Facilitating language documentation through incremental development of digital archive infrastructure

A dissertation submitted to the Graduate Division of the University of Hawaiʻi at Mānoa
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Linguistics

May 2023

by
Kavon Hooshiar

Dissertation Committee:

Gary Holton, Chairperson
Nick Thieberger
Bradley McDonnell
Kamil Ud Deen
Rich Gazan

to Grandma Dewey

and to the Papuans
for your spirit, your love, and your guidance

asal trupela
timur barat i bilong yupela

Acknowledgments

First of all, I would like to thank my parents, because without them I wouldn't be here. Babajon, thank you for everything you've taught me about the world, and for always supporting me in every way you could. Mom, I know that I am who I am because of you. I still dream of you, and speaking German with you. I keep your books as a reminder of our shared values.

I descend from a line of strong women that has always been a guiding light to me. Elizabeth Fluegge, born in Ganse, begot Charles Dewey, born in Des Moines, who married Beverly Jean Dewey, born in Ellensburg, who begot Jan Elaine Dewey Hooshiar, born in San Jose. Herachique Hofsepian, born in Tehran, begot Kamran Hooshiar, born in Tehran. Jan and Kamran Hooshiar begot Elizabeth Hooshiar, my dear sister, born in San Jose. Sissy, thank you for sheparding me, even when I kicked and screamed. Thank you for making sure I know they are proud, and thank you more for *being* proud of me.

I am ever so grateful to Dr. Andrea Berez-Kroeker, Dr. Katie Drager, and Dr. Gary Holton for teaching and mentoring me. Gary, thanks for having a beer with me. I couldn't have hoped for a better advisor and chair. You deserve more credit for your mentorship than my words can grant you here.

I would also like to thank the many professors and archivists of the Linguistics Department and beyond who educated me, including Dr. Lyle Campbell, Dr. Ken Rehg, Dr. William O'Grady, Dr. Victoria Anderson, Dr. Yuko Otsuka, the late Dr. Bob Blust, Dr. James Woodward, Dr. Kamil Deen, Dr. Brad McDonnell, Dr. Nick Thieberger, Dr. Kent Sakoda, Dr. Rich Gazan, Dr. Susan Smythe Kung, Dr. Mandana Seyfeddinipur, Wendy Camber, and Dr. Ladan Hamedani.

Thank you to all the Papuans who welcomed me, guided me, gave me a home, and worked with me. It was when I set foot on New Guinea for the second time in a place completely new to me that I realized I was at home. There is nothing I could ever give you that would equal what you gave me. I would especially like to thank Joe and Philip Talair of Amimai, Ben and John Ruli of Mane, Timo of Takar, and Osea of Betaf, for working with me and vouching for me. Anton – language and culture could not conceal the feeling that you are family to me. Thank you for sharing your knowledge, your life with me. Choosing to say goodbye to you was one of the hardest things I've done. I can never forget you.

Throughout this journey of multiple lifetimes I've been touched by so many whom I cherish and would like to honor here:

My dear Annette (Anet) Carty-Mole – you knew me so truly that you introduced me to this field. Dr. Don Stilo, Dr. Linda Konnerth, Chikari Tisso, Dr. Tom Owen-Smith, Dr. Lauren Gawne, Dr. Gwendolyn Hyslop, Dr. Mark Donahue, and Dr. Amos Teo – you welcomed me with open arms into your work and your lives and gave me my beginning in linguistics.

Kuʻu mau kumu no Hawaiʻi nei, who, in spite of history, intergenerational trauma, circumstance and my naievity, took me under their wing with grace: Kealoha Kelekolio, Clint Awai, Sabrina Kamakakaulani Gramberg, Dr. Kaliko Baker, Dr. Keawe Lopes, N. Haʻalilio Solomon, and Kaʻiulani Murphy. Mahalo nui loa. I am forever in your debt.

All the participants of LDTC who always managed to remind me how important all of this is, and especially my

intrepid partner in co-directing, Sejung Yang, as well as Philip Waisen, a great language expert and an even better friend.

The amazing people I met and bonded with in Manokwari: Yusuf Sawaki, Jeanete Lekeneny, Simon, Papua, the late Andreas Deda, Laura Arnold, Bruma Rios Mendoza, George Saad, and Eline Visser, and Fabian Cordero Peralta.

John Elliott – at this point I can't imagine who I would be if I hadn't met you.

Nick Dikas – the man of many nicknames, which I use with love – you keep on believing in me and it's gotten me here. And Julia Woolley – thank you for being there to tell me you know the feeling.

Melody Ann Ross, my friend, my sister, my compadre – my life is so much richer to have shared the road with you.

Christine Kelly – words don't describe all that you've meant to me. You connect me to the spirit world.

Ryan Shelby, for sharing all you've had and all you could. Anna Belew, for being the best at keeping me sane. Charitini Karadamou, for your special kind of energy. My fellow band members of Enemata for making music with me: Emily Krause, Alex Nagata, Nick Wood, and Jim Jacobs. Han(n)a Kim, for being so real. Ryan Henke, for reminding me who's my boy. Ingvild Badhwar, for calling me out and for caring about the result. Emily Jo Noschese, for being you. Jaeci Hall, for honoring me with sage. Jenna Macy, for sharing your flame with me. Penny Kremer, for taking me out to sea and giving me new life. Amber Camp, for being my dissertation support group, and for the thousand times you were supportive for any and all reasons. Emiliegh Tena, for inspiring me with words. Karolin Obert, for reminding me to feel. Raina Heaton, for believing in me.

And the many other members of the UH Mānoa community that have shared the ups and downs with me: Catherine Lee Brockway, Brad Rentz, Mihoko Sawada, Jacob Hakim, Colleen O'Brien, Alireza Eshraghi, Elham Monfaredi, Nora Garrod, Hamed Dehnavi, Andrew Pick, Dannii Yarbrough, Yen-Ling Chen, Leah Pappas, Blake, Constance Nicks, Victoria Chen, Bryn Hauk, Jess Charest, Michelle Kamigaki-Baron, Bethany Kalei, Reza Rahimnejad, Bahram Sanginabadi, Maseeh Ganjali, Maryam Raha, James Grama, Sam Rarrick, Linda Lambrecht, Jordan Antonio, Sigit Haryadi, Jim Yoshioka, Colleen Patton, and Brittany Wilson.

And also Zoë, Josh, Sarah, Michelle, and Amanda: thank you for letting me do my part in helping you earn your honorary degrees. Zoë, thanks for keeping me psyched.

It strikes me that I've been at this so long that a) so many of you are Doctor (though doctors you may be, classmates and friends you're still to me), and b) this list has become rather long, while I'm sure it's incomplete (I may have missed you in this list, yet I still miss you in my heart).

I must also give special thanks to my friends and family who put me up on the last leg of this journey: Nick and Julia, Elizabeth and Jon, Auntie Carol, Gary and Linh, Dad and Denise, and finally Linda Noeske.

In addition to all the amazing people who have helped me, I would like to thank the Bilinski Foundation and the National Endowment for the Humanities for supporting this research.

Last, but certainly not least, I would like to thank the acknowledgments section. In my darkest hours you never let that light at the end of the tunnel be extinguished. It's my priviledge to be writing you and I'm so happy I'm finally here with you.

Abstract

This dissertation engages in the methodology of language documentation by exploring how technology augments and/or hinders the processes of language documentation, with a specific focus on digital archive infrastructure. It seeks to make incremental improvements to the design of this infrastructure and thus the processes of language documentation. To do so, it asks: how can the design of digital infrastructure for archives be modified to increase archives' abilities to uphold principles of better practice in language documentation without drastically increasing the demand for resources placed on archives to deploy this infrastructure?

To answer this and related questions, I conducted a literature review, and I participated in conducting a series of workshops for users of the Native American Languages (NAL) archive to engage them in discussions and elicit feedback on what they want from digital archive infrastructure. With the information collected from these steps, I developed a design for digital archive infrastructure, in collaboration with NAL staff and IT development personnel at the University of Oklahoma, which incrementally modifies existing designs with the goal of furthering the effectiveness of language archives at preserving language data and providing appropriate access to these data in ways that empower Indigenous language communities.

The literature review identified key principles defined by the field for better practices in language documentation, and divided these principles into three categories from the perspective of archives: bringing data into the archive, getting data out of the archive, and cultivating relationships among stakeholders in language documentation and users of the archive. With these categories established, it identified the movement of data out of archives and cultivating relationships as areas of focus in the development of digital archive infrastructure. Finally, it presented recent and ongoing developments of archive infrastructure, and key aspects of modern web application software that can enable future developments.

The results from the workshops with NAL archive users confirmed the areas of focus identified in the literature review, in the sense that the interests, desires, and needs expressed by workshop participants mostly fell into these areas, specifically providing access to collections and maintaining relationships between Indigenous communities and the archive. These discussions motivated the inclusion of specific features in the design for digital archive infrastructure, including metadata fields, specific search and browse capabilities, and a system of user roles for moderation of collections and items in order to enable context specific co-curation.

Based on the information gathered from the literature review and workshops, a design for digital archive infrastructure was developed that modifies existing designs with the use of features from modern web applications in order to further the archive's ability to implement better practices by enabling different forms of access, co-curation, iterative archiving, and digital return.

In summary, this study engaged in the development of the methodology of language documentation by seeking to understand its foundational principles for better practice, and developing a design for digital archive infrastructure that further enables archives to enact these principles.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Language documentation is a social act; language is recorded by and with members of Indigenous communities and speakers of endangered languages, for their benefit and for posterity. Language documentation is a scientific act; recording language is necessary to make linguistic analyses reproducible and thus enact the scientific method. These two assertions hold simultaneously, and language recordings are the keystone of both acts.

Technology plays a central role in the methodology of language documentation today, because language recordings are made in digital formats and thus processed and preserved using computers. Meanwhile, as a field, language documentation develops and maintains a set of better practices - goals and aspirations for conducting language documentations ethically and effectively. Technology lies at the intersection of the nature of language recordings and the goals of the field; computer software is involved in almost every step and every process of modern language documentation, including making recordings, maintaining them, maintaining records of them, consuming them, sharing them, and protecting them and the rights of those who own them by virtue of creating them through the act of using language.

At their best, software tools augment the abilities of people to conduct language documentation by facilitating them in the tasks they need to accomplish, leading them and encouraging them to strive for better practices, and automating the tasks that are not intrinsically language documentation, but arise from the constraints of using computers and digital formats themselves. Software tools designed for language documentation that are widely used usually do facilitate users in accomplishing the tasks for which these tools are designed. However, best practices evolve, and technology developments bring about new methods for automating and streamlining tasks related to digital data and formats. With these changes come new opportunities to further develop designs for such software tools.

For software tools to better facilitate and augment the processes of language documentation, their design needs to be aware of and informed by these processes and their constraints, which are sufficiently complex and specific to warrant the development of specialized software in the first place. As a result of this relationship, the development of software for language documentation, and not just its use, is necessarily a facet of the methodology of language

documentation itself.

One of the crucial types of software that is developed, maintained and deployed in the practice of language documentation is the software that powers digital language archives. Language archives are repositories for language data maintained by institutions and charged with the long term preservation of these data (Henke and Berez-Kroeker, 2016:412, from Johnson 2004:143). The digital age and born-digital media that came with it brought new opportunities and challenges to the practice of archiving.

Golla (1995:152) predicted early on that stewardship of language data in archives would be a major task in the digital age, as well as the transmission of the knowledge needed to use these data. One of the main tasks of archives since then has been stewardship through digitizing legacy materials. Kaipuleohone (Albarillo and Thieberger, 2009:3), the Pacific and Regional Archive for Digital Sources in Endangered Cultures (PARADISEC) (Barwick and Thieberger, 2018:139), and the Native American Languages (NAL) archive (Linn, 2014:62) are a few examples of the many archives that have engaged in digitizing legacy materials during this period. This task requires specialized equipment, considerable time, and expertise (Linn, 2014:62), and having built this capacity, archives have engaged with local cultural centers (Barwick and Thieberger, 2018:139), Indigenous communities (Linn, 2014:62), and academic departments (Albarillo and Thieberger, 2009:3) to digitize their materials and make them available in the archives.

In addition to digitized legacy materials, archives faced a growing need to accommodate born-digital materials. These became ubiquitous along with technology like cheap and portable recording devices, storage media, computers, and phones, and together these have led to an abundance of digital language data. Storing language data in digital formats gave archives the ability to accommodate the growing amount of language data being created (Nathan, 2011:267), as well as increase access to collections by serving them over the internet (Trilsbeek and Wittenburg, 2006:325). Archives became digital in order to accommodate the developments in how language data were stored and to reap the benefits of digital technology. In addition to digitizing legacy media and supporting digital data, they did so by deploying software tools in the form of web applications with websites that provide for preservation of and access to digital media.

Early on in this process, Bird and Simons (2003b) codified better practices for how to manage digital language data while engaging in language documentation, including creating data and metadata with open standards and making data discoverable and accessible to all. These practices were implemented through the creation of the Open Language Archives Community (OLAC), which developed a metadata schema standard for language archives and a harvesting and indexing service for the metadata of language archvies (Bird and Simons, 2003a). More recently, similar ideas for better practices have been upheld by the broader scientific community through the development of the FAIR principles (Wilkinson et al., 2016), which emphasize the importance of the Findability, Accessibility, Interoperability, and Reusability of data in science. Additionally, the CARE principles (Carroll et al., 2021) extended these better practices as they relate to Indigenous digital heritage and communities.

As archives have come online through their digital infrastructure, including web applications for managing and serving digital data, they have used this infrastructure to uphold principles for better practices of data management in language documentation. Archives accomplished this in a number of ways. They grappled with and fine tuned their

metadata schemata in order to strike a balance between being sufficiently complex to describe data while also being simple enough to encourage depositors to create these metadata and to enable archivists to manage them (Barwick and Thieberger, 2018:136; Trilsbeek and Wittenburg, 2006:323). To encourage and facilitate depositors in creating these metadata, they built software tools designed to enable depositors to create metadata according to the archives' schemata in preparation for their deposit, such as LAMUS (Wittenburg et al., 2005:9), Arbil, SayMore (Moeller, 2014), CMDI Maker (Rau, 2016), and, most recently, Lameta (Hatton et al., 2021). In addition to providing tools for depositors, archives also focus on training depositors in how to prepare deposits (Nathan, 2011:262). Archives seek the use of open and well described formats for data (Barwick and Thieberger, 2018:137; Trilsbeek and Wittenburg, 2006:322; Conathan, 2011:250), and follow metadata standards in the broader scientific community in implementing their metadata schemata and protocols for harvesting these metadata to make their collections more discoverable, such as the OLAC metadata schema, an extension of the Dublin Core metadata schema, and the Open Archives Initiative (Barwick and Thieberger, 2018:138; Albarillo and Thieberger, 2009:6).

While the seven dimensions of portability and the FAIR principles for data management and stewardship focus primarily on making data in science available to all, language archives operate in the context of managing language data that are created and owned by Indigenous communities. In order to manage this complex set of constraints, archives have developed systems of different types of access restrictions for data and integrated these types into their metadata schemata (Trilsbeek and Wittenburg, 2006:333; Albarillo and Thieberger, 2009:7). Common types of access restrictions include: a requirement to authenticate as a user in order to view the data, a requirement to seek the depositor's permission to view the data, and a requirement to wait for a designated period of time to pass in order to view the data. Implementing such systems allows archives to provide as much access as possible, through the ease of internet access and mechanisms for discoverability, while simultaneously restricting access as much as necessary to meet the needs of depositors and language communities.

In order to make data in their collections discoverable, archives implement search functions for their metadata (Barwick and Thieberger, 2018:136; Conathan, 2011:246), assign persistent identfiers to their collections and data (Barwick and Thieberger, 2018:137), which are special links that are maintained so that they always take the user to the intended destination, and they enable their metadata to be harvested and indexed in broader search engines (Barwick and Thieberger, 2018:137; Albarillo and Thieberger, 2009:7). To enable access to the data in their collections, archives provide digital data files in high resolution preservation formats as well as lower resolution formats for ease of access. Additionally, they allow for data to be downloaded for use on the users' local machines, as well as streamed for convenient viewing (Barwick and Thieberger, 2018:138; Trilsbeek and Wittenburg, 2006:317).

Overall, archives have worked to create and maintain the digital infrastructure needed to accept deposits of complex data and then provide immediate access to these data with appropriate restrictions, while accommodating the workflows of different types of users, such as deposits, archivists, and users (Trilsbeek and Wittenburg, 2006). Moreover, they have done so while maintaining the core role of archives of preserving data. They responded to the need created by the age of digital data by building the protocols and processes to make archives digital and bring them online. In doing so, archives have filled the role of enabling linguistic research to be grounded in language data,

as predicted by Evans and Dench (2006:24) and reinforced by Berez-Kroeker et al. (2017). Additionally, they have worked to provide access to Indigenous communities and enable them in their own use of language data (Holton, 2012:108), as well as repatriate data to Indigenous communities (Thieberger, 2019; Barwick and Thieberger, 2018:139). Throughout this period of innovations in digital infrastructure for archives, archives have addressed many of the needs of stakeholders for managing language data. Due to the complexity of this infrastructure, and that of the data it is designed to manage, archiving language data has increasingly become a specialized area that requires technical expertise as well as an understanding of the data being managed (Arnold et al., 2020).

Still, there appears to be room for further development for archives, specifically in terms of the data flowing into archives, addressing the desires of Indigenous communities, and methods used to provide access to data in archives.

First, while archives have provided an invaluable service to practitioners of language documentation and scientific researchers by preserving recordings of language acts, only a minority of these recordings are preserved alongside corresponding transcriptions and translations. This limits the utility of deposited materials, not only for researchers who would seek to reproduce linguistic claims, but also for language community members who wish to view the materials, especially L2 speakers and heritage speakers. This situation is not primarily caused by a deficiency in archives – rather, it is likely the result of a conflict that arises between a desirable practice of archiving recordings as soon as they are available, and waiting to archive transcriptions and translations until they are deemed complete or ready, which may take somewhere between 10-100 minutes per minute of recording (Himmelmann, 2018:34). Nevertheless, it may still be valuable to consider what archives could do in developing their own infrastructure to encourage depositors to include transcriptions and translations in their collections.

Second, there is a sense that Indigenous communities have desires that may not be fully meet through the practices of language documentation, including the preservation of language data in archives. Various Indigenous groups and individuals have expressed this concern. For example, in describing the Breath of Life workshops, which are events designed to enable Indigenous community members to access and make use of archived language data, Leonard (2017) uses a framework of reclamation, claiming that Indigenous ownership of language data is, to some extent, lost when data are archived with non-Indigenous institutions. Looking at software tools, Brinklow (2021) claims that language technology will fail to serve Indigenous people as long as they are not included in its development. In the broader scientific context, the CARE Principles for Indigenous Data Governance are goals for better practices that were developed in response to the FAIR Principles, not only to augment them, but out of concern that they did not accommodate Indigenous peoples' needs for data management. The CARE Principles focus primarily on building relationships between existing institutions and Indigenous communities, building capacity in Indigenous institutions, and training Indigenous individuals in order to empower Indigenous communities in the management and use of their data.

The theme that emerges from these examples is that Indigenous people have desires to be included in the design and management of the processes that lead to the management of language data. Based on the pattern and timing of such examples, there appears to be a disconnect between Indigenous communities and academic enterprises that is systemic and has yet to be fully resolved. This context provides a reminder for archives that different stakeholders

in the practices of language documentation and archiving can have different expectations for how archives manage language data, and that these expectations can form complex sets of constraints when taken simultaneously. In each example in the previous paragraph, Indigenous people are asking to have more of a seat at the table. At the same time, some archives have put in significant effort to enable different stakeholders, including Indigenous communities, to participate in their processes of stewardship (Theimer, 2011; Linn, 2014; Garrett, 2014). The extant desire of Indigenous peoples to participate in and have ownership over processes of data management leaves room for archives to ask how they can continue to address this desire.

Another theme that emerges from Indigenous perspectives is a desire for more control over access, both in terms of providing more access in the ways that benefit Indigenous peoples, as in Leonard's framework of reclamation, and in terms of providing more control over access restrictions, as in the case of the CARE principles. While archives do focus on providing access for data as well as access restrictions to data where appropriate, this is another area where further development can be focused. Specifically in digital infrastructure, access and access restrictions have been designed for general use cases. For example, the goal of providing access has been focused more on giving people the ability to access data, and less on the ways in which specific stakeholders would like to access these data. In the case of access restrictions, digital archive infrastructure often relies on a restriction type that asks depositors to approve individual requests for data. This is a very powerful general solution to complex and diverse needs of different stakeholders and communities for restrictions (Trilsbeek and Wittenburg, 2006:317). However, it also places the power and responsibility of managing access restrictions in the hands of depositors, who may not be able to moderate these requests and may not feel comfortable with this responsibility. Thus, while providing access and access restrictions have been achieved by archives, there is still room for development in the methods and mechanisms for achieving them, given the calls for improvements by Indigenous community members.

Meanwhile, developments in web-based software tools enable new ways to provide interfaces to data, and this gives an opportunity for expanding the capabilities of digital archive infrastructure without increasing the demand for resources to run this software. On the one hand, rapidly changing technology represents a burden on institutions with limited funding, such as archives, who must grapple with constant maintenance and the risk of their software becoming incompatible with new frameworks. On the other hand, developments in software bring with them benefits that can be gained if newer frameworks are embraced. In a recent doctoral work in linguistics focused on adopting newer technology in language conservation work, Bettinson (2019) promotes the use of of Web 2.0 technology in language documentation and conservation. Web 2.0 refers to web based software applications that are dynamically generated through consisting of a database to store contentful information, and algorithms that combine those data with formatting information on the fly when a website user makes a request for content. This framework provides a logical structure to think about managing data through web-based software applications, with data stored in a database and processing to bring data in and out of that database. Making use of such a framework gives developers access to a plethora of open source software tools and features that have been born of the ever changing landscape of web-based applications and web infrastructure in general. By virtue of managing collections and providing websites as interfaces to these collections, archives are already engaged with Web 2.0 technology and

using web-based software applications for their digital infrastructure. However, as this technology evolves, there are more opportunities for utilizing it that archives can consider in their continued development.

Language archives are in fact continuing to change and develop. For example, the Archive of the Indigenous Languages of Latin America, hosted at the University of Texas at Austin, completed a redesign of their digital infrastructure in 2015, and the Archive Manager, Dr. Susan Kung (personal communication) has a list of features to request for the "next upgrade." As another example, The Language Archive, hosted at the Max Plank Institute in the Netherlands is currently carrying out an exploratory project as part of the European Text+ initiative to make its capacity to preserve language data more robust by duplicating the language data of its collections at an entirely different institution. These examples show how archives are continually developing their capacities to fulfill their role of preserving data and providing access to these data.

Archives pursue these developments to meet the changing needs of stakeholders, including Indigenous communities, in spite of the fact that financial and personnel resources are extremely limited for these institutions, which are often housed at universities and rely on soft money to fund these developments. They may not have dedicated employees for these tasks, but professors and staff doing this work as part of their larger duties. Often, these archivists are linguists, and yet they are charged with enabling various user groups who are not linguists (Holton, 2012:108) and use cases that are not predictable (Holton, 2011).

1.1 Research question

The primary research question posed in this dissertation is: How can the design of digital infrastructure for archives be modified to increase archives' abilities to uphold principles of better practice in language documentation without drastically increasing the demand for resources placed on archives to deploy this infrastructure?

To answer this question, this work engages with the methodology of language documentation by identifying goals for data management contained in the principles for better practices defined in the field, and exploring modifications to software used in this methodology in an attempt to improve its outcomes. In doing so, it recognizes the achievements of archives thus far at maintaining their core functions while developing their digital infrastructure, and moreover in the context of the constraint of limited resources in which they operate. This work engages with different users of archives, and especially Indigenous community members, seeking their feedback on what they would like from developments in digital archive infrastructure. In doing so, it continues the ongoing process of building relationships with these stakeholders. This is a crucial aspect of empowering Indigenous people through developments in digital archive infrastructure, especially given that the author cannot represent Indigenous people (see §1.3). Additionally, this work explores the application of additional components of the dynamic web and modern web frameworks to the design of digital archive infrastructure in order to answer the research question.

This work answers the research question in the context of a case study of the Native American Languages (NAL) archive. The NAL archive currently does not provide online access to its collections, and is currently engaged in a project to provide such access. For this reason, the NAL archive presents a useful opportunity to pursue developments

in digital archive infrastructure. Moreover, NAL actively engages in participatory archiving and collaborating with Indigenous communities (Linn, 2014). In this context, the NAL archive is well situated to engage with Indigenous communities during the development of its digital infrastructure.

In summary, NAL presents a useful case study for this work because it currently has a project and funding to develop its digital archive infrastructure, and it has room to grow in terms of providing access and collaborating with Indigenous communities in this development. This case study helps to ground this work in the reality of an existing archive that is seeking to develop its digital infrastructure and has real needs and constraints that it seeks to meet.

In pursuing the goal of incremental development of digital archive infrastructure, this work does not presume that present outcomes are unacceptable. Rather, just as better practices for the field are developed over time, so too should practitioners of language documentation seek to gradually improve the tools that enable its methodology.

1.2 Structure of the dissertation

The content of this dissertation is presented in five chapters. Chapter 2 summarizes the relevant literature on the roles and needs of different stakeholders in language documentation, the role of archives, the nature of language data, principles for better practices in language documentation, and aspects of modern web-based software tools that are relevant in pursuing new modifications to the design of software tools for archives. Chapter 3 presents the results of a series of workshops that the Native American Languages (NAL) archive hosted for its users to give feedback on what they want to see from new digital infrastructure for that archive. These workshops engage with principles of better practices in language documentation by building and maintaining relationships with Indigenous language communities and seeking to empower them in the way language data are managed, made available, and used. Chapter 4 defines a design for digital archive infrastructure that I developed in collaboration with NAL staff and IT development personnel at the University of Oklahoma. This design attempts to answer the question of how such infrastructure can be modified to improve outcomes, while being informed by the results from previous chapters. Finally, Chapter 5 summarizes the results of each chapter, the answer to the question posed by this dissertation, the contributions of this study to the field, and directions for future work.

1.3 My positionality in this research

I am primarily a linguist and PhD candidate in linguistics at the University of Hawai‘i at Mānoa, with no affilations to an Indigenous community. I identify as European/Persian/Armenian-American, and I have spent most of my life as a settler living on the occupied land of Native Hawaiians and Ohlone Native Americans. Living according to the Golden Rule has always been my most core value, and before I began pursing a career in linguistics, I spent years living with different communities in different societies that I would have considered “others” as a child, and this expanded my in-group to include all humanity in a very real sense. I made the decision to pursue linguistics based on my love of the beauty and complexity of human language, and as a direct result of my first introduction to the field

of language documentation. My path toward developing technology for language documentation emerged through my personal longing for tools that better met my needs and the needs of those I was in contact with, as well as my training and experience in industry developing software.

Thus, I do not have any right to represent Indigenous communities in relation to developing models for software for language documentation that meet their needs. Moreover, I agree with Brinklow (2021), that technology is inherently colonial unless Indigenous communities are participating in and leading its development, and thus I cannot effect decolonization. However, I believe that the motivation of individual and group actors does influence the outcomes of systemic processes. I am motivated by my specific set of skills and experiences across language documentation and software development, but the thread that holds this motivation together is my pursuit of the Golden Rule. I want to build software that suits others as well as it suits me, and I know I can only be a small part of the solution. In order to enact positive change, I charge myself to continue listening, always.

Chapter 2

Background

The field of language documentation is based on the creation and management of language data (Himmelmann, 2006:15), and language archives have long played a crucial role in preserving these data. This chapter provides background information on who the stakeholders are in language documentation projects, the role archives play, what language data are, the value statements that have been defined in the field, and technical aspects of managing language data.

§2.1 presents an overview of who participates in language documentation by presenting definitions of types of stakeholders and their roles, as well as some of their interactions. §2.2 defines language archives, and presents their role in language documentation and their position in language documentation workflows. §2.3 defines language data, and §2.4 presents an overview of the metadata archives use to describe language data, as well as metadata standards for language archives. §2.5 presents developments of better practices in the data science of language documentation, including portability, reproducibility, the citation and attribution of language data, and the FAIR and CARE principles. §2.6 describes recent developments in archive infrastructure through the example of the redesign of the infrastructure for the Archive of the Indigenous Languages of Latin America. §2.7 presents some important software features that are relevant for digital archive infrastructure. Finally, §2.8 introduces the Native American Languages archive, which serves as the case study for this dissertation.

2.1 Stakeholders

Various groups of people have roles and interests in language documentation. These include speakers or signers of the language being documented as well as other members of communities associated with the language (collectively the language community), linguists, archivists, funding bodies, universities and educational institutions, NGOs, government institutions, and software developers (Arka, 2018:132; Sawaki and Arka, 2018:263). Stakeholders can be individuals and groups, and individuals can have affiliations to multiple types of stakeholders (Arka, 2018:132). There is often considerable variation in the values and perspectives of individuals and groups that identify as each type of stakeholder. For example, practices considered ethical can vary widely between different language commu-

nities, as well as the norms and expectations of who has the authority to represent these groups (Holton, 2009:169). To better understand and define types of stakeholders, Nathan (2015:59-61) calls for empirical approaches to defining these types, for example through asking archive users questions about their role with respect to language data during registration, as in the case of the Endangered languages Archive (ELAR).

Of the different types of commonly defined stakeholders, the language community and linguists have the most direct interaction with language data. These two groups are not always dichotomous. Individuals who identify as community members can have training and experience as linguists, and linguists can either identify with the language community or have experience with and understanding of their cultural practices, norms, and expections. In this way the role of language community member and linguist can be represented as two ends of a spectrum, and the act of language documentation gives every individual more experience with the roles they did not originally take on, bringing every individual closer to the middle of that spectrum (Hooshiar, 2017:79). Nevertheless, the collaboration of these different stakeholders can be problematic due to "conflicting ideologies, history, and resource availability" (Chelliah, 2018:248). These differences can influence what language data are created. For example, an individual acting solely as a linguist may have little regard for the type and quality of the non-linguistic information being documented, whereas this information will be valuable to an individual acting solely as a language community member (Holton, 2012). Leonard and Haynes (2010:274) argue that this should be addressed through models of collaboration between different types of stakeholders that value the needs and expertise of the different types of stakeholders equally.

Models for defining the process and outcomes of language documentation projects have increasingly been designed and shaped to meet the needs and desires of the language community (Czaykowska-Higgins, 2009:17), as defined in §2.1. Of primary concern are the communities' preferences toward what data are created, how and by whom those data can be accessed, what technologies will be used (digital and analog), and what training and compensation will be involved. Crucially, these preferences will be different across and within different language communities (Holton, 2009:173), so the design of each language documentation project will differ. To truly decolonize language documentation research (see Smith, 1999:117), the language community should take on an integral role in reviewing proposals and results of language documentation projects (Leonard, 2018:62).

As stakeholders, language communities have needs regarding the qualties of language data created, the organization of those data into collections, the specifics of access to language data, the speed at which language data become accessible, and facilitating the use of language data beyond simple access, among others.

According to Holton (2012:106) community members may be interested in the types of language data that encode cultural knowledge as well as those that encode linguistic knowledge, even if those data were originally created with the intent of recording linguistic code. For example, ethnobotanical and astronomical and biographical narratives encode knowledge of cultural practices and people that are valuable to language communities. Language commmunities can also have specific needs with regards to the organization of language data into collections, prefering an emic system of categorizing and presenting language data to one based on categorical systems of linguistics (Widlok, 2013:187).

Community needs regarding access can be complex and usually vary across and within communities (Shepard, 2016:465). Making language data easily accessible to community members serves various needs beyond access to cultural knowledge. It facilitates ownership of a community's intellectual property, which is crucial for communities to achieve self-determination (Shepard, 2016:463), and it can help community members refer to the wisdom of others where speaking on behalf of those others would be culturally inappropriate (Widlok, 2013:187). At the same time, communities may not want the default level of access to be open to everyone (Widlok, 2013:190), and often a language documentation project requires a complex set of access restrictions for specific language data to be accessed and specific individuals and groups to access those data. Finally, the speed with which access becomes a reality is important. According to Widlok (2013:189), in order to be an active participant in a language documentation project, a language community needs to be able to review and respond to newly created language data quickly, which requires data processing and access protocols to be in place and efficient. Additionally, facilitating this capability for moderation should not place new burdens on the language community which may impede the language documentation project.

As crucial as access to language data is, community needs go beyond access itself. In asking if data preservation and access are enough for language commmunities, Shepard (2016:462) presents the following additional desideratum. Language data can serve communities in the goals of education in the language as well as supporting land tenure claims, and according to Shepard (2016:466-473), resources for accessing these data should be built with such goals in mind. While preserving data for future use has been considered important, it is now important to understand the ways in which that future is here, and thus support the current needs of communities to use their language data (Shepard, 2016:462).

Overall, Widlok (2013:191) summarizes the needs of communities members by saying that they should not bear the burden of understanding the technology which has facilitated the creation and management of large quantities of digital language data – the technology should be designed to meet their needs, even those that are unanticipated.

2.2 Language archives

As Shepard (following Derrida, 1995) reminds us, the term archive comes from "a Greek word meaning a place of convergence, where things commence and where authority is commanded" (Shepard, 2016:458). An archive is "a trusted repository created and maintained by an institution with a demonstrated commitment to permanence and the long-term preservation of archived resources" (Henke and Berez-Kroeker, 2016:412, from Johnson 2004:143). Archiving language data has a history going back to the 19th century, and a summary of this history before the digital age is given in Henke and Berez-Kroeker (2016:413-417). The first archives were collections of textual materials and early recording media such as wax cylinders, deposited with institutions charged with protecting them but not necessarily providing broad access to them (Henke and Berez-Kroeker, 2016:413-414).

Archives are collections of records, either from a single source or project, or from different sources with a common theme (Woodbury, 2014:19). "Potential records are appraised, and if selected, they are accessioned, arranged and

described by means of metadata, guides and finding aids of various kinds, which make them accessible" (Woodbury, 2014:19). The role of language archives increased with the development of language documentation as a subfield in linguistics that followed Himmelmann (1998). Digital archives developed at the end of the 20th century as digital media and the internet became ubiquitous. "Born-digital" media is now the norm, and analog archival materials are being digitized, making digital archives essential to preservation of language data. Since this digital revolution, the concept of participatory archives has been developed in an attempt to further address the needs of community members described in §2.1.

2.2.1 Participatory archives and co-curation

As digital resources and digital archives became more common, various stakeholders raised issues with the archives' ability to meet users' needs not only of accessibility, but also of dissemination. According to Widlok (2013), digital archives brought with them the promise of preventing data stores from becoming "data cemeteries," i.e. places where data are compiled that become inaccessible, but fulfilling this promise requires the creation of sufficient metadata, and the quality of metadata needs to be sufficiently emic to the communities producing the data to facilitate their access to it (Widlok, 2013:187-188).

The importance of metadata is multiplied when it is observed that users want to access resources for various reasons beyond what the depositors of those resources envisioned (Widlok, 2013:189; Holton, 2011), including the documentation of Indigenous placenames (Holton, 2011:159), and the creation of language education materials (Holton, 2012:108). Thus, to achieve a high quality of metadata, their creators should be mindful that future access of the data could be to curate non-linguistic information contained in the data or to create derived language materials (Holton, 2012:109). Additionally, various stakeholders can participate in increasing accessibility by producing guides to the collections and transcripts that are as detailed as possible.

Various authors have pushed the discussion beyond just the need for high quality metadata to the need to include diverse stakeholders in the process of language data creation and maintanence (Theimer, 2011; Linn, 2014; Garrett, 2014; Nathan, 2015). Wasson et al. (2016:642) found that most language archives are not meeting the needs of most users, and proposed user centered design of language archives as a way of addressing this issue. In serving language communities, access means more than links to data, but requires communities to have control over the process of data management and distribution Holton (2017). A participatory archive model is a proposed solution to the need for more community control.

Theimer (2011:9) defines a participatory archive as one where people other than archive professionals contribute knowledge or resources, which results in an increased understanding about language data. Linn (2014:61) defines a Community-Based Language Archive as an archive or collection that maintains and disseminates documentation that is conducted for, with, and by language communities. Linn adds that it is the continued interaction of the community with language resources through the available means of dissemination that truly constitues accessibility, rather than simply having access to these resources. Similarly, Garrett (2014:69) defines a Participant-Driven Language Archive as one that assigns rights and responsibilities to community members through direct, web-based

relationships. Nathan (2015:74-75) provides an example of feedback channels created by assigning such roles. Specifically, when archive users were required to request access from depositors, this opened a line of communication that informed users and created relationships around language resources.

Another approach to involving different stakeholders is co-curation, which focuses not only on the repository of data but the goal of building community through collaboration as a means to share the knowledge in the repository (Mutibwa et al., 2020:173). The findings of Mutibwa et al.'s case study on co-curation highlight that, in addition to being able to browse and search collections, it was important to create physical and virtual spaces to discuss collections, as well as to be able to document these discussions themselves.

Participatory archives and co-curation are themselves forms of collaboration among different stakeholders that occur in the physical or virtual spaces of archives. Leonard and Haynes (2010:274) argue that for collaboration to be successful, the needs and expertise of different types of stakeholders, especially language communities and researchers, should be valued equally.

Participatory archives and co-curation are important ways to engage diverse stakeholders, and specifically those stakeholders with ownership over the data, who have traditionally faced higher risks of their voices going unheard. At the same time, archives find themselves in diverse contexts, and there is no one size fits all framework for engaging stakeholders. Participatory archives and co-curation are frameworks for engaging stakeholders that are motivated by a desire to increase the quality and quantity of such engagement in ways that have not been the norm in the past, and thus are meaningful approaches to consider in the context in which an archive finds itself. While engaging diverse stakeholders to collaborate in the archival processes of depositing and curation have been discussed in the literature, there has not been a discussion about how such collaboration could be enabled by software design for digital archive infrastructure, to the author's knowlege. One approach to such a software design is described in §4.4.6.

2.2.2 Digital return

Digital return is a cover term for processes that return digital cultural heritage resources to Indigenous communities, and definitions of what constitutes digital return range from providing online access to resources that have been dislocated from a community, to conveying legal ownership of resources to a community, to relinquishing all copies of resources not returned to a community (Barwick et al., 2019:3). Processes of digital return are complex and specific to Indigenous communities. For example, digital return may involve navigating complex interactions of non-Indigenous legal frameworks of ownership and Indigenous rights to their cultural heritage (Barwick et al., 2019:8). Additionally, digital resources preserved by non-Indigenous institutions are usually cataloged with metadata based on non-Indigenous ontologies, which can undermine an Indigenous community's ability to use these resources and transmit this knowledge to younger generations (Barwick et al., 2019:14).

Language archives are institutions that participate in digital return in various ways. Providing online access to digital resources is one way of returning these resources to Indigenous communities, but there are other more proactive methods that can have more value for Indigenous communities. For example, the Pacific and Regional

Archive for Digital Sources in Endangered Cultures (PARADISEC) has developed software to export collections for specific Indigenous communities and hardware that communities can keep in their home locations to access these collections even without internet access and with limited electricity (Thieberger, 2019; Barwick and Thieberger, 2018:139). As another example, the Breath of Life workshops bring together Indigenous community members to give them access to archival materials and also train them in how to use these materials (Holton, 2017:19). The development of the Mukurtu CMS is an example of digital return through software design. Mukurtu aims to give Indigenous communities control over their data through the use of a repository management system that allows them to curate their data and manage metadata according to their needs (Holton, 2017:19). The software design presented in Chapter 4 is motivated by both the prospects of archival participation in digital return, and software design for repository management systems as a method to enable digital return.

2.2.3 The archive's position in language documentation workflows

The role of the language archive is to manage, preserve, and protect language data, and in doing so, serve various stakeholders in language documentation. This role is essential to the practice of language documentation (Himmelmann, 2006:15; Austin, 2014:60; Garrett, 2014:75; Nathan, 2015:56; Berez-Kroeker and Henke, 2018:349). At the same time, emphasis is placed on the need for archived language data to be more accessible (Nathan, 2015:76) and cited (Berez-Kroeker et al., 2018:14-15), and the need for archives to serve different stakeholders more effectively Garrett (2014:82).

In spite of the importance these authors place on existing archives as well as potential improvements to archives, the fact that archives are not more central to the actual practice of language documentation comes to light in technical discussions of workflows for the processing of language data and design proposals for software tools for these processes. For example, Hanke (2017:57) describes the process of language documentation as a series of linear steps from the production of raw data, through recording, metadata creation, annotation, and transport from the field, and ending with archiving the data, as shown in Figure 2.1. The fact that not all language data that should be in archives reaches archives is explained as a culmination of issues that occur at each step of the process leading to archiving, including the lack of automated digital metadata management, data loss and mismanagement, funding issues, interpersonal issues, etc. (Hanke, 2017:57-63). The fact that archiving happens only after all other steps are completed is left as an assumption, and the possibility of archiving being additionally available at intermediate steps is not considered.

Once data is in an archive, it appears that software developers and other stakeholders can have different views on what happens with it. Nathan (2015:56), in advocating for archives to increase their reach to diverse groups of stakeholders, presents a model that situates archives in the middle of various ingestion and dissemination processes by and for various groups, as shown in Figure 2.2. This model assumes that accessing the data in archives is just as important as depositing data into archives.

In contrast to Nathan's model, Hanke's (2017:115) proposal for a modified workflow that motivates the development of the software tool Aikuma, shown in Figure 2.3, retains the archive as an end of its own, and adds a different

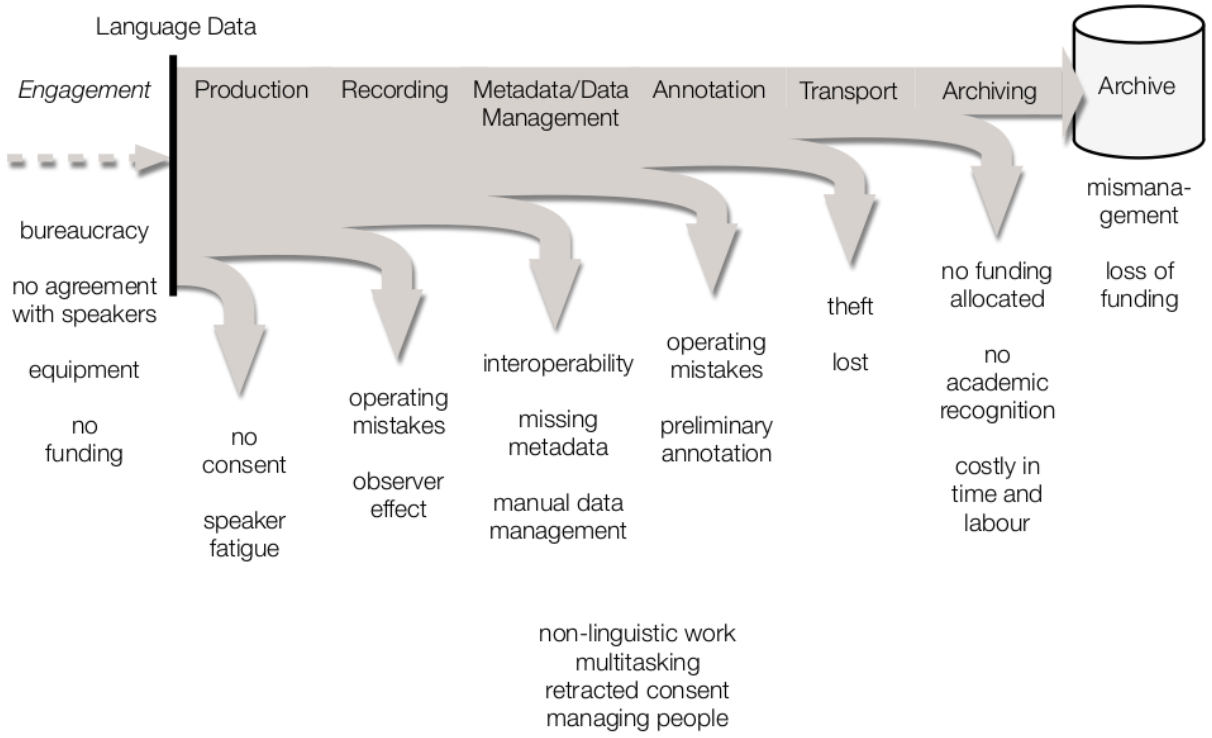

Figure 2.1: Phases of a language documentation workflow and issues associated with each phase. Archiving is considered the final step of the workflow (Source: Hanke, 2017:57).

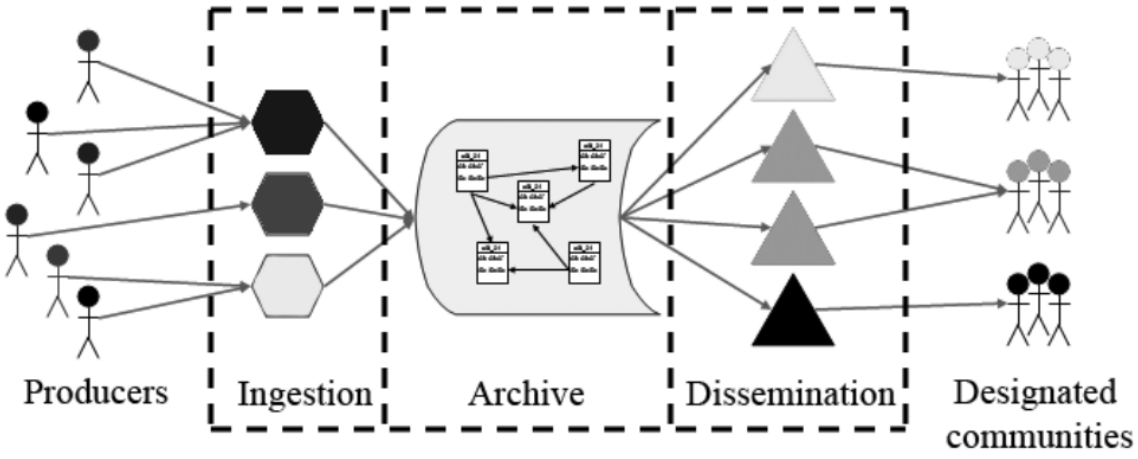

Figure 2.2: In modeling the role of the language archive, Nathan (2015) positions the archive in the middle of ingestion and dissemination processes by and for various stakeholder groups. (Source: Nathan, 2015:56).

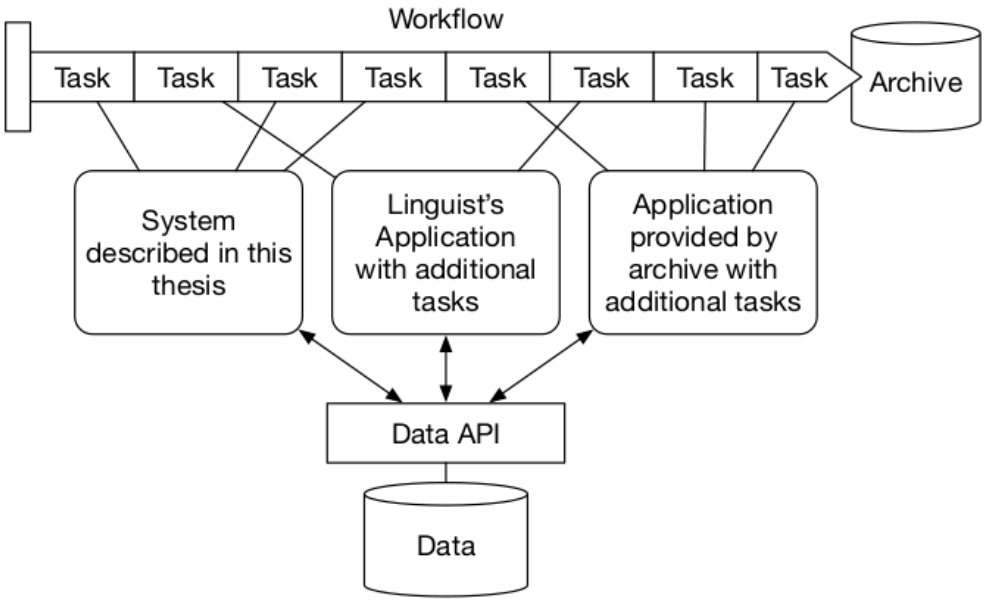

Figure 2.3: The model for Hanke's (2017) software development adds a second data store with an access layer in addition to the archive, which does not have a corresponding access layer. (Source: Hanke, 2017:115).

data storage component, labeled simply as data, with a data access layer, known as an Application Programing Interface (API) between the data and the various processes of language documentation that lead to eventual archiving. In this model, which describes the integration of Aikuma into language documentation workflows that also include depositing data in a language archive, data producers are asked to deposit their data twice, once in a database, and once in the archive. Access to the data is done through the access layer to the database, and never through accessing the archive. Moreover, the modular expandability to consumption of data in this model is through the API to the database and not the archive. In addition to having to deposit data in multiple places, this model creates redundancies and additional overheard, as both the archive and the database are taking some responsibility for storing data, and changes in one data store may need to be reflected in the other. Furthermore, resources for software development are spread thinner to develop both the database and archive as well as their corresponding access layers.

Bettinson and Bird (2017:162) use a similar model to illustrate their approach to software development for language documentation, shown in Figure 2.4. This model further emphasizes the importance of the API layer for accessing data, which is the interface between the data and various processes that users would be interested in, such as viewing the data in mobile apps, data processing and machine learning. In this model, there are again two points of data storage. The first, labeled "Language data API storage" exists explicitly to enable the access to the data through the API being created as part of this software development being presented. Adjacent to these new developments sits the language archive, which in this model is both literally and figuratively sidelined. Rather than storing the data in a way that is useful to access through various APIs for the various processes desired by stakeholders, in this model the archive's sole role is to store data for no one to use. Thus, this model makes no attempt to improve the accessibility

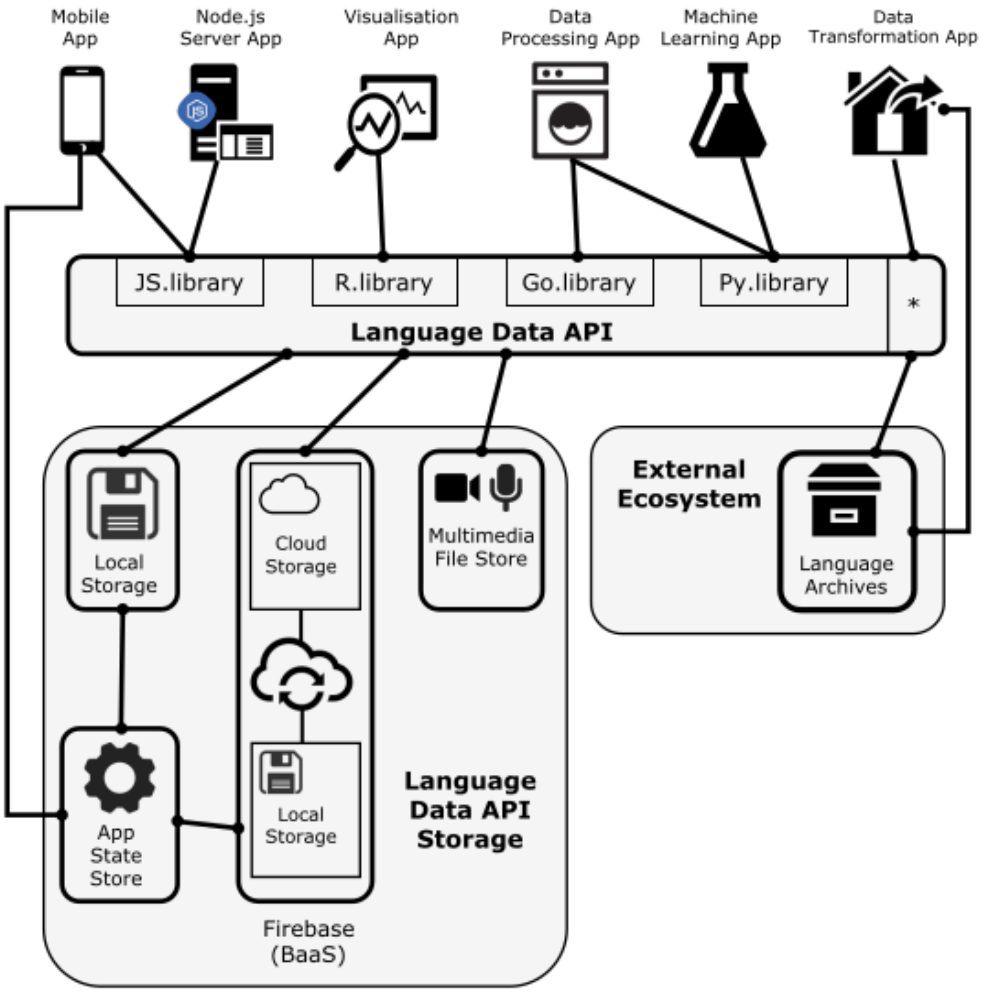

Figure 2.4: Bettinson and Bird's (2017) model for software development emphasizes the importance of an API to access data by different applications, and includes "Language data API storage," while the archive is only accessed for purposes of depositing data. (Source: Bettinson and Bird, 2017:162).

and dissemintation of language data through archives, and rather attempts to augment these capabilities using an alternative database that does not offer the same guarantees of data preservation as a language archive offers.

Baldwin et al. (2016:402) present a model for data processing, shown in Figure 2.5, that has similar steps to those in Figure 2.1, including processing data, transcribing it, depositing it, and in this case, accessing the data as well. However, this model stands out by virtue of dropping the language archive entirely, and storing the data solely in a database managed by the users themselves, which may not have the same guarantees of data preservation that are provided by language archives. This model assumes that participants in the project have the resources and will to manage and maintain a database for language data by themselves, including long-term data preservation.

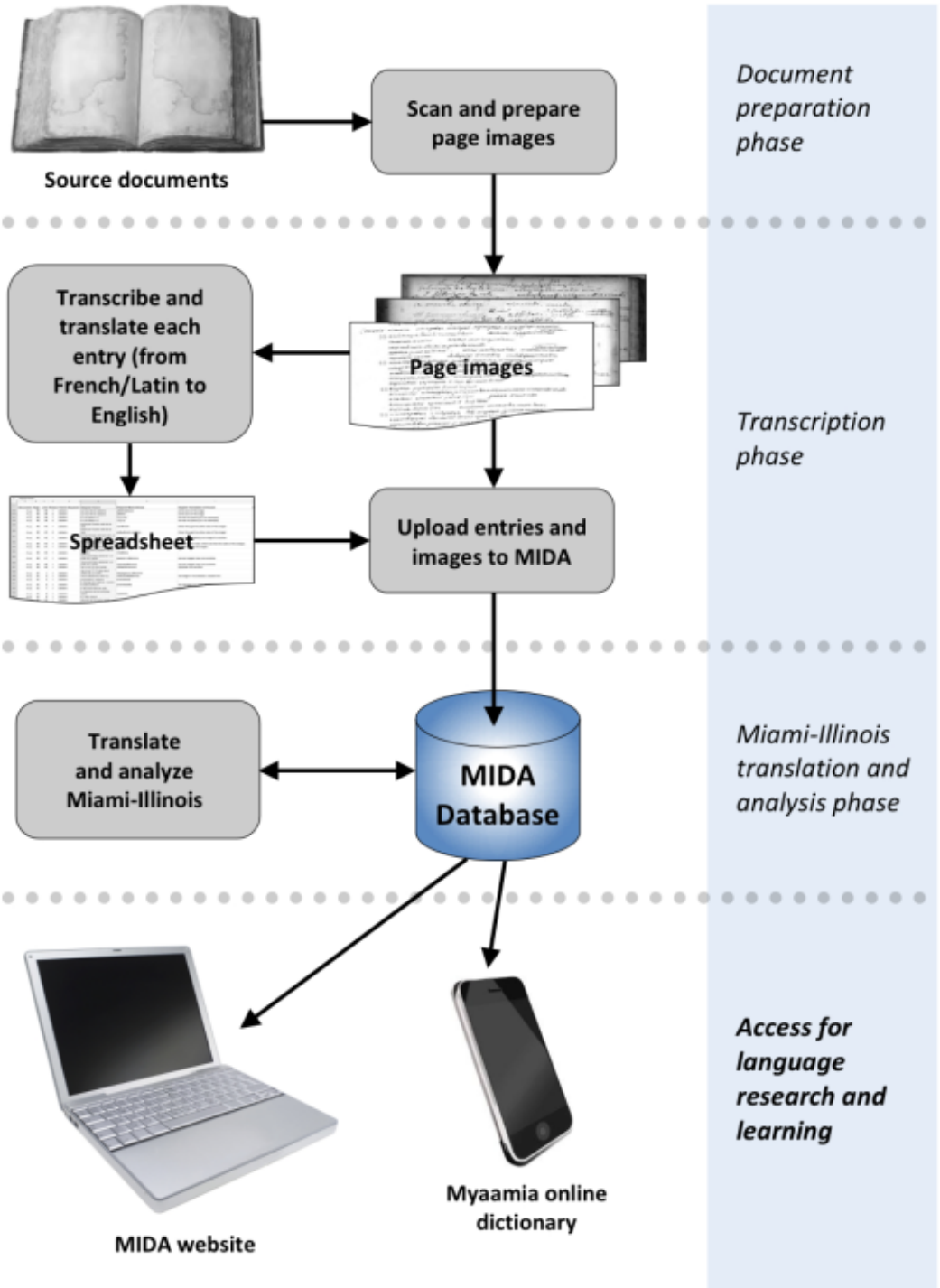

Figure 2.5: Baldwin et al. (2016) presents a model for a language documentation that uses an independent database as the sole data store, in lieu of a language archive. (Source: Baldwin et al., 2016:402).

2.3 What are language data?

Language data are the data used in the scientific practice of linguistics, and the creation, processing, management, and respect of language data make up the core practices of language documentation. Language data can be defined descriptively by dividing them into categories such as: raw, primary and secondary language data (Himmelmann, 2012).

2.3.1 Levels of data

Linguistic data are data that represent and describe language acts. A language act is an event where one or more people speak or sign a language. Linguistic data can be: audio and video recordings of language acts; metadata describing the context of language acts such as who was involved, when and where they occurred, and what languages were used; transcriptions and translations of language acts; and other, more abstract lingusitc descriptions of language acts, such as phonemic, prosodic, morphosyntactic and lexical analyses.

Himmelmann (2012) divides linguistic data into three main categories: raw data, primary data, and secondary data. Raw data are recordings of language acts. These data are unique, in the sense that any language act can only occur once and thus any recording of said language act can never be replicated. Additionally, raw data are particular, in the sense that they have a history associated with when, where, and by whom they were created. Primary data are transcriptions and translations of raw data. They are non-unique because transcriptions and translations can be made multiple times by different groups of people, and particularly because the results of creating these data will differ depending on the the historical context of their creation. Moreover, even within one documentation project, there can be multiple versions of transcriptions that are simultaneously valid (Marten and Petzell, 2016:117). Secondary, or structural data are the descriptions of languages based on these primary data. They are non-unique and non-particular, or in other words general, because they are claims about the grammar of languages that are expected to be consistently derived from particular primary data.

Two characterstics that distinguish raw data from both primary and secondary data are digital file size, and digital filetype. The digital file size for raw data is relatively large, as these data are stored as high quality audio and video files. In contrast, all primary data are stored as text, which is generally orders of magnitude smaller than raw data. The respeaking of raw data is a noteworthy edge case in this dichotomy. Respeaking is primary data associated with the raw data that it is derived from, but it is also raw data that can have primary data assocated with it, such as transcription.

2.3.2 Data levels represented in archives

Based on the charactersitcs of the different data levels defined in Himmelmann (2012), the imperative to preserve these data in archives is strongest for raw data. However, archiving data of the other levels is desirable, especially in the case of primary data, which increase the value and accessibility of archival deposits, as discussed in §2.2.1.

As one example of the distribution of these data levels in a language archive, the Native American Languages

(NAL) archive has a total of 3259 items in its collections at the time of this writing. Based on a survey of these items, conducted by the author in November 2022, 918 of the items contain audio or video recordings, while 174 items (19% of 918) contain corresponding primary data.

The limited number of items containing primary data is concerning in the sense that these data do not have well established methods for being published outside of archives, and moreover the archive can be seen as a primary location to deposit them. However, it also makes sense that these data would be non-existent, the time it takes to develop transcriptions (see Himmelmann, 2018:34), funding constraints, professional obligations of academics to focus on a litany of other tasks, and the feeling that documentation is incomplete (Hall, 2022:86).

To expand on the last point, Hall reports feelings of "guilt, embarrassment, and worse" over his documentary analysis, recognizing that some annotations are "confused, incomplete, or simply wrong." In sharing this, he stresses the importance of transferring data to familiar, legible formats. Here, Hall is arguing for the use of standardized, well described data formats, especially because this can help shed light on weaknesses in analysis, even if this results in negative feelings for the researcher. This thought process can be extended to the practice of archiving. Archives also promote the use of standardized, well described formats for data representation (described further in §2.5), and archiving also sheds light on data by allow them to be accessed by others. This process makes documentary analysis more scientific by allowing claims to be independently verified (described further in §2.5.2). Additionally, applying Hall's assertion, archiving is also better for the researcher because it makes it more likely that they will notice mistakes in their analysis, even if they have negative feelings about this process.

In the case of archiving, the process is designed to be permanent, in the sense that materials should not be edited or deleted once they are deposited. This is for many important reasons, like ensuring data are preserved and citable. However, this means that in addition to asking researchers to make their analyses publicly accessible, both to themselves and others, the archival process asks them to do so by making their analyses permanent records. I can report that through my personal communication with linguistics students and other creators of primary language data, I have observed a similar theme of stress that emerges through the conflict between the negative feelings about incomplete analyses and the perception that the archival process results in making permanent and final claims.

Given the under representation of primary data in archives, as evidenced for example by NAL, it is meaningful to explore ways to increase the rate at which primary data are deposited alongside their corresponding raw data. One such way is through software design, which is presented in Chapter 4. I suspect that the factors discussed previously in this section all have a larger effect on the extent of non-existent primary data in archives than any hypothetical shortcoming of the software applications implemented by language archives. Nevertheless it can be interesting to consider how developments in this software might improve the situation.

2.4 Metadata for language archives

Metadata that describe records of language acts are essential for making those data discoverable and accessible from archives. This section presents an overview of the developments in the broader scientific community and

within linguistics that have motivated the design of metadata schemata in language archives, as well as the types of metadata stored by language archives.

2.4.1 Developments in the broader scientific community

Efforts to standardize metadata used by archives have been undertaken since before it was common for language archives to store and serve data digitally. The Dublin Core (DC) metadata schema was conceived in 1996, and contains fifteen elements which are designed to be broadly applicable to a wide variety of resources (Kunze and Baker, 2007). These elements are: title, creator, subject, description, publisher, contributor, date, type, format, identifier, source, language, relation, coverage, and rights. Each of these elements is optional and repeatable in describing a resource. These elements can be qualified using refinements and encoding schemes. Refinements qualify the element, while encoding schemes qualify the values the element can take. For example, a refinement of contributor could be specifying two elements for creator and depositor, while an encoding scheme could be a controlled vocabulary for language that only allows for languages associated with ISO codes.

Elements in the DC metadata schema are written in the Extensible Markup Language (XML), which, like HTML, uses tags (text enclosed in angle brackets) to create structured text. A tag in XML consists of a name, attributes, and a value. Consider the following example of a DC element in XML:

```
<dc:date xsi:type="dcterms:W3CDTF">2014-06-27</dc:date>
```

In this example, "dc:date" is the name, "xsi:type" is an attribute whose value is "dcterms:W3CDTF", and "2014-06-27" is the value of the element. In markup languages like XML, tags occur in pairs of opening and closing tags, where the name of the tag occurs at the beginning of the opening tag as well as in the closing tag, after a forward slash. Attributes are placed after the name in the opening tag, and the value is placed between the opening and closing tags. Thus, the above example is a DC date element with an attribute that specifies an encoding scheme for dates, such that dates must be supplied in the format specified by that scheme. The value specified uses the format YYYY-MM-DD in order to be valid according to the encoding scheme. Approximate dates and date ranges can also be entered according to this encoding scheme.

In addition to building standards for metadata, the broader scientific community also developed standards for digital archives. The Open Archives Initiative (OAI), which started in 1999, provides a metadata standard for digital archives to follow in order to be interoperable in terms of access (Bird and Simons, 2003a:378). This standard includes two major components: implementing the DC metadata schema (including extending it as necessary), and implementing the OAI Protocol for Metadata Harvest (OAI-PMH). OAI-PMH is a protocol for communication between archives and harvesting services, which are services that pull metadata from various archives in order to index them in a single location. This allows for the creation of federated search services which allow users to search for resources using common metadata categories in many archives simultaneously[1].

[1]See list of OAI registered service providers at: https://www.openarchives.org/service/listproviders.html

2.4.2 Developments among language archives

Since the beginning of the adoption of digital formats and storage for language data, archivists and practitioners of language documentation have worked to develop digital methods for storing descriptive and other metadata for language data. The Open Language Archives Community (OLAC) was founded in 2000 by archivists, researchers, and software developers, with the goals of developing better practices for digital archiving and developing a network of digital language archives that conform to the OAI and participate in metadata harvesting protocols (Bird and Simons, 2003a:376). In their development of better practices for digital archiving, OLAC created a metadata schema specifically to describe the complex and structured data produced in language documentation sufficiently, and in doing so make language data easy to find. Additionally, OLAC provides a harvesting service for language archives that provide metadata using this schema.

The OLAC metadata schema was designed following principles of the Open Archives Initiative (OAI) by extending the DC metadata schema. By utilizing and extending the structured XML format of DC, the OLAC metadata schema provides an open standard that archives can use to ensure metadata meets that standard.

Table 2.1: OLAC metadata fields, as of July 11, 2008 (Simons et al., 2008)

Field	**OLAC refinements**	**OLAC encoding schemes**
Contributor	Role	
Coverage	Spatial, temporal	Points, boxes, periods, countries, geographic names
Creator		
Date	Available, created, accepted, copyrighted, submitted, issued, modified, valid	times and dates, periods
Description	abstract, table of contents	
Format	extent, medium	media type
Identifier	Bibliographic citation	URI
Language		content languages
Publisher		
Relation	conforms to, has format, is format of, has part, is part of, references, is referenced by, replaces, is replaced by, requires, is required by, has version, is version of	URI
Rights	Access rights, license	
Source		URI
Subject		subject headings, subject languages, linguistic fields
Title	Alternative	
Type		DCMI, linguistic, discourse

The OLAC metadata schema (current as of 2008) consists of the fields shown in Table 2.1, along with their refinements and encoding schemes. While metadata standards and XML implementations have evolved significantly since the OLAC standards were published, the semantics of the OLAC metadata fields have remained essentially constant and continue to be implemented by language archives. As of this writing, there are 63 archives participating in the community, and 39 currently comform to the OLAC standard[2]. As described in §2.4.1, refinements modify the mean-

[2]http://www.language-archives.org/archives

ing of the field they are applied to, while encoding schemes restrict the values that are valid for a field. The fields are all optional and repeatable, and can be applied with or without any refinement. Both refinements and encoding schemes are expressed as attributes in the XML tags they are applied to.

Looking at specific fields in Table 2.1, contributor describes people who contributed to the development of the resource. The field can be refined to specify the role the contributor had. The values for role are limited by a controlled vocabulary specific to the OLAC schema, which includes: annotator author, compiler, consultant, data inputter, depositor, developer, editor, illustrator, interpreter, interviewer, participant, performer, photographer, recorder, researcher, research participant, responder, signer, singer, speaker, sponsor, transcriber, and translator. The value selected from among these options is expressed as an additional attribute in the XML tag, while the name of the contributor is the value of the tag.

The coverage field is refined by two DC terms, spatial and temporal, which allow for the specification of locations as well as durations. Encoding schemes for the spatial refinement come from the DC, and include specifying geographic coordinates as a point or a box, as well as controlled vocabularies for countries and geographic names. For the temporal refinement, a scheme from the DC is used that specifies a time interval.

The creator field has no refinements or encoding schemes. It is recommended to use a refinement of contributor rather than this field, unless none of the roles for contributor are as relevant as the meaning of creator, which is a person that had a significant creative contribution to the work.

Dates can be refined to allow for dates for different events to be specified, including: available, created, accepted, copyrighted, submitted, issued, modified, and valid. The encoding scheme used for dates is from the DC, and allows for a range of temporal specificity. For increased specificity, a date and time can be specified, as well as a date and time in a specific time zone. For decreased specificity, approximate dates can be specified, such as a year and month, or a year. Additionally, date ranges are also allowed.

The description field can be refined as either an abstract or table of contents, which are both from the DC. This allows for a general description to be given, using the unrefined field, as well as an abstract or a description of the resource's contents, such as digital files. There are no encoding schemes for this element.

The format field can be refined as either an extent or medium. Extent is used to express the size of the resource. The relevant units for extent vary depending on the resource in question. For example, they could be a duration for audio or video recordings, or a number of pages for text based resources. Medium is the physical or logical carrier of the information in the resource. For digital resources, an encoding schema from the DC is recommended, which is a controlled vocabulary for media types, known also as Multipurpose Internet Mail Extensions (MIME) types. MIME types are different than, but related to, file extensions. Both provide information about the type of information contained in a digital file. However, file extensions are mainly a guide for computers to know how to read files. They are not an authoritative measure of the type of information contained in the file, as, for example, they can be changed by renaming the file, which does not change the underlying information in the file. File extensions can also be ambiguously used by multiple types of files. Thus, MIME types are used to identify file types in a more authoritative and explicit way that is useful for metadata for resource discovery.

Identifier is a field that is intended to be used as is for a unique identifier, which unambiguously refers to the resource. The value should be in the form of a Uniform Resource Identifier (URI) that resolves to the resource (see §2.4.4). This field can also be refined as a bibliographic citation.

The language field has no refinements, and uses an OLAC-specific encoding scheme that limits the valid values for the olac:code attribute of the field to languages with ISO 639-3 codes. This encoding scheme was created to ensure that all ISO codes would be available as selections. The content language field is used to specify all the languages associated with the resource. For example, a dictionary written in English about Cheyenne would have "chy" and "eng" as the values for the olac:code attribute.

The publisher of the resource is specified with a DC field with no refinements or encoding schemes. The value for publisher specifies a person, organization, or service as plain text.

The related field allows for connections to be established in the metadata between different resources, or between resources and standards. The refinements to this field come from the DC and establish the type of relationship being specified[3].

The rights field can be refined with DC terms for access rights and license, and has no encoding schemes.

The source field has no refinements, and uses the URI encoding scheme. This field overlaps semanticaly with the related field using the refinements for being a version of a resource or a different format of a resource. However, it is intended to be used for derivative works where the type, creator and/or title are significantly different to that of the original. Compare this to cases where the related field would be preferred, for example when there are two video files in different file formats that capture the same data, and one file was created by exporting the other file in a different format.

The subject field has no refinements and three encoding schemes. These schemes allow for specifying the subject according to a set of subject headings from the Library of Congress, which is a DC controlled vocabulary, as well as the language and linguistic field that are the subjects of study for the resource, which both use OLAC-specific encoding schemes. The language field refined by subject describes the subject language, in contrast to the content language field, which describes all the languages associated with the resource. For example, an audio recording with spoken metadata in English and a primary text in Cheyenne would have "chy" as the value for the olac:code attribute on the subject language field, and both "chy" and "eng" as the value for the olac:code attribute on the content language field. The subject field is important for language data because it enables searches for resources about a language.

Titles can be specified using the title field, and alternative titles can be specified with a DC refinement. This allows for different titles, for example in different languages. This field has no encoding schemes, because titles are free-form text.

Finally, the type field has no refinements and three encoding schemes. These encoding schemes allow the DC Metadata Initiative (DCMI) type to be specified using a DC controlled vocabulary, as well as the linguistic type and discourse type using OLAC-specific controlled vocabularies.

In addition to developing a metadata schema for language archives, OLAC provides a harvesting service for meta-

[3]https://www.dublincore.org/specifications/dublin-core/relation-element/

data from participating language archives, based on the OAI Protocol for Metadata Harvesting (https://www.openarchives.org/pmh/). This allows users to search for language resources among all participating archives. This is valuable for language archives because many such archives exist and thus there is no central repository of language data. Having a harvesting service allows for centralized discovery of language resources in the context of distributed storage and preservation of these resources. Additionally, since each of these archives operates in different contexts, they will have diverse needs and constraints for the metadata schema they maintain in their own infrastructure. Nevertheless, they can comply with the OLAC and OAI protocols by mapping their internal metadata to conform to the OLAC schema for the purpose of harvesting.

Thus, OLAC provides a minimal metadata standard that can be used by all language archives to allow them to follow the OLAC harvesting protocol, even if they use a different standard for their metadata internally. The International Standards for Language Engineering (ISLE) Metadata Initiative (IMDI), is another metadata standard that is widely used by archives in the CLARIN infrastructure in Europe (Broeder et al., 2010:43). It was developed as part of the DOBES project, and aims to be a more comprehensive metadata system (Conathan, 2011), as compared to OLAC, which was designed to facilitate aggregation of metadata and searchability, rather than to be exhaustive (Paterson, 2021:31). For example, IMDI is built to support the bundling of files into sessions with common metadata that refer to language acts, whereas OLAC uses a flat structure for resources and allows them to be connected using the related field. IMDI allows for more complex structure and detail in metadata, but it is not a suitable candidate for a minimal standard that archives can map their internal metadata to.

In spite of the existence of different standards for metadata schema, all language archives can participate in the OLAC harvesting service by mapping their internal schema to the OLAC schema for harvesting. The Language Archive (TLA), which is located in the Netherlands and hosts the DOBES collections, is an example of such an archive. Internally, it uses an IMDI metadata schema, while it is still harvested by OLAC through exposing an OAI-PMH feed.

Another initiative that encourages interoperability among language archives is the Digital Endangered Languages and Musics Archives Network (DELAMAN), which was established in 2003 to promote interaction among language archives (Henke and Berez-Kroeker, 2016:420). DELAMAN requires member archives to be OLAC compliant, that is, to implement metadata harvesting protocols, or have a plan in place to become compliant.

2.4.3 Types of metadata

Beyond the core elements defined by DC and OLAC, the types of metadata collected and stored by language archives are complex and diverse, but the metadata for which collection is prioritized by language archives can be divided into three main categories: descriptive, administrative, and hierarchical metadata. As the name suggests, descriptive metadata describe the what, who, when, and how of raw data, and many of the most common descriptors of raw data fall into this category. Examples include: title, languages signed and/or spoken, content type, dates recorded and deposited, people involved, geographical location, and associated projects and/or grants. This category of metadata is valid for the raw data regardless of where and how the data are stored and managed. The second category of metadata, administrative, includes metadata necessary for the storage and management of data by an archive or

other organization. These metadata include unique identifiers within an archive, access restriction information, and accession information for libraries. The distinction of these two categories of metadata is especially relevant in the case of transporting data to other repositories, for example when repatriating data to language communities. When transported to such a new context, descriptive metadata remain applicable because they are intrinsic properties of the data, whereas administrative metadata may become obsolete, as they are not intrinsic to the data but rather associated with the context of the original storage location. While the boundary between these two categories is generally clear, one example of an edge case is the access restrictions associated with raw data. Different archives have different schemata to categorize different access levels, and therefore moving data across such organizations may render the specific text of a datum's access restrictions invalid, even if the intent of that text remains crucial to the safe and ethical management of the datum. For this reason, descriptive text such as notes about intended access restrictions can be helpful, as well as proper care given to translating such metadata for new contexts.

So far, the metadata have been described and defined as they relate to a single record. Additional properties of data are manifested by the relationships between different data, and these relationships lead to a hierarchical structure that is represented with additional metadata and/or hierarchical design of databases that manage language data. A raw datum like a recording of a language act can manifest as multiple digital files, for example in a lossless format for posterity and a lossy, compressed format for ease of online viewing. As another example, an audio file can be derived by exporting the audio from a video recording of a language act. Diverse raw data can be associated by virtue of representing the same language act, such as when an audio file, a video file and/or digital photographs are created from different devices that directly recorded the language act. Raw data can be associated with primary data that represent the same language act, which is a common result of transcribing and translating recordings of language acts. Additionally, data that represent different language acts can be associated for various logical reasons, such as when different people recite the same story on different occasions.

Many language archives (such as PARADISEC, ELAR, Kaipuleohone, and NAL) have metadata schemata with three levels of hierarchical structure (Paterson, 2021:35,112,122). The first level is referred to by various different terms by different archives, including files (ELAR), essence files (PARADISEC), and bitstreams (Kaipuleohone), as shown in Figure 2.2. For the sake of clarity, this level will be referred to here as files. The term "files" refers either to actual digital files, or in the case of PARADISEC, to the group of digital files that are manifestations of the same captured raw data (like lossless and lossy versions of a recording). The second level is referred to by various different terms, including bundles (ELAR) and items (PARADISEC and Kaipuleohone), and will be referred to here as items, for the sake of clarity. Items group together files for any of the reasons mentioned in the previous paragraph, and these decisions vary by archive and even by deposit within an archive (Paterson, 2021:35,113,122). Finally, items are grouped into the third level, called collections, for thematic reasons, for example because they were part of the same research project or grant, feature the same language, or were deposited by the same person.

Table 2.2: Comparison of different terms used by different archives to describe hierarchical data

Present analysis	ELAR	PARADISEC	Kaipuleohone
files	files	essence files	bitstreams
items	bundles	items	items
collections	collections	collections	collections

2.4.4 Persistent identifiers and uniform resource identifiers

As mentioned in §2.4.2, a unique identifier is an unambiguous reference to a resource within a particular context. Persistent identifiers (PIDs) are unique identifiers that reliably refer to the resource over time and within a global context (in other words, beyond just a single archive). The maintenance of PIDs predates the digital age. For example, ISBN numbers are PIDs for books (Hakala, 2010).

As digital resources became available on the internet, this created a need for PIDs to be resolvable, or in other words to lead to the resource's location on the internet. URLs are unique locations on the internet, but they lack the trait of being persistent in the sense that the resource they refer to may change over time. Uniform resource identifiers (URIs) are designed to mitigate this process, known as link rot, by being stable over time (McMurry et al., 2017:5). This stability is often achieved by virtue of being maintained by an organization that ensures URIs remain resolvable, including doi.org and handle.net (McMurry et al., 2017:6). Digital Object Identifiers (DOIs) and handle.net Handles are both implementations of the Handle specification for PIDs that are maintained by doi.org and handle.net, respectively. An example of a DOI is:

10.1111/22222

In this DOI, the number 10 identifies the URI as a DOI, the number 1111 represents the organization that has requested the DOI, which usually houses the resource, and 22222 identifies that resource among the resources associated with the organization represented by 1111. Taken together, the sequence 10.1111/22222 is a PID for a resource. In addition to the DOI itself, doi.org maintains the protocol for URIs based on each DOI, such that the URI corresponding to the example DOI is:

https://doi.org/10.1111/22222

Like a URL, the URI resolves to the resource location, however, unlike a URL, it does so reliably over time. Additionally, the URI contains the DOI.

Organizations like doi.org and handle.net maintain the protocols for resolving their respective URIs. Beyond maintaining URIs, there is the task of minting them, or generating new URIs. Handle.net is provides Handle minting directly, while doi.org is a registrar for organizations that issue DOIs, such as DataCite. In each case, organizations that mint URIs charge a fee for this service.

While DOIs and Handles are PIs that are commonly used by DELAMAN archives, they are not free to create. The model used for URIs on the internet is that work is required to maintain links between URIs and the objects they reference, and thus organizations who issue them should be paid to do this work. While this makes sense on

the one hand, on the other hand it poses a challenge for organizations with limited funding who wish to uphold the better practices for managing language data, which include creating the ability to cite resources with references that will always remain valid. While it is true there are other services that software applications rely on which are not free, like web hosting and domain names, these other services often provided internally by the institutions that language archives are hosted by, such as universities. In contrast, DOIs and handle.net Handles are provided by external organizations. Some universities maintain a subscription with handle.net or a DOI issuer, allowing language archives to issue Handles for free or at reduced cost compared to paying for a subscription. However, in the case of the NAL archive specifically, introduced further in §2.8, there is no such subscription available.

2.4.5 Traditional knowledge labels

Traditional Knowledge (TK) labels are a form of metadata that were designed to represent Indigenous methods for sharing knowledge and resources in the contexts where Indigenous knowledge has been recorded and is now housed by non-Indigenous institutions (Anderson and Christen, 2013:111). They provide users with a mechanism to understand how to use resources in ways that fit an Indigenous community's wishes and methods for sharing knowledge. While not legally enforceable, they provide a mechanism to describe what is considered fair and equitable to the Indigenous community that has ownership over resources, thus enabling digital return (as described in §2.2.2). This is in contrast to the legal framework of mitigating loss to copyright holders (Anderson and Christen, 2013:117). In comparison to Creative Commons licenses, which seek to facilitate reuse of materials within the legal framework of copyright law, TK labels are designed to enable Indigenous communities to identify appropriate ways to use their resources that are specific to their communities.

The TK labels are divided into three groups, as presented on the Local Contexts website (localcontexts.org): provenance labels, protocol labels, and permission labels. Provenance labels are used to identify the group who is the primary cultural authority for the resource. For example, the TK Clan label specifies that circulation is subject to conditions related to clan membership, while TK Clan specifies that circulation is subject to conditions related to family membership. The full list of labels in this group is: TK Attribution, TK Clan, TK Family, TK Multiple Communities, TK Community Voice, and TK Creative.

The Protocol labels are used to invite users to respect traditional protocols associated with the use of a resource. For example, TK Seasonal is used for resources with knowledge that is usually only shared at certain times of year, and TK Women General is used for resources with knowledge that is normally only accessed by women. The full list of labels in this group is: TK Verified, TK Non-Verified, TK Seasonal, TK Women General, TK Men General, TK Men Restricted, TK Women Restricted, TK Culturally Sensitive, and TK Secret / Sacred.

Finally, the permission labels indicate what uses are generally approved for a resource and/or what activities require the user to engage directly with the community. For example, TK Non-Commercial is for resources that should only be used in non-commercial ways, and TK Community Use Only is for resources that should not be used by people outside the community or a sub-group of the community. The full list of labels in this group is: TK Open to Commercialization, TK Non-Commercial, TK Community Use Only, TK Outreach, and TK Open to Collaboration.

Each of these labels is designed to be customizable to fit the needs of specific Indigenous communities. To accomplish this, each label is applied with a corresponding free-text statement that explains how the label is being applied. While TK labels show great promise for guiding user access to archival collections about Indigenous languages, they have yet to be widely adopted by language archives.

2.5 Developing better practices in language documentation and data management

As the field of language documentation has developed, so too have descriptions of the science of data management associated with the field. Key developments that inform the outcomes of the present work include the recommendations for data to be portable, citable, and attributable.

2.5.1 Portability

Bird and Simons (2003b) identified value statements related to seven issues with language data management, as well as recommendations for each. These value statements are divided into the issues of content, format, discovery, access, citation, preservation, and rights.

The content of a language documentation should be rich and broad. It should use terminology that is standardized and well documented so that users can understand how that terminology compares to what is used in other documentations. The formats used for language data, as well as the documentation of those formats, should be open and non-propriety, as well as both human and machine readable. The existence of language data as well as its relevance to the user should be made discoverable though the creation of metadata, which meets quality standards including a preference toward the use of controlled vocabularies rather than free-form text. In addition to being discoverable, language data should be accessible to users regardless of which software or hardware they have access to, or if they have access to the internet. Language documentations as well as component language data should be citable in order to give credit to their creators and establish their provenance, and these citations should be durable over time. Language documentations should be preserved for the long-term. Finally, the rights of ownership and access for language data should be made clear and documented.

2.5.2 Reproducibility

A task force of 41 members of the linguistics community convened to discuss reproducibility and the future of linguistics (Berez-Kroeker et al., 2018). Berez-Kroeker (2015) adopts the term "reproducibility" from Gezelter (2009), defined as the ability to reuse data in order to test a previous research conclusion. This is in contrast to the related term replicability, which is the ability to test research claims by recreating the data used. The task force concluded that achieving reproducibility in linguistics can only be possible with proper citation and attribution of language data. They recommended that researchers educate themselves about data management, develop a relationship with

an archive, and critically evaluate their personal reluctance to share data. Additionally, they recommeded academic insitututions like departments, committees, and journals to adopt policies that encourage citation of language data and its attribution to those who create them (Berez-Kroeker et al., 2018:14-15).

Gawne et al. (2017:174), focusing on methodological description as a metric for reproducibility, surveyed 50 dissertations and 50 descriptive grammars from 2003-2013 and found that more than half of them do not describe the location of the data they use. They point out that defining best practice for data management in language documentation has not been enough to mitigate this issue, and suggest that better education as well as a culture shift are needed to do so.

To expand on the results from Gawne et al. (2017), Berez-Kroeker et al. (2017) surveyed 270 articles of nine top linguistics journals from the same ten year period for reproducibility. Berez-Kroeker et al. focus on two metrics of transparency they deem important for reproducibility: transparency about the methods in creating, collecting, and analyzing source data, as well as transparency about source data. They find that more than half of the data have no citation of any kind, and proclaim that "we are in danger of being a social science asking its audience to take our word for it" (Berez-Kroeker et al., 2017).

2.5.3 FAIR principles

The FAIR Guiding Principles for scientific data management and stewardship (Wilkinson et al., 2016) were created as a cross-disciplinary movement to codify better practices in data management in science. The FAIR principles provide a list that echoes many of the recommendations provided by the work described in §§2.5.1-2.5.2. The principles are divided into four categories: Findable, Accessible, Interoperable, and Reusable. Each category is broken down further into items, as described below.

The first major principle is that data and metadata should be findable. To do that, the data should be described with rich metadata, and together these datasets should be given identifiers that persist across time and are globally unique. Moreover, these identifiers should be an explicit part of the metadata describing the data. Finally, the data and metadata should be indexed such that they are searchable on a resource like a website. This principle relates especially to the format and discovery dimensions of portability in §2.5.1, as well as the persistence of citation dimension and the importance of citation for reproducibility in §2.5.2.

The next major principle is that data and their associated metadata should be accessible. This means that once the data are discovered, they can be retrieved by users through a standard protocol. To accomplish this, these data and metadata should first be accessible through their identifier, meaning for example that if the identifier is a URL, that URL leads directly to those data and metadata. Once visited, data should be accessible by common and open source software tools, for example downloading files through a web browser or streaming video through a web browser. Additionally, these protocols should allow for authentication where necessary, for example by logging in to the website as a requirement for accessing data. Finally, metadata should be accessible even when data are no longer available. This principle relates especially to the access and openness of format dimensions in §2.5.1.

The third major principle is data and their associated metadata should be interoperable. This means that the

methods and language for representing the data and metadata are accessible to all, by virtue of being well described and shared openly, as well as being applicable across datasets. This interoperability applies broadly across levels of data and metadata. For example, in developing primary language data like transcriptions, this would including using standards for glossing like the Leipzig glossing rules. This would also include using standards for metadata like the DC standard. This principle relates especially to the content and format dimensions in §2.5.1.

The last major principle is that data should reusable, which is described as the ultimate goal of these principles. Data and metadata should be richly described with many attributes, which enables reuse by making the dataset robust for future unknown uses. This relates to the future uses of data described in §2.2.1, and specifically by Holton (2011). Additionally, data and metadata should be provided with detailed information about where they came from and how they can be used in the future. This principle relates especially to the access and rights dimensions of portability in §2.5.1, as well as the provenance of citation dimension and the importance of citation for reproducibility in §2.5.2.

2.5.4 CARE principles

The CARE Principles for Indigenous Data Governance (Carroll et al., 2021) are a response to the FAIR principles that extends them in order to engage directly with the needs of Indigenous peoples. In addition to adopting the FAIR principles, the CARE principles consist of four categories: Collective benefit, Authority to control, Responsibility, and Ethics.

The first major principle is that systems for data dissemination should be designed in ways that Indigenous peoples benefit from these data. To do so, institutions should support the development of Indigenous institutions that can add value to data use practices. Additionally, institutions should support the use of data by Indigenous peoples, as well as support the use of data in ways that provide equitable benefits to Indigenous peoples.

The second major principle is that Indigenous peoples should have control of processes of data creation, access, and governance. This means informed consent should be obtained in the creation of data, and access to data should be given to Indigenous peoples, and Indigenous peoples should be active participants in developing protocols for data access that are appropriate for their communities. This relates to the participatory archive model discussed in §2.2.1.

The third major principle is that institutions engaging with data from Indigenous peoples are responsible for developing positive relationships with Indigenous peoples, expanding digital literacy among Indigenous individuals, and making sure data created are respectful of Indigenous peoples' worldviews.

The final major prinicple is that the use of data should be informed by the ethics of Indigenous peoples in order to minimize harm and maximize the benefit of this use across time.

2.5.5 Practices in relation to archives

In looking at the better practices presented in §§2.5.1-2.5.4 from the perspective of what an archive can do to follow them, three over-arching categories emerge. From this perspective, the practices focus on processes of data

and metadata coming into the archive (which can also be summarized as depositing), processes of data and metadata going out of the archive (which can also be summarized as access), and beyond that, processes of cultivating relationships related to these data flows. Table 2.3 summarizes these categories for the dimensions of portability described in §2.5.1, and Table 2.4 summarizes these categories for the rest of the practices, described in §§2.5.2-2.5.4.

Table 2.3: Categorizing portability principles as related to deposit, access, or cultivating relationships

Practice	Dataflows into archives	Dataflows out of archives	Cultivating relationships
Portability			
Content			
Terminology	✓✓		
Accountability	✓✓		
Richness	✓✓		
Format			
Openness	✓✓		
Documentation	✓✓		
Machine-readable	✓✓		
Human-readable	✓✓		
Discovery			
Existence		✓✓	
Relevance	✓✓		
Access			
Complete	✓✓		
Unimpeded	✓	✓	
Universal		✓✓	
Citation			
Credit, provenance	✓	✓	
Persistence	✓	✓	
Immutability	✓✓		
Components	✓✓		
Preservation			
Long term		✓✓	
Complete	✓	✓	
Rights			
Documentation	✓✓		
Research		✓✓	

In order to produce the summaries in Table 2.3-Table 2.4, a standard approach was taken, which was by definition over simplifying. The goal was to assign each practice to exactly one of the three categories by determining the primary focus of the practice. If one category could be identified as primary, this category is labeled "✓✓." In the rare cases where two categories were equally primary with no clear winner, each category is labeled "✓." Finally, in the exception that all three categories were equally relevant, each category was labeled "-."

A couple of examples can illustrate further how this classification process worked. First, looking at 2.3, the two components of the discovery dimension are categorized differently even though they are closely related. The "existence" component is classified with a focus on access because the text of this component focuses on the act of listing language resources. In contrast, the "relevance" component is classified with a focus on deposit because the text of this component focuses on the generation of metadata. As another example, consider the first component of

Table 2.4: Categorizing citation, FAIR, and CARE principles as related to deposit, access, or cultivating relationships

Practice	Dataflows into archives	Dataflows out of archives	Cultivating relationships
Citation and Attribution			
Citation	✓✓		
Culture shift			✓✓
FAIR			
Findable			
Persistent identifiers	✓✓		
Rich metadata	✓✓		
ID in metadata	✓✓		
Searchable		✓✓	
Accessible			
Retrievable	✓✓		
Metadata		✓✓	
Interoperable			
Language	✓✓		
Vocabularies	✓✓		
References	✓✓		
Reusable			
Rich metadata	✓✓		
CARE			
Collective Benefit			
Inclusive development			✓✓
Governance and engagement		✓✓	
Equitable outcomes	-	-	-
Authority to Control			
Rights and interests	✓✓		
Data		✓✓	
Governance of data	✓✓		
Responsibility			
Relationships			✓✓
Capability and capacity			✓✓
Indigenous			✓✓
Ethics			
Benefit v. harm	✓✓		
Justice			✓✓
Future use	✓✓		

the Accessible principle and the third component of the Ethics principle in Table 2.4. While both are about access, they focus on the creation and management of metadata needed to enable this access, which is essential for deposit.

One clear pattern emerges from this summary. Specifically, the practices focused on depositing with archives are described with more specific actions for how to enact them, and it is clear to point to ways in which archives have worked toward enacting them. As a primary example, many of the practices classified as deposit make recommendations about the generation of metadata for data deposits, and archives have developed sufficient and complex metadata schemata to provide for such metadata. Additionally, archives have implemented things like persistent identifies and access restriction levels. Finally, institutions have developed requirements for stakeholders to archive either in order to earn doctorate degrees or receive funding.

In contrast, access and relationship building are necessarily more open-ended in nature. Access principles like the

access dimension of portability and the final component of the Ethics principle call for access by future users whose needs are not predictable. Of course, some access principles are concretely defined and have been implemented by archives, as in for example the use of standard protocols such as OLAC and OAI-PMH to harvest and index searchable metadata. However, there is still an overall difference in these categories such that access principles are more open-ended in interpretation and enactment. Additionally, the cultivation of relationships is inherently open-ended and context specific.

From the contrast of the three categories presented in this summary of principles for better practices, we can identify access and cultivating relationships as areas of interest for developments among archives. The principles in the depositing category were established earlier, stem from a longer history of institutional archives, and are more likely to include specific actions that have been taken by many language archives. In contrast, the principles for access and relationship building are newer, offer fewer concrete actions, and are inherently more dependent on the context in which an archive operates.

It is nevertheless important to continue to uphold the depositing principles, and the way these are enacted in archive software is explored further in the next section. Multiple approaches toward achieving the principles of access are pursued, including planning for the allotment of resources to the development of a user access portal, described in Chapter 4, and implementing a machine-readable access portal, introduced in §2.7.3 and described in Chapter 4. Finally, striving for the principles of cultivating relationships is discussed further in Chapter 3.

2.6 Recent developments in language archives

This section presents recent developments in language archives and their digital infrastructure using one illustrative example, which is the redesign of the digital infrastructure for the Archive of the Indigenous Languages of Latin America (AILLA), which took place in 2015. AILLA's archive manager, Dr. Susan Kung (personal communication), provided details of the features requested by AILLA for this redesign.

Following the categories defined in §2.5, the features requested in the redesign can be categorized as focusing on either bringing data into the archive (deposit and data management) or getting data out of the archive (discoverability and access), with a couple exceptions falling into a third category of user interactions with the website.

In terms of deposit and data management, AILLA requested: metadata that links objects in the hierarchical levels of collections and files; support for custom file naming conventions, a depositor portal that includes support for uploading large files, and an extension of the website's Spanish language localization to the depositor portal.

For discoverability and access, AILLA requested: search for collections, sorting search results, viewing files by file extension, displaying language alternative names, font support for diacritics, audio and video streaming, a map of speech communities through location metadata, and the ability of users to bookmark search results.

Finally, two features requested that span both deposit and access or pertain to general use of the archive website are: the ability for users to retrieve and reset their account passwords, and a social media feed on the homepage.

Additionally, while the design is now completed, AILLA has a list of currently desired features it would like to

implement in the future, including: advanced search, batch-ingestion for metadata, bulk download of media files and metadata, bookmarking search results (which was not developed in the last redesign), and localization in the Portuguese language.

Some of the features requested by AILLA motivate discussions with archive users presented in Chapter 3 about what features they are interested in, including: search for collections and custom views of search results, advanced search, audio and video streaming, the map of speech communities through location metadata, and a social media feed on the homepage.

Additionally, many of the requests motivate features included in the design for digital archive infrastructure presented in Chapter 4, including: metadata relating data levels, the depositor portal, various forms of search, audio and video streaming, batch-ingestion for metadata, and bulk download of media files and metadata.

2.7 Aspects of software design

This section describes some aspects of software design that inform the software design for an archive presented in Chapter 4. These include modeling data flows in software applications, which leads into application programming interfaces, as well as user experiences, open source software projects, and version control systems.

2.7.1 Dataflows through archives

Modeling data flows through software applications can be idealized in terms of three steps: data in, data processing, and data out. Language archives with a digital presence through a website powered by a web application can be modelled in terms of these three steps of data flows. Generally, a language archive has a database that includes language data and metadata. Some of these metadata describe the inherent properties of the data, and some describe how these data should be treated and disseminated appropriately. The output data are web pages that display the metadata and either display the data or allow the user to download the data. The input data include user inputs like search queries and requests for data and metadata, as well as the data and metadata themselves. In the latter case, some archives may have an interface only for authorized archive personnel to input these data, while other archives may have a public user interface for depositors to input data.

These interfaces that present information from the database to the user are called user interfaces (UIs). Not all software applications need user interfaces, but many have them, and they often come in the form of web pages. While the information used by the application is stored opaquely in a database, for example an SQL database, UIs are web pages with text elements that can display information stored in the database to the user, and form elements that can take information from the user and store that information in the database. In the former case, the web pages combine data with formatting elements to present the data in a human-readable form. In the latter case, the web pages use data validation techniques to ensure that the human user has provided information in a way that is machine-readable, for example a standard structure for dates.

Another way that data can flow through software applications, including digital archive infrastructure, is through

microservices. Microservices are small applications that can be developed and deployed independently of each other, and that communicate with each other using messaging protocols (Nadareishvili et al., 2016:7). In this context, applications are considered small relative to an alternative design of one monolithic application that accomplishes all the functions of a set of microservices. A microservice does not need to consist of the full set of components that would make up a monolithic web application, including a database and a UI that allows humans to move data in and out of that database. One example of a configuration for software tools that would be a microservice is software that runs on a web server like a web appication but does not use a database, because it retrieves information from another application instead of storing the data itself. Another example of a microservice, which is even smaller than the previous example by virtue of having even less of the components of a web application, is any section of code that is atomic by virtue of enacting the three steps for data flows, including: logic for processing data and a protocol for input and output of data to and from that processing step. Such a microservice can be designed to be easily integrated into web applications, so that developers of web applications can achieve the functionality provided by the microservice with minimal software development of their own.

2.7.2 Repository management systems

Repository management systems are a type of software application that are well suited for implementing digital language infrastructure. They provide functionality and features that archives commonly need, like bundling data in items with files, allowing users to make deposits of data with this structure and their corresponding metadata, associating contributors with their ORCIDs, presenting metadata for items and files in web pages, enabling metadata harvest, and allowing the data deposited to be openly accessible or have restrictions to their access.

One example of a repository management system is InvenioRDM[4], which is designed to be a turn-key repository management system. Turn-key here means the software package is designed to require few steps to install and deploy. InvenioRDM is written in Python and uses the Flask web development framework. This software package is a variant of the Invenio software project, along with InvenioILS that focuses on turn-key functionality for libraries. The Invenio project begin in 2019 to provide an open development release of the software that was build by CERN to create the Zenodo archive website[5]. This means that the Zenodo archive, which serves the broader scientific community, is powered by a software application that is almost identical to InvenioRDM. Currently, there are no DELAMAN archives that implement this software package for their digital infrastructure.

Another example of a repository management system is Fedora Commons[6]. Rather than being designed with turn-key functionality in mind, it provides different software modules that can be configured and deployed together to meet the needs of different use cases. Fedora Commons is an open source and open development project that begin in 2003. Currently, multiple DELAMAN archives implement this software package for their digital infrastructure, including AILLA and The Language Archive.

[4]https://inveniordm.web.cern.ch/
[5]https://zenodo.org/
[6]https://fedora.lyrasis.org/

2.7.3 Application programming interfaces

The model for software applications described so far in this section includes a database, processes to move data in and out of the database, and interfaces with human users for input and output of data. However, in reality, software applictions do not exist in isolation from other applications, and a modification to this model allows applications to interface with each other, so that data out for one application is data in for another. This is done through the use of an application programming interface (API). APIs allow applications to communicate with microservices as well as other applications, and an API is used to implement the messaging protocol used by a microservice, as described in §2.7.1.

An API is simply an interface that an application provides to the data in its database that is designed to be machine-readable and to be accesses by other applications. As an example, take an archive that provides a web page for each item in its collections. This webpage has metadata fields displayed in a predetermined format and links to download data files. These web pages provide a human-readable UI for the item. An API provides a system for obtaining the same information in a machine-readable format. Requests for data through either interface are made through the same mechanism, which is URLs. For example, the human user could make a request by visiting the URL languagearchive.org/collection/itemID, where "itemID" is a unique identifier for the item. Based on this request, the server would return a web page file and the browser would display this file to the user. An API request is also made via a URL, for example languagearchive.org/api/collection/itemID. Upon receiving this request, the application returns the same data, but as structured text, for example an XML or JSON document. This request is designed to be made by another application, which accepts the structured text and extracts the data for use in its own processing steps.

This approach of creating interfaces between applications has provided lots of different kinds of opportunies for developers. One such opportunity is to increase efficiency by not rewriting code in each application, but using the results of other applications. For example, there are openly accessible APIs for general data such as information about countries. Any application that wants to be able to gather information about a country (such as location, population, geographic size, etc) without having to maintain this information it its database can query this API. There are also open APIs for language metadata for all languages with ISO codes. APIs can also be used for harvesting and indexing metadata. To do this, an archive can have an API which provides a URL that allows a harvesting application to request the metadata it requires. As a final example, APIs enable the use of microservices. When data are accessible through APIs in machine-readable formats, a small isolated piece of code can receive its input data via a request to another appliction's API and output its own data via its own API. Custom UIs can themselves be an example of a microservice enabled by APIs, in that a web application need not manage data in its own internal database, but instead it can retrieve data from another application through an API and then display these data as part of the UI it generates.

2.7.4 User experience

User experience (UX) has become a common phrase in the software devolopment industry, and is defined as the experience a software product creates for people who use it (Garrett, 2011). UX describes only the user's interaction with the software, and not its internal workings or explicit functionalities. Because of the nature of software, much of the user's experience is through a user interface (UI), which includes all the surface features of webpages, like links, buttons, forms, layouts, fonts and colors, as opposed to the the data that those webpages are populated with, and the logic used to populate them. Even when software is designed to accomplish well-defined functionalities, the "correct," or preferred user interface is determined by the psychology and behavior of the users (Garrett, 2011:8). Users tend to blame themselves when they are unable to use technology, so that software with a poor UX drives away users even if its functionality is of a high quality (Garrett, 2011:10).

In order to better define the term UX, Law et al. (2009:719-720) asked 275 researchers and practitioners from academia and industry to rate a series of statements about UX on a five-level Likert scale ("strongly disagree", "disagree", "neutral", "agree", and "strongly agree"). The results showed that UX researchers and practictioners agreed that a user's experience is not only subjective, but dependent on their social groups and community, as well as their relationship with the orginization providing them with this experience (Law et al., 2009:726).

Nathan (2015) provides multiple suggestions for improving aspects of UX when interacting with language data and language archives. Firstly, interfaces should be provided in the lingua franca(s) of target communities (Nathan, 2015:60). While the process of creating such localizations is resource intensive, archives can strive to achieve this where feasible. In many cases, the inclusion of a localization for an additional European language beyond English, such as Spanish for an archive with an areal focus on Latin America, is a cost effective step toward achieving this goal. Additionally, the ability to browse collections based on textual metadata and maps is important, beyond simply being given a search box (Nathan, 2015:60). Nathan focuses on "multiple content rendering," or the idea of presenting the same language data in different ways depending on the audience and desired experience. For example, videos can have subtitles in different languages, linguistic transcriptions and annotations can be provided in more or less detail, and texts can be presented in easy-to-read fonts or printable PDFs. As a UI feature that can be implemented without modifying underlying databases and data storage systems, multiple content rendering provides a relatively inexpensive way for archives, which operate with limited resources, to increase ease of access to and dissemination of language data.

2.7.5 Open source software

Open source software is software whose underlying source code is distributed freely for anyone to read and use. Open source software is a prerequisite to an open development model where software developments are published to open source repositories that track these changes, documentation of how the software is designed is prioritized, and contributions can be made by anyone who is willing and able. Not all open source software is developed this way, and simply publishing code in a public online repository does not lead to development under this model. An

open development model is in some sense analogous to academic models of scientific research, where results are shared so that others can make contributions and provide feedback to improve upon those results (Bonaccorsi and Rossi, 2003:8). In terms of software, this feedback results in bug fixes and maintainence which can persist as long as the software has utility to its users, in contrast to a proprietary model where developers make unilateral decisions on when and if to make changes, often motivated by funding or profits (von Krogh and von Hippel, 2006:979). Moreover, these benefits occur even though most projects have a very small minority of contributors making the vast majority of contributions, and much of the engagement is by passive observers of changes (von Krogh and von Hippel, 2006:979). Within the language documentation community, Bird and Simons (2003b:579) call for an "open source revolution," where this model can facilitate agreement on data formats and linguistics types and thus increase interoperability among archives.

2.7.6 Versioning and version control

Version control is the process of managing different versions of information (O'Sullivan, 2009:1). In its simplest form version control is a manual process to track changes, such as keeping different versions of a file and renaming them with a naming convention like "file_v1, file_v2, etc" or "file_2021_01, file_2021_02, etc." Manual version control puts the burden on the user, and is thus prone to mistakes. Additionally, manual version control protocols can create unexpected issues with data management, for example when an archive deposit contains multiple versions of an annotation file with different names (e.g. somename.eaf and somename_v2.eaf) that depend on an audio recording with the same name (somename.wav and a non-existent somename_v2.wav), but only a single recording exists (Babinski and Bowern, 2021).

Version control can also be accomplished with software that manages versions automatically with little input from users. Two common methods for software to accomplish versioning are maintaining each version of files, and maintaining only the differences between files so that previous versions can be reconstructed when necessary.

The first type of software version control involves maintaining different versions of files in addition to metadata about which sets of files constitute different versions. For example, if a versioned repository had five files named 1.eaf, 2.eaf .... 5.eaf, and the file 2.eaf was replaced with a new version, the version software would then contain six files, including both versions of 2.eaf, and metadata that explain there are two versions of the repository overall, as well as which files belong to each of this versions. In this case, all files besides those associated with 2.eaf belong to both versions, while the first version of 2.eaf belongs to the first version of the repository, and the second version of 2.eaf belongs to the second version of the repository.

Another, more abstract way to manage versions of files is to maintain only a single version of each file in the repository, and additionally maintain records of each change to each file, so that every version of every file can be reconstructed, if desired by the user. This process is efficient for plain text files, such as ELAN files, in which the files can be broken down into digital pieces that represent the text characters in the file, and these files can by parsed piece by piece, in a linear fashion. In such a scenario, changes can be defined as the removal and addition of chucks of theirs pieces, which is a parsimonious way to store this information. This parsimony makes the algorithm efficient

and scalable for text files. In contrast, this algorithm is not efficient for media files, because their digital pieces cannot be parsed piece by piece, so that the smallest possible change leads to storing the entirety of both copies of the file in the historical data.

Maintaining changes to files is more efficient, especially when sharing the entire version history among collaborators, while maintaining whole files for each version is less abstract, because it relates more closely to the way a human user would implement version control without software tools to manage this.

When considering the use of versioning in language archives, it is important to consider which of the types of versioning systems would be the most effective compromise between the characteristics of versioning that would be useful and the resources required to implement such a solution. This cost of implementation can be mitigated if a sufficient versioning system is integrated into software packages used by the archive.

The field of language documentation can benefit from the use of version control of language data because this research "is a long-term process of accumulating and revising knowledge" (Widlok, 2013:186), and with the passage of time more information may become available which enriches archived resources or reveals errors (Garrett, 2014:70). Bird and Liberman (2001:54) also call for the use of annotation systems that allow for maintenance of annotations, which can be accomplished with version control systems. One example of an archive that implements a versioning system for deposits is Zenodo, which is an archive that does not focus specifically on language data.

2.8 The Native American Languages archive

The Native American Languages (NAL) archive is a language archive housed at the Sam Noble Museum at the University of Oklahoma, and is a DELAMAN member archive. Linn (2014:53) describes NAL as community-based archive, in that it maintains and disseminates documentation that is conducted for, with, and by language communities. It was started in 2003 from approximately 130 donations, and has ongoing collaborations with organizations and collections such as the Wichita Language Project collections, Lenape Talking Dictionary, Chahta Anumpa Aiikhvna, Muscogee Nation language projects, the Chicksaw Nation language collections, the Marcia Haag collection, the Jonathan Amith Mesoamerican collections, the Sky Campbell collection, the Alaina Tahlate collection, and the Nebraska Indian Community College.

Since its founding, NAL has engaged in digitization efforts for its collections, and currently with born-digital materials it contains about 9,000 items in 175 Native North American languages[7] totaling about 12 TB of data. NAL actively takes requests for materials, through which it receives about 200 in person visits per year and provides about 2,000 copies of items per year.

NAL does not currently provide access to its collection online, and thus requests for data are made in person or via email and are honored manually by archive staff. The archive has received two NEH grants[8] for the planning and implementation of a project to bring collections online. One of the outcomes of these grants was that NAL conducted a series of workshops to gather feedback from users on what they want online access to look like, and the results

[7]https://samnoblemuseum.ou.edu/collections-and-research/native-american-languages/

[8]NEH grant numbers: PW-269366-20 and PW-285221-22, to University of Oklahoma; Principal Investigator: Raina Heaton

of these workshops are presented in Chapter 3. Another outcome was that a plan and design was made for digital archive infrastructure to accomplish online access, and this plan and design is presented in Chapter 4.

2.9 Chapter summary

Language archives have played a crucial role in achieving better practices in language documentation (§2.2), and they continue to have this role while operating with limited funding. These practices include making language data accessible and facilitating their dissemination to the appropriate stakeholders, as well as allowing research claims made about language data to be reproduced through this accessibility (§2.5).

In the context of this success, there are desiderata for further improvements to the processes of archiving and language data management, including increasing the rate that primary data are archived alongside their corresponding raw data (§2.3.2) and granting language communities more control over the management of their data where this is desirable and feasible (§2.1). Additionally, models for the development of modern software tools for language data management risk isolating language archives by limiting their integration into the workflows of these models (§2.2.3).

Various frameworks for better practices in language documentation have been proposed, and their principles include: providing rich metadata with data in well described and sufficiently standard formats, making data discoverable through the use of these metadata, software interfaces to index and search these matadata, preserving data for the long term, providing access to these data both as openly as possible, and with as many restrictions as necessary to honor the needs of data creators and Indigenous stakeholders, facilitating the citation of data in research and the attribution of its authors, and empowering Indigenous peoples to contribute to determining how data are created, participate in the governance of data, and benefit equitably from the use of data. Principles related to deposit describe more concrete actions that have more clearly been implemented by archives, while principles for access and cultivating relationships describe less concrete actions, are more context dependent, and have less clearly been implemented by archives. Thus, access and cultivating relationships are identified as areas of focus for future development among archives (§2.5).

Recent developments among archives echo the pursuit of these better practices, as for example with the redesign of AILLA's digital archive infrastructure. AILLA sought to improve their infrastructure for deposit and management of data through improvements to their metadata schema and depositor portal, as well as enabling batch-ingestion for metadata. Additionally, they sought to improve the access they provide through improved features for searching collections and displaying search results, audio and video streaming of data, bulk download of media files and metadata, and an interface to display mapped location information for data (§2.6).

Software applications can be modeled as having data coming in, processing that data, and having data going out. These data are stored in a database, and there is a user interface both to bring in data and take out data, usually in the form of a series of webpages (§2.7.1). APIs are interfaces to applications and their database, designed for applications to retrieve data in machine-readable formats from other applications. The use of APIs can increase

efficiency, enable metadata harvesting, and enable custom UIs and other microservices (§2.7.3). One advantage of applying custom UIs as microservices is the ability to create different UXs to meet the needs of different user groups while requiring less resources to deploy these microservices than if they were each part of standalone applications. Still, UX is a subjective measure of user satisfaction that depends highly on user perceptions of and relationships with the organization providing the software (§2.7.4). This highlights the importance of cultivating relationships presented in §2.5.

In addition to using microservices built off APIs, another way to mitigate the amount of resources required to implement a software application is to use open source software packages that accomplish many of the functions needed for the software application. InvenioRDM is such an open source software package for the application of archives because it is designed for repository management (§2.7.5). InvenioRDM provides features beyond repository management, including versioning of deposits (§2.7.6), which can facilitate iterative archiving and thus encourage depositors to come back to deposit primary data. Utilizing this feature in this way is presented in Chapter 4.

This chapter established some areas of interest for developments in language archives, including increasing the quality of access to archived data, cultivating relationships with stakeholders of language documentation and archiving projects, and increasing the amount of raw data in archives that has associated primary data available. NAL is an archive that recently began a project to increase access to its collections, (§2.8), and the rest of this dissertation presents some of the outcomes of that project in the context of the areas of interest for development presented in this chapter. Chapter 3 presents workshops with users that were designed to cultivate relationships with archive users and center the design of digital archive infrastructure around their feedback.

Chapter 3

User-centered design

Chapter 2 (§2.5) identified the process of cultivating relationships as an important area to focus on while developing digital archive infrastructure. Since 2020, the NAL archive, introduced in §2.8, has been engaged in a project to provide online access to its collections. This project has engaged with the process and principles of cultivating relationships through user-centered design (Wasson et al., 2016:643), meaning that NAL has sought and will continue to seek feedback from its user base throughout this project.

NAL hosted a series of workshops with members of its user base and community in 2020, as an initial phase of gathering feedback at the beginning of the development project. This chapter presents some of the findings from these workshops, as well as how they fed into the planning and design of digital archive infrastructure presented in Chapter 4.

§3.1 presents the context in which these workshops took place, including the format of the workshops and the questions posed to attendees. §3.2 presents an overview of the responses to the questions posed, and §3.4 presents my reflections on these results. §3.5 presents key takeaways from the results that are relevant to the development project.

3.1 Context

As described in §2.8, the NAL archive currently does not provide online access to its collections. However, NAL has been engaged in a project to develop digital archive infrastructure that provides online access, and this project has been funded through two NEH grants[1]. One of the tenets of the project established early on and while applying for funding is the use of user-centered design, which means ensuring the development is carried out with feedback from the user base at various stages of the development. In this context, NAL hosted a series of workshops in 2020 to seek initial feedback from its user base about what they would like to see from the NAL website and how they expect online access to be implemented. This was the first phase of seeking feedback for user-centered design, which by design occurred before there was a website for users to interact with. Subsequent stages of gathering feedback

[1]NEH grant numbers: PW-269366-20 and PW-285221-22, to University of Oklahoma; Principal Investigator: Raina Heaton

are planned, in which the users will have opportunities to interact with the digital infrastructure and website and provide feedback on what has been developed, before the project concludes.

3.1.1 Workshop format

The format of this event had to be modified due to the circumstances of the global pandemic that was taking place at that time. The event was originally planned as a single workshop, to take place in person in Oklahoma, and give users an opportunity to interact in person and discuss what they wanted to see as outcomes of this project.

Instead of using an in-person format, the event was reorganized as a series of workshops that took place virtually. Having multiple workshops gave NAL the ability to accommodate the different schedules of attendees, and having more events with fewer people each allowed for intimate discussions given the constraint of meeting in a virtual chat room.

3.1.2 Questions

During each of the workshops a series of questions was posed in order to elicit opinions, feedback, and encourage discussion among attendees. The questions were asked in one of three formats. In the first format, a question was posed and the floor was immediately opened to attendees to begin a discussion. The second format included a multiple choice question, which was administered through the Zoom meeting interface. Workshop attendees were asked to choose an answer to the question using the on-screen interface, and after all answers were in, the results were shared, and the floor was open to discussion. The final format was used for exactly one question. The question was posed in the context of a group activity that took place during the workshop. Attendees were split into breakout rooms and asked to try out an archive website together and observe what they liked and disliked about that resource. Afterwards, attendees reconvened in the main room and summarized their discussions. In addition to the direct answers to the multiple choice questions, the answers provided by attendees in the open discussion segments were recorded through note taking.

In addition to the three types of questions posed at the workshops, feedback was sought from participants after the workshops in two different ways. First, a series of five questions was sent to participants as a text document after they completed a workshop, for them to answer in written form and return to us at their convenience. Second, wire frames were designed after the workshops concluded, which incorported participants' responses during the workshops, and these were sent to participants to solicit their comments and feedback.

The questions asked can be divided into six categories by subject: search and browse, access, community-specific metadata, co-curation, digital return, and summary questions. The questions are presented below, organized by subject, with reference to their format type.

3.1.2.1 Search and browse

The first subject, search and browse, asks participants how they would like to search for collections, and if and how they would like to browse through collections. The basic premise of search in terms of the digital infrastructure is that there are various metadata fields that can be queried with a text string for partial or exact matches, which leads to an output of collections, items, or files that match that query. The UI can highlight certain metadata fields for search over others by providing form fields to search only specific metadata fields, in what is often referred to as an "advanced search." Additionally and alternatively, the UI can search many metadata fields simultaneously with one query to find matches across all terms, and this is often referred to as a "basic search." While the basic search is computationally more complex, the terms "advanced" and "basic" refer to the user experience (UX), defined in §2.7.4, in which advanced search normally provides many form fields while basic search provides a single form field for users to interact with.

In contrast to search, the basic premise of browse is that the UI includes curated lists of resources based on predefined search criteria, for example providing a list of results that match queries for any given language. In this UX, the user is browsing collections, items, and files by language. While the query to produce browse results is fundamentally the same as with search, the UI is designed to highlight the experience of browsing through languages, for example by providing a list of languages that the user can click on, rather than asking them to type a language into an empty form field.

The questions generally do not assume such technical knowledge of UIs, but rather that participants have experience using websites with search and browse capabilities in some form. The first question in this category uses the open discussion format, and elicits common preferences for finding materials through browse and search.

1. When using a language archive, describe how you typically search for materials. What kinds of things are you usually looking for, and how specific do you get with the search criteria?

The second question provides a more concrete follow up to the open discussion format by providing a list of metadata fields for search and/or browse and asking participants which ones they would like to have available. This question is in the multiple choice format, where multiple answers are allowed.

2. Take a look at the following list of features. Please mark ALL those you'd like to see developed for NAL (By contributor, by date created, by dialect, by language family/subgroup, by map/location, by collection, by file type, by genre).

The next question asks specifically about if participants are interested in browse capabilities, and if so using which metadata fields. It does so using the open discussion format in order to allow participants to answer creatively on which types of browse are important to them.

3. Would you like to see pages with pre-assembled sets of resources (so far a page with resources grouped by language, a page with all the grammars and dictionaries, and a page with teaching materials, a page with

songs/music, and a page with language planning materials)? Any suggestions for other useful topics for browsing?

The next question is an open discussion question that asks about a related type of metadata, which is personal metadata for contributors to collections. The goal was to gauge how interested participants are in displaying personal information about contributors on the website.

4. Would you like to see pages for people represented in the collection, with a short bio, picture, and list of items they are affiliated with?

The final question in this category was in the breakout format, where participants entered small breakout groups and were asked to view one of the three archive websites and discuss anything they liked or disliked about the resource.

5. Check out one of the following websites for other language archives:
   The California Language Archive (https://cla.berkeley.edu/)
   Alaska Native Language Archive (https://www.uaf.edu/anla/)
   Computational Resource of South Asian Languages (https://corsal.unt.edu/)
   Is there anything that you've particularly liked about the websites for other language archives? Disliked?

3.1.2.2 Access

The access category includes questions about accessing data in the collections once it is discovered by users through search and/or browse.

The first question gauges participants' interest in having access to data in different formats, specifically small easy to download access copies of files versus large archival quality files. This question uses the multiple choice format with yes and no answer options.

6. In addition to small, easily-downloadable files, would you be interested in having very large, archival-quality files directly available for download? (The other option is that large, high-quality files could be requested and sent separately).

The next question is also about formats for displaying data, in this case primary data. Specifically, this question asks if participants are interested in being able to view transcription files directly in the browser. This question uses the multiple choice format with yes and no answer options.

7. Would you like to see us develop (in conjunction with language programs) a separate web-based viewer for the transcripts and translations in NAL? See for example here: http://northernpomolanguagetools.com/texts/story.php?story=3.

The final two access questions were asked in the follow-up questionnaire sent to participants to answer in writing. They are general questions to gauge participants' interest in having NAL provide online access to its questions, and

they were designed to give participants a private, safe space to express any positive or negative feedback about this development.

8. Are you excited about the prospect of being able to access the language materials in NAL online? What do you see as the benefit to your community?

9. What apprehensions do you have (if any)?

3.1.2.3 Community-specific metadata

Questions about community-specific metadata were geared toward Traditional Knowledge (TK) labels, described in §2.4.5, which are a type of metadata developed to give Indigenous peoples more control over their digital cultural heritage (Anderson and Christen, 2013:106). The first question asks if participants would like NAL to implement TK labels, and the second question asks if participants would be interested in using TK labels themselves when depositing with NAL. Both questions were formatted as multiple choice questions with yes and no answers.

10. Would you like to see NAL put TK labels (https://localcontexts.org/tk-labels/) on some or all of the collections?

11. If you are a depositor, would you be interested in collaborating with us to add TK labels to the materials in your collection?

3.1.2.4 Co-curation

The next category of questions is co-curation, which is a type of information gathering for collections where information for collections is created or provided by people other than the original depositors. For example, language community members could see a recording of another community member and have information about the recording's context or contents that they would like to provide to be included in the collection. This has implications for the governance and management of collections in the archive, and these questions were aimed at eliciting a discussion about co-curation using different mechanisms in order to gauge participants' interest and reactions.

The first question in this category was the most general, and was posed as an open ended discussion question about allowing users to comment on materials.

12. What do you think about allowing visitors to the site to comment on materials in the collection?

The next two questions ask participants to imagine that NAL has a mechanism in mind to receive information to augment exist metadata through co-curation. The first question asks whether they would be willing to participate in this co-curation by vetting the information received, and is formatted as a multiple choice question with yes and no answers. The second question is a multiple choice question for those who answered no to the first question, and asks who they think should vet this information.

13. As mentioned in the background document, we are developing a way for knowledgeable users to submit metadata to items in the collection. That information will need to be vetted before it goes into the official

record. The question is, would you be interested in approving/rejecting any suggested additions/changes to the information for your collection that come from users of the website?

14. If you answered NO, who do you think should vet those? (Someone from the tribe, Someone I appoint, NAL staff, Tribal Entity, other)

The last two questions ask about co-curation via a less controlled mechanism than providing metadata, specifically through allowing for general comments on the website. The first question asks if this is something participants would like to see developed, and the second asks how they imagine it would look. Both questions are formatted as multiple choice, with the first question having yes and no answers and the second having a list of possibilities.

15. In addition to allowing users to add catalog information about each item, would you be interested in seeing us develop a place where people can add general comments/discussion?

16. If so, how do you think that should look? (Separate downloadable file, twitter-style feed, box at bottom)

3.1.2.5 Digital return

There was one question about digital return that asked if participants have ideas about if their community would benefit from repatriating materials, and if so, where they might see that taking place.

17. Thinking about your community, could they benefit from having a local copy of NAL resources? Where do you think we could potentially put a computer with collection materials?

3.1.2.6 Summary questions

The last category of questions contains three questions that either summarize topics of the workshop or ask for general feedback. They were given to participants in the written questionnaire that followed their participation in the workshops.

18. What topics/features discussed during the workshop are the most important to you/do you think will be most useful?

19. Is there anything you can think of that you would like to see developed wasn't mentioned in the workshop or this questionnaire?

20. Any other comments or things you'd like us to know?

3.1.3 My involvement

I participated alongside NAL staff in planning and running these workshops. I was not involved in designing the format or choosing who to invite, however, I participated in designing the set of questions to be asked during the workshops by giving my feedback on their content and format. Additionally, I was a note taker during all of the workshops.

3.2 Results

The results from each of the questions are summarized below. Quantitative answers are given graphically and qualitative answers are paraphrased. The questions are presented in the order in which they were introduced in §3.1.2, and each question is followed by a summary of its results.

3.2.1 Search and browse

1. When using a language archive, describe how you typically search for materials. What kinds of things are you usually looking for, and how specific do you get with the search criteria?

    - Several people only search by language name
    - Search for 'texts'
    - Search by people's names or family names
    - Search for related languages
    - Search by dialect or band/clan
    - Search by location/map
    - Filter by gender, or other speaker characteristics
    - Filter by date range
    - Get notified, or have a dedicated place to look for, new materials when they become available
    - Something like Amazon's "people who searched for X also searched for Y" function to facilitate discovery

The answers to Question 1 give clues as to what metadata fields are important to focus on when highlighting search and browse capabilities in the UI. Specifically, language is an important metadata field along with some basic fields for contributors like name and tribal affiliations. Language family relationships and geographic location are two types of fields that were highlighted as important, which lead to more complex search queries. While more resource intensive to develop, they can also lead to UI features that users are excited about, like seeing items on a map that are related to the current item by virtue of geographic location. Finally, the "Amazon-like" search for related items would be resource intensive to develop, in part because it would require more metadata about previous searches, and practically speaking, these metadata would need to be sufficiently anonymized before storing in the database. Thus, this is targeted as a low-priority feature of interest.

2. Take a look at the following list of features. Please mark ALL those you'd like to see developed for NAL (By contributor, by date created, by dialect, by language family/subgroup, by map/location, by collection, by file type, by genre).

Figure 3.1 shows the results of naming specific metadata fields that participants might be interested in for searching and browsing. Being able to search by genre came out as the most requested feature, followed by file type and

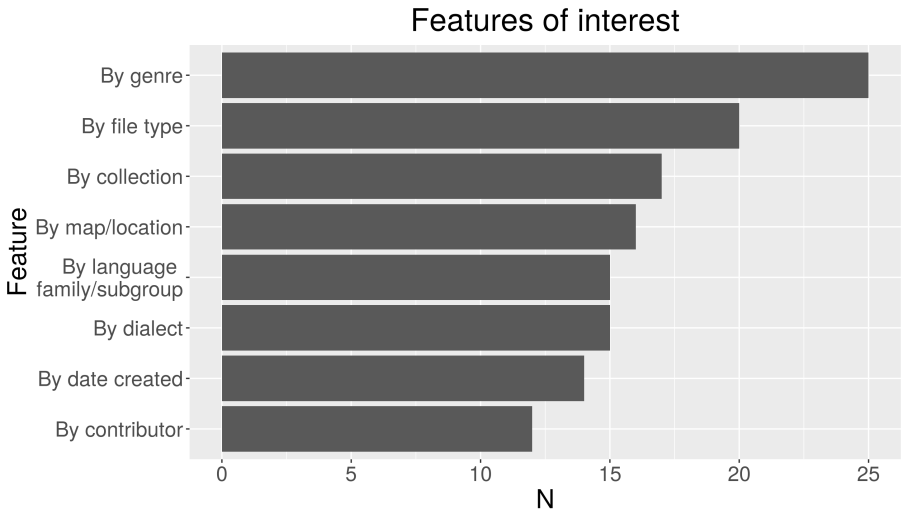

Figure 3.1: All features that participants reported being interested in seeing

associated collection. Again, searching by geographic location and related languages were important to many people as well.

3. Would you like to see pages with pre-assembled sets of resources (so far a page with resources grouped by language, a page with all the grammars and dictionaries, and a page with teaching materials, a page with songs/music, and a page with language planning materials)? Any suggestions for other useful topics for browsing?

    - personal narrative vs traditional story
    - songs, music, movies by type
    - languages, that include/account for alternate language and tribe names
    - browsing by Glottolog-like language taxonomy, including subgroups
    - teaching materials organized by method and type, e.g. materials for teachers, for students, and for home
    - browse by person's name
    - browse using a map

Question 3 asks about browsing features that participants are interested in, and the discussion brought to light specific combinations of genres that are interesting to some people, like personal narrative vs traditional story, media genres, and teaching materials organized by characteristics specific to that genre. This is especially interesting in the latter case, where the archive does not have standardized metadata fields for these distinctions. If the interest is confirmed in future discussions, it would be valuable to pursue the development of browsing guides based on automated curation of teaching materials using, for example, regular expressions to search among the metadata for parameters of interest. To accomplish this, teaching materials could be coded with key phrases like "materials for teachers" and "materials for students." that the the search query would look for.

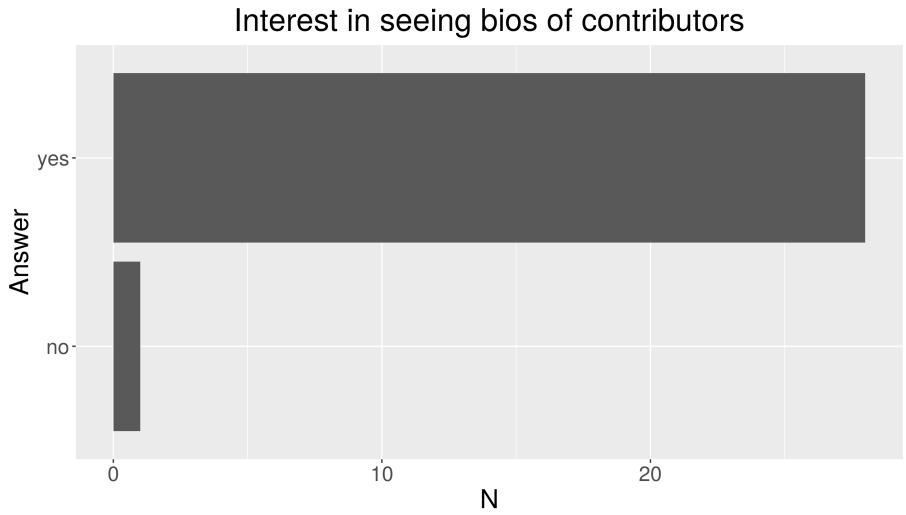

Figure 3.2: Participants' interest in seeing bios for contributors to collections

4. Would you like to see pages for people represented in the collection, with a short bio, picture, and list of items they are affiliated with?

The results from Question 4, shown in Figure 3.2, confirm that the desire to see and search for personal metadata of contributors far outweighs any potential for privacy concerns. Nevertheless, the archive will maintain the ability to anonymize any contributors that wish to be anonymous.

5. Is there anything that you've particularly liked about the websites for other language archives? Disliked?

    - It is important that the site be visually appealing. Most language archive sites are sterile and generic.
    - Open-ended search boxes are great if you know what you are looking for, but not helpful if you don't. Add information about the various ways to search next to the search box. Perhaps give suggestions of examples of what to search for.
    - Many people liked an obvious, simple search box on the home page.
    - The keyword search needs to be supplemented by a detailed/advanced search.
    - NAL was also easy to deposit with in the past.
    - The Smithsonian is also easy to access.
    - Several people liked the CLA's map search.
    - People like to browse, and don't like to answer a lot of questions to see what is there. Having a video to watch about how to navigate (CORSAL) was helpful.
    - More Native designs, possibly also writing on the homepage. In general seeing less English.

The discussion on what participants liked about other archive websites revealed a mix of opinions about what makes a positive UX. Some people expressed a desire for visually appealing websites that are easy to use, which are

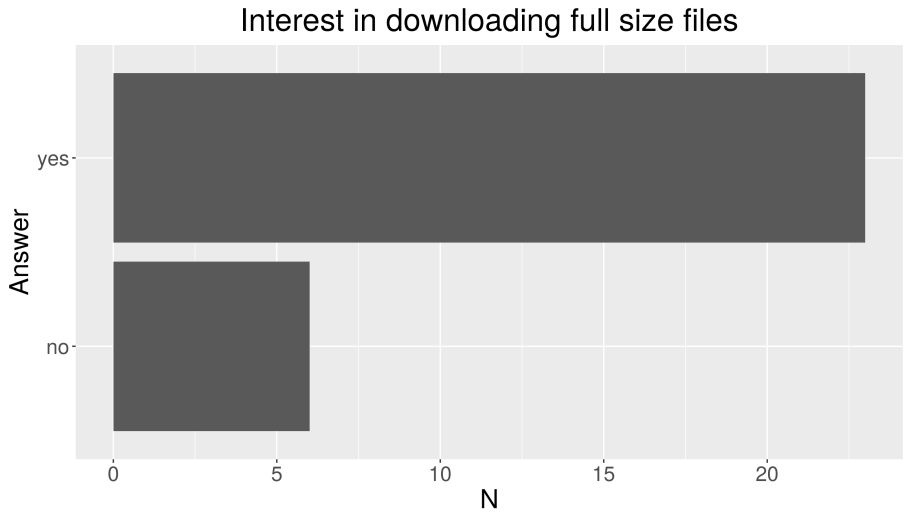

Figure 3.3: Participants' interest in being able to download full size files

statements that are hard to translate to specific design criteria. Nonetheless, it became clear from the discussion that for many people, ease of use is achieved through a basic search field that lets you search through many metadata fields simultaneously. However, there were mixed views here, in that other participants had a clear desire to have advanced search fields as well. Multiple participants also agreed on the value of search suggestions, in either style of search. Finally, some people were interested in seeing more Indigenous design elements on the websites.

3.2.2 Access

6. In addition to small, easily-downloadable files, would you be interested in having very large, archival-quality files directly available for download? (The other option is that large, high-quality files could be requested and sent separately).

Figure 3.3 shows the results for Question 6, that most people want to be able to download full sized files in archive collections. This makes it clear that even if web-based viewers are developed, as for example for video streaming of small access copies of videos, links to download the full files should be available and prominently displayed.

7. Would you like to see us develop (in conjunction with language programs) a separate web-based viewer for the transcripts and translations in NAL? See for example here: http://northernpomolanguagetools.com/texts/story.php?story=3.

Question 7 asked participants if they were interested in web-based viewers for transcriptions and translations, and Figure 3.4 shows that participants were almost all interested in this feature. This provides motivation for NAL to provide such functionality by, for example, incorporating an ELAN viewer that has already been built as a microservice in the digital infrastructure.

8. Are you excited about the prospect of being able to access the language materials in NAL online? What do you see as the benefit to your community?

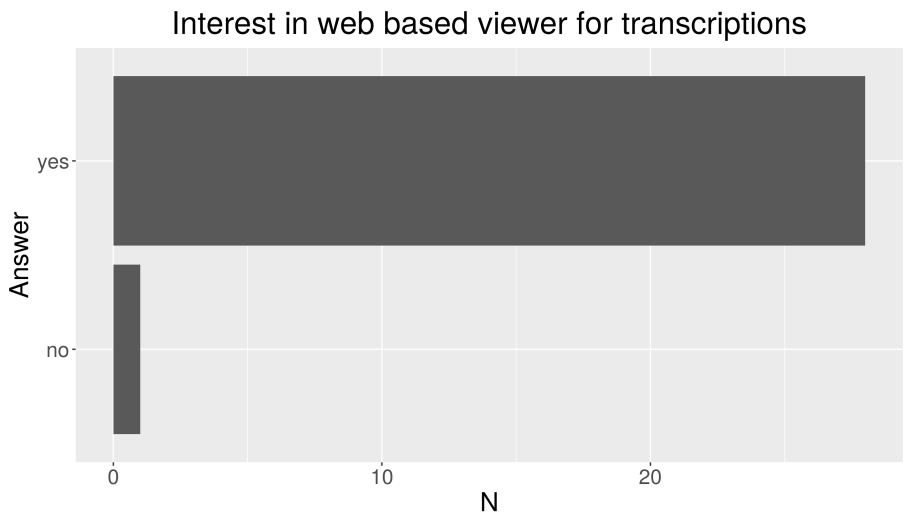

Figure 3.4: Participants' interest in a web-based viewer for transcriptions and translations

- Useful to tribal members who have computer access, particularly those who don't live near tribal resources or have access to native speakers; there's a lot of online interest in Native languages for many spiritual, political and personal reasons and forms of healing.
- Useful to students doing research or looking for resources.
- For the academic community, being able to access NAL collections online will be a huge help, both for research and community engagement. NAL has multiple, very significant collections, so being able to access them all in one place and search across them will be a huge help in terms of finding relevant data to assist with developing pedagogical materials and research articles.
- Useful to curriculum developers and teachers of Native languages.
- There will be more opportunity to consult with language professionals who run the collections.
- People often feel more comfortable exploring an online resource than coming to visit an unfamiliar institution in person.
- This will increase the discoverability of these resources, particularly tribes trying to figure out what's out there belonging to their community.
- This can also be a way for the community members to see what the NAL can do for their community when it comes to recording and preservation.
- Having easier access to the language might also help to promote people in wanting to learn it.
- Continued development will yield other opportunities not yet explored here.
- Online presence will help people know the collection exists.
- People will be comfortable knowing these materials are in a safe place.

The answers to Question 8 show that many people who attended the workshops have a positive view of providing online access to the collections, especially because community members are likely to have internet access but not as likely to have access to travel to the collections in person. Some people also expressed the view that this development would have a positive effect on cultivating relationships between NAL and the language communities.

9. What apprehensions do you have (if any)?

    - 12 responses had no apprehensions
    - Apprehension about materials important to tribes being made public online, that people could potentially exploit these resources, or commercialize them, despite the access protocols and usage agreements.
    - Cultural appropriation, both by non-Natives and those claiming federal recognition based on stolen Indigenous IP
    - To be done well, the museum should work closely with tribal organizations
    - Non-tribal members worry about running the risk of making some things public that shouldn't be, even with the best of intentions
    - Fear that the effort will fall short in ease of use or accessibility.
    - That this type of project requires long-term commitment from many entities in both time and energy that doesn't always materialize.
    - That some materials are from second language speakers and contain errors, but that people accessing the resources may take them as correct.

The apprehensions expressed in the answers to Question 9 affirm that language community members and other NAL users expect the archive to perform the role that institutional archives perform while honoring the principles described in §2.5, especially those categorized with a focus on depositing. Specifically, participants were concerned for upholding Indigenous rights over data and enforcing appropriate access restrictions for those data. Beyond access restrictions, they are also concerned about misusing data, and this affirmed NAL's plan to require users to register with the website and agree to terms of use, as a way of providing some means to respond to misuse of the data. Finally, the comments echo the principles of cultivating relationships as an important factor in success of such a project that NAL is undertaking.

3.2.3 Community-specific metadata

10. Would you like to see NAL put TK labels (https://localcontexts.org/tk-labels/) on some or all of the collections?

11. If you are a depositor, would you be interested in collaborating with us to add TK labels to the materials in your collection?

The answers to Questions 10-11, presented in Figures 3.5-3.6, show that many participants were interested in having NAL implement TK labels, which motivates NAL to include these labels as part of its metadata schema.

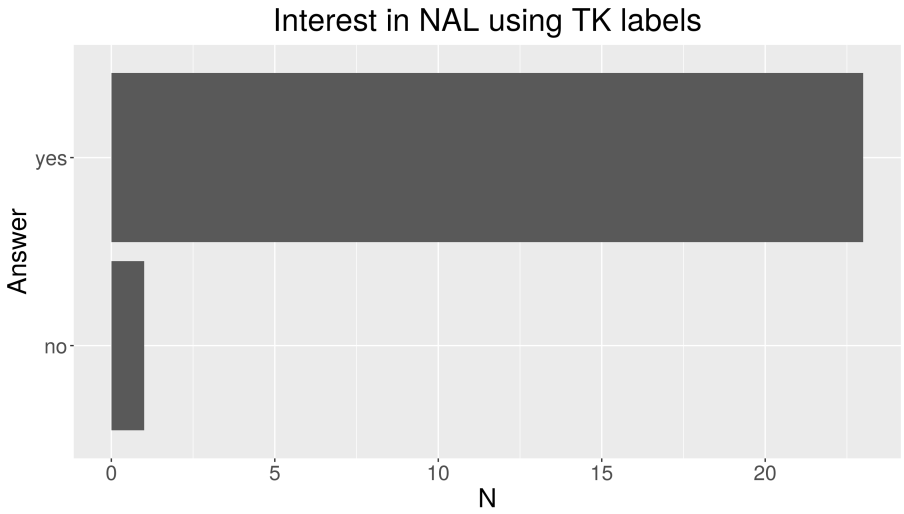

Figure 3.5: Participants' interest in NAL using TK labels

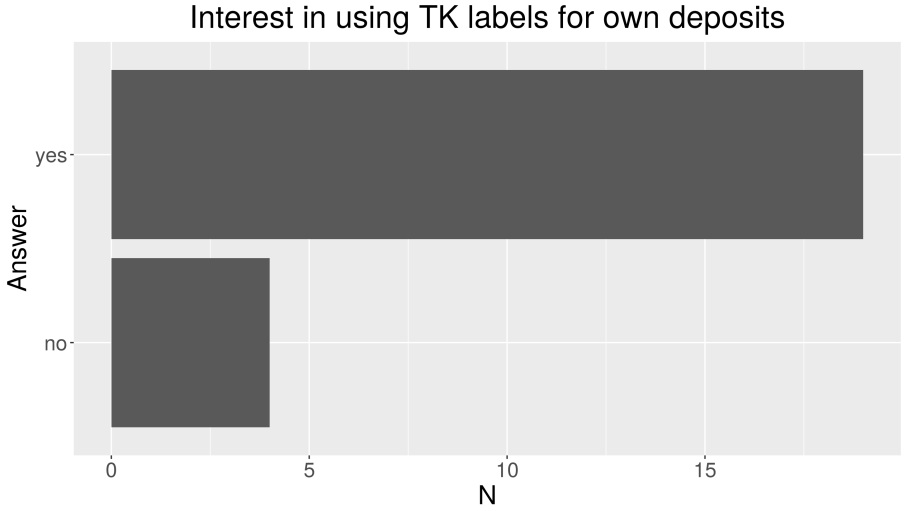

Figure 3.6: Participants' interest in using TK labels for their own deposits

3.2.4 Co-curation

12. What do you think about allowing visitors to the site to comment on materials in the collection?

    - While this sounds good in theory, it has to be controlled/vetted/moderated.
    - Moderation systems are hard to keep up. Depends on the volume and types of comments. If you have limited resources it can get out of hand.
    - Maintaining external moderation would require long-term sustained commitment from communities.
    - You have to be really careful, historically generates a lot of spam.
    - comments for the purpose of meeting user's needs would be useful (like what they might want to know about materials), but if its open to any comments it could be problematic.
    - A well-placed "Ask the curator" function could accomplish a lot of the benefit of this.
    - Instead you could periodically have sessions for commenting that are temporary and moderated.
    - Maybe if the collection had another interface like a social media account it could be better to have that there.
    - What would value-added examples of commenting even look like?
    - This might be a nice feature but should have a low priority vs. search and general usability of site.
    - Possibly have just a twitter feed on the general site instead of a comment box for each item.

The discussion of commenting prompted by Question 12 showed that there is a lot of concern among different participants about the benefit of having a commenting feature, how it would be implemented, and the sustainability of maintaining this feature and moderating comments.

13. As mentioned in the background document, we are developing a way for knowledgeable users to submit metadata to items in the collection. That information will need to be vetted before it goes into the official record. The question is, would you be interested in approving/rejecting any suggested additions/changes to the information for your collection that come from users of the website?

14. If you answered NO, who do you think should vet those? (Someone from the tribe, Someone I appoint, NAL staff, Tribal Entity, other)

The results for Questions 13-14, shown in Figures 3.7-3.8, illustrate the mixed responses to the idea of vetting information that was provided by others for collections. More people said they were willing to participate in the vetting process than not, but it is clear that not everyone is in this position, nor do they have consistent views on who should be responsible for this task.

15. In addition to allowing users to add catalog information about each item, would you be interested in seeing us develop a place where people can add general comments/discussion?

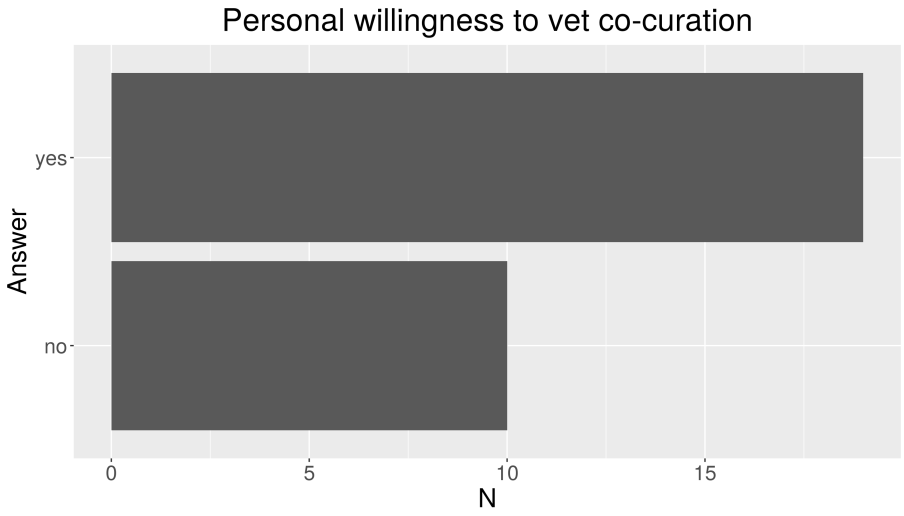

Figure 3.7: Participants' willingness to vet co-curation for their own collections

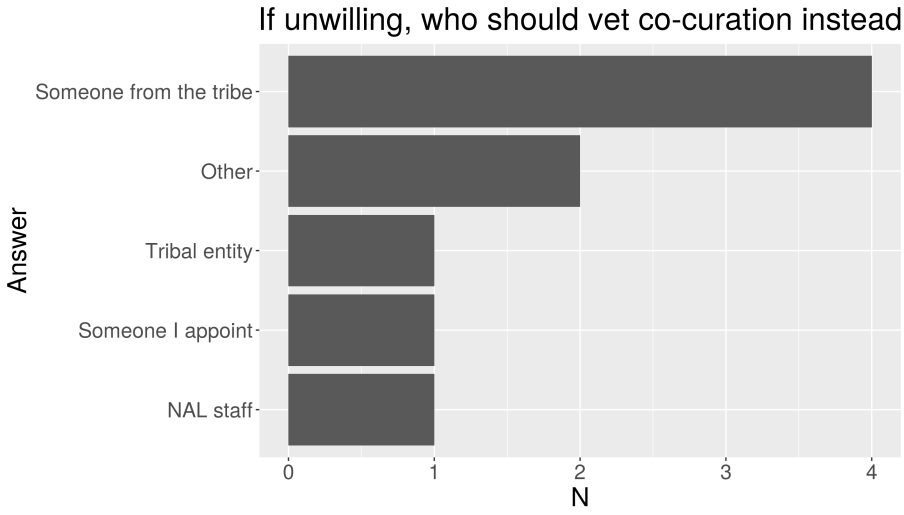

Figure 3.8: Participants' views on who should vet co-curation for their own collections, if not themselves

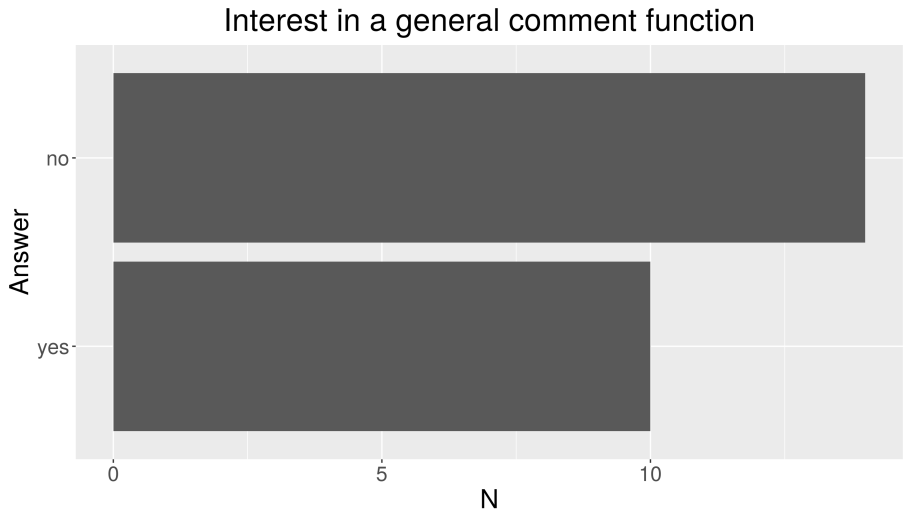

Figure 3.9: Participants' interest in having a general commenting feature

16. If so, how do you think that should look? (Separate downloadable file, twitter-style feed, box at bottom)

Figures 3.9-3.10 show the results for Questions 14-15. More people said they were not in favor of a general comments feature than those who were in favor. This echoes the apprehension in the previous discussion on commenting. Based on this result, NAL decided to forgo development of this feature.

3.2.5 Digital return

17. Thinking about your community, could they benefit from having a local copy of NAL resources? Where do you think we could potentially put a computer with collection materials?

    - We are very spread out [so there isn't a central location]. Putting it online is the best we can do.
    - We already have our own portal/archive for the resources in question.
    - We have a community library where people who don't have personal computers can access the internet, so they could just access it from there.
    - I know we have a library down here. It would be on us to make sure the community knows how to browse NAL
    - By and large our language learners are tech savvy so they don't need any special provision, the website is sufficient.
    - We are building a community repository, so we could use your archive-side infrastructure potentially.
    - Our local servers aren't particularly secure, so better to use yours.

Question 17 asked about participants' interest in digital return of data, and for the most part, the responses suggested this is not a high priority, and that online access to NAL makes more sense. In one exception, a participant

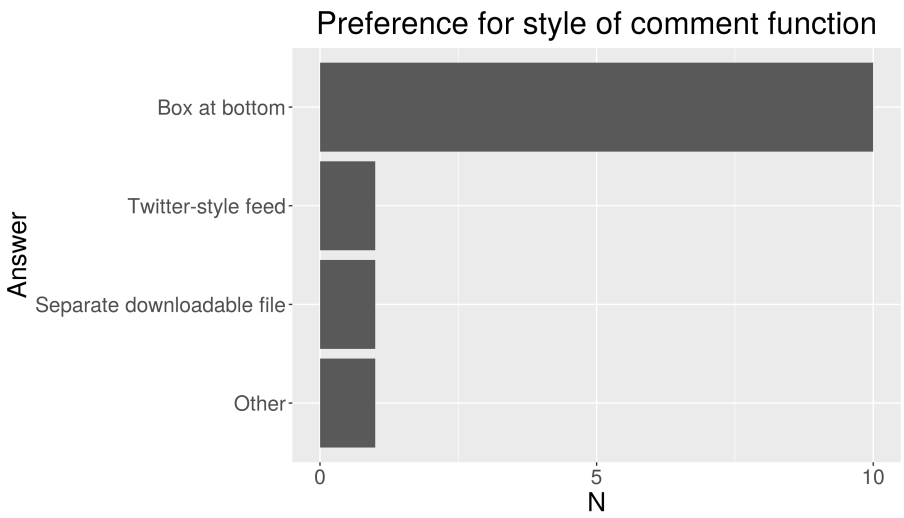

Figure 3.10: Participants' preference for the style of a general commenting feature

mentioned a desire to build a community repository as well as interest in using NAL software to do so. This motivates NAL's commitment to make the software application it develops open source.

3.2.6 Summary questions

18. What topics/features discussed during the workshop are the most important to you/do you think will be most useful?

    - 6 people said that the ease of browsing and/or downloading the materials is most important; many users will not be tech savvy or familiar with archives.
    - Less focus on preservation and access and more focus on discovery and delivery.
    - Learned about some technical aspects of language and archiving, like unique codes of resources, databases.
    - We are imagining a site that is more vibrant, that is focused on reaching out to learners of the languages.
    - NAL could become the go-to place to see what other communities with similar or related languages are doing and get inspired.
    - Access control and defining user permissions is very important, and discussing it with people so that they understand.
    - Adding more information about what is in each file would be helpful to the user.
    - Advanced search features/search tools for collections.
    - Benefit to participants and NAL in bringing together a group of like-minded people from different tribes and backgrounds.
    - Enjoyed learning about what the Sam Noble can do and the direction it wants to go, and that this involves depositors and tribes in the planning and discussions.

- I'm glad depositors have input.
- This is a good way to create a relationship with Indigenous peoples, and to let people know what services NAL provides. This is a start to get away from the notion that museums are a gatekeeper to Indigenous culture and knowledge.
- A place for depositors to arrange and tag their items is important.
- Complete and updated metadata tags is a current priority.
- Maintaining the integrity of the collection as the underlying technology changes every 5 years or so.
- Comparing different archives was helpful.

The answers to Question 18 mainly highlight that participants echo the value of the principles presented in §2.5. Specifically, they value discoverability and access, with necessary access restrictions. Moreover, they value the positive UX that comes with easy to use and powerful search tools for discovering data. Finally, the commitment to building relationships with language communities is valuable to them, and they viewed the workshops as an act of this process.

19. Is there anything you can think of that you would like to see developed wasn't mentioned in the workshop or this questionnaire?

    - 16 Nos
    - Development of policies e.g. related to health where a sick or ill person may listen to the sacred songs for healing needs.
    - Consider facilitating some of these projects in other ways. For example, time-aligned texts could be submitted in collaboration with a publication series, possibly through the museum or the Native Nations Center. The text series could also address other relevant topics of interest to tribes, e.g. why Oklahoma languages are so important, updates on particular revitalization efforts, challenges that need to be overcome & some solutions, reports on what communities have already done and where they want to go, issues in educational linguistics, language rights, survey design, inter alia.
    - It would be great if the NAL collections continue to go in a more dynamic and interactive direction. Ideally, the collections should play a role in the language revitalization efforts of every community that is interested in working with the NAL collections at OU. It's up to the NAL collections to articulate ways it can partner with communities beyond archiving.
    - Consider how NAL development might support born-digital materials created in the context of a tribe specific revitalization efforts or archival needs.
    - NAL collections can continue to provide workshops and engage in outreach efforts to address issues in documentation, description, and pedagogy, sharing not only collections materials but also linguistic/language knowledge relevant to language policy and pedagogy.

- Consider doing an electronic and a print guide to NAL holdings and services.
- Tribes should look at doing partnership agreements with NAL so that they can clearly articulate proprietary copyright concerns but also leave an opening for future development and use as a public site for our materials.
- Consider doing a call for movies and videos; several tribes are doing dubbing in Native languages.
- In future workshops it would be interesting to have some participants who have developed working online dictionaries or language lessons take part to discuss how they prepared the material they put online.

Among many specific ideas expressed in the answers to Question 19, one main theme that emerged is that participants suggested that NAL make efforts to work with language communities to use data in ways that are valuable to them, such as pedagogy, maintenance, and revitalization.

20. Any other comments or things you'd like us to know?

- 17 thankful, grateful for doing this work and including us, and are excited for the next steps.
- Look into ElasticSearch as a search database.
- Create mechanisms to generate forms/lists online of material that meets a given criteria. A simple [ ] next to an item might be used to export the content to an XML document and convert to PDF.
- Facilitate line-by-line playback of selected material.
- Work with an ASR specialist for multilingual phone recognition.
- Create workshops that assist with tribal policies on digital resources.
- This project is going in the right direction by including tribes and listening to their suggestions.
- The workshops brought up some interesting issues around sharing, where some tribes do not want the language displayed outside tribal contexts, as well as comments on problems with the way social media is being used by some tribal members, pitting some families against others within the tribe.

Finally, the answers to Question 20 presented many different ideas, but a theme that emerged once more is the value of NAL working with Indigenous communities and cultivating relationships with these communities.

3.3 Feedback on wire frames

After the workshops were completed, a set of wire frames were developed by an IT developement group at OU (described further in §4.1), which where motivated by the responses of participants during the workshops. These wire frames were built as a concept for a new NAL website that will be built as part of the project to bring online access to the NAL archive's collections. The website focuses specifically on access for general users, while the processes of

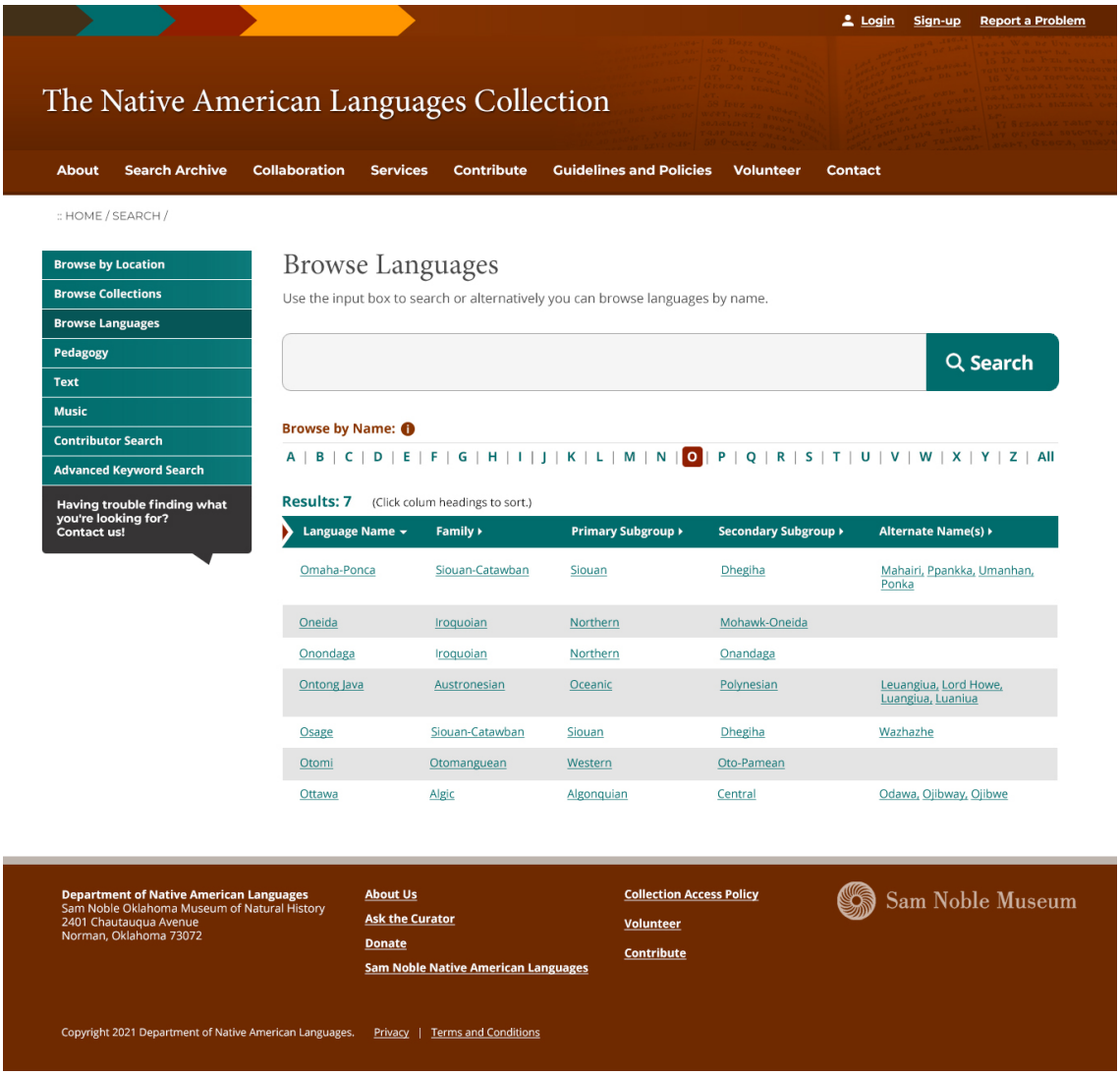

Figure 3.11: Wire frame for browsing based on languages

repository management are carried out by other components of archive's digital infrastructure (described further in Chapter 4). The wire frames were sent to workshop participants, who were asked to give comments and/or feedback via email.

The wire frames include views for collections, items, and people, as well as search and browse functionalities to accomodate the needs of users based on their responses in the workshops. Figure 3.11 shows the wire frame for browsing by language, and Figure 3.12 shows the wireframe for searching by location.

A full list of responses to the prompt for feedback on the wire frames is as follows:

- Looks great.
- This is amazing—great work!
- Thank you for sending the mock-ups. They look great. I think it's very clear and user friendly. I don't see

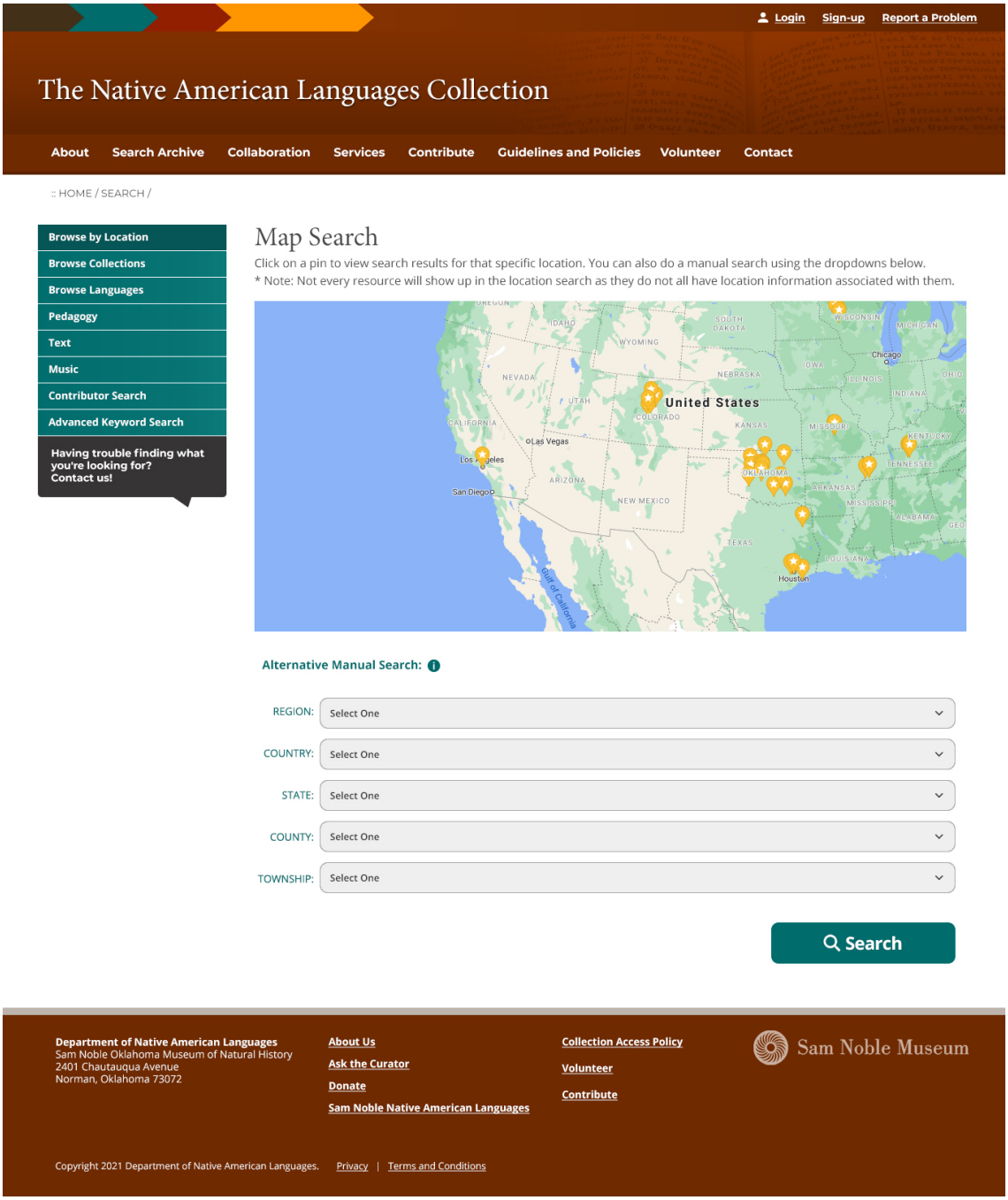

Figure 3.12: Wire frame for searching based on location

anything that stands out at all so far. I love that there are so many different ways to search for something. Keep up the good work! I look forward to seeing it in action.

- Wow. Those sample pages look great. They are clear, professional, and user-friendly. An excellent upgrade and fine contribution!
- Thanks so much for the update and wonderful screenshots. I want to go over them ASAP as I am sure that they will inspire a lot of ideas.
- This looks great! I love the color scheme and how clean/user friendly everything is.

The responses of workshop participants to the wire frames were limited, but unanimously positive. Participants reported that the website depicted by the wire frames looked user friendly, with one participant reporting they liked the color scheme used, and another participant reporting they thought the website depicted looks professional. Additionally, one participant reporting liking the many ways to search for resources represented in the wire frames, which reflects the topic of search and browse discussed at the workshops.

3.4 Discussion

The responses of participants both during and after the workshops provided a glimpse of what NAL users want from an archive in terms of functionalities and UX. One of the main takeaways from these results taken together is there lack of deviation from my expectations. While I did not make predictions for the answers to the workshop questions, I can say in hindsight whether specific answers fit my expectations for archive users, or if they are otherwise surprising. For the most part, the answers to questions in the subjects of search and browse, access, and community specific metadata fit my expectations.

On the subject of search and browse, NAL users on the whole were interested in both basic and advanced search as well as browse features, which fits my expectations. This expectation is based on the fact that, in my experience, basic search is convenient, and advanced search is necessary occasionally. Additionally, implementing both basic and advanced search is a common feature of many websites I use. I would expect a large enough group of people surveyed to include some people that value basic search, advanced search, and both. The metadata for browsing that were most requested were languages, geographic location, and genre. My strongest expectation would be for language to be important, because of the fact that the data at hand are presented as language data and stored in a language archive. Additionally, I am not surprised that location and genre were also common responses.

On the access subject, NAL users were interested in browser-based functions for viewing files, which fits my expectation that today's computer users are conditioned to expect a browser-based UX due to the ubiquity of browser based web applications. Additionally, some NAL users expressed concern that online access to data would increase the prevalence of misuse of Indigenous cultural heritage, which fits my expectations, based on the fact that Indigenous communities can often have mistrust of non-Indigenous institutions (Leonard, 2018; Smith, 1999). Additionally,

it motivates the security measures taken by existing digital language archives, as well as those presented in Chapter 4.

Finally, on the subject of community specific metadata, NAL users were in favor of the implementation of TK labels, which fits my expectation that Indigenous users of language data would be interested in having the ability to use TK labels. This expectation is based on the level of interest in TK labels that I have perceived from participants in events such as the International Conference on Language Documentation & Conservation, hosted by the University of Hawai'i at Mānoa, and Relationships, Reciprocity, and Responsibilities: Indigenous Studies in Archives and Beyond, hosted by the American Philosophical Society. Additionally, the extent to which NAL users were in favor of TK labels is noteworthy, as the support was almost unanimous.

Co-curation, digital return, and the wire frames were subjects where the responses were surprising or noteworthy.

On the subject of co-curation, it was noteworthy how split the participants were in allowing comments to existing deposits, as well as how to vet this information. It was surprising how easy the conversation on this topic became heated. It seemed that, collectively, the groups of participants agreed that this information could be valuable if the implementation for curating this information was done well, and that the method for accomplishing such an implementation is unclear.

On the subject of repatriation, it was surprising that few NAL users were interested in the physical repatriation of materials to their communities. This result suggests that online access is valuable to and sufficient for users with reliable internet access, and motivates a focus on digital return through archive access, discussed in §5.3.4.

Finally, it was surprising that the responses to the wire frames, while positive, were minimal in both number and content. This suggests that regular users may not glean much from wire frames, and motivates the solicitation of further feedback from NAL users when a prototype of the website is available for them to use.

3.5 Chapter summary

This chapter presented the results of a series of workshops that NAL conducted with users of its archive in order to solicit feedback about what would be desirable for a website that provides online access to its collections. This feedback was sought as a first stage of user-centered design of archive infrastructure and a website for the NAL archive, where user-centered design means incorporating feedback during design and development. The workshops were a series of virtual meetings where invited participants answered multiple choice questions and engaged in discussions about open ended questions. The questions spanned topics that included: search and browse preferences when using archives, interest in specific features related to access, interest in using community-specific metadata fields, interest in and willingness to participate in co-curation, relevance of digital return, and summary questions (§3.1).

The answers and discussion that came from the workshops motivate NAL's plan for designing digital archive infrastructure, both in affirming certain features to pursue and certain features to avoid. The workshops motivate the

development of both basic search and advanced search functions, and beyond that a focus on UI features for browsing collections based on genre of items, related languages, and geographic location. Additionally, the workshops motivate the development of TK labels as part of the archive's metadata schema.

Based on the workshops, developing an open ended commenting feature will not be pursued. However, in order to provide for co-curation in a more narrow sense of accepting additions to metadata, the plan for infrastructure includes roles for moderation that can be assigned to specific users for specific collections or items, such that co-curation can be enabled by the software in specific cases where there are people willing to engage.

The discussions from the workshops also affirm some design decisions NAL had made prior to the workshops, including hiring an Indigenous graphic designer to participate in developing NAL's website, releasing the software application for NAL's digital infrastructure with an open source license, and continuing to focus on building relationships with language communities and language community members (§3.2).

While many of the results fit my expectations for NAL users, some noteworthy results are that: co-curation through accepting and vetting information from third parties is a contentious issue; physical repatriation of materials to communities is not a strong interest; and the responses to wire frames were positive but further work is needed to solicit feedback on the NAL website (§3.4).

In summary, the information obtained from these workshops fed into the design for new digital archive infrastructure for the NAL archive, and this design is presented in Chapter 4.

Chapter 4

A design for digital archive infrastructure

This chapter presents a design for digital archive infrastructure that is an integral part of the plan for a new website for the Native American Languages (NAL) archive. This work began in 2020, and some aspects are completed while others are ongoing. I worked with NAL to develop a model for their existing data, and migrate their metadata to an SQL database. Additionally, we made a plan to develop a website for the archive and manage the data in its collections going forward. This plan compartmentalized the development into two modules, based on constraints of the archive and its various audiences. One of the modules is targeted for use by archivists and depositors, while the other is targeted toward users. Following Trilsbeek and Wittenburg (2006), this chapter uses the terms archivist, depositor, and user to describe different people that interact with the archive. Archivists manage the archive and its collections, depositors participate in depositing data and metadata with the archive, and users are interested in accessing data. These terms are used to distinguish different use cases for and interactions with the archive. However, a given individual might take on some or all of these roles at different times. The word user is used in related contexts, for example with the user experience of any individual interacting with the archive, the user accounts that archivists, depositors, and users all have when interacting with the archive, and the user base of an archive, which is the group of all people interacting with the archive. When contrasted with depositor, however, the term user refers to an individual primarily interested in accessing data rather than depositing it.

This chapter presents design and then process in each section. While it is possible to abstract away the design completely from the process, this obscures details of the logic that led to the design that are highlighted in the process. These are the details of decisions made based on competing goals and complex constraints, and making these decisions is the process.

§4.1 presents an overview of the design process, how aspects of the process influenced the design itself, and which parts of the process are completed or otherwise ongoing. §4.2 defines the overarching principles that guided the design. §4.3 describes the metadata schema used by the design and its mapping to the OLAC metadata schema. §§4.4-4.5 provide details of the processes of deposit and access in this design, or in other words, the inputs to and outputs from the archive. §4.7 describes the current state of implementation of the design. Finally, §4.8 presents

future developments for the software design, based on compatibility with existing tools and infrastructure.

4.1 The design process

While ideally the design should be valid independent of the processes that fed into the design, the reality is that certain compromises between conflicting goals and constraints did feed into the design, and the processes behind these design decisions help to capture the logic of these decisions. For this reason, a summary of the processes involved in the design of the software infrastructure is presented in this section. Additionally, more detailed information on these processes is presented at the end of each subsequent section, where these details are important in explaining the design. These details are important because this software infrastructure is designed to be open source and shared with other communities with like minded goals to use for their own systems. Thus, these procedural details provide a guide for where the design might need to be altered to meet the needs of other communities.

The people involved in the design process can be divided into four groups: The NAL archive staff, led by Dr. Raina Heaton; myself as a developer; a University of Oklahoma (OU) IT team as developers; and members of NAL's user base that attended workshops and development meetings during the course of this process. The archive staff organized and led the project for developing NAL's digital archive infrastructure, and supplied requests for features and constraints for the design based on NAL's preexisting processes and data. They supplied this information to the developers, including myself and the OU IT team. I was tasked with developing a repository management system for the archive, while the OU IT team was tasked with developing an access portal the website, which are software applications that are described in more detail in §4.2.1. Finally, NAL's user base provided feedback in the beginning of the design process in order to motivate the design based on this feedback.

During this process, Dr. Heaton and I met regularly to discuss details of the design of the archive infrastructure, in order to ensure the design accommodated NAL's specific needs and met best practices for archives in language data, described in §§2.4-2.5. These include: interoperability with existing metadata and harvesting standards (Bird and Simons, 2003a:379); long-term preservation of data (Henke and Berez-Kroeker, 2016:412); facilitating access and simultaneously restricting it where necessary to meet the needs of Indigenous communities and depositors (Wilkinson et al., 2016; Carroll et al., 2021); and ensuring data are citable (Berez-Kroeker et al., 2018:14). Dr. Heaton met with the OU IT team multiple times, to discover what role they might play in this project, establish a relationship, present NAL's needs for a front-facing website, and come to an agreement about the services they would provide. I met with Dr. Heaton and the OU IT team three times in long-format meetings to establish the details of the general plan for the design and software architecture of the web applications that would provide the archive infrastructure and front-facing websites, as well as the coordination between these components. I continue to meet with specific members of the OU IT team as we implement the design to address issues as they develop and to ensure the coordination between software modules is sufficient. Finally, members of NALs user base attended workshops with Dr. Heaton, the OU IT team, and I during each stage of the process in order to give them a platform to give us feedback on our design. This feedback was considered crucial by definition, as the planning and development stages were

defined to be iterative. That is, the planning and development stages included feedback gathering from NAL's user base in order to ensure their needs were being met by the design of the new website and the underlying archive infrastructure.

This process has spanned two grants awarded by NEH[1] (see: §2.8). The planning and design phases occurred mainly during the planning grant and the application phase of the implementation grant. There were four phases of planning and design: personnel, software architecture, and overarching design designs; modeling and migrating existing metadata; gather feedback from NAL users; and developing a detailed plan for the software applications that will comprise NAL's digital archive infrastructure. Revisions of the design as well as implementation are ongoing. The implementation is in the first of three years of the grant at the time of this writing. As of this writing, a development version of the archive infrastructure is implemented, and the features it currently includes are described in §4.7.

The first stage of the design focused on personnel decisions and understanding what resources the people and teams could provide. This included the decision of Dr. Heaton and me to collaborate, based on my experience with documentary linguistics and software development. Additionally, Dr. Heaton had meetings with various members of the OU campus who have IT related roles to determine who to include for different aspects of the project. I participated in some of these meetings as well in order to discuss details about software architecture and design, server administration, and language data.

Based on various goals and constraints related to better practices in language documentation, the needs of NAL's audience, OU infrastructure, and finances, we designed on a plan for building a website for NAL that was modular, leveraged existing open source and open development software, and was extensible in specific ways, as presented in §§4.2.2-4.2.3. In this plan, I would develop a repository management web application focused on features related to archive data management and depositing to the archive, while the OU IT group would develop a web application to facilitate access for users of the archive.

The second design stage was focused on developing a model of the existing metadata for NAL collections and migrating these data to an SQL database. The goals of this phase were to increase access to these metadata for the archive staff, and to migrate these metadata to a format that would easily transition to the next phase of the project, which would be developing the archive infrastructure for the new website.

This was an iterative process, in which I collaborated with Dr. Heaton and archive staff. In each iteration, we collected a sample of metadata, I built and/or modified an SQL database model to accommodate these data, and I received feedback from archive personnel about the validity and robustness of the model. In subsequent iterations, we incorporated edge cases in the data in order to accommodate all the existing data.

The third stage of the development focused primarily on gathering information from the NAL user base on their desires and expectations for the website and the user experience (UX), as defined in §2.7.4, that it would provide them. This information was obtained primarily in two ways. First, a group of workshops were hosted by NAL to ask questions of users about their desires and expectations of an archive, as well as their feedback on using aspects

[1]NEH grant numbers: PW-269366-20 and PW-285221-22, to University of Oklahoma; Principal Investigator: Raina Heaton

of existing archives. The results of these discussions are presented in Chapter 3. Second, at the end of this stage, the NAL user base was asked to provide feedback on wire frames for the website. These wire frames were created through collaboration between Dr. Heaton and the OU IT group as part of the planning grant. The wire frames consisted of visual mockups for pages of the website that would provide access to the collections in the archive. The conclusions reached from the discussions in the workshops and the feedback on the wire frames informed design decisions, and more details on such decisions are presented throughout the rest of this chapter.

The final stage of development, which was dependent on all the other stages, was the detailed development of the design for the repository management web application and the public access web application. These designs were created by myself and the OU IT group, respectively, based on the modularization that was built into the design from the first stage. We communicated during these developments to ensure the applications would work in coordination, and this coordination was based mainly on communicating the metadata schema of the collections designed in the second stage. Finally, we incorporated the feedback we received as part of the third stage, making this design the first step in our iterative design approach.

The four stages occurred in the larger context of two NEH grants, and the second of these grants is ongoing. The implementation of the design developed in the fourth stage is currently in year one of three, following the timeline of the grant, so that many of the details of the design have yet to be implemented.

As of this writing, the repository management web application and the public access web application have been implemented by creating and modifying a development instance of InvenioRDM with sample data from NAL that is open access. Many of the features currently implemented are provided by the base InvenioRDM software package, while a select few of the modifications to InvenioRDM that are proposed in the design have been implemented. Many of the modifications to InvenioRDM that are proposed in the design have yet to be implemented. The current implementation is only available to the developers and NAL staff because it is under active development.

§§4.3-4.5 focus primarily on my contributions to the design, which include the metadata schema, the repository management application, and the communication between application modules. Each of these sections summarizes which features presented are provided by the base InvenioRDM software package, which features are to be implemented through modifications to InvenioRDM, and which of these modifications have been implemented thus far. §4.5 additionally summarizes some features of the public access website designed by the OU IT group, where they are relevant to the cohesion of the modular design.

4.2 Design principles and software architecture

The overarching design principles that emerged from the initial stage of design included using a modular design, leveraging existing open source software and building open source software, and extensibility of access. Additionally, these principles led to initial choices for software architecture in the design.

4.2.1 Modular design

The design of the archive website and underlying infrastructure is modular, and consists of two modules that are independent web-based applications that communicate through an application programming interface (API) and access the same underlying database. The first web application, the repository management application (RMA), is focused on providing the features for depositing data and metadata as well as maintaining those data and metadata. The second web application, the access portal application (APA) is focused on providing user friendly access to the data managed by the archive. Using the metadata for access restrictions for both items and files, as well as user accounts, both web applications provide access to the data only where appropriate. The RMA serves data through a JSON-based REST API, while the APA reads these data through calls to the API. In this way, the APA need not store the data.

Figure 4.1 illustrates the two modules and their relationships to each other, users, and other software applications. The RMA is written in the Python language, using the Flask web framework, and extends the InvenioRDM software application through a series of modifications to InvenioRDM. The APA is also written in Python, using the Django web framework. Both applications have a user interface (UI) in the form of a website. The RMA is primarily targeted for use by depositors and archivists, and includes a depositor portal (described in §4.4), while the APA is primarily targeted for use by users. However, users, depositors, and archivists can view both websites (but users do not have access to the depositor portal). In addition to a UI, the RMA has an API that the APA uses to retrieve data and metadata, rather than the APA storing the data in a database of its own. Communication between the two modules is asymmetric: the APA always initiates communication by issuing a request to the API, and the RMA issues a response to the APA. This configuration is a property of REST APIs in general, and it makes the design more efficient in the sense of not needing to build an API component into the APA. The RMA also has the potential to communicate through its API with hypothetical additional access portal applications, which could be built in the future to serve the needs of specific user groups.

As an example of the data flow in this design, consider a deposit of language data that is made by any stakeholder in a language documentation project, such as a speaker and/or linguist. Before making the deposit, the depositor must sign up for a user account using the RMA website, agree to the terms of using the archive, and an archivist must assign their account permission to use the depositor system, thus making their account a depositor user account, as described in §4.4.1. To make the deposit, the depositor uploads these data and create metadata through a browser-based UI provided by the RMA, and these data and metadata are stored in the database managed by the RMA. Upon completion of the deposit, these data are transferred to a private server by software run on the archive's servers, which constitutes part of the archive's infrastructure. This private server is not exposed to the internet, and houses master copies of the collections. In the case that some of these data have access restrictions that require their exposure to the internet be mitigated, these data are stored only on the private server. The copies of these data that were created on the server hosting the archive software are removed and replaced with references to these files. Once deposited, the UI provided by the RMA allows the depositor to view the collections they deposited and make edits when necessary. This UI focuses on presenting the complete set of metadata for these collections, and also

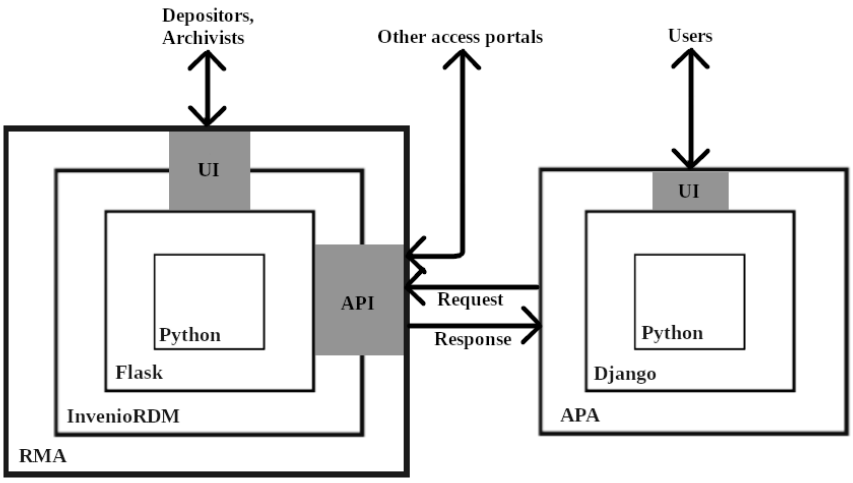

Figure 4.1: The repository management application (RMA) and the access portal application (APA) are the two modules comprising the digital archive infrastructure, shown here in relation to each other and with their software dependencies.

provides a way to download files.

In addition to reaching authenticated users through the UI provided by the RMA, the data and metadata for the deposit are also accessible through the API provided by the RMA. The second module in the design, the APA, calls this API to access the data and metadata of the collection when a user requests it. The user also needs to make a user account and agree to the archive's terms in order to access the collection. Once authenticated, the user accesses the data through the UI provided by the APA. The APA focuses on being easy to use for less tech-savvy users, as well as increasing discoverability through search and browse features. The user can see a subset of the metadata most relevant to users, preview the files in the collection, and download the files that are relevant to them, as well as the metadata for any given item. Additionally, the user is provided with a link to view the same item or collection page in the UI of the other module.

This division of labor between the two modules is useful for various reasons. First, compartmentalizing the needed tasks into two modules allows for a design that leverages existing software that already focuses on accomplishing some of the tasks of each module. Second, dividing these tasks between two modules also suits the skills and expertise of the people implementing this design. Specifically, as a linguist, I am knowledge in the better practices of language documentation and language data management that need to be incorporated in designing the RDM. Meanwhile, the OU IT team has extensive experience in designing web applications with UIs tailored to users' needs, which is the primary goal of the APA. Third, while the modularized design includes an API out of a need for the modules to communicate, the inclusion of this API naturally helps in achieving extensibility in the design, because the system is ready to communicate with additional modules through this API.

4.2.2 Leveraging open source and open development software

The ability to use open source software that makes use of software standards is critical to the design. First, using open source software, such as InvenioRDM, is critical because, as an archive, NAL has limited resources at its disposal, and thus limited ability to commission new software development. It can be argued that using open source software is not necessarily cheaper than paying for propriety software as a service provided by an organization, which could be for-profit or otherwise. This will depend on the rates charged by such organizations for their service, as well as the rates charged by those who would develop a solution using open source software and deploy it. However, NAL has access to resources for open source development by virtue of being associated with OU, and NAL currently has a funding structure that makes this the most financially feasible option. Second, it is important that open source packages used are built on open standards for web design, because using such software mitigates the cost of custom developments by increasing the chances that developers have experience with the software packages. The use of open standards also increases the chances that the software packages are compatible with other web infrastructure into the future, and thus the chances that the current project can be maintained in the future.

In order to leverage open source and open development software, the design calls for the use of the Django web framework for the APA, and the InvenioRDM repository management platform (see: §2.7.5) for the RMA. as a repository management system, InvenioRDM offers many of the features that are called for by the design of the RMA, which means that development of the RMA can focus on the small set of features as well as modifications to existing features that are needed to extend InvenioRDM to meet the requirements for this module.

As discussed in §2.7.1, web frameworks only assume that data (such as the metadata of collections) are stored in a database, such as an SQL database, and/or retrieved through API calls, but do not make assumptions about the data model beyond that. Additionally, they do not come pre-packaged with UIs for users to edit data in the database. Web development frameworks prioritize deliverables that are useful for almost every website, such that a wide range of developers can utilize the same software tools to develop web applications. These deliverables include rapid prototyping of web page templates that are used to generate web pages automatically and dynamically by calling data from a database, as well as providing modern advanced UI features designed to improve UXs.

The goals of the user access module align well with the features provided by general purpose web development frameworks, because the metadata of the collections is served to this module in a serialized format by the API, and these data do not need to be edited by the users. Additionally, a goal of the design of this module is to provide a UI that meets the specific needs of NAL's audience, and web frameworks are designed to provide broad and powerful customization of the web page templates that create the UI.

Meanwhile, the goals of the RMA align with repository management systems, like Fedora Commons and InvenioRDM. Repository management systems make more assumptions about the underlying data model than do general web frameworks, as described in §2.7.2, and these assumptions are relevant for the data model in the case of NAL and other language archives. For example, they assume data are bundled in items with files that constitute a deposit made by one person or a group of people. These people are able to manage the deposits, including editing when necessary. Another relevant assumption of the data model is that data may be openly accessible or have restrictions

to their access.

In terms of UIs to the data, InvenioRDM for example provides a deposit system, which allows depositors to manage the underlying database by submitting deposits and editing those deposits. Crucially, depositors are not allowed to manage and edit other people's deposits, which is relevant for NAL and other language archives. This example can be viewed in contrast to web application software that is not designed for repository management. For example, in the case of Wordpress, which is a web application that uses a data model that is relevant for blogs, content generators can edit each other's submissions unless user access roles are explicitly defined with custom code, such as plugins. In terms of access, InvenioRDM has a UI that prioritizes the structure and details of the deposits and their metadata in a way that is useful for depositors and researchers, but not necessarily for other audiences.

Together the two modules provide an underlying database, UIs for the different use cases of depositors and users that consist of web pages that access the data dynamically, and a web based UI to edit the database – and they do so in a way that meets the archive's needs for all three of these components. The ability to leverage open source software packages comes from modularizing the design, because there are valuable open source software packages that fit the needs of each module. Repository management systems provide an appropriate data model for language archives, but development of their UI features is constrained by their included UIs. Meanwhile, web frameworks do not include these specific implementations for data models or UIs, making development of UIs less constrained, and requiring more resources to develop the data model that already exists in repository management systems.

The selection of InvenioRDM specifically as opposed to other repository management systems is motivated by: differences between InvenioRDM and the most likely alternative option considered, Fedora Commons; a more general goal of increasing the participation of stakeholders other than developers in software design; and personnel constraints in the NAL context. In comparison with Fedora Commons, InvenioRDM is designed as a turn key repository (see §2.7.2). This means that it is designed to be deployed with little development as is feasible for a specific use case. Since this use case aligns closely with the goals of the RMA module, making use of InvenioRDM for the RMA limits the amount of development needed to implement the design.

Additionally, InvenioRDM is written in Python, in contrast to Fedora, which is written in Java. Python is a language used more commonly than Java in linguistic analyses among practitioners of language documentation. This is an assertion that is not documented explicitly in the literature, but is evidenced by methods classes in linguistics departments using Python and/or R, as opposed to Java and other languages. I recognize that there is a high barrier to entry for non-developers to contribute to software design, and I have no expectation that the selection of InvenioRDM for the RMA will fundamentally change the outlook for collaboration of different stakeholders in software design. However, such a selection can lower the barrier to entry for such collaboration. This does not require non-coders to code. In contrast, a more plausible opportunity for collaboration that the selection of InvenioRDM enables is the development of documentation for the software, and feedback generated through reading and developing this documentation. At the surface level, such documentation should be easy to read for any user without coding experience. However, at a certain level, documentation of software is not only about using the software, but also includes examples of how the software is built and maintained. Given that individuals that are not developers will fall on a

spectrum in terms of their coding knowledge, when the discussions around software design include a programming language that more people have some familiarity with, this could lower the barrier to entry enough for people at some point in the middle of the spectrum of coding knowledge to begin to participate.

In addition to the broader, more theoretical aim in selecting a software package written in Python, there is a practical benefit to this selection in the context of NAL and the personnel involved in the case study. The OU IT group are professional developers that specialize in developing applications in Django and the Python language. Additionally, I have used Python in most of my recent professional and academic development and coding experience. Regardless of this shared experience and expertise, the design is modularized in a way that allows the two modules to be implemented in different languages. In fact, this is an advantage of modularized design and the use of APIs. However, development of the modules is only the first stage in the life cycle of the software applications. Maintaining the software applications is at least as important, and should continue after the current grant for the NAL archive has concluded. This means that the future of who will maintain the software and how this will be funded remains uncertain. This is a constraint of NAL relying on soft money, as well as potential changes to OU and archive personnel that cannot be predicted. In addition to this context, the design assumes that both the RMA and the APA will be hosted by NAL, which means it is likely that a single OU employee will be given this maintenance task at any given time. Given these constraints, and the assumption of a modular system, there is a value in designing the modules to be written in the same language, because it increases the chances that a given IT professional is able to maintain the system. This value alone does not motivate the choice of any repository management system for the RMA. In fact, the compatibility of InvenioRDM with the goals of the RMA and the limited amount of development that this compatibility leads to is the primary factor. Nevertheless, the value of increasing the likelihood of maintaining the software applications until the end of their life cycle, or even increasing that life cycle, is important.

Ultimately, the goals of the present project of developing digital archive infrastructure can be achieved with or without a modularized design, and with or without open source software. However, in the context presented by the case study of the NAL archive, the modularized design enables the selection of a software package for the RMA module that implements the features needed for repository management as close to the requirements as possible, and it simultaneously enables to selection of a web framework for the APA that enables less constrained customization of the UI, which enables more efficient development for the specific team tasked with developing this module. This design assumes that user needs are different than depositor needs, which motivates the design of different UIs, and given the context of NAL, the modular design facilitates this development.

Additionally, the choice of open source software packages is itself a better practice (Holton et al., 2017:4). While the design for both modules uses open source software, the design for the RMA is noteworthy because InvenioRDM is actively developed using an open development model at the time of this writing. Having open documentation not only of the software, but also the process of its development, leads to a wealth of resources for developers using this software. Moreover, it opens the door to the possibility of collaboration with the the InvenioRDM developers, which can lead to more sustainability of the features developed in the current project. This process is described in more detail in §4.6.

4.2.3 Extensibility of access

The design of the archive infrastructure is extensible in terms of increasing access to specific communities through UXs specifically tailored to those communities' needs. This is because the design includes the development of an API for the RMA which is called by the APA to access metadata and data. Thus, this API can be used in future projects to develop community-specific user portals. For example, any community that wants to include data they have deposited with the archive on their own website can coordinate with the archive to use the API to display these data on their website. This applies to new website development projects, but also applies just as well to adding additional pages to an existing community website that call the API.

4.2.4 Data security

Data security is an important factor constraining the design. As a basic measure for security, the design is deployed in docker containers. This helps to secure data both from unauthorized access and faults during the deployment of software updates. Beyond this, there is an expectation of some of NAL's user base that the archive will limit the exposure of data to the internet and to public servers, especially data with access restrictions.

This constraint is enforced in the design by making data storage and management the responsibility of only one of the modules, specifically the RMA. The database for the system is managed by this module, and data security measures are a priority of this module, including the enforcement of access rights and restrictions through user accounts, the encryption built into the InvenioRDM software package, and the sequestration of restricted data to private servers.

In contrast, the user access module does not store data, and thus it can avoid such complexity in its design. This module accesses the data and metadata through calls to the API. While it is allowed to store metadata by way of caching them in order to optimize search queries, the data are never stored by this module, but rather accessed on the user's machine directly when the API is called.

Thus, this design reduces the number of potential failure points in the system in terms of data security, which is valuable especially in the case of extending the scope of access described in the previous section. Specifically, consider the case in which a community wants to install their own instance of the software created by implementing this design. They could install either or both modules, depending on their needs. In this example, assume this group wants to access data they have already deposited with the archive. They want to build and host their own user access portal to these data, and to do so they will start with the user access module as a template for their access portal. Thus, they install only the user access module of the software. In this case, it is especially valuable that this module does not store the data on the server hosting the web application, because it is therefore not redundantly responsible for managing the data and the security of these data. Thus, this design mitigates the need for the community in this case to be aware of and handle data security issues, and meets the need for data security required of the archive.

It is worth noting here that data security can describe securing data against things like theft as well as securing data against loss. While it can be argued that securing data against loss is always mitigated by storing data in

duplicate, the expectations of NAL from its user base include both types of data security. Thus, securing data against theft is addressed by measures such as the one described in this section, while securing data against loss is addressed by storing data redundantly on NALs private server as well as writing backup copies to tape using infrastructure provided by OU.

4.2.5 Personnel resources

While personnel details need not be part of the design in theory, they are in practice. The modular nature of the design fits the personnel resources available to NAL. As a linguist, I am familiar with the requirements for managing data according to better practices in language documentation and I am able to translate this to needs for software development. Meanwhile, the software development team at OU is experienced in general website development and qualified to translate the UX feedback from the NAL user community into front end functionality. Additionally, as this design is for a project situated at OU, NAL benefits from utilizing the personnel resources that are also associated with OU.

While this motivation and division of personnel resources is specific to the context NAL finds itself in, it may also be true that this division is analogous to other groups of developers interested in using the software developed using this design for their own communities. Specifically, there may be a small overlap between the group of people interested in managing the data and the group of people interesting in accessing the data. If developers are sought from these groups of people, they may be more likely to be interested in and qualified for contributing to one of the modules in the design, rather than a hypothetical non-modularized design.

4.3 Metadata

The design for the NAL archive includes a metadata scheme for the deposits to the archive. This schema is based on the assumption that deposits consist of digital files, which can have their own metadata, and that these files are grouped together in larger logical units that can also have metadata. As described in §4.3.1 below, these larger units are items and collections.

The goals of the design of the metadata scheme were to: accommodate existing data in the archive, conform to the protocols of the Open Archives Initiative and the Open Language Archives Community (OLAC), include the metadata specific to the accession process conducted by NAL in conjunction with the OU library, and finally, where feasible, minimize modifications to the schema native to the InvenioRDM software package for the sake of efficient development.

The metadata schema developed in order to accomplish these goals is presented in §§4.3.1-4.3.4, including the hierarchical structure of the metadata, the fields at each hierarchical level, and examples of the complexity and edge cases at each level. As of this writing, the metadata schema as been implemented, while the process for harvesting metadata following OLAC remains for future development.

4.3.1 Hierarchical levels (Collection, Item, File)

The metadata have three hierarchical levels: collection, item, and files. This structure is motivated by being common among language archives, as descrbed in §2.4.3 (Paterson, 2021:35,112,122), as well as by the NAL archive's existing metadata structure.

The existing metadata structure when this project began had only items in the database explicitly. A logical level of collections existed through the unique identifiers for the items, which had the structure [collection unique identifier-enumerator], for example ABC-001, where "ABC" is the unique identifier for the collection, and "001" is the enumerator. Thus, all items that were logically part of the same collection had the same collection-level unique identifier in their unique identifiers, which were named "catalog numbers." The logical level of files existed only through the existence of files in the archive's file system whose names were prefixed with the catalog number for the item they were associated with.

The metadata schema adopted includes relational tables for collection, item, and file, where each file is associated explicitly with an item, and each item is associated explicitly with a collection.

Files are the digital files deposited with the archive, and are often audio files like .wav files, video files like .mpeg files, rich text documents like .docx files, pdf files, and transcription files like .eaf files.

Items are the first logical grouping of files. The prototypical usage of this level is to group together files that describe and/or relate to the same language act. This is not a strict requirement for the archive, however, and there are other uses for the level of the items in the archives existing data, including for example files that relate to the same project, and files recorded on the same day.

Collections are the logical grouping of items, and thus the second logical grouping above files. The prototypical usage of this level is to group together items that are deposited by the same depositor or group of depositors. In the existing data in the NAL archive, this grouping often also corresponds to items deposited together in the same deposit, which happens when only one deposit is made for a given collection.

4.3.2 Collection Metadata

Collection level metadata are a relatively small set of fields that are general to the collection and not specific items. These fields are summarized in Table 4.1.

The first field in Table 4.1 is the COLLECTION UNIQUE IDENTIFIER[2], which takes the form of a three letter alpha code chosen by the archive when a new collection is being created. This field refers unambiguously to a given collection. In this design, each object, such as a collection, item, or file, has one field defined to serve this role as a unique identifier. This field is meant to be human-readable, and uniquely identifies objects on user interfaces meant for human consumption, in citations, and in harvesting by OLAC. However, in the arena of code development for SQL databases, it is necessary to define a field that acts as a "primary key" for a table, but this primary key is distinct from the unique identifier field presented here and for all other tables in the design. In fact, it is an essential part of the design that the human-readable unique field and the computer-readable primary key are distinct. One of

[2]Fields in the metadata schema for the present design are presented in small caps, for clarity.

Table 4.1: Collection metadata fields

Field	Data type	Notes
Collection unique identifier	3-digit alpha code	Unique identifier
URI	URI	Automatically generated (can be manually entered)
Cite as	Text	Automatically generated
Abstract	Text	
Acquisition information	Text	
Description, scope, and contents	Text	
People	Person unique identifiers	
Languages	Language unique identifier	Automatically generated from items in collection
Number of items	Number	Automatically generated from items in collection
Extent of items	Text	Automatically generated from items in collection
Collection date range	Date range	Automatically generated from items in collection

the practical reasons this distinction is valuable is that, in the SQL implementation, it is a best practice to ensure that primary keys only ever refer to a single object across the entire lifespan of the database. Thus, this distinction between a unique identifier field and a primary key allows for the scenario in which an object, such as a collection, is deleted and then recreated again later. In this scenario, the two objects created sequentially would have the same COLLECTION UNIQUE IDENTIFIER value but different primary keys. Thus, this design and implementation allows for the desired functionality of being able to recreate the same object while also following the best practices of database design.

The next two fields are the URI and a citation for the collection. The URI is in the form of a URI which provides a unique identifier that is resolvable to a location on the internet, as described in §2.4.4. In other words, this is a link that a user can go to in any web browser that brings them to the archive web page for this collection. The process for generating these URIs in the RMA is explained in §4.4.7. The citation, or CITE AS field, is a text field which provides a citation for the collection following the Tromsø recommendations for citing language data (Gawne et al., 2021). The citation is automatically generated through the process described in §4.4.3.2. Both collections and items are designed to be cited independently, so that URI and CITE AS are also provided for item level metadata, which are described in §4.3.3.

The next three collection fields in Table 4.1 are for an abstract, a description of how the collection data were acquired by the archive, and a description of the scope and contents of the collection. These are text fields that are to be provided by the depositor.

The PEOPLE field is a set of references to objects stored in the people table, which is described further in §4.3.7. This field allows the depositor to select people, along with their roles (described in §4.3.3), to be associated with the collection in general, as opposed to specific items.

The final four fields in Table 4.1 are derived from information contained in the metadata for items in the collection. Thus, these metadata fields are not created by depositors, but rather are generated automatically by the RMA.

The first field, LANGUAGES, is a list of references to objects in a separate table which stores language objects, which is described in §4.3.5. This list is an aggregation of the list of languages assigned to each item contained in the collection.

The next two fields are the NUMBER OF ITEMS and EXTENT OF ITEMS. The NUMBER OF ITEMS is a simple tally of the items in the collection. The EXTENT OF ITEMS is a prose statement derived from the extents of the items in the collections. For example, if the collection has items that have a total of three audio and/or video files that are five minutes and 24 seconds in length, as well as 2 PDF files that are 23 pages in length, the value of EXTENT OF ITEMS would be "00:05:24 and 23 pages." If the items contain no media files with a time duration, the time duration part of this value is dropped, and if the items contain no media files with an amount of pages, the page number part of this value is dropped.

Finally, the COLLECTION DATE RANGE is a date range for the whole collection, such that the beginning date of the range is set equal to the earliest beginning date for any item in the collection, and the ending date in the range is set equal to the latest ending date for any item in the collection.

4.3.3 Item metadata

The item level has the most complex set of metadata of the three levels, which is presented in Table 4.2. The metadata fields are presented alongside their metadata type as defined in §2.4.3, their data type, the OLAC field and refinements and/or encoding schemes used to map the field to the OLAC metadata schema (in the form "field →refinement/encoding scheme"), and notes about other properties of the field, if any. The notes are given with abbreviations, such that "UID" stands for unique identifier, and "AG" stands for automatically generated.

The PARENT COLLECTION field is a reference to an existing collection object. The UNIQUE IDENTIFIER, or catalog number as it is called in the NAL archive, unambiguously refers only to the current item from across all the items in the archive. It takes the form of [collection unique identifier-enumerator], for example ABC-001. Having unique identifiers for items is not strictly necessary from a software perspective, because items are each stored in a database table with unique primary keys, but it makes the design more human-readable by making it clear from the main item descriptor what collection that item belongs to. Additionally, the unique identifier for items allows resources to be unambiguously identified in citations as well as when harvested by OLAC. Finally, this convention fits with the naming convention for files and the broader convention for naming at all three hierarchical levels, which are described in §4.3.5.

The ACCESS LEVEL field is a controlled vocabulary with four text values, given in Table 4.3. This controlled vocabulary is based on the categories currently used by the NAL archive. The first access level means that files can be viewed freely and in any format, both in the context of streaming data online and downloading full copies of the data. The second access level means that data are only viewable in the context of steaming but not as files for download. It is important to note that technically it is impossible to stop users from capturing data streams to their computers. However, this design calls for not providing this option specifically, and the terms presented for uses to agree to include following the conditions of these access levels. Additionally, in order to mitigate unexpected data access, the deposit system presents the depositor with information about the the security risks associated with each level at the time of deposit, and this process is described further in §4.4.3. The third level means that files are available after a specific date. Finally, the fourth level is a catch all for restrictions to access that are determined based on other

Table 4.2: Item metadata fields

Field	Type	Data type	OLAC mapping	Notes
Parent collection	Hierarchical	Collection unique identifier		
Unique identifier	Administrative	Text	Identifier	UID
URI	Administrative	URI	Identifier →URI	AG
Cite as	Administrative	Text	Identifier →citation	AG
Access level	Administrative	Controlled vocabulary	Rights →access rights	
License	Administrative	Controlled vocabulary	Rights →license	
TK labels	Descriptive	Controlled vocabulary		
English title	Descriptive	Text	Title	
Indigenous title	Descriptive	Text	Title →alternative	
Description, scope, and content	Descriptive	Text	Description	
Genre	Descriptive	Controlled vocabulary	Type →linguistic type	
General content type	Descriptive	Controlled vocabulary	Type →DCMI type	
Languages	Descriptive	Language unique identifiers	Language	
People	Descriptive	Person unique identifiers	Contributor →role	
Location	Descriptive	Location unique identifiers	Coverage →spatial	
Recording context	Descriptive	Text		
Public event	Descriptive	Boolean		
Creation date	Descriptive	Date	Date →created	
Acquisition notes	Administrative	Text		
Project/grant	Administrative	Text		
Collection date	Descriptive	Date	Date →accepted	
Collecting notes	Descriptive	Text		
Deposit date	Administrative	Date	Date →submitted	
List of files	Hierarchical	File unique identifiers		AG
Access level restrictions	Administrative	Text	Rights →access rights	
Copyrighted notes	Administrative	Text		
Permission to publish online	Administrative	Text		
Original format medium	Administrative	Text		
Digitized on	Administrative	Text		
Software used	Administrative	Text		
Conservation Recommendation	Administrative	Text		
Location of original	Administrative	Text		
Other information	Administrative	Text		
Call number	Administrative	Text		
Accession number	Administrative	Text		
Accession date	Administrative	Text		
Type of accession	Administrative	Text		
Publisher	Administrative	Text		
Publisher address	Administrative	Text		
ISBN	Administrative	Text		
LOC catalog number	Administrative	Text		
Total number of pages, physical description	Administrative	Text		

metadata. The logic for restricting these data is described in §4.5.1. The data for items with the fourth access level are not accessible either through streaming or download if any of these restricting conditions are met.

Table 4.3: Description of access level categories for NAL

Access level number	Access level description
1	Open access
2	Materials available to view onsite but no copies may be distributed
3	Access protected by a time limit
4	Depositor (or someone else) controls access to the resource

In addition to access levels, the TK LABELS field is a controlled vocabulary that allows items to be associated with Traditional Knowledge (TK) labels (see §2.4.5). The controlled vocabulary terms include: TK Attribution, TK Clan, TK Family, TK Multiple Communities, TK Community Voice, TK Creative, TK Verified, TK Non-Verified, TK Seasonal, TK Women General, TK Men General, TK Men Restricted, TK Women Restricted, TK Culturally Sensitive, TK Secret / Sacred, TK Open to Commercialization, TK Non-Commercial, TK Community Use Only, TK Outreach, TK Open to Collaboration. Any number of the terms in this controlled vocabulary can be associated with an item, and each time a TK label is selected, a descriptive text can be applied for this association.

The GENRE field uses a controlled vocabularies where one or more of a set of terms should be selected in order to describe the item based on predefined categories. The controlled vocabulary is based on the NAL archive's existing set of categories. The list of genres consists of: article, book, ceremony, ceremonial, chant, conversation, correspondence, curse, dataset, debate, description, document, drama, educational material, elicitation, ethnography, field notes, grammar, greeting/leave-taking, history, instructions, instrumental music, interview, lexicon, meeting, myth, narrative, oratory, photograph, poetry, prayer, procedure, proverb, reader, recipe, ritual song, sketch, song, speech play, thesis, transcript, translation, unintelligible speech, and wordlist. These are mapped to the OLAC cateogories for linguistic type as shown in Table 4.4. Some of the GENRE categories do not have an appropriate equivalent cateogy in linguistic type, and these do not lead to a tag being applied in the OLAC record. When multiple genres are selected that have the same OLAC mapping, this leads to one tag for all of these genres in the OLAC record. When multiple genres are selected that have different linguistic types in the mapping to OLAC, each of those linguistic types leads to a tag for that type in the corresponding OLAC record.

The GENERAL CONTENT TYPE field uses a controlled vocabulary where one of a set of terms should be selected in order to describe the item based on predefined categories. The controlled vocabulary is based on the NAL archive's existing set of categories, and these categories map onto DCMI type. The categories (and their mappings) are: audio (sound), audio/video (moving image), book (text), manuscript (text), ephemera (text), and website (interactive resource).

The LANGUAGES field is a set of references to objects stored in the languages table, which is described further in §4.3.5. This field allows the item to be associated with any number of languages in the archive database.

The PEOPLE field is a set of references to objects stored in the people table, which is described further in §4.3.7. This field allows the item to be associated with any number of people in the archive database. For each person selected from the people database, a role must be selected for that person in relation to the item. One or more of the following list of roles must be selected: Annotator, Author, Collector, Compiler, Consultant, Data inputter, Editor, Filmer,

Table 4.4: Mapping of the GENRE controlled vocabulary to OLAC data categories

Genre category	OLAC data category
Article	
Book	
Ceremony	Primary text
Ceremonial	Primary text
Chant	Song
Conversation	Primary text
Correspondence	Primary text
Curse	Primary text
Dataset	
Debate	Primary text
Description	Language description
Document	
Drama	Primary text
Educational material	
Elicitation	Primary text
Ethnography	
Field notes	Language description
Grammar	Language description
Greeting/leave-taking	Primary text
History	Primary text
Instructions	Primary text
Instrumental music	Instrumental music
Interview	Primary text
Lexicon	Lexicon
Meeting	Primary text
Myth	Primary text
Narrative	Primary text
Oratory	Primary text
Photograph	
Poetry	Primary text
Prayer	Primary text
Procedure	Primary text
Proverb	Primary text
Reader	
Recipe	Primary text
Ritual song	Song
Sketch	Language description
Song	Song
Speech play	Primary text
Thesis	
Transcript	Primary text
Translation	Primary text
Unintelligible speech	
Wordlist	Lexicon

Illustrator, Interlocutor, Interpreter, Interviewer, Performer, Photographer, Publisher, Recorder, Research participant, Researcher, Responder, Signer, Speaker, Sponsor, Transcriber, and Translator. In the software implementation, this is accomplished using a link table, which stores metadata fields that manifest when an object from the item table is associated with an object in the people table – in this case the role of that person in relation to the item. Specifically,

every assignment of a person to an item is an object in this link table with a primary key, and these objects have a role field that stores the metadata about this relationship. In addition to the main PEOPLE field for items, there are two other fields that are references to people in the people table: COLLECTOR and DEPOSITOR. These fields are calculated automatically when the PEOPLE field is declared, based on which people specified in the PEOPLE field have collector and depositor roles, respectively, if any. These automated fields are redundant, but serve to optimize queries to the database by the applications as well as calls via the API, described further in §4.5.3.

There are four dates in the item metadata: CREATION DATE, COLLECTION DATE, DEPOSIT DATE, and ACCESSION DATE. Each of these dates is stored in multiple formats in order to be compatible with the existing metadata in the archive as well as interoperable with the output processes of API calls and metadata aggregation. The first format is a text format in the form: YYYY-MM-DD/YYYY-MM-DD. This format is compatible with the Dublin Core metadata schema, and facilitates human-readability. On either side of the slash are the beginning and end dates of a date range. Each of the dates has a four digit year followed by a month and a day. The day can be dropped and the month can be dropped. The ability to drop the days and months allows for approximate dates, which is needed to capture the existing metadata in the archive. For example, an existing date "August, 2000" can be described as "2000-08," while an existing date "1990s" can be described as "1990-1999." From this field, the RMA also redundantly calculates the beginning and end dates and stores them in a date format native to Python (specifically, as datetime.date instance), in order to facilitate database queries on dates and date ranges. For example, if a user was interested in all items from the year 2000, the query would be for all items that have date ranges beginning in or before 2000 and ending in or after 2000.

There are three fields to represent the location and context in which the item was created: LOCATION, RECORDING CONTEXT, and PUBLIC EVENT. Each item can be associated with any number of locations through the LOCATION field, which has references to location objects stored in a locations table. Each location object has the following fields: MUNICIPALITY OR TOWNSHIP, COUNTY OR PARISH, STATE OR PROVINCE, COUNTRY OR TERRITORY, GLOBAL REGION, LATITUDE, LONGITUDE, and DCMI POINT. At least some of the fields are required in order to create a location object. That is, either coordinates are required, or at least one of the descriptive fields (MUNICIPALITY OR TOWNSHIP through GLOBAL REGION) is required. The DCMI POINT field is automatically calculated following the Dublin Core metadata schema, and using the form "east=[latitude]; north=[longitude]; name=[descriptive fields]." The "east" and "north" values are taken from the latitude and longitude fields if those fields are provided, while the "name" value is taken from the descriptive fields in the order they are listed above, if any are defined. For example, if the field values were defined as "municipality or township=Norman, state or province=Oklahoma, country or territory=USA, global region=North America," the "name" value would be "Norman, Oklahoma, USA, North America." In addition to the LOCATION field, the RECORDING CONTEXT field is a text field to provide more information about the context in which the recording took place in the location, and the PUBLIC EVENT field is a boolean field to specify if the event was open to the public or not.

The LIST OF FILES in the metadata for items is an automatically calculated list of references to objects in the files table that are associated with a given item. This allows for efficient queries to the database about which files are

associate with items.

The remainder of the descriptive metadata fields for items are text fields, including: INDIGENOUS TITLE; ENGLISH TITLE; DESCRIPTION, SCOPE, AND CONTENT; ACQUISITION NOTES; PROJECT/GRANT; and COLLECTING NOTES. INDIGENOUS TITLE is for titles that are in the language that is the subject of the item, while ENGLISH TITLE is for titles that are natively in English and English translations of Indigenous titles. DESCRIPTION, SCOPE, AND CONTENT is used for a prose description of the item's scope and contents. ACQUISITION NOTES is for information on how the archive obtained the resource, while COLLECTING NOTES is for information on how the depositor obtained the resource. Finally, PROJECT/GRANT is for information on the project that led to the creation, collection, and/or depositing of the resource and funding for this project.

There are additional administrative fields that provide more information about how the archive should enforce access restrictions for the item. These are free form text fields that provide the archivist guidance on managing the data, and generally they describe information that was obtained as prerequisites to publishing the data in the archive. These fields are ACCESS LEVEL RESTRICTIONS, which is for further specifications to the access level selected, COPYRIGHTED NOTES, which is notes on the copyright status of the item, and PERMISSION TO PUBLISH ONLINE, which is relevant for legacy items deposited until the proposed online deposit system exists.

Some of the item level metadata fields are specific to NAL's processes. The following text fields are for items that were digitized by the archive: ORIGINAL FORMAT MEDIUM, DIGITIZED ON, SOFTWARE USED, CONSERVATION RECOMMENDATION, LOCATION OF ORIGINAL, and OTHER INFORMATION. Additionally, NAL participates in an accessioning process for its items in coordination with the OU library, and the following text fields relate to this process: CALL NUMBER, ACCESSION NUMBER, ACCESSION DATA, and TYPE OF ACCESSION.

Finally, the book category of metadata includes the following text fields that apply to the subset of items that are books: PUBLISHER, PUBLISHER ADDRESS, ISBN, LOC CATALOG NUMBER, and TOTAL NUMBER OF PAGES AND PHYSICAL DESCRIPTION.

4.3.4 File metadata

Files have a small set of metadata fields specific to the nature of the digital files being stored, including FILE TYPE, EXTENT, and FILE SIZE. Additionally, files have metadata fields in common with items, including PEOPLE, LANGUAGES, and ACCESS LEVEL. The set of metadata fields at the file level is given in Table 4.5, alongside their metadata type as defined in §2.4.3, their data type, the OLAC field and refinements and/or encoding schemes used to map the field to the OLAC metadata schema, and notes about other properties of the field, if any. The notes are given with abbreviations, such that "UID" stands for unique identifier, "AG" stands for automatically generated, and "AG*" stands for automatically generated, and can be manually entered.

The PARENT ITEM field is a reference to an existing item object. There are two FILE PATH fields in order to accomodate the application structure and harvesting through OLAC. The first FILEPATH is the unique identifier for files in the archive. This is logically consistent with the file structure of the database and server housing the files, in which the file paths are unique. The value in this field is an instance of python file path, rather than plain text, such

Table 4.5: File metadata fields

Field	Type	Data type	OLAC mapping	Notes
Parent item	Hierarchical	Item unique identifier		
File path	Administrative	File path		UID, AG
File path	Administrative	URI	Description →contents	AG
File name	Administrative	Text		AG
File type	Administrative	Media type	Format →medium	AG*
Checksum	Administrative	Text		AG
Enumerator	Administrative	Text		
Title	Descriptive	Text		
Extent	Descriptive	Text	Format →extent	AG*
File size	Administrative	Number		AG
A/V Specification	Administrative	Text		
Access level	Administrative	Controlled vocabulary		
Languages	Descriptive	Language unique identifiers		
People	Descriptive	Person unique identifiers		

that it must be a valid path that points to a file. Thus, the file path includes the folders and subfolders that house the file as well as the file extension. To avoid sharing server configuration publicly, the FILE PATH field is only used internally. The second FILE PATH is a URI that points to the location of the file as it is served rather than stored on the server. This is appropriate for locating the file over the internet and for harvesting metadata. The FILE NAME field includes the base file name and the file extension, and is calculated automatically from the file path. The FILE TYPE value is calculated automatically from the file's native metadata and/or its extension. This value can also be entered manually, in case it cannot be calculated automatically. The CHECKSUM field stores checksum values automatically generated by the RMA. The ENUMERATOR field is a text field used for files that are enumerated, like tracks of a CD. Additionally, two free-form text fields are TITLE and A/V SPECIFICATION.

The EXTENT of the file is a text field that is automatically generated when a file is uploaded and can also be modified. It describes either the duration of media files or the number of pages of documents. For example, if the file is an audio and/or video file that is 1 minute and 3 seconds in length, the value for EXTENT will be "00:01:03." If the file is a PDF file that is 10 pages in length, the value for EXTENT will be "10 pages." These values are calculated using the native metadata of the file, first by checking the media type of the file and checking for the corresponding duration or number of pages in the cases that the media type has such native metadata. This process allows for a compromise between automation and flexibility. With this process, the automation of EXTENT is possible, and in cases where the RMA reads the EXTENT fields, it attempts to parse their values for this format, and if it is successful it uses the corresponding logical extent (i.e. a duration or number of pages) in its processing. At the same time, the user can elect to change the text to something else if neither duration nor pages is relevant for a given file, and the RMA will simply ignore this value in its processing of things like total EXTENT OF ITEMS (and their files) in a collection.

The three metadata fields that files have in common with items are: PEOPLE, LANGUAGES, and ACCESS LEVEL. The decision to have these fields at both the item and file levels arose from a compromise between different design ideals and constraints. First, it would be ideal for fields to occur at the hierarchical level that they are most directly

related to, or most directly represent. However, this immediately poses the issue of how to determine this for a given field, and this is dependent on the definiton of the levels, as described in §2.4.3. It would also be ideal to avoid redundancies in the fields, not only to limit the work that depositors must do in generating metadata, but also because this makes the RMA more efficient. In addition to these ideals to aim for, it is also true that generating metadata is subjective and can depend on the person creating the metadata. As an example, there may be many people speaking in a recording, but some may be considered outside of the focus of the recording, or in the background, and thus excluded. Additionally, in the case that PEOPLE is available as a field at both the file and item level, different depositors might have different strategies for who gets included at the item level. One depositor could include everyone who is associated with every file. Another depositor could include only the people that are the main focus of the item and/or the majority of the files.

Including PEOPLE only at the item level decreases the complexity of the metadata schema and the effort required to create the metadata, however, it eliminates the possibility of differentiating which people are associated with which files. In contrast, including PEOPLE only at the file level allows each file to be described independently, and a summary of all the people in each file can be generated by the RMA automatically. However, this does not leave room for the depositor to specify their own version of the summary for PEOPLE at the item level. Specifying PEOPLE at both levels is another alternative that gives the depositor more freedom to specify the metadata as they prefer, but raises the question of what to do if depositors only create metadata at one level. Asking them to create the metadata at both levels requires them to repeat their work. However, this issue can be addressed through the RMA, by automatically generating this field based on supplied metadata and allowing the depositor to overwrite the automated values as they see fit.

The situation is similar for LANGUAGES and ACCESS LEVEL. Having the three fields at the file level allows the metadata to capture which languages, people, and access levels apply to which levels in the case that they are different, rather than simply applying the aggregate list of people and languages, as well as a derived access level, to the item level. At the same time, having the fields at the item level allows the metadata to represent the case that these fields were only ever applied to the item level and it is thus assumed that they apply to all the corresponding files. This case is relevant, for example, for items that were deposited before the archive had a metadata schema that represented files. However, the distinction is not just relevant for legacy deposits. For all deposits, it allows depositors to distinguish these metadata at the different levels.

In order to account for the complexity of having LANGUAGES, PEOPLE, and ACCESS LEVEL at both the item and file levels, the design for the RMA includes additional automatically calculated fields that are not directly editable, but used to present information in the UIs. Additionally, the logic for calculating these fields is different for LANGUAGES and PEOPLE versus ACCESS LEVEL.

For LANGUAGES and PEOPLE, the RMA maintains aggregate lists of languages and people that are assigned at the file level. For example, in the case of LANGUAGES, the RMA aggregates the list of languages assigned to each file associated with a given item and then assigns this generated aggregate language list to that item. This is separate from LANGUAGES on the item, which the depositor can edit freely. In the UI, the metadata can then be presented to

the user in a way that distinguishes between the languages assigned on the item and languages assigned on files. For example, if a depositor assigns Cheyenne to LANGUAGES at the item level, and additionally assigns both Cheyenne and English to LANGUAGES on a file associated with that item, the metadata presented in the UI would read "Cheyenne (From specific files: English)."

While the same logic applies for the PEOPLE field on items and files, the logic is different for ACCESS LEVEL across items and files. Access levels can be assigned at both the item and file level, and in this case both the item level and file level metadata should determine the effective access level for a given file, where the most restrictive level assigned at either hierarchical level should be taken as the effective access level for that file. Thus, this calculated effective access level is stored as a separate field at the file level and the RMA can use this effective access level for its access restriction processes. Meanwhile, a range of access levels is calculated from each file and stored in a separate field at the item level, which can be presented as a summary in the UI for the item view.

4.3.5 Naming conventions for the hierarchical levels

The design prescribes naming conventions at all three hierarchical levels. As described in §4.3.2, the naming convention for collections is a three-character code, as in "abc." The naming convention for items, as described in §4.3.3, is [collection unique identifier-enumerator], as in ABC-001. The naming convention for the file level is equivalent to the naming convention for the literal digital files, and this is [collection unique identifier-enumerator_file.extension], as in ABC-001_001.txt. The "file" part of the file name can be an enumerator or any text that leads to a unique file name.

This convention emerged out of a desideratum from the NAL archive staff to ensure file names have exactly three levels, which corresponds to the logic of hierarchical levels used by other language archives.

Two edge cases that the convention prescribes against but are nonetheless supported by the design are as follows. The first is when an item has only one file and that file name is identical to the item name plus a file extension. For example, in this case the item is ABC-001 and its single file is ABC-001.txt. The second case occurs when an item has multiple files that have the same "file" text and are thus only distinguished by their file extension. One common example of this is when an item describes a speech act that was captured as raw data with an audio file and transcribed as primary data with a ELAN file. In this case, if the item was ABC-002, the files would be ABC-002.wav and ABC-002.eaf.

These edge cases violate the desideratum of file names with three components. At the same time, they are patterns observed in the archive's existing data. Forcing the convention as a strict requirement would mean rewriting existing data, which is problematic in terms of using limited human resources, but especially in the case that depositors may be hard to reach for their consent on this change. At the same time, having a strictly enforced naming convention would mean that the RMA's algorithms for processing data could be simpler by not needing to accommodate the edge cases. As a compromise, the design accepts the complexity of the edge cases, but certain automated processing algorithms are only available for files that follow the convention. Specifically, when building a deposit using the depositor system described in §§4.4.2-4.4.3, the RMA can automatically sort uploaded files into corresponding items

that it generates based on the filenames if they follow the naming convention. If the files are not named following the convention and thus not parsable in this way, this process requires additional depositor input.

4.3.6 Language metadata

Languages are logical entities that are associated with collections, items, and files as described in §§4.3.2-4.3.4. In addition to these associations, languages have a complex set of intrinsic metadata. In order to avoid storing these language metadata redundantly on each collection, item, and file, the design includes a database table for languages themselves. When a language is associated with a collection, item, or file, it is done so through a reference to the language object found in the languages table. The metadata fields for language objects are given in Table 4.6.

Table 4.6: Language metadata fields

Field	Data type	Notes
Language name	Text	
Alternative names	Text	
Glottocode (or ISO code)	Controlled vocabulary	Unique ID
Family	Text	
Primary subgroup	Text	
Secondary subgroup	Text	
Region	Text	
Dialects	Text	
Location	Location unique identifiers	
Notes	Text	

The metadata fields for languages mostly have values that are not unique to the archive, and thus true of languages outside of the database, with the exception being the NOTES field. Given this, it would be convenient to use these general language data rather than maintain a languages table internally. However, from a software development perspective, it is useful to maintain these data in the RMA's database. It stands to reason that there must be some logical entity that represents a language in the database, if only to assign that entity to a collection, item, or file. In its simplest form, this could be a unique identifier like a Glottocode, which would sufficiently disambiguate which language was being assigned. If an API for languages based on Glottocodes was exposed to the internet, the RMA could call such an API in order to populate web pages with the languages information based on the Glottocode. However, this turns out to be inefficient to an extent that is undesirable. For example, such a design would require a query to an external server every time a collection, item, or file view is loaded in a web page, in the case that object has at least one language associated with it. This example provides some nuance to the goal of avoiding the storage of information redundantly in database design. This goal applies to the internal structure of a database, and not to any larger entity, like a group of databases across the internet.

In contrast to the above example, the use of a language table in the RMA is designed for database efficiency as well as automating the entry of publicly available information into the database (rather than obtaining this information at the stage of web page generation). To accomplish this, the languages table uses Glottocodes as a unique identifer. New language objects are not created directly by depositors, but by archivists and/or automated processes during

the deposit submission process. When an archivist creates a new language object in the languages table, they must supply the unique identifier as either a Glottocode or an ISO code in the UI for this field. The UI provides an auto-complete function to suggest valid Glottocodes and ISO codes from a controlled vocabulary based on their text input, and allows them to choose one of these options. The UI also provides data validation to ensure that a valid option is entered. If a Glottocode is provided, the RMA writes that value to the unique identifier field. If an ISO code is provided, the RMA looks up the associated Glottocode from a reference table and writes that value to the unique identifier field.

The reference table for languages is a table stored by the RMA that is designed not to be edited by depositors or archivists, but rather used as a reference for specific processing steps carried out by the RMA. The table contains the set of language metadata provided by Glottolog, including Glottocodes, ISO codes, language names, family information, and so on. When a new object in the main language table is created and associated with a Glottocode, the other fields are automatically populated with information from the reference table. This applies to all fields except for NOTES. At that point, the archivist can modify the fields as necessary and provide the notes field value if necessary.

Having both the main language table and the Glottolog reference table allows for database efficiency, automated generation of known metadata, and control over which languages are represented in the archive. Additionally, the reference table can be updated as information on Glottolog changes. Finally, the language table allows for convenient display of different language metadata in different contexts in the UIs, described further in §4.5.3.

4.3.7 Person metadata

Like languages, people are logical entities with associated metadata, and are stored in a separate people table. The fields for each person object are given in Table 4.7.

Table 4.7: Person metadata fields

Field	Data type	Notes
Collaborator ID	Number	Unique ID
Family Name	Text	
Given Name	Text	
Nickname	Text	
ORCID	Text	
Privacy	Controlled vocabulary	
Native/first languages	Language unique identifiers	
Other languages	Language unique identifiers	
Tribal affiliations	Text	
Clan or society	Text	
Place of origin	Location	
Date of birth	Date	
Date of death	Date	
Gender	Controlled vocabulary	
Other info	Text	
Groups	Group unique identifiers	

The COLLABORATOR ID is the unique identifier for person objects. This is an arbitrary enumerator that incre-

ments with each person created in the database such that each person has a unique number. Like with other object tables, this is not serving the function of a primary key in the database. Rather, this number provides a convenient disambiguator for human users in cases where multiple people exist in the database with the same name. The names of people are defined with the fields FAMILY NAME, GIVEN NAME, and NICKNAME. These fields are designed to be compatible with the Dublin Core schema.

The PRIVACY field is a controlled vocabulary with different settings that correspond to levels of privacy for the treatment of personal information. There are three privacy levels in the controlled vocabulary: listed, unlisted, and anonymous. Personal information can appear in two forms on the archive website. The first is in relation to collections, items, and files they are associated with, and the second is in the form of a view of the person object in the UI, which can be described in familiar terms as a personal profile page. The PRIVACY value of listed corresponds to the least privacy of the three settings. In this case, the person's name will appear on collections, items, and files that they are associated with, along with their roles in relation to those objects. Additionally, their person metadata will be visible on a web page that is created to be a view of their person object, which is accessible, for example, by clicking on their name where it is listed as associated with a collection, item or file. The second setting is unlisted, which corresponds to a middling amount of privacy. In this case, the person's name will appear in association with collections, items, and files, but there is not a so-called profile page corresponding to their person object. In fact, there is a page for their person object but the only person metadata it lists is their name and the collections, items, and files they are associated with. Finally, the most private setting is anonymous. An anonymous person appears with collections, items, and files they are associated with as "anonymous" along with their role in relation to these objects. While there is a view of every collection, item, and file they are associated with, neither their name nor any other person metadata is listed on this or any view.

The metadata fields for person objects include language references, dates, and locations, which have been described in detail in §4.3.3. The language fields are NATIVE/FIRST LANGUAGES and OTHER LANGUAGES. Both of these are sets of references to language objects. The date fields are for DATE OF BIRTH and DATE OF DEATH. Like with dates on items, these can be a range of dates, which the RMA uses to manage approximate dates for these events for people. This allows depositors to enter approximate information for age like a decade, as in "this person is in their 50's." The mechanics of this data entry are presented in §4.4.3. Finally, the PLACE OF ORIGIN field works similarly to the LOCATION field on items, where a location can be defined that is as specific as geographic coordinates or as general as a country or region of the world.

Finally, there are a group of fields that relate to a person's affiliations to groups. The two fields named TRIBAL AFFILIATIONS and CLAN OR SOCIETY are text fields that allow for free form explanations of affiliations. Groups is a list of references to group objects in a database table for groups that comes with the InvenioRDM software package. Using these group affiliations, it is possible to restrict access or moderation rights to the set of people that are associated with a specific group.

The plan for the NAL archive is to use this design without explicitly systematizing the link between tribal, clan, or society and group affiliations. The NAL archive is not in a position to moderate these affiliations, and moreover

they are sufficiently complex to mitigate the possibility of systematizing them in a way that satisfies all members of all groups. However, for the sake of argument, we can look at what this process would entail, what process NAL will use instead, and the resulting differences from these approaches.

If database and application design was the primary concern in this case, it would make sense to synthesize the groups that emerge from the plain text metadata in the current archive collections for TRIBAL AFFILIATIONS and CLAN OR SOCIETY and then define a controlled vocabulary from those groups that people could choose from in order to establish their group affiliations. Then, items in the archive could be searched for based on these affiliations. For example, if an item was associated with a person in a specific group, a user of the archive UI could search for other items that are similarly associated with that group. Moreover, if a collection was deposited on the condition that it only be viewed by members of a certain tribe or other group, this could be accomplished through the RMA by limiting access based on a person's association with that group (see §4.5.1.1).

It is easy to see the implications and potential issues with this approach, and especially with the last example. In the case of NAL, the archivists do not have the authority to define the groups, let alone the appropriate membership of those groups, and how these affiliations might change over time. Additionally, even if they did, they do not have the resources to moderate these affiliations, which would include making sure that people are not associated with groups that they do not have the right to be associated with. At the same time, the archive has developed and maintained the plain text metadata for tribe, clan, and society affiliations. Thus, it is possible to take a less explicit approach with non-systematized search, and allow users to search for collections, items, or files associated with people that have a specific text string (or approximate text string) in their metadata for these fields. Additionally, the GROUPS metadata field remains in the software architecture of the design, so that in the case that the archive wants to make a group that can be used by the RMA to control access to or moderation of specific collections, the archive still has the option to do this. One example of where this function would still be useful to NAL is for collaborating groups, for which the archivist could define a group and allow those with affiliation to this group to make revisions to an existing collection. This process is explained in detail in §4.4.

The group affiliation functionality introduced here is valuable for any archive or organization that wants to have access and management rights that are controlled in part by the software and moderated by the archivist, as described in §4.4. Additionally, it is flexible in order to accommodate other organizations that want to use the software and have the authority and desire to manage these rights more explicitly. This type of extensibility of the system is described further in §4.6.

4.4 The deposit process

This section describes the process of depositing materials with an archive using the infrastructure provided by the RMA. It describes details of who makes deposits, how they make the deposits, and what happens after the deposits are made. The term deposit refers to the data and metadata submitted by the depositor for one collection during one session, and collections can consist of multiple deposits. The details of the depositing process are framed in the

context of which details are provided by the InvenioRDM software application and which details are to be developed based on the current design.

Currently, basic user account functionality has been implemented, while the process of granting moderation and editing rights for deposits made by other depositors has yet to be implemented (§4.4.1, §4.4.4, §4.4.6). The first iteration of the depositor portal has been implemented, while changes to its UI to reflect the metadata schema, automated generation of metadata during deposit, and batch editing have yet to be implemented (§4.4.3). Versioning has been implemented (§4.4.5), and the system for assigning URIs has yet to be implemented (§4.4.7).

4.4.1 User accounts for depositors

The RMA requires users, depositors, and archivists to authenticate with a user account to perform certain actions, like using the depositor portal and accessing data, although authentication is not required to access metadata. In creating an account, the account holder must pass a verification test to confirm they are human rather than a bot. Additionally, they must agree to the terms of the archive in order to proceed. They must then be authenticated through signing into their account to use the depositor portal and access data.

In the process of creating a user account, the user can specify values for the metadata described in §4.3.7, including privacy restrictions and ORCID. During the creation of a user account, the RMA has a set of procedures used to link the user account to a person object in the database. First, if the user account is associated with an ORCID that is also associated with a person object in the database, this is sufficient to automatically link the account to this person object. Second, if the user account is not related to any person object in the database, a new person object is created and the user account is linked to this object. Finally, if the name matches the name on a person object, the new person object created during the account creation is associated with the existing object, and the archivist is able to confirm that these accounts are duplicates in a separate process which merges these objects into one.

In order to limit misuse of the depositor portal, the software further distinguishes between depositor user accounts and other user accounts. Archivists can assign the depositor priviledge to a user account and thus convert it to a depositor user account. Once a depositor is logged in with a depositor user account, they can use the depositor portal to create new deposits, described in §§4.4.2-4.4.3. Once they create a deposit, they are able to revise it because their user account and person metadata are associated with the collection they deposited. In this sense they have a management role with the data in this specific collection. Depositors can also be granted a management role to collections that other depositors created by associating their user account with a group assigned to manage that collection, which is described further in §4.4.6.

The InvenioRDM software package comes with a user accounts system, the ability to define groups and assign user accounts to those groups, and an authentication system based on ORCID, so that users can create accounts associated with their existing ORCIDs. Additionally, depositors can create deposits once they are authenticated and manage these deposits based on their management rights by virtue of creating the deposit.

The features called for by the design that must be developed in addition to those that come with InvenioRDM are the additional person database table with its metadata structure and the processes for associating user accounts with

people in that table, both those that are automated and those that require archivist input through a UI. Additionally, the functionality of assigning user accounts to groups needs to be extended so that these associations can be used to provide management rights to multiple depositors for the same collections.

The development of a people database beyond the user account function is not redundant, but rather relevant for archives focused on managing language data, because with these data there are logical and practical differences in the nature of archive users and contributors to language data. The primary difference is that depositors are a small subset of the contributors to the data being deposited, and thus many contributors will not be website users. While it is possible to make user accounts for every contributor in an automated process, this does not give those people access to their accounts in the event that they wish to use the website. Since website use is an opt-in process, it makes more sense to store their metadata from the deposit in a people table and then allow for the association of this metadata with their user account in the event that they opt in to using the website. Conversely, another practical issue arises in the event that a user deletes their account. The ability to delete one's own user account is a standard feature provided by software applications, including InvenioRDM, and if the metadata for people associated with archive data was stored only in user accounts, it would be deleted by default if the user account is deleted. In summary, keeping user accounts separate from contributors to archive data as logical entities makes sense because of the differences between these entities.

4.4.2 Uploading data through a web-based user interface

The RMA provides a deposit system by way of a UI for uploading language data. To use this system, a depositor creates a user account and logs in. Once authenticated, they are presented with a button in the UI to start a new collection, and at this point they are taken to a web page with prompts to fill in collection level metadata as well as a web form to upload files. This follows the standard UI conventions for modern websites, where there is a browse button as well as drag and drop functionality. The browse button opens a window on the depositor's machine that allows them to browse their local file for the files they want to include in the collection. The UI has a box which invites the depositor to drag and drop files from their local file system navigation window into the box.

Once files have been selected through the browse or drag and drop mechanisms, they begin to be uploaded to the system in the background, and the status of this process is shown in the UI. During the beginning of the upload process, the files' native metadata is read so that the RMA can begin certain automated processes for creating metadata and completing the deposit, described further in §4.4.3.

The first process completed by the RMA is to attempt to parse the filenames of the files selected based on their compliance with the naming convention described in §4.3.5. If the files follow this naming convention their relations to hypothetical items can be inferred algorithmically. Thus, the RMA proceeds in one of two ways based on whether the condition of compliance with the filename convention is met.

If the RMA is able to parse the filenames of the files selected, it automatically creates items based on these filenames. Then, it associates these items with the collection being created as their PARENT COLLECTION. Next, it creates file objects for each file selected and associates those file objects with the appropriate filename. Finally, it

calculates the COLLECTION UNIQUE IDENTIFIER value based on the filenames and fills in this field in the collection metadata.

If the filenames are not parsed, the RMA creates one item as a child of the new collection, and then creates file objects for each file that is associated with that single item.

The metadata created in this process is displayed in the UI as follows. In addition to the fields for collection metadata, there is a box for each item with the item name of that item displayed prominently, and a list of the new files that are associated with that item in that item box.

Once these processes are complete, the UI provides a way for the depositor to rename files. When completing this operation, it prompts the depositor with information on how to conform to the filename convention. When a filename is changed and the new filename meets the file naming convention, its relationship to the collection is reevaluated, and it is moved to the appropriate item when this item has changed. If this item does not exist, a new item is made to complete this operation.

In this way, the system is flexible enough to handle non-conforming filenames, but at the same time provides depositors feedback on how to conform to the convention, and gives them the benefits of automation once they do so, which in this case are the automation of the creation of logical item entities and associating files with the appropriate items. This flexibility is needed to accomodate existing materials in the archive, however, the data validation for file names can be enforced strictly for new deposits.

The InvenioRDM software package has much of the core infrastructure needed to produce the UI and some of the metadata structure described here. The features to be developed based on the design center around the relatively complex metadata schema used in this design and the automation processes that decrease the amount of work required by the depositor to complete the deposit.

Figure 4.2 provides a screenshot of the deposit system provided by InvenioRDM. This screenshot captures the UI after the depositor has elected to make a new deposit. It shows various functions described in the design in this section. It allows the depositor to create what it labels an "upload" once the depositor authenticates through a user account. Clicking to create an upload brings the depositor to an upload submission form that has a file selection system with browse and drag and drop functionalities. Multiple files can be selected and uploaded. Additionally, the page prompts the depositor to fill in various fields for metadata on the uploads, including the upload type, digital object identifier, title, authors, language, and access restrictions. The UI also has more sections with other metadata fields sorted by topic, and these sections are hidden until revealed by the depositor by clicking on them.

Framing these features provided by the InvenioRDM software in terms of the design presented in this section and through this chapter, the upload system provided by InvenioRDM most directly corresponds to a UI for uploading data and creating metadata for a single item and files associated with that item. Thus, the development for the design beyond this software package is primarily about developing a collection level metadata table, the association between the collection metadata and the existing item and file data, and the algorithmic processes involved in processing these metadata and presenting them through the UI.

The metadata schema for InvenioRDM includes tables for the upload and the files associated with the upload

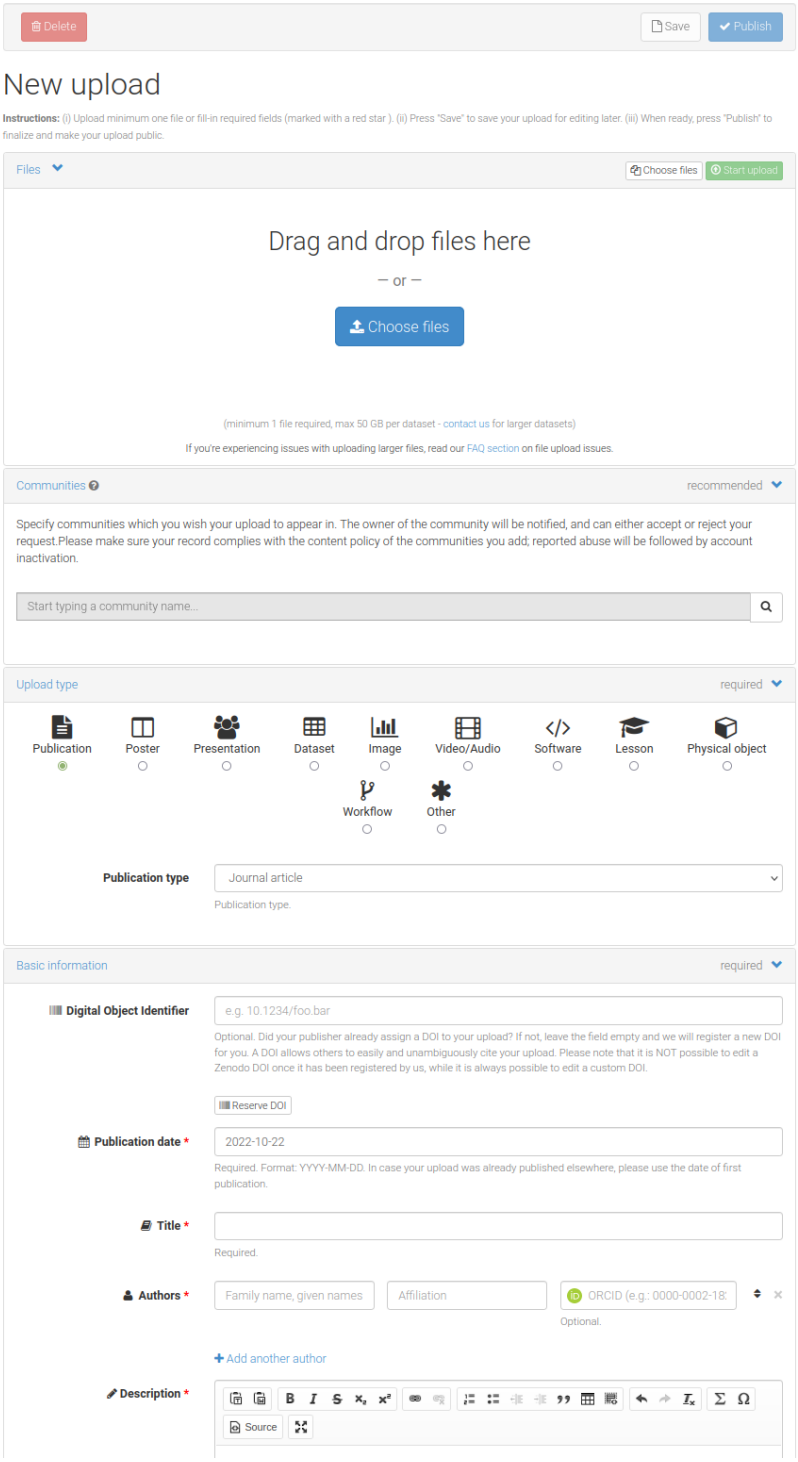

Figure 4.2: UI for uploading in InvenioRDM

which can be taken as the tables for items and files in this design and modified to meet the schema defined in §§4.3.3-4.3.4. Beyond that, an additional table is needed for collections. The UI for creating an upload is used as a view for editing an item, and a new view is needing for editing a new collection. This new view is what the depositor is taken to when they choose to make a new deposit. The processes for parsing filenames need to be added. These processes lead to the creation of new items, and these are what correspond to uploads in InvenioRDM. Thus, each item created can be edited through a UI view that correspond to the upload edit view shown in Figure 4.2.

4.4.3 Adding metadata through a web-based user interface

The deposit system also provides a UI for entering all the metadata associated with a collection, items, and files. The previous section described how the deposit system ingested digital files and created database entries for collections, items and files, using a combination of automated calculations and depositor input. The UI for the deposit system provides views for editing collections, items, and files, such that every object created for the deposit has a corresponding web page with form fields that allow the depositor to input metadata, the format of these fields corresponds to the data type of the metadata field they apply to, as is the case with all modern websites. For example, a text field has a simple box to enter text, while a date field allows the depositor to click on a button and have a calendar appear to give them a graphical interface to choose a date. As another example, fields that have a controlled vocabulary as their data format may have a drop down menu listing these options, or radio buttons with graphics and text to help the depositor select one option. In addition to these views for metadata entry, the deposit system has a spreadsheet entry view for batch editing of metadata. Finally, the RMA processes metadata automatically in specific cases to mitigate the amount of work required of the depositor to complete the deposit.

4.4.3.1 User interface primitives from InvenioRDM

Looking first at the UI provided by the InvenioRDM software package can provide useful context for the features that need to be developed as part of this design. It turns out that InvenioRDM uses metadata fields with many of the same data types that are called for in the present design, and thus it has many UI features for these data types that can be utilized in this design. For example, Figure 4.3 shows the UI for entering the type of the upload in InvenioRDM.

This UI provides text and icons for each of a small group of types that make up a controlled vocabulary. The selection process involves clicking on one of the radio buttons, used when only one option can be selected. We can call this part of the web page a UI primitive, meaning the UI and the underlying data type for which it provides an interface. This UI primitive is a good fit to adopt for the metadata field GENERAL CONTENT TYPE in the current design, which is also a controlled vocabulary where only one option can be selected, and the terms have some overlap with the terms used in InvenioRDM. Making this change for the development would involve changing the metadata schema as well as the template web page for this UI and the graphics files it loads for the icons. This modification is a relatively small tweak compared to the work that went into developing this UI primitive in the first place, which makes this a good example of the efficiency of adopting open source and open development software packages instead of building everything from scratch. Moreover, the InvenioRDM documentation includes an explanation for

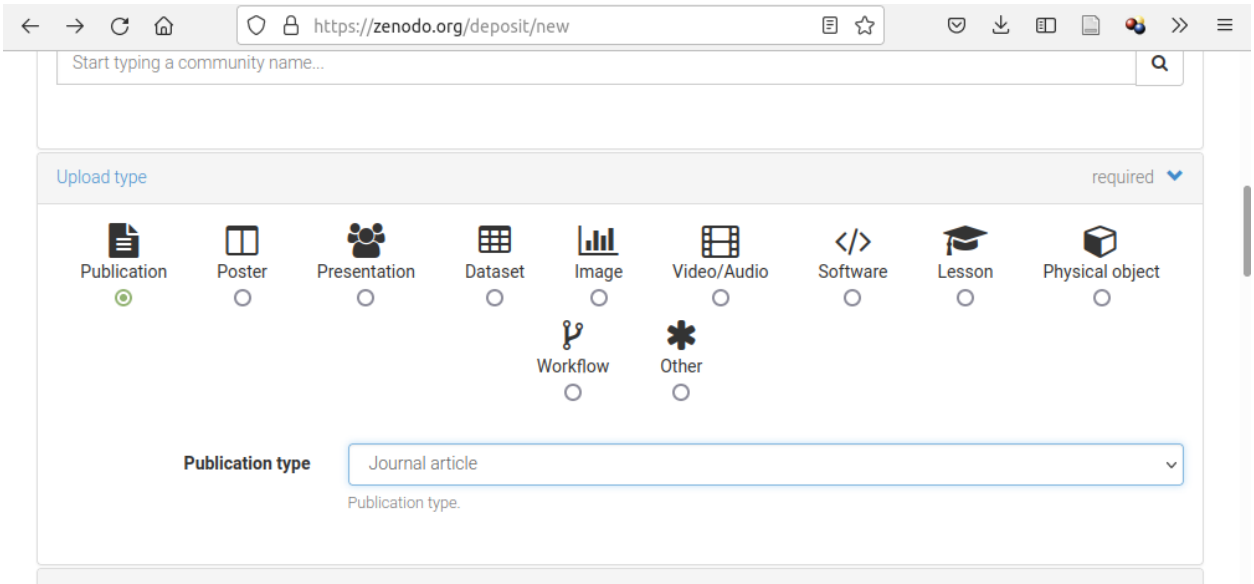

Figure 4.3: UI for selecting upload type in InvenioRDM

how to make modifications like changing the metadata schema and web page templates used by the software.

Figure 4.4 provides another example of a UI primitive that can be adopted by this design. In this case, it is for the publication type of an upload. This is a controlled vocabulary with more terms than the vocabulary for type, and the UI provides a drop down menu for the depositor to select the publication type. This UI primitive would be useful, for example, for the GENRE field for items in the current design.

Another UI primative, this time for dates, is presented in Figure 4.5, which shows the entry for publication date in the InvenioRDM UI. This system allows the depositor to enter the date in two different ways, either as text they type manually, or by selecting a date from the calendar. While this is a common feature in data entry in modern websites, this UI provides a particularly seamless instance of this UI feature, in the sense that initially there is a only a text box, and when the depositor clicks on the text box the calendar appears without interrupting the text entry operation. The depositor can then either type or click on the calendar to interact with it and find the date. Clicking on the calendar populates the text field with the date in the valid text format, which informs the depositor about how to enter valid dates as text. Conversely, typing in a valid date shifts the calendar to represent that date.

Modifying this UI primitive in order to accommodate the dates described in §4.3.3, which allow for date ranges and approximate dates, requires the following steps. First, there are two dates in the database: the beginning date and end date for a date range. Date entry for the date range could be accomplished by putting two of the date entry forms shown in Figure 4.5 side by side, and a button that alternates between exact date and date range. When exact range is selected, the second date, shown to the right, would be greyed out, and when date range is selected, both dates would be available for data entry. Additionally, there could be a button on top of the current date form that allows the depositor to specify they are entering an approximate date. In this case, an additional text box would appear for the approximate date, and the two date boxes would remain, showing the range that the approximate

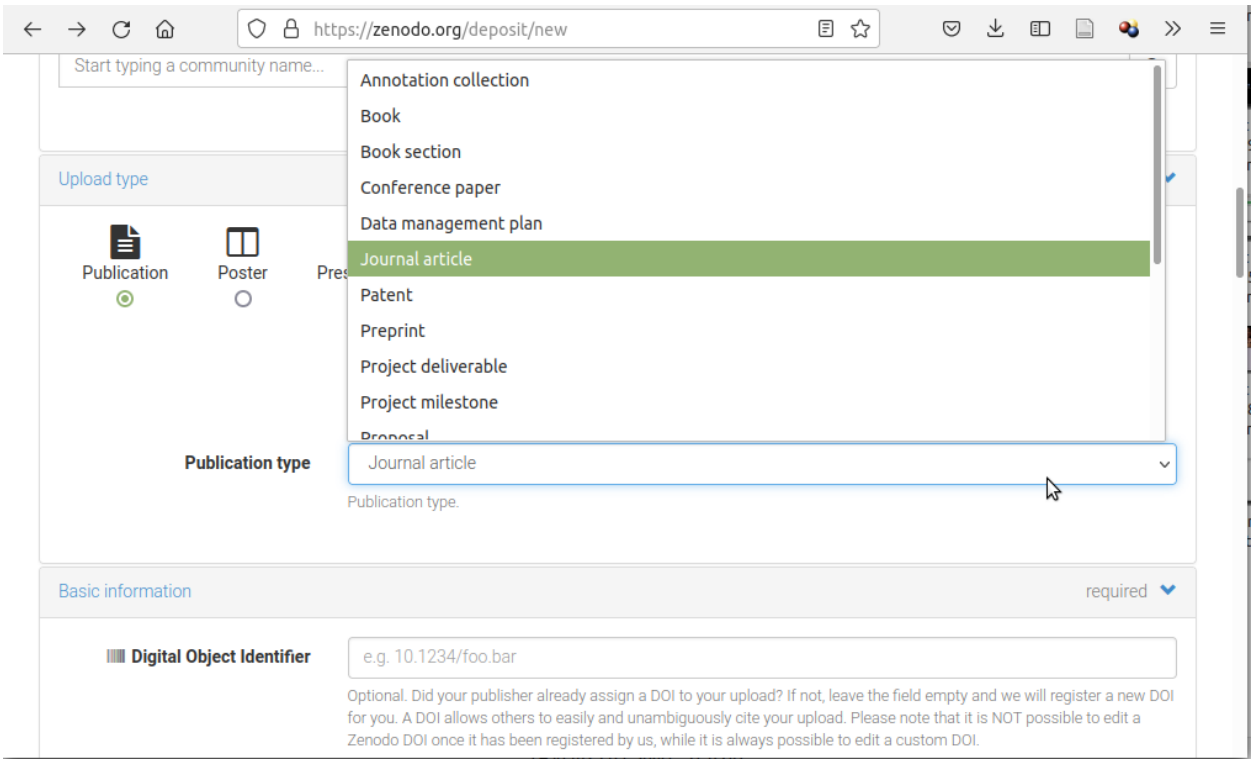

Figure 4.4: UI for selecting publication type in InvenioRDM

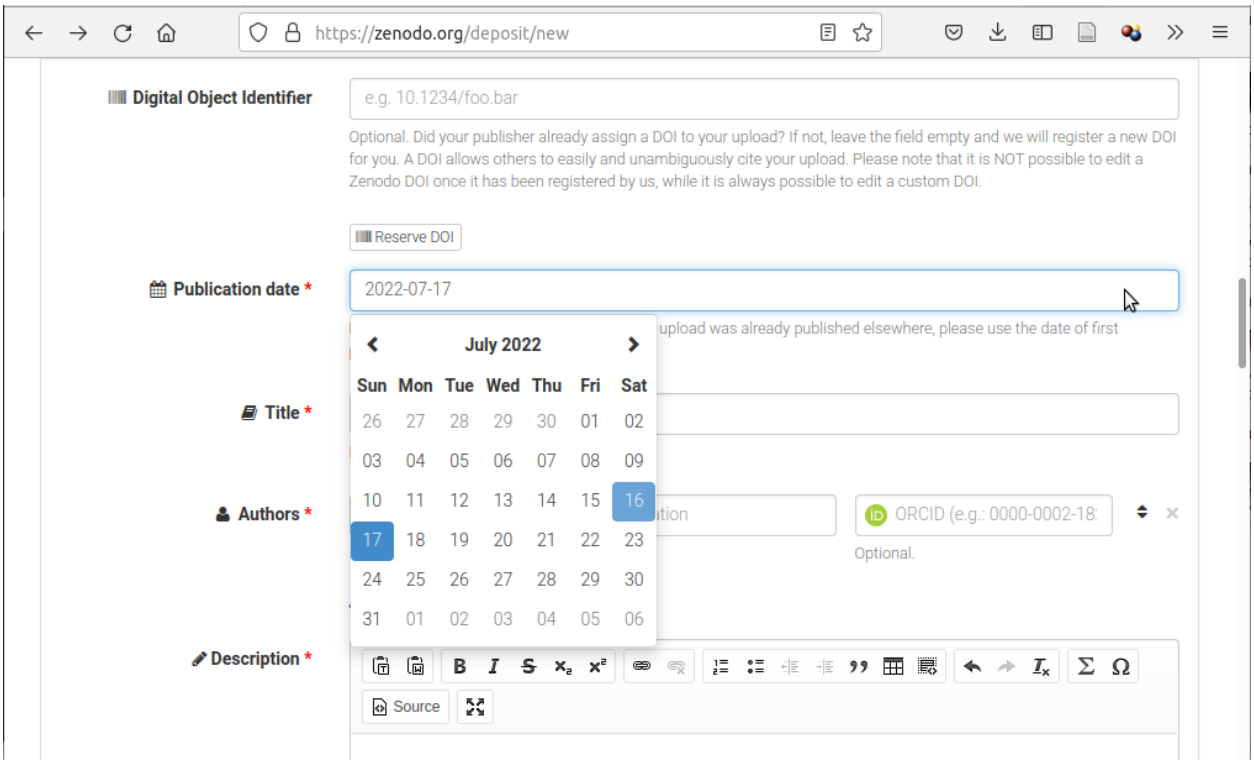

Figure 4.5: UI for selecting publication date in InvenioRDM

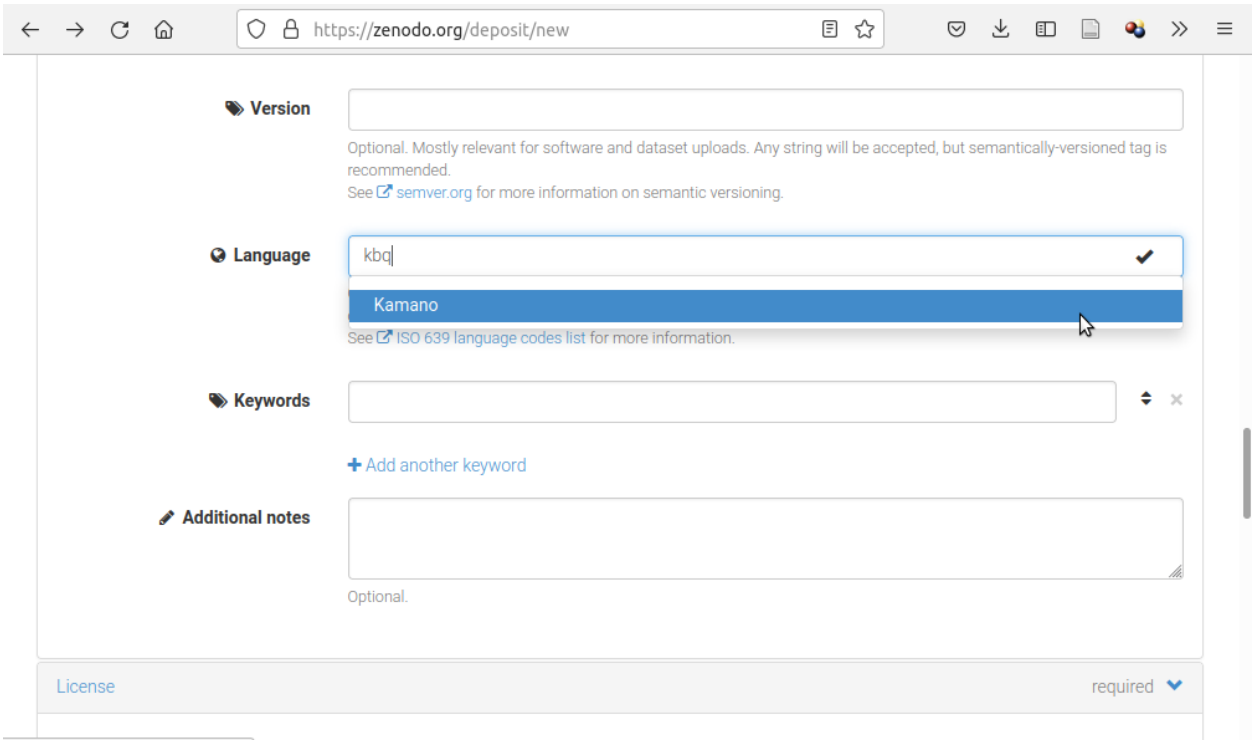

Figure 4.6: UI for selecting language in InvenioRDM

date being entered corresponds to. In this configuration, text entries like "1990's" would be considered valid by the form (see §4.3.3), and the date range this corresponds to would be entered into the beginning and end date, which in this example would be "1990-01-01" and "1999-12-31." This would allow data entry for approximate dates, which accommodates NALs existing data, as well as the reality of older language data, like items that need to be digitized, which reach the archive with metadata of limited quality and/or specificity.

Another example of complexity in data types of metadata fields is that of languages associated with collections, items, and/or files. The UI primitive for language in InvenioRDM is shown in Figure 4.6. In this UI, the field for language provides data validation with a list of language names that correspond to ISO codes. Once a valid name is typed in the field, the value for the language object associated with an ISO code is entered in the database, and this is presented to the depositor as plain text in the UI. If a string that is not equal to an official name of a language with an ISO code is typed into the box, no value is entered in the database, and the text in the UI is deleted once the depositor clicks away from the field. Additionally, alternate names are not used in this data validation system, so that typing an alternative name of a language in the field will not result in successful data entry. In this way, the UI forces the depositor to enter valid data.

Meanwhile, the UI facilitates the depositor in entering valid data using a method similar to that for dates, which is illustrated in Figure 4.5. That is, when a string is entered into the text box that is a match, or partial match, of one of the names on the list of valid language names, a box appears on the page below the text box with links to languages in the database. Clicking on one of these links applies that language as the value for this field and enters

the text of that language name into the data entry box.

In order to modify the InvenioRDM software for the present design, the metadata schema needs to be changed as described in §4.3.3 and §4.3.5, and this UI primitive should be modified and adopted for these fields.

First, the metadata schema for items and files requires that multiple languages can be associated with any give item or file, whereas the InvenioRDM software is designed to allow only one language to be added to an upload. The database relationship needs to modified to allow a "many to many" relationship between the item table (or file table) and the language table.

Next, the language list used for data validation needs to be modified to be based primarily on Glottocodes and secondarily on ISO codes. Beyond that, this list should also include alternate names for languages. In this way, typing language names, alternate names, ISO codes, and Glottocodes would all be ways to achieve the application of the language to the metadata field. In the database, the association would be to a language object which has a Glottocode as its primary identifier. The application of a language to an item or file is actually the association with an entry in the language table, which is different than the table of languages available to depositors in the depositor portal. To restate the difference motivating this design from §4.3.5, the table for data entry represents the general information in the world about language metadata, while the language object table represents the language metadata maintained by the archive and archivists. In the event that a depositor selects a language that does not currently have a corresponding object in the language table, such an object is generated automatically by the RMA.

The added complexity in the list of available languages is handled in the UI by the suggestions box that appears based on the text typed by the depositor. This added complexity, which includes the acceptance of alternate language names, increases the chances that a string typed by the depositor is a match or partial match to multiple languages. This is especially true if the string typed contains few characters. To handle this situation, the UI primitive should be modified such that the links in the drop down box with language suggestions provide more metadata for each language being suggested. Currently, these links contain only the text of the language name, as in the example in Figure 4.6 with the link text "Kamano." In contrast, the modified UI provides suggestions with the language name, Glottocode, ISO code, and region, so that the link for the Kamano language would have the text, "Kamano (kama1370 | kbq, Oceania)." Finally, the UI needs to be modified such that it accepts multiple languages for a language field. Fortunately, the UI primitive for contributors in the InvenioRDM UI already achieves this. The UI provided by InvenioRDM for selecting contributors is shown in Figure 4.7. Taken as a UI primitive, it provides the needed functions for assigning people to items and files in the current design. It provides data entry for names and also allows the depositor to assign multiple contributors, which is also the additional functionality needed languages in the current design. Additionally, the UI allows the depositor to select a role for each contributor, which corresponds to the database structure that allows roles to be assigned based on the association of a contributor to an upload. This is the database structure needed for languages and people in the current design.

Additionally, the current design calls for items and files to be associated with person objects, rather than plain text for their names, as described in §4.3.7. The UI for this modification includes providing suggestions for people objects in the database based on matches or partial matches to people objects in the database. This UI feature appears

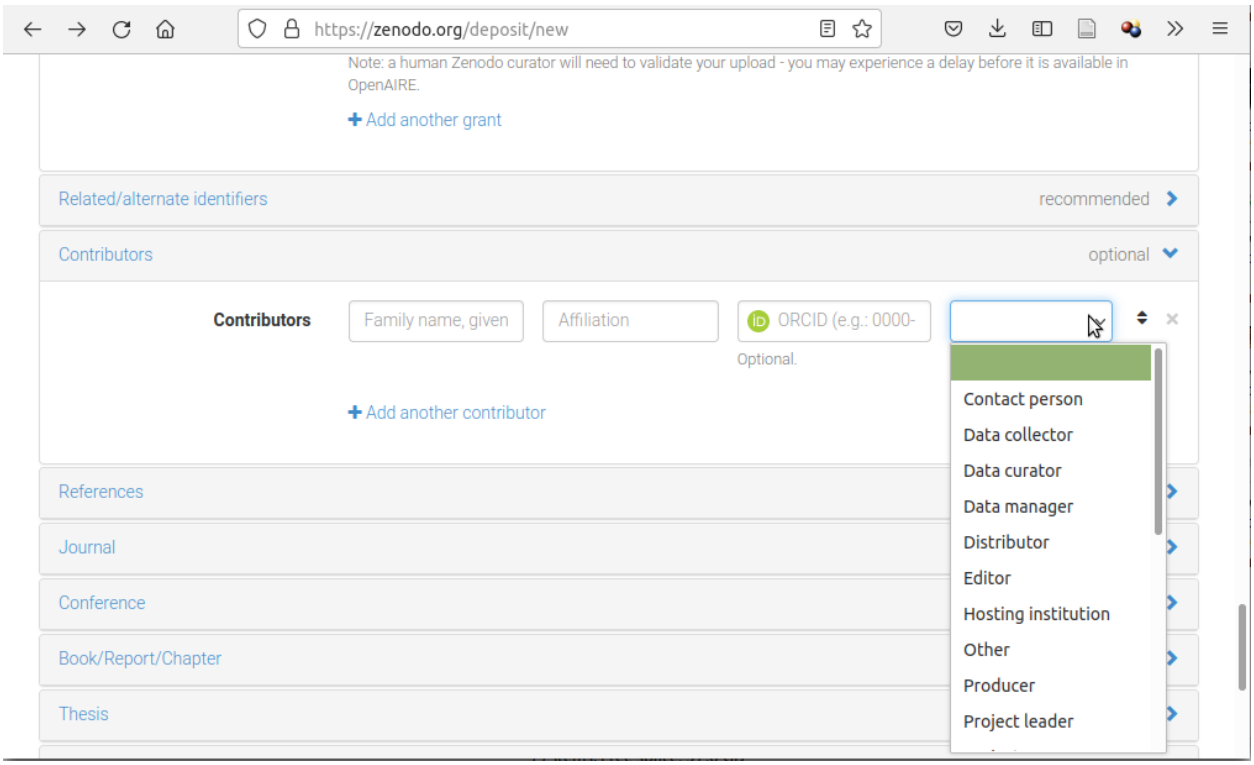

Figure 4.7: UI for selecting contributors in InvenioRDM

similarly to that of the UI primitive for languages in Figure 4.6. In the drop down box populated with suggestions for people objects, the text of links includes the person's full name and languages spoken in order to help the depositor disambiguate. For example, the link text for a person named John Smith who speaks Cheyenne and English would appear as "John Smith (Cheyenne, English)." In order to further facilitate selection of people from this suggestion list, they are sorted by a metric of relevance. This metric sorts people who are already associated with the collection first, and people who are not already associated with the collection but who are associated with languages that are already associated with the collection next. Finally, in order to provide privacy for people and restrict access to their personal information, only those people objects that have a privacy setting of "listed" will appear in these search suggestions, with the exception of those people already associated with the collection. This configuration will likely result in the creation of duplicate person objects in the database, as in when a depositor enters a name for an unlisted person who is not already associated with the collection. In this case, the UI does not allow them to connect their text entry to the existing person object, even though this connection would be correct, and instead creates a new object with duplicate information. After the creation of such a duplicate, input from an archivist is then needed to mark these two person objects as duplicates so that they can be merged by the RMA. These extra steps are considered desirable in order to protect the privacy needs of people in the database.

As a final example of UI primitives, Figure 4.8 shows the UI provided by InvenioRDM for entering metadata specific to books. Many of the details of this UI and the database structure it provides an interface for correspond to

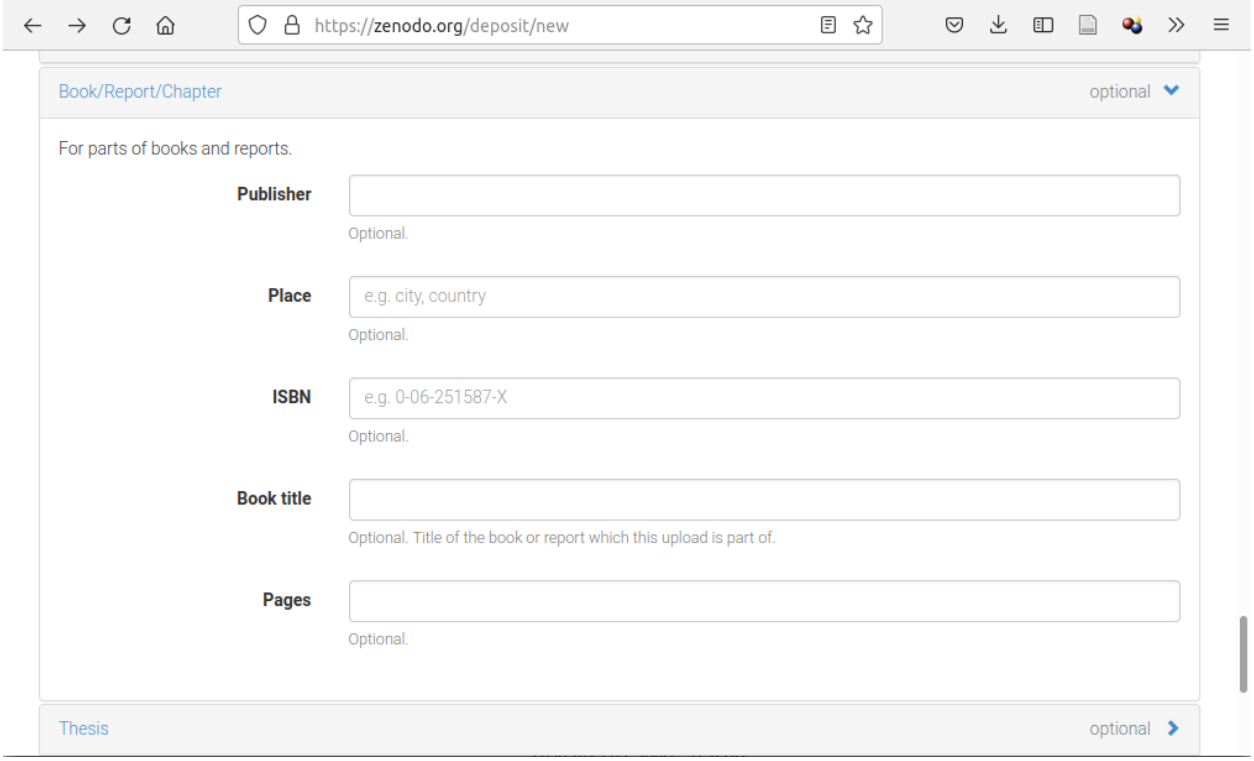

Figure 4.8: UI for inputting book related metadata in InvenioRDM

the present design. First, the book related fields are metadata for uploads, which corresponds to the current design and its book fields for items. Next, the book specific metadata are assumed to be only relevant for a subset of uploads, and thus they are categorized as book related and displayed in a specific section of the UI that can be hidden. In Figure 4.8, the button for hiding or showing these fields is in the upper right of the image, labeled "optional." Thus, in this example the UI primitive provides the UI and database relations needed by the present design, and thus can be used as is. Moreover, features like the collapsible boxes for specific categories of metadata can be used for other categories of lesser used metadata in the present design, which are defined in §4.3.3.

4.4.3.2 Automated creation of metadata

When possible, the RMA uses automated processes to generate metadata. The term automated is used here to mean that the RMA completes the task without depositor input. In other cases, metadata creation is semi automated, in the sense of requiring some depositor input but mitigating the number of explicit actions required by the depositor, where actions are clicks and text entry. Some of these processes have been mentioned in previous parts of §4.4, and all such processes are treated in more detail here.

As mentioned in §4.4.2, the depositor system provides a UI for uploading files, and this interface occurs on the view for creating a collection. In this step of the deposit, the RMA parses filenames in order to distribute them into corresponding items. The UI allows the depositor to rename files to follow the file naming convention, at which

point the RMA attempts to parse the new file name and distribute it to the correct item as necessary. In this view, the depositor can also make new items and drag the files to the boxes that represent those items in order to distribute files manually.

During this process, the RMA is using the information obtained through the UI to create the metadata necessary to link the collection with items, and the items with files. In other words, the depositor is not required to specify in any other way that these relations exist besides through the UI, which is designed to be quick and easy for depositors. The RMA manages the relations of entities in the collection by assigning a reference to an item object to the PARENT ITEM field of each file that is a logical child of said item object. Likewise, it assigns PARENT COLLECTION values to each relevant item's metadata. This is a semi automated metadata creation process in that it may require some depositor input in the case that filenames do not follow the file naming convention, and it uses either filenames or depositor interaction to generate the explicit metadata in the database. These metadata are fundamental to the interaction of the depositor with the data and metadata, because they are used to provide links between different views of the logical entities in the collections, as well as summaries of which entities are associated, and queries for associated entities.

At the collection level, many of the metadata fields are not fundamental properties of the collection, but are aggregations of properties of the item and files associated with collections, as defined in Table 4.1. As such these data are redundant, but rather than calculating the values every time they are requested by a user through loading a web page, they are calculated once every time the underlying data change. This increases efficiency of the RMA, as well as queries to the database made through the API, which is described in §4.5.3. These fields include: the list of languages in items and files associated with a collection, the list of people in items and files associated with a collection, the number of items associated with a collection, the total of the extent of all files associated with a collection and the range of all the dates of the items associated with a collection. The process of generating these metadata is fully automated by the RMA.

There are two fields that directly describe both collections and items that are generated through automated processes. The collection URI and item URI values are in the form of URIs that are generated automatically by the RMA for new and revised collections and items. This process is fully automated, unless the depositor wants to intervene to enter a custom URI. These URIs are created through a process of registering them with an external web service. as described in §2.4.4, and this process is explained in §4.4.7. The other field is CITE AS, which is fully automated by the RMA. The citations generated follow the Tromsø recommendations for citing language data (Gawne et al., 2021). To generate the citations, the RMA reads the values from PEOPLE and their roles, CREATION DATE, UNIQUE IDENTIFIER (collection or item), and TITLE (collection or item). The RMA takes people with an author role as authors for the purpose of building the citations. Having read these data, the RMA inserts these values into a string template that follows the syntax of the citation convention, using an algorithm for processing these insertions. For example, the types of metadata are separated by periods, and for types of metadata with multiple items, like people, the people are separated by commas. Additionally, the first name in the list of people needs to be flipped so that the last name comes first and the last name and first name are separated by a comma. Finally, edge cases in syntax need

to be addressed. For example, if a type of metadata is missing, the period in the template needs to be removed so that there is not duplicate punctuation, and the last person in the list of people should not be followed by a comma.

On items, the depositor is assigned to all items they deposit through the depositor portal automatically. They are assigned as a person associated with the item whose role is depositor. The person object that represents the depositor is determined from the link between the user account of the person using the depositor system and the person object in the database.

Dates are generated with partial automation from the RMA, as described in §4.4.3.1. This automation comes in when selecting approximate dates. The RMA calculates explicit date ranges based on approximate date strings, and provides feedback on if these approximate dates provided by the depositor meet the data validation requirements.

To further facilitate the entry of approximate dates for person metadata, i.e. dates of birth and death, the UI for entering these dates includes buttons the depositor can press with texts that read "20s," "30s," "40s," etc, up to "90s." These texts represent commonly used ranges for ages rather than dates. When the depositor clicks one of these buttons, the date range is automatically set to a 10 year range for birth dates. When this field is entered in the context of a depositing an item, the date range is calculated relative to the creation date of the item. For example, if the item was created on 2004-05-05 and the depositor selects the "50s" button for a persons age, this date range begins on 1944-05-06 and ends on 1954-05-05. In the case that these metadata are entered on the web page for editing the person's metadata, which would generally be done by an archivist with access to the full table of people objects, these ranges are calculated relative to the date of data entry, and an additional date entry field is provided to change that date from today to any date.

Many of the metadata fields for files are calculated automatically, including FILE NAME, FILE PATH, FILE TYPE, EXTENT, and FILE SIZE. The InvenioRDM software package already includes processes to read all of these except EXTENT. To add EXTENT to the list of automatically generated metadata for files, the RMA needs to read a given file's native metadata, depending on the type that it determined for the file. In the case that the file is an audio or video file, this metadata will include a timestamp for the duration of the media in the file. In this case that this is a rich text document or PDF, the file data will include a number of pages. After reading the metadata for files of these types, the RMA needs to store this information in the EXTENT field for a given file, following the convention described in §4.3.4.

While editing the metadata items and files, the UI provides buttons to the right of each field that provide the depositor the ability to duplicate the value that is entered in that field to the current object's siblings. In other words, if the field is associated with an item, this function duplicates the value in the field on all other items associated with the current item's parent collection. Otherwise, if the field is associated with a file, this function duplicates the value in the field on all other files associated with the current file's parent item. In order to accomplish this, the UI first presents a popup box to the depositor, which shows a list of each sibling item (or file), the value for the metadata field each sibling currently has, and the value it will have after the operation. The depositor can select or deselect specific siblings to apply the operation to, and the preview value for that sibling based on whether it is selected in undergo the operation. In other words, if a sibling is selected, the value after the operation will be equal

to the current item (or file), whereas if the sibling is deselected, the value after the operation will be equal to the value before the operation. Once the selection process is complete, the depositor can click an "apply" button to apply these changes, or a "cancel" button to cancel the operation at any time. This process provides a semi automated way to duplicate metadata across items and files.

4.4.3.3 Batch editing

The duplication process described at the end of the last section is a type of batch editing, where batch editing is a function were the same operation is applied to multiple different entities with one unified action. In addition to this, the design provides a more general method for batch editing of metadata through a spreadsheet based UI for metadata editing.

The batch editing UI and functionality can be described as spreadsheet based simply because the UI for this feature resembles a spreadsheet, with a grid of rows and columns of cells for data entry. This UI provides a small set of basic features commonly associated with editing spreadsheets, including typing in cells for data entry, scrolling through the cells, selecting groups of cells, and copying and pasting data from one cell to others, or from a group of cells to another group of cells. While the UI allows these operations to happen naturally, in also provides data validation by not accepting values on a cell by cell basis unless they are valid.

The spreadsheet editing UI appears when this function is selected by the depositor, either on a web page for editing a collection or a web page for editing an item. In the former case, the batch edit button provided is for editing all items in that collection, such that each item is displayed in the spreadsheet view on one row. In the latter case, the batch edit button provided is for editing all files in that item, and each row in the spreadsheet view represents one file.

Taking the former case as an example, the spreadsheet UI appears as a full screen pop up box and each row is populated with the metadata of one item. Column headers, given in the first row, represent the name of each metadata field, and these headers are read only. With this configuration, the spreadsheet gives an overview of the metadata for all the items in the collection. For fields whose data type is plain text, it is straightforward to imagine how the batch editing would work. For example, the depositor could type in a value for PROJECT/GRANT and then copy it, highlight the cells corresponding to this field for the other items, and paste this value into those cells. For such fields where no data validation is needed, this is a sufficient process for batch editing these fields. Once done, the depositor could click on an "apply" button at the bottom of this spreadsheet popup and the RMA could proceed with the batch operation of applying every value in a cell that has changed to the corresponding place in the database.

Implementing other data types in this spreadsheet UI is more complex but still feasible. It requires handling programmatic values, implementing data validation with corresponding feedback to the depositor, and accepting a slightly higher learning curve for the depositor.

Looking first at representing different data types in the spreadsheet, consider languages, dates, and locations. In the case of languages, the data type is references to other database objects, and the question arises of how these references might be presented in the spreadsheet. It turns out this representation is similar to the one on the simple

web page view for editing a single item. Specifically, languages are represented with hypertext, where a text string is presented to the depositor in the UI, and this string is the visible part of a tagged entity in a markup language like HTML or XML. Thus, the value in the cell resembles text, but is actually a reference to a language object. With this representation, depositors can add new languages through typing language names or unique identifers. Additionally, depositors can copy languages and paste them across items.

The next data type to consider is dates. In this case, all dates are represented as data ranges, written in the text format YYYY-MM-DD/YYYY-MM-DD. The depositor can type dates in this form, and the are accepted as long as they are valid. The depositor can type only one date and this will be accepted. Certain shorthands for approximate dates are also accepted by the data validation, including YYYY and YY-MM/YY-MM. As the depositor types a date, a calendar appears below the cell, just as it does in the simple metadata editing page for an item, in order to give the depositor that alternative graphical method of date selection. Based on these details, the UX for the depositor in data entry is similar to that of the single item edit page. The main differences arise from the fact that this entry is happening in the confined space of a spreadsheet cell, so the complex display of fields for descriptive text in addition to the two date fields for a date range, as described in §4.4.3.1, is not feasible within this cell. This may lead to higher learning curve for depositors, who must enter data in such a cell with less feedback from the UI. However, the spreadsheet batch editing feature is a more advanced feature than the single item edit page, and it is assumed that the depositor will gain some familiarity with data entry using that simpler system first. However, in case this is not the case, there is also a tooltip feature for these complex cells, so that the depositor can hover over the date they have entered in the cell and see more verbose representations of this information in a pop up box.

The final data type to consider is locations. The data entry for locations in the spreadsheet view proceeds similar to that of dates, with one notable exception. Based on differences in the nature of the data represented by these different data types, there is a much higher chance that more depositor input is required to enter a location than is required to enter a language. According to the process described in §4.3.5, the underlying processes for accepting depositor input to associate a language with a file is completely automated, even if this event is the first time this language is being associated with an item in the archive. The same is not true for locations, which are also represented as objects in a separate table in the database. Rather than storing all possible locations in the database, which would be impractical and infeasible, locations are stored as they are used. This means when a depositor types in the name of a town for example, the database can suggest a preexisting location with that town's name and geographic coordinates chosen to be at the center of that town. In this case, the operation proceeds without further depositor input. However, if the depositor defines a location with geographic coordinates because, for example, they want to be more exact than providing a town location, this location likely will not exist in the database, even though these coordinates represent a valid location. In this case, the UI prompts the depositor with a warning that this location will be created without other location metadata, by highlighting the location in yellow. This follows commonly used color cues for software applications, where red is for errors, as in data validation errors, and yellow is for warnings, like in this case where the data is valid but may require further attention. In this case, the depositor can right click on the location highlighted in yellow and a pop up box appears that allows them to enter other metadata for that location they are

creating.

With these interactions with complex data types in place, the depositor can proceed to use the spreadsheet and get the benefit of copying and pasting complex metadata due to the fact that they have been represented in the two dimensional format of a spreadsheet, where rows represent items and columns represent fields of any data type. For convenience, the depositor can right click on a column or row and revert all changes to that column or row, respectively. Alternatively, they can ignore the row and it will be hidden and changes will not be taken up by the RMA to the database. This gives the depositor a way to focus on only the metadata fields on which they wish to perform batch operations. Moreover, this layout of visible and hidden columns can be remembered as the depositor's preference for future use.

Once the depositor is satisfied with the changes they have made, they can click an "apply" button at the bottom of the spreadsheet box, and the RMA will proceed to modify the database accordingly as long as all the changes to the data are valid. In the case that some cells do not pass their validity checks, the RMA provides a prompt in the form of a pop up box that specifies the cells that need to be fixed. The references to specific cells can be clicked on to take the depositor to those cells in the spreadsheet view.

In addition to providing a UI for editing data in the form of a spreadsheet, the batch editing process allows a user to import data from spreadsheets on their local machines. During this import process, data validity checks are done in order to map columns from the spreadsheet to the batch editing UI, where possible. The system allows for spreadsheets to be imported that were exported by Lameta.

4.4.4 Moderation of submissions

Once the depositor is finished uploading files to a collection, distributing those files to items, and inputting metadata for the collection, items, and files, they submit the collection for review by an archivist. The edit view for the collection provides a submit button to accomplish this. During this review process, the collection and its associated items and files are only visible to the depositor and those with an archivist role. This gives the archivist an opportunity to make sure the content is appropriate for inclusion in the archive based on the archive's criteria. During the review process, the archivist can review the data and metadata using the same UI available to the depositor, and that of publicly accessible collections. The design does not include software for responding to specific issues within the system, as in through direct messages through user accounts. However, the archivist can use the contact information provided by the depositor to contact them about potential issues with the deposit. Once any necessary edits are made and the submission is approved, the archivist can approve the collection with an approval button on a moderation UI that is available only to them. This is simply a list of collections in moderation status with buttons that correspond to the actions they can take, like approve, deny, and delete. Deny allows the collection to remain in the system to give the depositor a chance to save the metadata they created in this format of the deposit system. In deny status, the collection remains invisible to other users besides archivists. A certain number of days after a submission is denied, it will be automatically deleted. This number of days can be changed in the setting of the RMA, and it can be any number of days or never. The default is 30 days.

The InvenioRDM software application does not have such a moderating process already implemented. There are processes for admins to modify any existing upload, but there is not explicitly an archivist role that moderates every collection before it becomes accessible. To achieve this, the software needs to be modified to define a user group of archivists, so that certain user accounts can be assigned to this role. These user accounts are given editing rights to every collection. An additional access level needs to be defined that functions the same as InvenioRDM's closed access setting. This access level is called "under review," and a collection is given this access level by assigning a value to a special metadata field that controls the review process. This field can be set to either "approved, under review, or denied." Additionally a new UI component needs to be built which provides a view of all collections based on their review status, and highlights those under review, followed by those that have been denied. This UI provides buttons for the actions archivists take to complete the moderation process. Finally, in lieu of a more complex notification system to alert archivists about new collections, the RMA should send an email to the addresses of user accounts with an archivist role when a new collection has been submitted and thus come under review.

4.4.5 Support for versioning of archived data

Up to this point, this chapter has described the metadata schema used by the RMA, and the deposit system that provides a web based interface for depositors. The RMA also provides a limited system for versioning of archived data. The goals of this system are to provide the full history of changes made to the metadata of items and files added to the items, to manage new files added that are modifications to old files as versions in order to automate processes of maintaining these relationships between versions, and to facilitate the delivery of information from collections and their constituent items and files directly through the archive website UI as well as the RMA's API.

Recall that §2.7.6 described three different strategies for implementing version control: human managed; computer managed, maintaining whole files for different versions; and computer managed, maintaining only changes to files for each version. The design of the RMA calls for the use of a computer managed system for version control that maintains whole files for each version. This design enables versioning for the files associated with items, as well as the metadata for these files and items.

The present design leverages the existing tools of the InvenioRDM software package, which includes a versioning system for uploads that maintains whole files for each version. Figure 4.9 shows the UI for InvenioRDM that provides links to different versions of an upload, where users can view the metadata and files that constitute that version. The InvenioRDM application maintains versions of metadata as JSON files that it stores internally. Depositors do not interact with these files directly in the depositor portal, but they can be viewed by exporting the metadata for any given version of an item that is displayed through the UI. The InvenioRDM application also has URIs not only for the versions specifically, but also for the latest version. The design for these URIs is described in §4.4.7.

The design fits the use case for archives, because having explicit files for each version ensures they are citable and easily accessible. This is in contrast to a system that maintains only changes to files, because the software application is then needed to reconstruct previous versions, which raises concerns for accessibility if the software application becomes obsolete or unavailable. To further contrast the two computer managed systems, the system that maintains

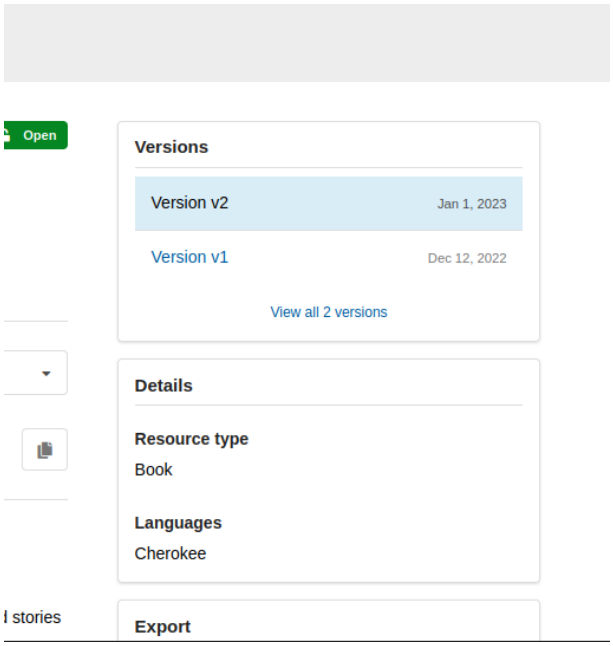

Figure 4.9: InvenioRDM UI feature for accessing different versions of an upload

only changes to files is useful for collaboration before depositing, in which collaborators want to work on the same files asynchronously. They can make many small changes to text files that result in many versions, and they rarely need to access specific historical versions of the files. In contrast to this situation, archive deposits that lead to new versions are going to be rare, and the ability to view each historical version is crucial. Generally, there is no expectation that raw data will change, which takes the form of media files. Primary data, such as transcriptions and translations formatted as ELAN files, may change as analysis changes, but these versions will be few and occur slowly.

The present design choice also has some advantages over a human managed system, both for depositing and access. For deposit, the present design choice can encourage iterative archiving through the UX it provides to depositors. Iterative archiving refers to the process in which depositors make more than one deposit to a collection, either to expand on previous items or provide new versions of files. One situation in which a depositor would want to deposit a new version of a file is when modifications have been made to a transcription or translation in a text file such as an ELAN file. Both the present design choice and the human managed versioning system allow the depositor to accomplish this. However, the human managed system accomplishes this through file naming conventions, such as appending "v2," "v3," etc to an original file name, which highlights the version history of the file. In contrast, the present design choice highlights the current version of the file to depositors and users first when they view the corresponding item, and then allows them to explore the different historical versions when they wish to. It can be argued that this is a superficial change, and that the human managed system allows the depositor to accomplish iterative archiving to the same degree as the present design. However, as described in Chapter 2, encouraging researchers to deposit language data is an ongoing issue, and institutions like universities, funding

agencies, and archives can contribute to cultural shifts in the field to encourage such deposits (Gawne et al., 2017:174). Moreover, primary data is more underrepresented in language archives than raw data 2.3.2, and the present design can contribute to a cultural shift toward archiving primary data when it is available, even when it is incomplete. This design allows archivists to encourage depositors to archive iteratively by ensuring them the archive is designed not only to accept revisions, but it automates this process and highlights the most recent version of the file in its UIs.

The shift to computer automation of file versions also provides practical benefits to users in their access of data in multiple use cases. The first comes when users download the set of files associated with an item, for example one that represents a language act. In this example, the full version history of the item contains two .wav files and four .eaf files, because the primary data for each raw data file has been updated once. In the human managed versioning system, the user accessing the files likely has to take some actions to manage the files in order to view them properly (see, for example, Babinski and Bowern, 2021:timestamp 7:52). For example, if the filenames are "a.wav, a.eaf, a_v2.eaf, b.wav, b.eaf, and b_v2.eaf, the user has to either a) rename a.eaf and b.eaf to arbitrary names and rename a_v2.eaf and b_v2.eaf to a.eaf and b.eaf, respectively, or b) reassociate the current version of the ELAN file with the .wav file. In both cases, the user accessing these data needs to understand the file naming schema for versioning enough to use it, as well as take explicit file management actions in order to access the data. In other words, the human managed versioning system puts the burden of version management not only on the small subset of users who deposit data, but on the larger group of users who access these data. In contrast, the software managed versioning system allows users to download only the set of files that correspond to the version they are interested in, and in either version, the files have consistent names, because the versions of files are not subject to the constraint of having unique files. If the user wants to compare different versions, they can download them in different folders and the files work without further effort. This does produce an inefficiency of downloading the .wav files twice. However, the user can minimize storage space or download time if they like by only downloading the .eaf of the second version they wish to see and then either copying or moving the .wav. Additionally, this situation does not entail teaching the users about versioning conventions for filenames, and it only arises when the user is interested in multiple versions of the file, as opposed to every time files are accessed, as in the case of a human managed versioning system.

The second use case of access that benefits from this software versioning system is when data is accessed through the RMA's API. The API is described in §4.5.3, however, suffice it to say here that the API is an interface the RMA provides for other software applications to access data, as opposed to a UI which is an interface for human users to access these same data. In this case, the API is designed to serve the data to website applications that serve as access portals to data in the archive. By design, these access portals are conceived of as meeting the use case of groups like language community members and non-linguists. To use stereotypical use cases as examples, users of the RMA's UI are more likely to be searching for the version of a primary data file that corresponds to the version cited in a research paper, whereas community members are more likely to want to hear what their fellow community members say with the most accurate transcriptions available, which they could assume is the latest version. The previous example shows that using a human managed versioning system asks users accessing data to bear the burden of learning this system and parsing the metadata of filenames. Humans can tolerate these burdens to a certain extent, but for some

the learning curve may be prohibitive. In this example, using a human managed versioning system would place the burden of parsing these metadata on computers. This adds complexity to the system, which requires a well defined parser for filenames. This complexity may be acceptable, but the problem arises when humans make mistakes by naming files in ways that do not conform to the file naming conventions for the versioning system. Computers cannot bear such burdens – they simply fail.

Using the computer managed versioning system solves this issue by providing a well defined, automated system for defining the current version of the files and maintaining this definition. This can then be used by the API to send the current version to access portals, which by design are assumed to want the latest version (but can also request others). This is a good example of the synergy that comes with using modern software tools. Both versioning systems and APIs are modern tools that add complexity to the software design. Considered in isolation, it may be hard to justify the benefits of one such component given the complexity it adds to the design. However, considered together, we can see how these tools were not built in isolation, but rather in the context that they are likely to be used together. In this case, using the computer managed versioning system reduces the overall complexity of the design under the constraint that an API is to be used as well, because less software needs to be developed to achieve interoperability between these tools.

In summary, the design includes a system for versioning for items and their files that is managed automatically by the RMA. The goals of this design include: making connections between versions explicit and automatically managed; making access to the files more convenient for users; making access functional for computers running other applications, like access portals for communities; and creating a system that contributes to a culture that encourages creators of primary data to deposit these data before they are "finished." Some of these benefits may seem small, like in the case of mitigating the need to change ELAN files or edit these files in order to utilize them. However, this is one of a number of benefits that, taken together, become compelling. Designing for interoperability between the versioning system and the RMA's API is a major benefit that allows the design to reap the benefits from the added complexity of the versioning system. Additionally, while the contribution to the cultural ethos around archiving is more of a hypothetical claim that would be hard to show directly, such a contribution would in fact be the biggest benefit in the sense of increasing access to language documentation outputs. Moreover, these benefits are more compelling when the cost of implementing these changes is low due to the choice to use the open source software package InvenioRDM that already implements these features, as well as documentation for how these features are implemented.

4.4.6 Collaboration in depositing and co-curation

As described in §2.2.1, various stakeholders have discussed the value of empowering language communities to have control over data management (Holton et al., 2017) and to participate in dissemination of data (Nathan, 2015:74-75). Additionally, archives can encourage collaboration in data management by assigning rights and responsibilities to community members through direct, web-based relationships (Garrett, 2014:69). Finally, it is valuable to create virtual spaces to discuss collections and document these discussions (Mutibwa et al., 2020:173).

The present design applies these discussions to archive infrastructure through collaboration in deposits and co-curation. Collaboration in this context is narrowly defined as multiple people contributing to an archive collection. Collaboration in general entails many interactions between collaborators to create and manage language data that do not involve the archive. However, collaboration in depositing and the maintenance of deposits can enable language communities in controlling the management of their data, and the archive software provides a mechanism to assign roles and responsibilities to depositors and community members. Co-curation in this context includes the processes of: accepting new information from language communities and other stakeholders about existing deposits; vetting this information, for which the individuals or groups with the authority to do so is community-specific (see §3.2.4); and incorporating this information as data or metadata, where appropriate. Using the components of the depositor system previously described, it is now straightforward to describe how the system is designed to allow for collaboration in deposits and co-curation.

Both collaboration on archive deposits and co-curation involve multiple people interacting with the depositor system of the RMA. We can consider for example two use cases that are relevant. The first is a prototypical collaboration where multiple people are involved in creating and maintaining a collection. Following the principle of archiving early, one person in the group archives data from an initial fieldwork event that occurs. Later, another fieldwork event is organized in which more data are created. At this time, the first person who deposited is no longer available to make another deposit, but the group still wants to add data to the existing collection. They have new items that should thematically go in the same collection because they belong to the same project, and they have new transcriptions of recordings that were previously archived, so they would like these files to be appended to existing items. Because the first person is no longer available, a second person wants to use the depositor system to accomplish these tasks. As a second example, a depositor makes a new collection as part of deposit, and specifies in the metadata for the collection that any of three other people are allowed to maintain the collection in the depositor's absence.

Both of these example cases can be accommodated with a similar work flow in the deposit system. Once a new collection is submitted for review, the RMA automatically creates a group for editors of this collection. The user account of the initial depositor is automatically assigned to this group. In the first example, when the collaborating group is ready to make a second deposit, they reach out to an archivist specifying that they would like to add to the deposit, and they declare a second member of the group as a depositor. The archivist asks the second depositor to make a user account and to provide the archivist with the account username. The archivist finds the user account in the system and adds this account to the group for maintaining the relevant collection. Now, when the second depositor opens the deposit system, they see the collection and its associated items the same way the first depositor would, and they can add new items and files as needed. During this process the second depositor becomes associated with new items and files as a depositor in the metadata. When they are finished with the update, they click the submit button on the collection editing view, and the collection is once again submitted to archivists for review.

The second use case example works almost the same as the first. The difference is that, in this second case, none of the original depositors are available, but a community member would like to add an item to the collection because

it is thematically relevant, rather than making a new collection. The archivist can give editing rights to the collection to the new person wishing to make the deposit, provided that the new depositor has the appropriate authority to take an editing role for the collection. The system also provides risk management through the automated moderation process, which applies to revisions as well as initial deposits. This process is available immediately without further software development, given the deposit system described thus far.

Using this system for moderating deposits and revisions by assigning moderation tasks to specific depositors gives the archive a way to address the complexity of collaborating in depositing and co-curation. In summary, both collaboration in depositing and co-curation in the form of vetting changes proposed by users are processes that involve multiple stakeholders having the ability to manage deposits simultaneously. The system proposed here allows the archivist to assign user accounts moderation roles for specific collections in the cases where qualified people are available and interested in participating. In the absence of such people, the archivist can take on the role of moderation as necessary.

4.4.7 Uniform Resource Identifiers

The design of the RMA includes three strategies for the use of PIs, described in §2.4.4, for each collection and item, including each version of each item. These are: manually supplied PIs through the depositor portal; assigning DOIs that are automatically generated and issued by DataCite, which is one of several organizations that issue DOIs (see §2.4.4); and assigning Handles that are automatically generated and issued by Handle.net. Including all three of these options in the design arises from a compromise between three different factors: the costs associated with issuing PIs and institutional constraints related to paying for these costs; the goal of making the RMA useful for other organizations beyond NAL, with the understanding that these organizations might have very limited funding, or also might have an existing relationship with common PI issuers; and the fact that the InvenioRDM application that the design modifies includes a feature for automatically generating DOIs issued by DataCite.

The first strategy is to manually supply PIs in the depositor portal. This option is for circumstances where an organization does not wish to use the other options, or has a relationship with an issuer of URIs other than DataCite or Handle.net.

The next strategy is using automatically generated DOIs issued by DataCite. This strategy is included in the design because DOIs are a commonly used URI among DELAMAN archives, and InvenioRDM includes an integration with DataCite that enables it to generate DOIs issued by DataCite by activating this feature in the settings and authenticating with DataCite. The fact that this strategy is already implemented in InvenioRDM makes it appealing, however, DataCite's price point is high and may be unaffordable for organizations without an existing relationship to the DOI issuer.

The third strategy is using automatically generated Handles issued by Handle.net. This strategy is appealing because Handles are issued at a cheaper price point and are also commonly used by DELAMAN archives. However, generating URIs from this source is not currently implemented with InvenioRDM, so it would require development resources to develop this functionality. It is worth noting that, at the time of this writing, the documentation of

InvenioRDM mentions a desire of the InvenioRDM development team to implement integrations with other issues of URIs besides DataCite, so there is a potential for collaboration in this realm.

The plan for the NAL archive includes two phases. In the first phase, NAL will request Handles outside of the RMA and manually assign them to URI in the depositor portal, which is to say NAL will use the first strategy described in this section. Once development is complete on integrating automated Handle generation into the RMA, NAL will switch to using this feature, which is the third strategy discussed in this section.

4.5 Access

Accessing data in the archive is accomplished both through the RMA and the APA. Functions that contribute to accessing data are distributed differently across both applications. Each application provides a UI for data access tailored to a specific audience, while the RMA also provides an API to serve data to the APA. This section highlights key features that contribute to the functionality of access.

Currently, access level restrictions are implemented in the RMA, while they have yet to be implemented in the APA, which only has access to items without restrictions while it is being implemented (§4.5.1). The RMA UI views for items and files are implemented, while those for collections, people, and languages have yet to be implemented (§4.5.2). The API is implemented, while the use of access keys to limit access to the API has yet to be implemented (§4.5.3). The detailed view for items and files in the APA has been implemented, while the other views and streaming capabilities have yet to be implemented (§4.5.4).

4.5.1 Managing access restrictions

One of the primary goals in designing the RMA is ensuring that access is achieved in an appropriate manner, which in other words means enforcing access restrictions effectively with the software, using metadata and human input where needed. §§4.3-4.4 provided detail on how access restrictions for the items and files of collections can be defined using the RMA's metadata schema and the deposit system's UI for these metadata. While much of the requisite infrastructure for managing access appropriately has been defined in these sections, this section describes details of four mechanisms for enforcing access restrictions, given the previously described information as context. These mechanisms are: access levels; granting access through group associations; restricting access to personal metadata through listing people anonymously; and managing files with server infrastructure.

The RMA and APA websites use user profiles to manage access to data. Firstly, viewing data requires a user to be authenticated through creating an account, logging in, and agreeing to terms of use provided by NAL. Only small previews of files that have no access restrictions can be viewed without logging in.

Openly accessible files are available to any authenticated user. Files with access restrictions are only available to authenticated users that have been granted access via user profile metadata.

Access to management of collections is also managed through user profiles. This privilege is granted automatically to the depositor of a collection, and can be granted to authorized depositors manually by the archivist, upon

request.

4.5.1.1 Access levels

As described in §4.3.3, there are four access levels that represent categories of access restrictions that are applied to items and files. The first level, open access, means that any authenticated user can view the data. The second level, website only, means that data can only be viewed on the website but not downloaded explicitly as files. The third level, embargo, provides an embargo on access until a certain date. The fourth level, restricted, stipulates that the depositor or another person has the authority to stipulate access.

The InvenioRDM software packages has access levels implemented that are similar to open, embargo, and restricted, so that the present access levels can be implemented with modifications to those existing levels. InvenioRDM has an open access level, but does not require users to log in as a condition. This level needs to modified to require users to be authenticated. To achieve the second access level in the design, the UI should not provide buttons to download files (see §4.5.2.1) when files or their corresponding items have this access level. Additionally the URLs these buttons link to should not lead to a successful file download operation if the file to be served has this access level. The third level requires no modifications to InvenioRDM, which has an embargo-based access restriction level. The fourth level is a catch all for access restrictions that depend on the approval of an authority, either the depositor or otherwise. InvenioRDM has a related level in which the depositor is responsible to authorize users who request access, and the website allows the depositor to declare in plain text what the restrictions are, while the user requesting access can write a plain text explanation justifying their request, which is sent to the depositor to review and moderate the decision to grant access. Implementing the fourth level of the current design requires taking this InvenioRDM access level and adding to it an automated system for granting access to specific users based on the group affiliation metadata described in §4.3.7.

4.5.1.2 Granting access through groups

The RMA uses various types of group affiliations to grant access through automated and semi-automated processes. These group affiliation are managed using the group metadata described in §4.3.7. In this system, groups can be defined for any reason, and the metadata for these groups includes a descriptive name, the user accounts and people associated with the group, and whether the group is publicly listed for all users to see or only for archivists to see.

Groups can be made to grant access to specific collections to all members of a group rather than a specific user. This ability to grant access programmatically through group affiliations combined with the manual process of granting specific users access through requests constitutes the features needed to implement the fourth access level in the design.

Beyond access, groups are also used to grant editing and moderation rights to collections. Archivists are an example of a set of users defined by a metadata group. This affiliation can only be granted by a system administrator or other archivist. Being a member of this group allows the archivist to moderate and edit any collection, as well as manage metadata in the person and language tables. Groups are also made automatically for each collection, so that

depositors can be added to these groups to gain editing privileges for a collection, as in the cases of collaboration described in §4.4.6.

4.5.1.3 Privacy protections for people metadata

In addition to restricting access to data, the system allows for restricting access to personal metadata. This is defined on a per person basis, using the metadata field for privacy settings defined in §4.3.7. Each person can choose between these settings, whether or not they have an archive user account. With a user account, they can change this setting in their user profile. As a non-user, they can specific their wish for anonymity to a depositor, who can then define them as anonymous when entering their data for a deposit. The three different settings for privacy are anonymous, unlisted, and listed.

Anonymous means a person's personal data, including their name, will never be shared through the website to anyone except archivists, and their name will appear as 'Anonymous' on items and files, with the role they played in association to these items and files. The only metadata shared is a list of items and files they are associated with, which is only accessible as a link from one of those item or file pages.

Unlisted means that a person's name is shown on items and files they are associated with, and the only information available publicly is their name and a list of associated items and files. Additionally, when a depositor attempts to add people to the metadata for items and files of a deposit, this persons name does not appear as a suggestion,

Finally, listed means that a person's name appears on item and file pages they are associated with, and they have a profile page with their other personal metadata along with a list of items and files they are associate with. Additionally, they do appear in a list of suggestions for people to add to item and file metadata, using an auto-complete function based on partial search, as described in §4.4.3.1.

4.5.1.4 Data security

Based on the NAL archive's use case and obligations to its user base, additional measures beyond the scope of the applications described here are taken to mitigate risks to data security. They are mentioned here for reference, because data security is an important component of ensuring access to data is done appropriately, and some data security measures occur at levels of IT infrastructure beyond the software of applications served over the internet.

Specifically, NALs plan for the use of this software design is to additionally restrict where files are located on their servers based on access restrictions. Freely accessible files are stored in the database of the RMA and are freely accessible to the RMA to serve to users via the website. In contrast, files with access restrictions are stored on a private server not available to the public. When these files are requested through either software application, the RMA makes a request to the private server for the file requested by the user, and stores a copy temporarily just for the purpose of serving the file to the authorized user for that specific session. These temporary access copies are destroyed promptly, and not relied on for preserving the data. This design is used to meet the requirement of NAL's audience to mitigate exposure of restricted data to the internet.

4.5.2 Repository management application user interface and depositor portal

The goal of access to archive data through the RMA is to present the metadata efficiently and provide a means to download files to users who have appropriate access to these files. This UI is intended for depositors, archivists, linguists, and people in similar roles who want to understand the logical structure of the archive data through their metadata. Additionally, they are likely to want to download data (that they have access rights to) that they are interested in to view on their own machines within their established workflows. The design for this UI is not new, but rather is an application of the InvenioRDM software package with necessary modifications for the current use case. Modifications to InvenioRDM's UI need to be made to accommodate the customizations to the metadata schema and the corresponding needs for viewing and editing these metadata, as described in §4.4.3.1 as well as through this chapter.

4.5.2.1 User interface features

To understand the features provided by the UI of the RMA, it helps to understand the web development concept of views. A view in this sense is a template for a facet of a UI, which usually takes the form of a web page. The web page viewed by the user is the facet of the UI, in that it is one of many web pages that are interrelated and constitute the interface to the database. The code that creates these facets is found in web pages, where HTML and CSS create visual templates for how different objects in the same database table will manifest on the web pages. For example, a view could be built with the goal of displaying metadata fields of an item. This view would have a visual template that is built with the knowledge of the metadata structure for these objects, like what metadata fields are in this structure and what data types these fields use. Once built this template serves as the base code for every web page the user interacts with that displays the metadata for a single item. For example, they might visit two urls: "https://examplearchive.org/abc/abc-001" and "https://examplearchive.org/abc/abc-002" and experience two separate web pages that show the metadata for items ABC-001 and ABC-002, respectively. However, for each view, the application stores one web page template file, which it populates dynamically with the metadata from the database only at the moment the user requests such a URL, and sends the finished web page populated with the requested data to the user's browser. Thus, in this example, the UI view can be described as a detailed view of items.

For the most part, the views presented by this UI either show the full set of a metadata for data base objects, allow depositors to edit these sets of metadata, provide lists of associations between objects in the data base, or show the results of search queries to the data base.

The UI has a detailed view for each hierarchical level of items, collections, and files. The collections view shows the collection metadata, aggregate metadata from associated items, a list of items in the collection, and a list of people who manage the collection. The items view shows the items metadata, a list of files, a list of versions, links to export the metadata in different formats, and a link to download the files associated with the item, as a zip file. The file view shows the file metadata, a link to download the file, and a preview of the file for certain file types, including audio, video, PDF and ELAN files.

The UI has some views to facilitate searching the archive's collections. There is a basic search results view, which

provides a list of items and collections that match basic searches. There is an advanced search view which provides a list of explicit search fields to search specific metadata. This not only helps narrow down searches, but gives the user input about what there search queries can be.

In addition to the views for collection, items, and files, there are detailed views for languages and people. These views show each collection and item that a person is associated with and each collection and item a language is associated with, to facilitate browsing related data. The detailed views for people can be restricted based on privacy settings, and they provide personal metadata only if the person object's privacy setting is set to "listed." This functions as a profile page when the person is listed. There is a view showing a list of all languages to facilitate the lookup of language metadata like Glottocodes. There is a view showing a list of all people, which is only viewable by archivists.

For each detailed view, there is an editing view for the corresponding database object. There is one edit view each for collection, items and files, and these were covered in more detail in §§4.4.2-4.4.3. There is a spreadsheet view for batch editing item metadata and a spreadsheet view for batch editing file metadata, described in §4.4.3.3. Together, these views make up the deposit system.

In addition, there are edit views only available to specific users, like the view for moderating collections, which is viewable only by archivists, and is described in §4.4.4. This provides a list of collections with metadata about their moderation status and links for actions to take related to this status. Beyond collections, items and files, there are edit views for language, person, and group objects, which are only viewable by archivists.

4.5.3 Application programming interface

The design of the RMA includes an API for the archive to serve metadata and links to files to other web applications. The format of the API is a JSON based REST API, because this has become a standard for API implementations (Sweigart, 2022). Given the description of APIs above, it is straightforward to describe what it is the API serves in this design. It serves the same data that are delivered to each of the views described in §4.5.2.1. These are the metadata of collections, items, files, people, languages, and locations. For each database table there is a way to call the API to receive a list of objects in the table according to their unique identifier, as well as a request all the metadata fields for a specific object in each table. These are the same queries made to the database in order to populate the web page templates for the views in §4.5.2.1. The difference is that instead of populating the web page templates, this same data is translated to a JSON format by an function called a serializer, and delivered as a text string with the data in this format. Table 4.8 provides a list of the requests that can be made to the API and the information it returns in its response.

It is important to note here that the use of JSON is not in conflict with the use of XML for data that has become a standard in language documentation, as for example in the E-MELD standard and the XML based ELAN data structure. The language used by an API describes how data will be communicated between the two applications, and not how the data will be stored once received by the requesting party. In this sense, the serializer is simply an alogrithm for converting the metadata stored in a SQL database to a JSON string. A prototypical application of APIs for web development is for a web application to request such JSON formatted data and consume them by

Table 4.8: List of API requests and responses

Request	Response
.../api/collections	list of collections
.../api/collection/[pid]	metadata for one collection
.../api/items	list of items
.../api/item/[pid]	metadata for one item
.../api/files	list of files
.../api/file/[pid]	metadata for one file
.../api/people	list of people
.../api/person/[pid]	metadata for one person
.../api/languages	list of languages
.../api/language/[pid]	metadata for one language

immediately converting them to Javascript objects for further processing. Thus, the use of a JSON based API does not commit developers to converting existing data stored in other formats. In contrast, it facilitates the use of other open source software, as well as coordination with other web developers, who use JSON based APIs because they are the standard. To summarize this point, the JSON used by the API here is part of a communication protocol, whereas the XML used in ELAN documents is a storage protocol.

Another important feature of the API's design is that it uses access keys to strictly control which applications can call the API. While the ability to enable access to other web applications has the potential for greater access, it also makes it easier to misuse the data, which is explicitly a concern for NAL users (see: §3.4). Thus, the API can only be accessed with a valid access key, which can only be obtained through communication with the archive directly.

The mechanics of how other web applications make requests is also straighforward. The application calls the API by visiting a URL. As this is a web application, it is not visiting the web page in a browser but it is still making a request to the server using the HTTP protocol. The API defines a series of URLs that can be visited to request database information. For example, a URL like "https://examplearchive.org/api/items/list" would be defined that, when visited returns the list of item objects in the data base by UNIQUE IDENTIFIER, and this information is translated to a JSON string. Similarly the URL "https://examplearchive.org/api/items/abc-001" would result in the server sending the metadata for item ABC-001 as a JSON string.

The APA, described in the next section, calls the API to retrieve metadata and links to data from the RMA. If this were the only application that used the API, it would be easy to argue that this modular design with two applications is inefficient, in that it requires an API for a second application that could be designed as additional views in the first application. However, the value of including the API in the system in the design comes in the form of the extensibility of the system. With the API in place, the RMA is ready to serve applications in the future, which is described further in §4.6.

The design requires a group of small modifications to the InvenioRDM software package to be implemented. InvenioRDM includes a JSON based REST API that serves the metadata. First, the modifications to the metadata schema, described in §4.3, need to be implemented. After that, the API calls need to be adjusted to serve metadata in this schema. Additional API calls can be made for the sake of increasing efficiency for applications that call the API.

Specifically, the API calls for detailed object metadata described in this section can be an inefficient way to retrieve metadata for many objects at once, for example each object that meets a certain search criteria in its metadata. Thus, API calls can be defined that include a list of all objects in a table as well as one or two metadata fields for each of these objects that are needed by the application making the call. These needs will come to light as the development of the APA proceeds with iterative feedback from NAL's user base, as well as the potential development of other applications.

4.5.4 Access portal user interface

The APA provides its own UI to the data of the database. The focus of this UI is providing access to community members and others who are less likely to be academic linguists and archivists. In this use case, the users also want to see what data are available, but are mainly interested in a small subset of the metadata that describe the data. Additionally, they are interested in viewing data in convenient formats, and don't have pre-existing workflows that necessitate they download data to interact with, but they might download some data once they have viewed it, mainly to have a copy.

This UI presents data from the same database as the UI for the RMA, because the APA retrieves the data from the RMA through its API. This means the APA does not need to be designed to manage the data redundantly. The focus is on presenting the data and metadata in a way that highlights different metadata and methods for access than that of the other UI.

Based on feedback from NAL's user base, presented in Chapter 3, the metadata of most interest are: ACCESS LEVEL, INDIGENOUS TITLE, ENGLISH TITLE, DESCRIPTION, SCOPE, AND CONTENT, GENRE, LANGUAGES, PEOPLE, LOCATION, RECORDING CONTEXT, PUBLIC EVENT, CREATION DATE, as well as implicitly the relationships between collections, items, and files. For this reason, the UI highlights these metadata on detailed views of collections, items, and files, through web page design and formatting. Examples of such highlighting include larger text, bold font wait, the use of colors within the designs color scheme that draw attention to the text, and position on the page. In contrast, metadata not in focus may be given less highlighted formatting, presented in page elements that are hidden by default and must be revealed by the user with a click, or missing from the page entirely.

The search and browse features identified as high priorities based on the feedback from NAL users presented in Chapter 3 are implemented in the web access portal. These include basic and advanced search, as well as browsing by related languages, genres, and geographic location.

The data are also presented differently in the web access UI. Previews of videos, audio, and text files are presented most prominently on the detailed UI views, such that the page for each file primarily provides the user a away to interact directly with the data in the browser with as little effort as possible. Links to download the files are prominently displayed, but still secondarily to the browser based access like streaming the video. Navigation buttons between files in an item are provided as left and right arrow buttons on the side of the detailed file view, so that users can easily scan through files to experience them. In essence, this UI focuses on satisfying a user who wants to directly experience the data and be immersed in it. This is in contrast to the UI provided by the RMA, which aims to satisfy a

user who wants to interact with the data from a birds eye view and see the relationships between the data that give it hierarchical structure.

The design for the two applications together does not assume that users will fit either of the two use cases described here, and thus the APA provides links in every view to the corresponding view in the RMA's UI. Together these UIs provide multiple content rendering (Nathan, 2015:60), in that they present the same data in two different ways that aim to meet different use cases of the user base. Individual users may have either of these needs at different times, and have access to each method of rendering the content.

4.6 Digital return

Digital return of data to Indigenous communities, described in §2.2.2, is facilitated by the design of the two applications in three main ways: through the convenience of downloading data and metadata, through the API provided by the RMA, and through the design of the applications to be used by other organizations.

First, the data management UI provides convenient links to download the metadata set for items in multiple standard formats as well as the data a zip file. The InvenioRDM software package allows for exports of metadata for specific items as XML and JSON files, and exports files associated with an item as a zip file. Furthermore, at the time of this writing, the InvenioRDM developers are implementing file storage in the software package according to the specifications of the Oxford Common File Layout,[3] which provides a method of file storage that is application-independent. Once this development is in place, the use of this upcoming version in the current design will result in the storage of metadata alongside data in the file system containing the archive's data. This ensures that metadata are available for their corresponding data even in the case that the web application is no longer available. Thus, the design enables the data and metadata to be delivered to Indigenous communities in standard formats that they can import into their own applications and services of their choice.

Second, the API described in §4.5.3 is by definition infinitely extensible. This has a practical value in facilitating digital return in that it enables communities who want to control their access to their data, while still utilizing the archive for the purpose of data management and preservation. For example, a community can build a website that functions like the APA and accesses language data through the API of the archive rather than hosting it in an internal database. This means they are not reliant on the archive to provide a UX specifically for them, but rather can design a UX to meet their specifications. This provides an analogous division of labor to that used in NALs plan for an archive website. Linguists can focus on the aspect of software development related to their expertise, which is language data management, while the community members can focus on the aspect of development that linguists are not as qualified for, which is their specific needs for a UX in interacting with these data. This value may seem to be diminished by the fact that the community in this example would need to have someone with software development skills. However, the division of labor mitigates the barrier to entry, because open source web frameworks and free or cheap website services abound, while hosting and proper data management are not free.

[3] https://ocfl.io/

While building an access portal web application may still seem like too high a barrier to entry for this mechanism of digital return, the same principle applies for communities who already operate their own websites. The archives API can be integrated into any single page or group of pages on their website. The only additional demand of resources in accomplishing this would be in understanding the protocol for making API calls. With this in place, the community website could implement things like listing files of their languages available in the archive, or providing previews of these files directly on their website.

Finally, the third method for digital return in this design is that the software in this design is intended to be used by other organizations. This means an organization can download the software as an open source repository and configure it for their organizations needs using the documentation provided. Based on the design, the expectation is that the access portal will need to be customized for that organization's users, while the RMA will need less modification to be used. In essence, this is because there is less variation in the nature of the relationships of language data itself, than in the variation of people's expectations for interacting with these data. The resulting digital return in this scenario is of the resources developed by linguists to manage language data effectively and appropriately. If an organization has some IT resources to run a web application on their own or cloud servers, they can use the applications designed here themselves and retain full control over their data and the software providing management and access to thes data. The software design presented in this chapter is intended for this use case as well as the use case of an academic archive, This is because so many aspects of these use case are fundamentally similar.

Part of the design of the software presented in this chapter is the plan to release the software so that it is not only open source, but it is also practical to use by other organizations. This plan includes three phases that are all distinct mechanisms for disseminating the software for use by other organizations.

In the first phase the software is developed as a github repository,[4] which includes an instance of the InvenioRDM software package that is created using an Invenio-cli command and modified as the design requires. Here, CLI stands for command line interface, so that the Invenio-cli is a command line interface for Unix-based systems that developers install on a computer like a server, and once installed, it provides a set of commands that can be run to install the web application so that it is ready to be served over the internet. In this phase, the repository is a specific instance of the software application, and as result, it contains certain configuration variables, like private keys, that are not shared. The repository includes documentation for how an organization should replicate these missing pieces as well as customize the instance for their own use.

In the second phase, the modifications called for in this design are built as modules for the InvenioRDM software system, so that the software in this design can be installed by installing InvenioRDM and these InvenioRDM modules. In essence, this is like installing a flavor of InvenioRDM. An open source repository[5] is provided which gives instructions on how to install this flavor of InvenioRDM and configure it for use by other organizations.

The final phase involves the development in a command line interface tool specific to this software project. This CLI is similar to the base Invenio-cli, except it is designed specific to install the customized flavor of InvenioRDM,

[4] https://github.com/kavonjon/archive-software
[5] see: note 4

which was implemented in the second phase, as conveniently as possible. Here, convenience is provided by the CLI by requiring less user input through assuming more about what the developer wants to install. In this case, the assumption is that the developer wants to install the specific flavor of InvenioRDM developed in the second phase, as opposed to other possible configurations of InvenioRDM.

Each of these phases represents progress towards more sustainability of the software packages as deliverables to other organizations, and ease of use for those organizations.

4.7 Current implementation

As described in section §4.1, the implementation of the design for digital archive infrastructure for the NAL archive is ongoing, and is in year one of a three-year implementation grant at the time of writing.

In the current stage of development, both the RMA and the APA are implemented as development versions that are sandboxed, meaning they are only accessible by the development team. The RMA is a working instance of InvenioRDM with sample data from NAL that is open access. The API is implemented, which allows the development of the APA to proceed with access to the sample data through the API.

Current development of the RMA is for the metadata schema, and subsequent adjustments to the API to serve these metadata. Future developments include all other modifications to InvenioRDM that are part of this design.

The software is deposited with the Zenodo archive[6] in its current state of development. While there is currently no convenient installation option, it can be compiled on a local machine. Future versions of the software will also be deposited at the same location.

4.8 Future developments

Much of the work implementing the design described in this chapter is in progress and will be completed by May, 2025, according to the schedule of the NEH grant[7] this work is a part of. This implementation work includes: the process for harvesting metadata following OLAC; the process of granting moderation and editing rights for deposits made by other depositors; changes to the depositor portal UI to reflect the metadata schema; automated generation of metadata during deposit and batch editing; the system for assigning URIs; access restrictions in the APA; the RMA UI views for collections, people, and languages; the use of access keys to limit access to the API; and streaming capabilities in the APA.

The design of the RMA lends itself to specific future developments, specifically for the metadata schema, related harvesting protocols, access and the deposit process.

[6] https://doi.org/10.5281/zenodo.7344249

[7] NEH grant number: PW-285221-22, to University of Oklahoma; Principal Investigator: Raina Heaton

4.8.1 Automated metadata generation

One of the limitations of the RMA's procedures for automating the generation of specific metadata is the way it calculates values for the EXTENT field for files. It is able to calculate the duration of audio and video files, as well as the number of pages of text documents that have pages. However, it does not have a method to calculate the extent of plain text files like ELAN files. Future development in this area should focus on how to define the extent for such documents. For example, the calculation could be based on the number of annotations, but this may not be a meaningful measure of extent, given that annotations can multiply in number with additional tiers, and at the same time annotations can be empty. There are currently Elan file viewers that provide metrics for annotations and number of tiers, which could be included in the design. Once a definition of extent for these documents is understood, the RMA can be modifed to calculate EXTENT for these files.

4.8.2 Generalizing metadata

The present design includes many compromises between the goal of making the RMA generally useful for other organizations that would deploy their own instance of the software, and ensuring the RMA meets the constraints of the NAL archive, which is the focus of this case study. Developing the design in the context of a language archive is an advantage because it avoids design in a vacuum – in other words, outside of the context for which the software is designed – but it inherently biases the design toward that context over others.

One of these biases is manifest in the metadata schema used by the design. Many of the fields that deviate from the OLAC metadata schema do so in order to accommodate NAL's existing data. One noteworthy example is the case of GENRE and GENERAL CONTENT TYPE, which map to OLAC:LINGUISTIC TYPE and DC:DCMI TYPE, respectively. This mapping has issues because the genres do not directly correspond to the linguistic types, but some of the genres correspond logically to some of the linguistic types. However, some of the genres also correspond to DCMI types.

Future developments should include modifications to the metadata schema that bring it closer in line with the OLAC schema. In the case of GENRE and GENERAL CONTENT TYPE, the metadata schema should be modified such that genres like "book" are moved to the GENERAL CONTENT TYPE field. Beyond this, the software can be made to be generally useful with an additional configuration file that allows for unneeded fields like NAL specific fields to be disabled in the RMA.

4.8.3 Metadata harvesting protocol

The metadata schema of the present design motivates some exploration into specific ways that the OLAC schema and harvesting service could be further developed. Three examples are with locations, genres, and versions.

In the case of locations, neither OLAC or DC implement polygon shapes for locations, but rather they implement boxes, or rectangular shapes for two-dimensional locations. While a rectangle that bounds a polygon is an approximation of that two-dimensional location, polygons are becoming more common in geographic data in web applications. Thus, it is worth exploring the possibility of proposing an update to either the OLAC or DC schema to

include polygons, using their processes for adding terms to their vocabularies.

While the GENRE field in the RMA's metadata schema has some inefficiencies based on NAL's existing data, the results from the workshops in Chapter 3 show that users are particularly interested in searching for resources based on specific genres that are not in OLAC:LINGUISTIC TYPE. Thus, it is worth exploring the possibility of proposing an update to the OLAC schema to include GENRE in some form, because this would allow users to search OLAC's index of language archives based on genres.

Finally, including version control in the RMA has interesting implications for metadata harvest. Specifically, which versions should be harvested. The current design for the RMA is to make only metadata for the latest versions of resources harvestable, for two reasons. First, the UI for OLAC's indexed search is not designed for versions of files, and it would change the UX of the website if search results were crowded with versions of a single resource. Second, discovery is not necessarily aided by indexing all versions, and citation is not hindered by indexing only the most recent version, because the URIs in citations are always resolvable to the correct version of the resource. Exploring whether versions of resources should be harvested should be done in coordination with OLAC and other language archives.

4.8.4 Microservices for access

With the design for the RMA and APA in place, this model can be extended by adding microservices (described in §2.7.1) that provide access in a variety of ways beyond those provided by the current design. Additionally, these microservices can be hosted anywhere, including by the archive itself, which would allow language communities and other groups to create customized forms of access with little overhead or resources.

To give one example of how this could work, consider a scenario where a language community would like to have a website that displays data from their language using specific categories that do not correspond to metadata fields in the archive or any categories that previous researchers used to describe the data. Some hypothetical categories could be: recordings in which people refer to a specific cultural practice, or recordings that include heated arguments. In these cases, the data included in these categories could easily span different items and collections. The desired output would be a website that groups data in new categories and allows users to browse by these new categories.

To meet this need, a microservice could be developed that is similar to the web access portal described in §4.5.4, in that it uses the API of the RMA to retrieve metadata and data without needing to store these data interally, but rather requests them to be directly displayed to the user in a website. However, unlike the APA, this application is very small. Its only features are as follows. It allows a user to authenticate through the archive and select which data are to be included in this presentation, while controlling which data are available based on access restrictions. It then allows the user to define simple categories that can be applied as tags to the data they selected. Finally, it creates a website that allows the user to browse the selected data by the tags applied to them.

As a microservice, this would be less resource intensive to build and maintain than a full application with a database, and the archive could allow users to generate an unlimited number of these custom access websites. This would mean a language community could define its own categories for how they wish to view data through a simple

UI, and they wouldn't need to deal with the IT complexity of hosting a full web application. Moreover, this would allow them to apply culturally relevant categories to data that the archive and outside researchers have not developed.

This type of application is promising because it empowers communities to control how they use their data, it is not resource intensive to develop or require community based IT resources, and it utilizes the archive infrastructure proposed in this chapter, specifically the API, which allows it to be simple and light weight.

4.8.5 Automated interchange with collaborative data and metadata development tools

The RMA provides the potential for an automated ingest feature for collections, based on the ingest feature implemented by the InvenioRDM software package. Using this feature, a depositor can import a tagged version of a git repository hosted on Github directly as a new deposit. To do so, the InvenioRDM provides a UI feature where a depositor can authenticate with their Github credentials via the InvenioRDM UI and then see a list of their git repositories available for this process. The depositor can then choose the appropriate git repository, and then select a subset of the files in the repository to be included in the deposit. Once selected, InvenioRDM retrieves the files from Github directly, and takes the depositor to the depositor portal, where they can fill in metadata for this the deposit. This feature provides depositors, especially collaborating groups, who develop their collection data and metadata on Github a way to automate and streamline their deposits to NAL.

While the use of git for collaboration in language documentation has received some attention recently (McDonnell, 2017), its use may be limited by steep learning curves and also limits on file sizes for the free version of the Github. In the future, the RMA can be modified to provide the same import features from other git servers, as well as locally hosted git repositories, if and when any such git servers become popular for language documentation.

4.9 Chapter summary

This chapter presents a design for digital infrastructure for archives that provides repository management, a depositor portal, and access through multiple systems and UIs. The design is motivated by the principles for better practices in language documentation and the developments in web-based software applications described in Chapter 2, as well as by feedback from NAL's users in the discussions described in Chapter 3.

As described in §2.5, many of the principles established for better practices in language documentation relate to bringing data into archives and managing these data. Two main areas in the design for archive infrastructure that address these principles are the metadata schema and the process of depositing data.

The metadata schema follows standards set by other archives, like using three hierarchical levels and conforming fields to the Dublin Core metadata standard. Simultaneously, it is designed to meet the needs of the NAL archive and its existing collections. It includes metadata necessary to enforce access restrictions, which is an essential component of best practices and a matter of importance to NAL users based on their feedback. Finally, it includes community-specific metadata in the form of TK labels, which was requested by NAL users in their feedback (§4.3).

The design provides a depositor portal for depositing and managing collections. It also uses version control to

allow for updates to metadata and data that retains all versions, in order to ensure data are preserved and always citable. This allows for co-curation, and by utilizing user accounts and assigning those user accounts roles for moderation for specific collections and items, it does so in a way that accommodates the different needs of different depositors and communities. This design aims to provide sustainability in co-curation by allowing the archivist to perform moderation alongside other depositors, in the cases that they prefer this or become unavailable for this work (§4.4).

The principles of access described in §2.5 are an area of particular interest and excitement for the users surveyed in Chapter 3. To follow these principles and meet this interest, the archive infrastructure uses multiple strategies in delivering access to data. First, it provides the UI for depositors and collection management, which is designed to give full accounts of metadata and make it easy to moderate changes based on co-curation. Additionally, it uses an API to provide a machine-readable interface to collections, such that the system is infinitely extensible in terms of access portals to the data. The design provides one such access portal in the form of a website for general users to access collections (§4.5).

There are multiple outcomes of developing the design presented in this chapter. First, the design contributes to language documentation by providing a marginal update to models for achieving the principles of deposit, data management, and access for archives, using developments in web-based software applications to do so. For NAL specifically, the process of developing this design has led to the development of a metadata management application with a relational SQL database that NAL staff currently use internally to honor requests for access. Additionally, NAL has a detailed plan in place to develop their digital infrastructure, including compartmentalized modules and personnel in place to develop each module (§4.1).

This design uses open source software in the form of InvenioRDM, which itself is an open development project (§2.7.5). The current project extends that software using an open development model, with collaborators on the team releasing the software using open source licenses. In the current stage of development, the software is a beta version for internal use that has the API implemented so that the different collaborators on the different modules have access to the other modules, and the modules can communicate.

The design is infinitely extensible in the sense that different access portals can be made by different groups and hosted in different places. These access portals can use the API to access data according to access restrictions and present them as different communities and groups desire. This allows, for example, for the creation of microservices that can request data from the archive and present it using categories specific to different communities and groups. Additionally, the design includes components that could provide more streamlined deposit in the future by ingesting data through git repositories (§4.8).

Chapter 5

Conclusion

This dissertation engages in iterative improvement of the methodology of language documentation, first by identifying goals for data management contained in the principles of better practices defined by the field, and second by developing a design for digital archive infrastructure with marginal modifications to existing designs that aim to increase archives' abilities to meet these goals. The use of software by different stakeholders is an important part of conducting most of the processes involved in language documentation. Moreover, the design of this software requires the input of language users and linguists in order for it to serve these and other stakeholders in conducting language documentation. Thus, as a linguist, I engaged in the methodology of language documentation by asking one overarching question: how can software tools in the form of digital archive infrastructure be modified to further achieve better practices in language documentation?

5.1 Summary of chapters

The dissertation is presented in five chapters that collectively provide an answer to the question posed, including a a background chapter with a literature review, a description of a series of workshops designed to gather feedback from archive users, and a description of a design for digital archive infrastructure that aims to provide marginal improvements over existing designs.

The literature survey in Chapter 2 introduced archives and the role they play in language documentation. It demonstrated that when models are described for new and developing software tools, they tend to place archives in an ancillary roles in terms of data flows, where data are meant to be deposited with archives, but archives are not seen as the primary way to access data. This situation creates a risk of sidelining archives in their role of serving different stakeholders, even if only through a perception that they are not useful for access, which is being reinforced by these models.

Next the literature survey describes what data and metadata are in language documentation, using levels of data defined by Himmelmann (2012), and common metadata fields used by archives. Taking a single archive as a case study, Kaipuleohone, it describes a pattern where raw data are most often not preserved in archives alongside

corresponding primary data in the form of transcriptions and translations.

This chapter explores principles for better practices that have been defined in the field, including Bird and Simons's (2003b) dimensions of portability, the principle of reproducibility and the need to cite raw and primary data presented by Berez-Kroeker et al. (2018), the FAIR principles for scientific data management (Wilkinson et al., 2016), and the CARE principles for Indigenous data governance (Carroll et al., 2021). It identifies three main categories that these principles focus on from the perspective of archives: bring data into the archive, both in terms of deposit and managing data through their metadata; having data flow out of the archive through access; and cultivating relationships with stakeholders, especially Indigenous communities, around the practices of the data flows through archives. One difference observed between the group of principles focusing on deposit and data management versus the principle focusing on both access and cultivating relationships is presented. Namely, principles focused on deposit describe more concrete actions that have been implemented archives. In contrast, the principles for access are more likely to call for access without describing specific actions, and the principles for cultivating relationships are often vague due to being inherently context specific and process oriented.

Archives actively pursue better practices, and the literature survey provides an illustrative example of AILLA, which, in a recent redesign of its digital archive infrastructure, sought to improve its infrastructure for deposit and management of data through improvements to its metadata schema and depositor portal, as well as enabling batch-ingestion for metadata. Additionally, it sought to improve the access it provides through improved features for searching collections and displaying search results, audio and video streaming of data, bulk download of media files and metadata, and an interface to display mapped location information for data.

The background chapter next surveys some aspects of modern web-based software tools that are relevant in answering the question posed by this work. This includes application programming interfaces (APIs), which provide a mechanism for facilitating access with little overhead; the importance of users' experience (UX) in outcomes when using software tools, and the effect of their relationships with software providers on this experience; open source software and its value in lowering the resources required to deploy software tools; and version control systems, and their relevance for iterative archiving and bringing more primary data into archives.

Finally, this chapter introduces the Native American Languages (NAL) archive, which is currently engaged in a project of bringing online access to its collections, and serves as a case study for this work.

Chapter 3 presents the results from a series of workshops conducted by NAL in 2020 to seek feedback from its users on how they would like the project for online access to proceed. This workshops series is a first phase of seeking feedback that forms the basis for the user-centered design that NAL is engaged in, which they seek feedback from users at different stages of developing their digital infrastructure. In this way this work, alongside the staff of the NAL archive, is engaging with the principles of better practices in language documentation that focus on cultivating relationships with stakeholders.

Some key findings from the discussions at the workshops motivated the design for digital archive infrastructure presented in this work. These include: metadata fields that should be highlighted in search and browse functionalities in order to make collections discoverable based on users' needs; users' interest in NAL implementing Traditional

Knowledge (TK) labels; a diverse set of opinions on co-curation, which motivate a design for infrastructure that allows the archive to enable depositors to vet new information where this is appropriate and to retain this responsibility where it is appropriate; users' interest in Indigenous design elements in user interfaces (UIs), which confirms the value of NAL's decision to hire an Indigenous graphic designer; and an interest in having the software developed available for others to use, which confirms the value of NAL's decision to release the software developed as open source software. Beyond this, workshop participants confirmed that they value the role the archive plays in preserving data and maintaining access restrictions where appropriate, and they view the workshops as a positive step toward cultivating relationships and empowering Indigenous communities.

Chapter 4 presented the design for digital archive infrastructure developed by myself in collaboration with NAL staff and IT developers from the University of Oklahoma (OU). This design was developed in the context of NAL's needs as a case study, but also with the goal of being applicable to language archives in general. The design is only marginally different than existing designs for archive infrastructure, and this is because archives are already successful at providing for the deposit of, management of, and access to data. The goal of the current design is to be one step in a continuous cycle of incremental improvements to such digital infrastructure.

As a result of being an incremental improvement on existing models, the chapter in part focuses on details that are similar to those of existing archives, including the design's metadata schema, its implementation of access restrictions and user accounts, its depositor portal, and its use of persistent identifiers. Additionally, it focuses on design details that deviate from common features among archives, including: UI features for its depositor portal, a software managed version control system to enable co-curation and the depositing of primary data through iterative archiving; and an API that allows for the creation of different UIs for the needs of different user groups. Additionally, the design leverages open source software in the form of the software package InvenioRDM. This mitigates the resources required to complete the project of deploying the designed infrastructure, and helps to ensure that the core features of repository management are achieved according to modern standards for software applications. The design extends this software package and is available as open source software itself, to enable other archives to use and/or incorporate elements from the design, as well as enable digital return by allowing communities to deploy the software for their own repositories.

5.2 Answers to the research question

The question posed by this dissertation was: How can the design of digital infrastructure for archives be modified to increase archives' abilities to uphold principles of better practice in language documentation without drastically increasing the demand for resources placed on archives to deploy this infrastructure?

This work provides several key answers to this question. First and foremost, building and maintaining relationships with stakeholders and Indigenous communities is crucial to ensuring that developments in designs for digital infrastructure meet the needs and expectations of users. The workshops presented in Chapter 3 allowed me and my collaborators to build and maintain such relationships, and they gave us insights into how to design digital

infrastructure in ways that meet users' needs.

In terms of design, one of the most important answers is to not reinvent the wheel, and to make sure archives are able to keep doing what they have historically done the best, which is preserving data and managing their metadata. The value of maintaining these core functionalities was confirmed by participants in the workshops, who in discussions spoke of an expectation of the archive to preserve data and manage appropriate access restrictions. One of the major ways in which the present design accomplishes this is through leveraging the open source software package InvenioRDM. Since this package is designed for repository management, it provides many of these core features, leaving the archive's very limited resources to be applied to the features that are specific to its use case.

In terms of features in the design that can be seen as modifications to common existing archive features, the most important feature in providing an answer to the research question is the API. First, it is used in the design to provide different UIs for different use cases, specifically that of depositors and co-curators vs that of users accessing collections. Moreover, this API can be used to create ever more access portals in the future to meet the unanticipated needs of specific communities, without demanding the archive infrastructure be reworked. Importantly, the design uses the API immediately to create the access portal for general users, which means implementing the design gives us direct experience with how to use the API to create other access portals. Looking at models for data flows in language documentation, this design is also a response to models that make the archive a dead end for data that are deposited with no expectation for access. In such models, data are simultaneously stored outside of archives in databases that do not necessarily meet the standards of archives, and new software tools are developed to provide an API for these other databases. In contrast, the current model provides and API for the archive itself, so that it can serve as a central point of access without additional overhead beyond maintaining the API itself.

In addition to the API, the version control system managed by the software directly is another feature that represents a modification to common designs which answers the research question. By allowing metadata to change and managing each version of metadata, the design allows for co-curation by accepting new information from language community members that describe collections. In combination with the feature of user roles for moderation that can be applied to specific users for specific collections and their items, the design allows for meeting the various needs of users established in the workshops. Specifically, the design allows for delegating moderation to certain collections and items where that is appropriate, and just as easily allows for archivists to carry out this moderation where that is deemed appropriate by depositors and communities. Additionally, the version control system allows for files to be added and new versions supplied, which enables iterative archiving with the goal of increasing the rate at which primary data are deposited alongside their raw data.

5.3 Contributions of this work

This work contributes to the field of language documentation both through enabling the NAL archive to achieve the principles of better practices in the field, and by incrementally developing designs for archive infrastructure that can be used by other archives and Indigenous communities to manage language data.

It is important to note that many of the aspects of the design presented in Chapter 4 are not new. Moreover, the reliance of the current design on previous work is a crucial component of the design. Existing designs for digital archive infrastructure are effective at accomplishing a wide array of tasks that enable language archives to fulfill their roles. The purpose of the current work is not to propose alternatives to existing designs, but rather to build on them. The aspects described in Chapter 4 that are not new to digital archive infrastructure include: metadata schemata suitable for language data, especially with regard to language metadata; harvesting protocols to make data discoverable; hierarchical levels of metadata and file naming systems that complement these levels; user accounts; depositor user interfaces for depositing data and creating metadata; URIs; and various types of access restrictions for data. Additionally, the use of APIs in web applications, including web applications that make up digital archive infrastructure, is a common practice. These APIs are not highlighted as part of digital archive infrastructure designs, but rather are implemented to achieve the necessary data flows in software.

The current design builds on existing better practices achieved by previous designs by packaging components of digital archive infrastructure in a novel way in order to achieve a limited set of improvements. The current work includes an API explicitly at a high level of the design, in order to achieve extensibility of access as well as a new model for digital return. Additionally, the current design is novel in that it includes a software-managed versioning system, in order to enable collaboration through co-curation as well as iterative archiving.

5.3.1 The Native American Languages archive

For NAL specifically, this work gave NAL a concrete plan and design for developing its digital archive infrastructure, bringing online access to its collection, and making its metadata harvestable by OLAC. As part of developing this plan and the design, NAL built and maintained relationships with its users and Indigenous communities through workshops it hosted to get feedback from users, thus implementing user-centered design (Wasson et al., 2016:643) and participatory archiving (Linn, 2014). Additionally, as part of this design process, I migrated NAL's data to a relational SQL database and built a web application that the NAL staff currently use to search collections and curate data in order to honor access requests from users. This migration helped prepare NAL's existing data for the designed infrastructure, and facilitates NAL in providing access until its collections are online.

5.3.2 Keeping archives at the center of language documentation

Language archives have long strived to serve practitioners of language documentation, and in doing so have played a central role in the scientific and social practices of language documentation. These efforts include: digitizing analog materials (Albarillo and Thieberger, 2009:3; Barwick and Thieberger, 2018:139; Linn, 2014:62); stewardship of language data (Golla, 1995:152); encouraging the creation of metadata and developing metadata standards (Bird and Simons, 2003a); enabling linguistic research to be grounded in language data (Berez-Kroeker et al., 2017; Evans and Dench, 2006:24); encouraging stakeholders with language data to archive these data (Berez-Kroeker, 2015); providing access through archive websites (Trilsbeek and Wittenburg, 2006) and discoverability through harvesting metadata (Bird and Simons, 2003a); and working with Indigenous communities (Theimer, 2011; Linn, 2014; Garrett, 2014).

These efforts have collectively placed language archives at the center of the practice of language documentation.

Software use and development have been crucial to many the efforts that have places language archives at the center of language documentation. The software used by language archives and in language documentation changes over time, and these changes bring risk and rewards. A recent example that is relevant to the present work is the development of dynamic web applications (see §2.7.1) that have become ubiquitous on the web. Such dynamic applications allow language archives to manage data through convenient UIs and serve data through automatically generated web pages. However, as practictioners of language documentation implement advanced features of modern web frameworks in software tools for specific tasks, there is an increasing risk of isolating archives from data flows in language documentation if these tools are not developed in coordination with archives.

The models of Hanke (2017:115) and Bettinson and Bird (2017:162), described in §2.2.3 are examples of software design that risk pushing archives toward the periphery of data flow and practices in language documentation. Figure 2.4 illustrates Bettinson and Bird's model, which includes a database for language data hosted and served by Google, and API to the application serving these data, and a series of applications and microservices to make use of these data. In this model, the archive receives data independently of these processes, and has no API for access or interaction with other applications.

The present development contrasts with models that risk isolating the archive by giving the archive an API to interact with other applications. Thus, software applications built to accomplish specific tasks can incorporate the archive into their data flows directly. In the case of Bettinson and Bird's model, where a database independent from an archive is needed for the creation and development of language data, the archive's API could be called by one of the other applications in the model to facilitate deposit and thus ensure the Google database does not become the primary data store in practice. This would be accomplished by a more advanced version of the API in the current design that accepts metadata through an API call and populates a draft of a deposit with these metadata. Additionally, for applications that seek to reuse data in archives, the present development allows for these applications to be built as microservices that call the API for data from the archive. Thus the API allows for access that is customizable for different user groups and Indigenous communities.

Rather than assuming that new software development for data workflows should be done around archives, which increases the risk of archives becoming "data cemeteries," where data are compiled that become inaccessible (Widlok, 2013:187-188), this model incorporates the archive into these software developments, empowering archives not only to continue to provide access over the ever-evolving web, but also to innovate new models for digital return.

5.3.3 Co-curation

The design for digital archive infrastructure presented in Chapter 4 includes a software versioning system that, together with user account roles for moderating deposits and TK labels, enables a novel system for co-curation among language archives. This system is motivated by NAL users' requests for co-curation that allows for new information and comments on existing collections while simultaneously moderating these potential changes, where the people doing this moderation may be different for different collections (§3.2.4).

The versioning system streamlines the process for new information about existing deposits to be incorporated into the archive's collections. A new version of an item can be declared in the system, either to add new versions of files or metadata. A new version of a file could be useful, for example, for primary texts with new tiers or modifications to existing tiers. A new version of metadata could be useful, for example, for adding TK labels or descriptive information that relates to Indigenous ontologies of information. While in existing designs, the metadata for such an item can be modified to add such information, the present design provides a mechanism to explicitly capture that such information was added or changed, as well as capturing metadata about that change, like who made it, who approved it, and when it was made.

The metadata about changes is created as part of the system which assigns editing and moderation roles to user accounts for specific collections and items. When a depositor with an editing role makes a change to an item or collection, they have the opportunity to comment on the change they made, and this triggers a review process whereby a depositor with a moderation role for that item or collection, or an archivist, must approve the change before it becomes a part of the archive. The person approving the change also has the opportunity to comment on why they approve the change. This creates a metadata record of the changes made after initial deposits, which forms a virtual space for co-curation (see: Mutibwa et al., 2020).

Finally, the inclusion of TK labels alongside the versioning and moderation systems allows Indigenous communities to participate in co-curation by applying appropriate TK labels to their digital heritage, while simultaneously having metadata in the archive that explicitly documents these changes and decisions, as well as who made them.

Furthermore, the application of the different types of TK labels, including provenance, protocol, and permission labels, can lead to co-curation in allowing the archivists, depositors, and developers to respond to this new information in specific ways relevant to each type of label. For example, if provenance information is provided in the text of a TK attribution label, this could lead depositors or archivists to add specific authors to the item in a subsequent version. If information about permissions is provided in a TK Non-commercial label, this could motivate depositors or archivists to add a non-commercial license to the item in a subsequent version, whereas if a TK Outreach label is applied, this could motivate depositors or archivists to change the access level of the item to restricted in a subsequent version. As a final example, if protocol information is provided frequently through TK Seasonal labels, this could motivate developers to create explicit cues in the website UI, such that archive users are alerted when they are viewing item pages with TK seasonal labels during time periods when the item should not be viewed.

Taken together, the software versioning system, management of editing and moderation rights for specific items and collections, and the use of TK labels create a novel system for co-curation that enables Indigenous communities to work together with depositors and archivists to curate the archive's collections.

5.3.4 New model for digital return

The power of the API used by the RMA to serve data to the APA is that it is extensible to other access portals yet to be developed. This creates the potential for new models for digital return based on a combination of providing access to data through archives, and simultaneously empowering Indigenous communities with software for access

portals that they can customize for their own metadata and ontologies (Barwick et al., 2019:14). An example of this process is explored in §4.8.4. If an Indigenous community wants more control over how they discover and interact with their data, they can develop their own access portal by modifying the APA or building any web application that uses the RMA's API to access data in the archive. Such a web application can query the archive for data containing an Indigenous community's language and then assign custom metadata to those data that meet the community's needs. This model for digital return is related to that of Murkurtu CMS (see §2.2.2), which also allows Indigenous communities to customize software to meet their needs for data management and curation. However, the model proposed here allows a community to develop custom methods for consuming their digital heritage resources, while still relying on the archive to preserve the resources themselves. In other words, the community gets to control their use of data while still getting the benefit of the archive's primary role of data preservation. This model for digital return will not be suitable for every Indigenous community, and indeed appropriate methods for digital return will vary across Indigenous communities (Barwick et al., 2019:3). For example, if a community wants to control data management and preservation in addition to controlling curation and consumption, Mukurtu CMS is a compelling option. However, many Indigenous communities have existing relationships with archives and rely on the archive's service of data preservation, as in the case of the NAL archive. The power of this new model for digital return is that it is infinitely extensible. Every community can have their own access portal that accesses data through the RMA's API like the APA does. Moreover, these access portals can be designed as microservices that streamline the customization process, making them easy to implement, even with limited resources.

5.4 Limitations of this work

The previous section summarized the benefits gained from developing a design for digital archive infrastructure, including increased quality of access, co-curation, and iterative archiving. However, much work remains to be done to ensure the archive remains centered in the processes of language documentation. The design proposed here augments more traditional designs with features from modern web-based applications in order to incrementally improve on archives' abilities to provide these outcomes, and this improvement will no doubt continue, both for NAL and other language archives. Moreover, in spite of these improvements, the software itself can never be the primary solution to the problem of how to implement better practices. In fact, software likely always plays a smaller role here than education, building relationships, and developing the culture around the work of language documentation.

As a good example of this limitation, implementing the API called for in the design will not by itself lead to every individual and group having the type and quality of access to data that they want or need. On the contrary, it is a small piece of the puzzle in accomplishing such a monumental goal. The API is like a digital pipeline that manages the flow of data out of the archive, with appropriate access restrictions, to other software applications. Achieving utility from this feature presupposes that other groups and institutions will acquire their own funding to build the software applications that connect to the other end of the pipeline and receive the data as they have been curated by the archive. This is a large and noteworthy limitation of the design and its use of an API.

In my personal experience, especially talking to various stakeholders at conferences, I find that many people get excited about this work and imagine its use by extrapolating the software's abilities far beyond what it is intended to accomplish. While the excitement is appreciated, and I believe this work is an important step toward accomplishing such goals, it is important to remember that this design adds functionality that is limited in scope, and this is desirable given archives' very limited resources. Nevertheless, this small change which requires only a small amount of additional resources to maintain is designed to enable others to build much smaller applications than they would be able to without the API, such as microservices for access, using their own resources. Thus, in addition to being limited in scope, this design anticipates future work by the current developers as well as other groups.

Another design feature that is limited in the outcomes it provides is the use of version control managed by the RMA. While it does enable co-curation, it only does so by mitigating the complexity of operations that depositors and archivists have to manage manually when modifying existing collections. Nevertheless, co-curation still requires collaboration and commitments on the part of stakeholders, which are larger requirements than the aspects mitigated by the software. Likewise, for iterative archiving, version control only makes the process slightly more streamlined, but we still need people to want to submit data later on as they become available, and they need to have the appropriate incentive structure to do so.

These examples highlight how relationships and the culture of doing the work are always more important than software. However, software is still important, and it is inevitable that software languages, frameworks, and packages change over time, so some resources should be applied to embracing these changes in order to get the benefits they bring. The alternative to embracing the change is letting software stagnate and ultimately spending the same resources on maintenance of outdated frameworks and packages.

In this context, this project has nevertheless reminded us how limited resources can be, even when we try to optimize for designs that have low demands for resources, and how funding structures aren't suited for maintenance and recurring costs. This is true in spite of how normal such costs are in this work, such as recurring costs for minting persistent identifiers and maintaining server infrastructure.

Finally, while the goal of the present software design is to be applicable to other organizations, extent to which this is achievable is limited by the unique context for each language archive. While language archives share the set of better practices in language documentation described in §2.5, the situation in which each archive operates is different. This makes it challenging to implement an overarching solution. Ultimately, the design needs to work for NAL first, and more work to generalize the solution can be done in the future.

5.5 Directions for future work

The most clear line for future work is the development of digital archive infrastructure for the NAL archive based on the design presented in Chapter 4. This is the most important outcome of the larger project that NAL is engaged in, and is necessary for NAL to meet its obligation to its users to bring online access to its collections.

As part of this development, the collaborators on the project will be fleshing out more details on how to implement

the API in order to meet the needs of the different modules. This will be important for the future work of extending the system of access through the development of other access portals to meet the needs of specific groups and communities.

Having implemented this design for NAL, future projects might focus on either implementing the software for other archives, or incorporating some of its features in the infrastructure of other archives. Additionally, Indigenous communities will be able to deploy the software designed by NAL for their own repositories.

Another line of inquiry that will become interesting as microservices are developed for customized access to NAL's collections is: what should be done with metadata that are developed in the use of these microservices? In the example given in §4.8.4, the microservice is a small web application that requests data from the archive through its API and presents these data using custom categories relevant for a specific user group. In this example, these categories are new metadata that do not exist in the archive but are relevant to the community using the web access portal. In the case that these metadata and categories are important to a community, that community may wish to have this information preserved in the archive alongside the existing collections. Thus, this creates a new feedback loop for bringing information into the archive and raises questions about how to achieve this, both in terms of the participation of stakeholders and the role of the archive's digital infrastructure in accomplishing such a task.

Finally, the model for digital return proposed in §5.3.4, which the design for archive infrastructure presented in Chapter 4 enables, can be further developed through integrations with Mukurtu CMS. This would take the form of a plugin for Mukurtu CMS that allows the application to ingest data from an archive directly through the RMA's API. This would benefit Indigenous communities who control their digital heritage using the Mukurtu CMS platform, and also have data in an archive they wish to incorporate in their instance of Mukurtu CMS. This process of moving data across platforms would be partially automated through the proposed plugin. When enabled, the plugin would extend the UI in Mukurtu CMS for ingesting data with a view of resources available to ingest from the archive running the RMA. To reach this view, the Mukurtu CMS user would authenticate with their RMA depositor account through a login view in Mukurtu CMS. Based on their RMA depositor account, the view in Mukurtu CMS would show a list of collections, items, and files for which the RMA depositor account has a moderation role. The Mukurtu CMS user would then be able to select resources for ingest, and confirm their choices. Mukurtu CMS would then proceed with the ingest as if uploading files from the user's local machine, but these files would be ingested directly from the RMA over the internet.

References

Albarillo, Emily E., and Nick Thieberger. 2009. Kaipuleohone, the University of Hawai'i's Ethnographic Archive. *Language Documentation & Conservation* 3(1): 154–181. http://hdl.handle.net/10125/4422.

Anderson, Jane, and Kimberly Christen. 2013. 'Chuck a Copyright on it': Dilemmas of Digital Return and the Possibilities for Traditional Knowledge Licenses and Labels. *Museum Anthropology Review* 7(1-2): 105–126.

Arka, I. Wayan. 2018. Reflections on the diversity of participation in language documentation. In *Language Documentation & Conservation Special Publication No.15: Reflections on Language Documentation 20 Years after Himmelmann 1998*. University of Hawai'i Press. http://hdl.handle.net/10125/24815.

Arnold, Denis, Bernhard Fisseni, Pawe\l Kamocki, Oliver Schonefeld, Marc Kupietz, and Thomas Schmidt. 2020. Addressing Cha (lle) nges in Long-Term Archiving of Large Corpora. In *Proceedings of the LREC 2020 Workshop, Language Resources and Evaluation Conference, 11–16 May 2020, 8th Workshop on Challenges in the Management of Large Corpora (CMLC-8)*, 1–9. Marseille, France: European Language Resources Association.

Austin, Peter K. 2014. Language documentation in the 21st century. *JournaLIPP* (3): 57–71. https://lipp.ub.uni-muenchen.de/lipp/article/view/190.

Babinski, Sarah, and Claire Bowern. 2021. Contemporary Digital Linguistics and the Archive: An Urgent Review. In *7th International Conference on Language Documentation and Conservation (ICLDC)*. University of Hawai'i at Mānoa. https://www.youtube.com/watch?v=aC06qrr1gcY.

Baldwin, Daryl, David J. Costa, and Douglas Troy. 2016. Myaamiaataweenki eekincikoonihkiinki eeyoonki aapisaataweenki: A Miami Language Digital Tool for Language Reclamation. *Language Documentation & Conservation* 10: 394–410. http://hdl.handle.net/10125/24713.

Barwick, Linda, Jennifer Green, Petronella Vaarzon-Morel, and Katya Zissermann. 2019. Conundrums and consequences: Doing digital archival returns in Australia. In *Language Documentation & Conservation Special Publication No.18: Archival Returns: Central Australia and Beyond*, ed. by Linda Barwick, Jennifer Green, and Petronella Vaarzon-Morel, 1–27. Honolulu & Sydney: University of Hawai'i Press & Sydney University Press. http://hdl.handle.net/10125/24875.

Barwick, Linda, and Nick Thieberger. 2018. Unlocking the archives. In *http://www.usyd.edu.au/disclaimer.shtml*. Foundation for Endangered Languages. https://ses.library.usyd.edu.au/handle/2123/20395.

Berez-Kroeker, Andrea, Lauren Gawne, Barbara F. Kelly, and Tyler Heston. 2017. A survey of current reproducibility practices in linguistics journals, 2003-2012 https://sites.google.com/a/hawaii.edu/data-citation/survey.

Berez-Kroeker, Andrea L. 2015. Reproducible research in descriptive linguistics: Integrating archiving and citation into the postgraduate curriculum at the University of Hawai'i at Mānoa. In *Research, Records, and Responsibility: Ten years of PARADISEC*, ed. by Amanda Harris, Linda Barwick, and Nick Thieberger, 39–51. Sydney: University of Sydney Press.

Berez-Kroeker, Andrea L., Lauren Gawne, Susan Smythe Kung, Barbara F. Kelly, Tyler Heston, Gary Holton, Peter Pulsifer, David I. Beaver, Shobhana Chelliah, Stanley Dubinsky, Richard P. Meier, Nick Thieberger, Keren Rice, and Anthony C. Woodbury. 2018. Reproducible research in linguistics: A position statement on data citation and attribution in our field. *Linguistics* 56(1): 1–18.

Berez-Kroeker, Andrea L., and Ryan Henke. 2018. Language Archiving. In *The Oxford Handbook of Endangered Languages*, ed. by Kenneth Rehg and Lyle Campbell. Oxford University Press, 1 edition.

Bettinson, Mat. 2019. Enabling Large-Scale Collaboration in Language Conservation. Doctoral Dissertation, University of Melbourne.

Bettinson, Mat, and Steven Bird. 2017. Developing a Suite of Mobile Applications for Collaborative Language Documentation. In *Proceedings of the 2nd Workshop on the Use of Computational Methods in the Study of Endangered Languages*, 156–164. Honolulu: Association for Computational Linguistics. https://www.aclweb.org/anthology/W17-0121.

Bird, Steven, and Mark Liberman. 2001. A formal framework for linguistic annotation. *Speech Communication* 33(1): 23–60. http://www.sciencedirect.com/science/article/pii/S0167639300000686.

Bird, Steven, and Gary Simons. 2003a. Extending Dublin Core Metadata to Support the Description and Discovery of Language Resources. *Computers and the Humanities* 37(4): 375–388. https://www.jstor.org/stable/30204912.

Bird, Steven, and Gary Simons. 2003b. Seven Dimensions of Portability for Language Documentation and Description. *Language* 79(3): 557–582.

Bonaccorsi, Andrea, and Cristina Rossi. 2003. Why Open Source software can succeed. *Research Policy* 32(7): 1243–1258. https://www.sciencedirect.com/science/article/pii/S0048733303000519.

Brinklow, Nathan Thanyehténhas. 2021. Indigenous language technologies: Anti-colonial oases in a colonizing (digital) world. *WINHEC: International Journal of Indigenous Education Scholarship* (1): 239–266.

Broeder, Daan, Marc Kemps-Snijders, Dieter Van Uytvanck, Menzo Windhouwer, Peter Withers, Peter Wittenburg, and Claus Zinn. 2010. A data category registry- and component-based metadata framework. 43–47. European Language Resources Association (ELRA). https://pure.mpg.de/pubman/faces/ViewItemOverviewPage.jsp?itemId=item_442673.

Carroll, Stephanie Russo, Edit Herczog, Maui Hudson, Keith Russell, and Shelley Stall. 2021. Operationalizing the CARE and FAIR Principles for Indigenous data futures. *Scientific Data* 8(1): 108. https://www.nature.com/articles/s41597-021-00892-0.

Chelliah, Shobhana. 2018. Reflections on language documentation in India. In *Language Documentation & Conservation Special Publication No.15: Reflections on Language Documentation 20 Years after Himmelmann 1998*, ed. by Bradley McDonnell, Andrea Berez-Kroeker, and Gary Holton. University of Hawai'i Press. http://hdl.handle.net/10125/24826.

Conathan, Lisa. 2011. Archiving and language documentation. In *The Cambridge Handbook of Endangered Languages*, ed. by Peter Austin and Julia Sallabank, 235–254. Cambridge, UNITED KINGDOM: Cambridge University Press. http://ebookcentral.proquest.com/lib/uhm/detail.action?docID=691813.

Czaykowska-Higgins, Ewa. 2009. Research Models, Community Engagement, and Linguistic Fieldwork: Reflections on Working within Canadian Indigenous Communities. *Language Documentation & Conservation* 3(1): 182–215. http://hdl.handle.net/10125/4423.

Derrida, Jacques. 1995. Archive fever: A Freudian impression. *Diacritics* 25(2): 9–63.

Evans, Nicholas, and Alan Dench. 2006. Introduction: Catching language. In *The stand- ing challenge of grammar writing (Trends in Linguistics Studies and Monographs 167)*, ed. by Felix K. Ameka, Alan Dench, and Nicholas Evans, 1–39. Berlin: Mouton de Gruyter.

Garrett, Edward. 2014. Participant-driven language archiving. In *Language Documentation and Description, vol 12: Special Issue on Language Documentation and Archiving.*, ed. by David Nathan and Peter Austin, 68–84. London: SOAS.

Garrett, Jesse James. 2011. *The Elements of User Experience: User-Centered Design for the Web and Beyond*. Berkeley, CA: New Riders.

Gawne, Lauren, Andrea Berez-Kroeker, Helene Andreassen, Philipp Conzett, Koenraad De Smedt, Christopher Cox, and Lauren Collister. 2021. Using the Tromsø Recommendations to cite data in language work. In *7th International Conference on Language Documentation and Conservation (ICLDC)*.

Gawne, Lauren, Barbara F. Kelly, Andrea L. Berez-Kroeker, and Tyler Heston. 2017. Putting practice into words: The state of data and methods transparency in grammatical descriptions. *Language Documentation & Conservation* 11: 157–189.

Gezelter, Dan. 2009. Being Scientific: Falsifiability, Verifiability, Empirical Tests, and Reproducibility. The OpenScience Project. https://openscience.org/being-scientific-fasifiability-verifiability-empirical-tests-and-reproducibility/.

Golla, Victor. 1995. The records of American Indian linguistics. In *Preserving the anthropological record*, ed. by Sydel Silverman and Nancy J. Parezo, 143–157. New York: Wenner-Gren Foundation for Anthropological Research.

Hakala, Juha. 2010. Persistent identifiers: an overview. KIM Technology Watch Report. http://www.persid.org/downloads/PI-intro-2010-09-22.pdf.

Hall, Patrick. 2022. Participatory Design in Digital Language Documentation: A Web Platform Approach. Doctoral Dissertation, University of California Santa Barbara. https://www.proquest.com/openview/db943d69bd368295036b639ce12d0e6d/1?pq-origsite=gscholar&cbl=18750&diss=y.

Hanke, Florian R. 2017. Computer Supported Collaborative Language Documentation. Doctoral Dissertation, University of Melbourne.

Hatton, John, Gary Holton, Mandana Seyfeddinipur, and Nick Thieberger. 2021. Lameta [software]. https://github.com/onset/laMETA/releases.

Henke, Ryan, and Andrea L. Berez-Kroeker. 2016. A Brief History of Archiving in Language Documentation, with an Annotated Bibliography. *Language Documentation & Conservation* 10: 411–457.

Himmelmann, Nikolaus. 1998. Documentary and descriptive linguistics. *Linguistics* 36(1): 161–195.

Himmelmann, Nikolaus P. 2006. Language documentation: What is it and what is it good for? In *Essentials of Language Documentation*, 1–30. De Gruyter Mouton. https://www.degruyter.com/document/doi/10.1515/9783110197730.1/html.

Himmelmann, Nikolaus P. 2012. Linguistic Data Types and the Interface between Language Documentation and Description. *Language Documentation & Conservation* 6: 187–207.

Himmelmann, Nikolaus P. 2018. Meeting the transcription challenge. In *Language Documentation & Conservation Special Publication No.15: Reflections on Language Documentation 20 Years after Himmelmann 1998*, ed. by Bradley McDonnell, Andrea L. Berez-Kroeker, and Gary Holton. University of Hawai'i Press.

Holton, Gary. 2009. Relatively Ethical: A Comparison of Linguistic Research Paradigms in Alaska and Indonesia. *Language Documentation & Conservation* 3(2): 161–175. http://hdl.handle.net/10125/4424.

Holton, Gary. 2011. "Unknown Unknowns" and the Retrieval Problem in Language Documentation and Archiving. *Language Documentation & Conservation* 5: 157–168. http://hdl.handle.net/10125/4496.

Holton, Gary. 2012. Language archives: They're not just for linguists any more. In *Language Documentation & Conservation Special Publication No.3: Potentials of Language Documentation: Methods, Analyses, and Utilization*, ed. by Frank Seifart, Geoffrey Haig, Nikolaus P. Himmelmann, Dagmar Jung, Anna Margetts, and Paul Trilsbeek, 111–117. University of Hawai'i Press.

Holton, Gary. 2017. From Community to Archive and Back: Language archives and digital return. In *FEL XXL: Communities in Control: Learning tools and strategies for multilingual endangered language communities*, ed. by Nicholas Ostler, Vera Ferreira, and Christopher Moseley, 17–23. Alcanena, Portugal.

Holton, Gary, Kavon Hooshiar, and Nick Thieberger. 2017. Developing collection management tools to create more robust and reliable linguistic data. In *Proceedings of the 2nd Workshop on the Use of Computational Methods in the Study of Endangered Languages*, ed. by Antti Arppe, Jeff Good, Mans Hulden, Jordan Lachler, Alexis Palmer, and Lane Schwartz, 33–38. Honolulu, Hawai'i: Association for Computational Linguistics. https://aclanthology.org/W17-0105.

Hooshiar, Kavon. 2017. The LDTC Model and its Impact: a Piece of the Puzzle in Language Conservation and Documentation. In *FEL XXL: Communities in Control: Learning tools and strategies for multilingual endangered language communities*, ed. by Nicholas Ostler, Vera Ferreira, and Christopher Moseley, 78–82. Hungerford, England: Foundation for Endangered Languages.

von Krogh, Georg, and Eric von Hippel. 2006. The Promise of Research on Open Source Software. *Management Science* 52(7): 975–983. https://pubsonline.informs.org/doi/abs/10.1287/mnsc.1060.0560.

Kunze, John, and Thomas Baker. 2007. The Dublin core metadata element set. IETF RFC 5013. https://www.rfc-editor.org/rfc/rfc5013.

Law, Lai-Chong, Virpi Roto, Marc Hassenzahl, Arnold Vermeeren, and Joke Kort. 2009. Understanding, scoping and defining user experience: A survey approach. In *Proceedings of the 27th International Conference on Human Factors in Computing Systems, CHI 2009, Boston, MA, USA, April 4-9, 2009*.

Leonard, Wesley. 2018. Reflections on (de)colonialism in language documentation. In *Language Documentation & Conservation Special Publication No.15: Reflections on Language Documentation 20 Years after Himmelmann 1998*, ed. by Bradley McDonnell, Andrea Berez-Kroeker, and Gary Holton, 55–65. University of Hawai'i Press. http://hdl.handle.net/10125/24808.

Leonard, Wesley Y. 2017. Producing language reclamation by decolonising 'language'. In *Language Documentation and Description*, ed. by Wesley Y. Leonard and Haley De Korne, volume 14, 15–36. London: EL Publishing.

Leonard, Wesley Y., and Erin Haynes. 2010. Making "collaboration" collaborative: An examination of perspectives that frame linguistic field research. *Language Documentation & Conservation* 4: 269–293. http://hdl.handle.net/10125/4482.

Linn, Mary S. 2014. Living archives: A community-based language archive model. *Language Documentation and Description* 12: 53–67.

Marten, Lutz, and Malin Petzell. 2016. Linguistic variation and the dynamics of language documentation: Editing in 'pure' Kagulu. In *Language Documentation & Conservation Special Publication No.10: African language documentation: new data, methods and approaches*, ed. by Mandana Seyfeddinipur, 105–129. University of Hawai'i Press.

McDonnell, Brad. 2017. Git use cases. In *1st workshop on Metadata Editing and Collection Management (MEaCoM)*. Alcanena, Portugal.

McMurry, Julie A., Nick Juty, Niklas Blomberg, Tony Burdett, Tom Conlin, Nathalie Conte, Mélanie Courtot, John Deck, Michel Dumontier, Donal K. Fellows, Alejandra Gonzalez-Beltran, Philipp Gormanns, Jeffrey Grethe, Janna Hastings, Jean-Karim Hériché, Henning Hermjakob, Jon C. Ison, Rafael C. Jimenez, Simon Jupp, John Kunze, Camille Laibe, Nicolas Le Novère, James Malone, Maria Jesus Martin, Johanna R. McEntyre, Chris Morris, Juha Muilu, Wolfgang Müller, Philippe Rocca-Serra, Susanna-Assunta Sansone, Murat Sariyar, Jacky L. Snoep, Stian Soiland-Reyes, Natalie J. Stanford, Neil Swainston, Nicole Washington, Alan R. Williams, Sarala M. Wimalaratne, Lilly M. Winfree, Katherine Wolstencroft, Carole Goble, Christopher J. Mungall, Melissa A. Haendel, and Helen Parkinson. 2017. Identifiers for the 21st century: How to design, provision, and reuse persistent identifiers to maximize utility and impact of life science data. *PLOS Biology* 15(6): e2001414. https://journals.plos.org/plosbiology/article?id=10.1371/journal.pbio.2001414.

Moeller, Sarah Ruth. 2014. Review of SayMore, a tool for Language Documentation Productivity. *Language Documentation & Conservation* 8: 66–74.

Mutibwa, Daniel H, Alison Hess, and Tom Jackson. 2020. Strokes of serendipity: Community co-curation and engagement with digital heritage. *Convergence* 26(1): 157–177. https://doi.org/10.1177/1354856518772030.

Nadareishvili, Irakli, Ronnie Mitra, Matt McLarty, and Mike Amundsen. 2016. *Microservice architecture: aligning principles, practices, and culture*. Sebastopol, CA: O'Reilly Media, Inc.

Nathan, David. 2011. Digital archiving. In *The Cambridge handbook of endangered languages*, ed. by Peter K. Austin and Julia Sallabank, 255–273. Cambridge: Cambridge University Press.

Nathan, David. 2015. On the reach of digital language archives. Sydney University Press. https://ses.library.usyd.edu.au/handle/2123/16673.

O'Sullivan, Bryan. 2009. *Mercurial: The Definitive Guide: The Definitive Guide*. "O'Reilly Media, Inc.". Google-Books-ID: upW0sR6BC_wC.

Paterson, Hugh J. III. 2021. Language Archive Records: Interoperability Of Referencing Practices And Metadata Models. Doctoral Dissertation, University of North Dakota. https://commons.und.edu/theses/3937.

Rau, Felix. 2016. CMDI Maker - the state and prospects of a HTML5 Web app. In *Tools Summit*. Melbourne, Australia.

Sawaki, Yusuf, and I. Wayan Arka. 2018. Reflections on linguistic fieldwork and language documentation in eastern Indonesia. In *Language Documentation & Conservation Special Publication No.15: Reflections on Language Documentation 20 Years after Himmelmann 1998*, ed. by Bradley McDonnell, Andrea Berez-Kroeker, and Gary Holton. University of Hawai'i Press. http://hdl.handle.net/10125/24827.

Shepard, Michael. 2016. The Value-Added Language Archive: Increasing Cultural Compatibility for Native American Communities http://hdl.handle.net/10125/24715.

Simons, Gary, Steven Bird, and Joan Spanne. 2008. OLAC Metadata Usage Guidelines. OLAC: Open Language Archives Community. http://www.language-archives.org/NOTE/usage.html.

Smith, Linda Tuhiwai. 1999. *Decolonizing Methodologies: Research and Indigenous Peoples*. Zed Books. Google-Books-ID: Nad7afStdr8C.

Sweigart, Al. 2022. 16. Working with CSV files and JSON data. Automate the Boring Stuff with Python. https://automatetheboringstuff.com/2e/chapter16/.

Theimer, Kate. 2011. Exploring the participatory archives: What, who, where, and why. In *Annual Meetings of the Society of American Archivists*. http://www.slideshare.net/ktheimer/theimer-participatory-archives-saa-2011.

Thieberger, Nick. 2019. The ongoing challenge of connecting speakers to archival language records. In *6th International Conference on Language Documentation and Conservation (ICLDC)*. Honolulu, Hawai'i. http://hdl.handle.net/10125/44786.

Trilsbeek, Paul, and Peter Wittenburg. 2006. Archiving challenges. In *Essentials of language documentation*, ed. by Jost Gippert, Nikolaus P. Himmelmann, and Ulrike Mosel, number 178 in Trends in Linguistics Studies and Monographs, 311–336. Berlin: Mouton de Gruyter.

Wasson, Christina, Gary Holton, and Heather S. Roth. 2016. Bringing User-Centered Design to the Field of Language Archives. *Language Documentation & Conservation* 10: 641–681.

Widlok, Thomas. 2013. Analogical Problems with Digital Data. *The Asia Pacific Journal of Anthropology* https://www-tandfonline-com.eres.library.manoa.hawaii.edu/doi/abs/10.1080/14442213.2013.768694.

Wilkinson, Mark D., Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E. Bourne, Jildau Bouwman, Anthony J. Brookes, Tim Clark, Mercè Crosas, Ingrid Dillo, Olivier Dumon, Scott Edmunds, Chris T. Evelo, Richard Finkers, Alejandra Gonzalez-Beltran, Alasdair J. G. Gray, Paul Groth, Carole Goble, Jeffrey S. Grethe, Jaap Heringa, Peter A. C. 't Hoen, Rob Hooft, Tobias Kuhn, Ruben Kok, Joost Kok, Scott J. Lusher, Maryann E. Martone, Albert Mons, Abel L. Packer, Bengt Persson, Philippe Rocca-Serra, Marco Roos, Rene van Schaik, Susanna-Assunta Sansone, Erik Schultes, Thierry Sengstag, Ted Slater, George Strawn, Morris A. Swertz, Mark Thompson, Johan van der Lei, Erik van Mulligen, Jan Velterop, Andra Waagmeester, Peter Wittenburg, Katherine Wolstencroft, Jun Zhao, and Barend Mons. 2016. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data* 3(1): 160018. https://www.nature.com/articles/sdata201618.

Wittenburg, Peter, Romuald Skiba, and Paul Trilsbeek. 2005. The language archive at the MPI: Contents, tools, and technologies. *Language Archives Newsletter* 5: 7–9.

Woodbury, Anthony C. 2014. Archives and audiences: Toward making endangered language documentations people can read, use, understand, and admire. In *Special Issue on Language Documentation and Archiving*, ed. by David Nathan and Peter Austin, number 12 in Language documentation and description, 19–36. London: SOAS. https://www.academia.edu/9194345/Woodbury_Anthony_C._2014._Archives_and_audiences_Toward_making_endangered_language_documentations_people_can_read_use_understand_and_admire._In_David_Nathan_and_Peter_K._Austin_eds_LD_and_D_12_Special_Issue_on_Language_Documentation_and_Archiving._London_SOAS._pp._19-36.