

Development of the Canadian Agri-food Lifecycle Data Centre with Data Format Interoperability Requirements

by

Matthew Stanley Fritter

B.A., The University of British Columbia, 2017

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

in

The College of Graduate Studies

(Computer Science)

THE UNIVERSITY OF BRITISH COLUMBIA

(Okanagan)

June 2020

© Matthew Stanley Fritter, 2020

The following individuals certify that they have read, and recommend to the College of Graduate Studies for acceptance, a thesis/dissertation entitled:

DEVELOPMENT OF THE CANADIAN AGRI-FOOD LIFECYCLE DATA
CENTRE WITH DATA FORMAT INTEROPERABILITY REQUIREMENTS

submitted by MATTHEW STANLEY FRITTER in partial fulfilment of the requirements of the degree of Master of Science

Dr. Ramon Lawrence, I. K. Barber School of Arts & Sciences
Supervisor

Dr. Nathan Pelletier, I. K. Barber School of Arts & Sciences, Faculty of Management
Supervisor

Dr. Yves Lucet, I. K. Barber School of Arts & Sciences
Supervisory Committee Member

Dr. Barb Marcolin, Faculty of Management
Supervisory Committee Member

Dr. Ying Zhu, Faculty of Management
University Examiner

Dr. Jonathan Little, Health & Social Development
External Examiner

Abstract

The field of Life Cycle Assessment (LCA) models the resource flows and emissions characteristic of real-world industrial, agricultural, and economic activities through the use of Life Cycle Inventory (LCI) datasets. As the amount of data available to LCA practitioners through national and commercial database initiatives increases, there have been growing concerns within the LCA community regarding the interoperability of LCI data. Choice of data format and nomenclature poses problems for re-usability, as a dataset may not cleanly integrate into an LCA model due to differences in nomenclature, or a practitioner's LCA software may simply not recognize the format type. This interoperability has been identified as one of the largest problems, along with data availability, in the LCA field.

The focus of this research was the development of a new national Life Cycle Inventory database: The Canadian Agri-food Life Cycle Data Center (CALDC), which will serve as a central repository for Canadian agri-food data. During the course of the research, information was solicited from existing LCA database providers to inform development, and potential solutions for the interoperability issues were researched and implemented within the CALDC. The development included a searchable public database repository, as well as a web application that allows users to create, modify, and publish new LCI datasets, known as SimpLCity.

A set of recommendations were drafted for new LCI database initiatives, with the goal of increasing the interoperability between databases and datasets and increasing the availability of data. These recommendations were used in the development of the CALDC, and also present potential future avenues for expansion and development, such as the implementation of Application Programming Interfaces (APIs) or the re-distribution of datasets through third-party data providers and initiatives.

The Canadian Agri-food Life Cycle Data Centre is now live, and is currently being used by researchers at both UBC and external stakeholder partners such as the Canadian Roundtable for Sustainable Beef (CRSB) to create and publish new publicly available agri-food data for LCA research.

Lay Summary

This research proposes and implements a new central database, known as the Canadian Agri-Food Life Cycle Data Centre (CALDC), which allows users to both contribute and access publicly available life cycle inventory datasets. These datasets describe the inputs and outputs of agricultural and industrial processes. Often, such datasets variously utilize one of several different, incompatible formats and naming systems. The datasets produced by the CALDC are designed to support multiple formats, be easily interoperable with different kinds of LCA software, and can be published publicly on the database, allowing data needed to model agricultural emissions to be easily created, shared, accessed, and used by researchers free of charge.

Preface

Parts of this thesis were accepted for publishing in the International Journal of Life Cycle Assessment (IJLCA), as of June 2020.

Table of Contents

Abstract	iii
Lay Summary	iv
Preface	v
Table of Contents	vi
List of Tables	ix
List of Figures	x
Acknowledgements	xii
Dedication	xiii
Chapter 1: Introduction	1
Chapter 2: Background	3
2.1 LCA Methodologies	3
2.2 The Life Cycle Inventory	5
2.3 Interoperability in LCI	8
2.3.1 Existing LCI Data Formats	8
2.3.2 Other Interoperability Issues	9
2.4 Third Party Database Initiatives and Programs	10
2.5 Summary	13
Chapter 3: Methodology	15
3.1 Initial Objectives	15
3.2 Technical Considerations	16
3.3 Interoperability Requirements	18
3.4 Third Party Providers & Initiatives	19

TABLE OF CONTENTS

3.5	Development & Testing	19
3.6	Summary	22
Chapter 4: Implementation		23
4.1	Technical Findings	23
4.1.1	Existing LCI Databases	23
4.1.2	Formats & Nomenclatures	25
4.2	Software & Architecture Choice	28
4.2.1	Web Application	28
4.2.2	Database	29
4.2.3	Stack Architecture	30
4.2.4	Database Design	31
4.3	Security & Maintenance Considerations	34
4.4	Development Timeline	37
Chapter 5: Features		39
5.1	Application Workflow	39
5.2	Feature Walkthrough	41
5.2.1	Logging In	42
5.2.2	User Home Page	42
5.2.3	Additional User Options	43
5.2.4	Browsing Datasets	43
5.2.5	Viewing Datasets	44
5.2.6	Managing Datasets	45
5.2.7	Creating a New Dataset	46
5.2.8	Editing a Dataset	47
5.2.9	Exporting a Dataset	49
5.2.10	Importing a Dataset	51
5.2.11	Support Page	52
5.2.12	Managing User Settings	52
5.2.13	Submitting User Feedback	53
5.2.14	Using the Admin Console	54
5.2.15	Public CALDC Homepage	56
5.2.16	Navigating the Public CALDC	58
5.2.17	Viewing and Downloading Public Datasets	59
Chapter 6: Discussion and Recommendations		63
6.1	Interoperability Recommendations	64
6.2	Technical Recommendations	67
6.3	Third Party Data Providers and Initiatives Recommendations	68

TABLE OF CONTENTS

Chapter 7: Conclusion	69
Bibliography	71
Appendix	78
Appendix A: Risk Assessment	79
A.1 Injection	79
A.2 Broken Authentication	79
A.3 Sensitive Data Exposure	80
A.4 XML External Entities	80
A.5 Broken Access Control	81
A.6 Security Misconfiguration	81
A.7 Cross-Site Scripting	82
A.8 Insecure Deserialization	82
A.9 Using Components with Known Vulnerabilities	83
A.10 Insufficient Logging & Monitoring	83

List of Tables

Table 4.1	Three major LCI databases compared.	25
Table 4.2	Average and Maximum Sizes of Datasets by Type. . .	33
Table 5.1	Summary of Features and Permissions.	41
Table 6.1	Total Counts as of July 2020.	64
Table 6.2	Recommendations for LCI Database Initiatives.	65
Table A.1	Injection Risks.	79
Table A.2	Broken Authentication Risks.	80
Table A.3	Sensitive Data Exposuren Risks.	80
Table A.4	XML External Entities.	81
Table A.5	Broken Access Control.	81
Table A.6	Security Misconfiguration.	82
Table A.7	Cross-Site Scripting.	82
Table A.8	Insecure Deserialization.	83
Table A.9	Using Components with Known Vulnerabilities.	83
Table A.10	Insufficient Logging & Monitoring.	84

List of Figures

Figure 3.1	Typical Web Software Stack	17
Figure 4.1	CALDC Software Stack	31
Figure 4.2	CALDC Database Tables	32
Figure 5.1	Public-facing Web User Flow	40
Figure 5.2	Private-facing Web User Flow	42
Figure 5.3	CALDC Splash Page	43
Figure 5.4	SimpLCity Home Page	44
Figure 5.5	User Menu	44
Figure 5.6	Dataset Browser	45
Figure 5.7	Viewing a Dataset	46
Figure 5.8	Dataset Completion Message	46
Figure 5.9	Empty Dataset Value Warnings	47
Figure 5.10	Viewing ILCD Reference Flows	47
Figure 5.11	Dataset Actions	48
Figure 5.12	Dataset Dependencies	48
Figure 5.13	Dataset Creation Selection	49
Figure 5.14	Dependency Warning	49
Figure 5.15	Editing Interface	50
Figure 5.16	Placeholders and Tooltips	51
Figure 5.17	Input Validation	51
Figure 5.18	Exchange Interface	52
Figure 5.19	Confirmation Page	52
Figure 5.20	Export Dialog	53
Figure 5.21	Exported Files	54
Figure 5.22	Import Interface	55
Figure 5.23	Support Page	56
Figure 5.24	User Settings	56
Figure 5.25	Feedback Form	57
Figure 5.26	Feedback Submission Message	57

LIST OF FIGURES

Figure 5.27	Submissions Table	58
Figure 5.28	Administrator Actions	58
Figure 5.29	Approved Datasets	59
Figure 5.30	Adding Administrators	59
Figure 5.31	Public Datasets Access	60
Figure 5.32	CALDC Public Homepage	60
Figure 5.33	Viewing Public Datasets	61
Figure 5.34	Exploring a Public Dataset	61
Figure 5.35	Downloading Public Datasets	62
Figure 5.36	ILCD Reference Flow in Public Viewer	62

Acknowledgements

This work was supported with funds from the Natural Sciences and Engineering Research Council of Canada (NSERC)/Egg Farmers of Canada (EFC) Industrial Research Chair in Sustainability.

I would like to acknowledge the help and guidance provided to me by Prof. Lawrence and Prof. Pelletier, without whom this thesis would not have been possible. I would also like to thank Prof. Lucet and Prof. Marcolin for their role on my graduate committee, and Prof. Fazackerley, Prof. Nastos, and Prof. Ould-Khessal for their continuing camaraderie.

Thank you also to Davoud Heidari, Ian Turner, and Vivek Arulnathan, my fellow PRISM lab researchers who contributed much to the development and testing of our software and whose feedback and insight was invaluable.

I would also like to thank all those who offered me insight into the world of Life Cycle Assessment, in particular Peter Arbuckle of the US Department of Agriculture, who has been an invaluable resource in this research.

Dedication

To my friends, my family, my coworkers, and my students who have accompanied me on this academic journey and have wished me luck.

Chapter 1

Introduction

Life Cycle Assessment (LCA) is a heavily data-reliant approach to modelling resource flows and emissions associated with the production and use of products and/or services, primarily for industrial and agricultural production. This technique allows entire supply chains to be modelled and the resulting emissions to be measured and aggregated into useful metrics. The development and distribution of datasets used in these models are contributed to by both public researchers and commercial businesses; most LCA practitioners choose to make use of a dedicated Life Cycle Inventory (LCI) database to support their modelling [CF06] – in particular with respect to inputs to the product system of interest. Such databases have been developed at both national and international levels, both as publicly available resources and commercially licensed products.

Although Life Cycle Assessment was conceived as a field in the 1960's, and developed throughout the 1980's and 1990's [HBOM17], there has been relatively little standardization in terms of data exchange formats, nomenclature, and technical implementation of LCI databases. Those standards that do exist such as ISO 14048, are broad enough that interoperability between two claimed ISO 14048-compliant LCI data sources may be poor. The lack of shared best practices in the production and formatting of LCI data is believed to have negatively impacted both the interoperability of the datasets, and their reusability [RBF⁺15]. As LCI datasets can potentially contain a large amount of data that is usually manually entered, the cost to produce these datasets, in terms of time and effort, is not trivial. Therefore, by promoting interoperability with different dataset formats and nomenclatures, the overall cost in time can be reduced and the sharing of LCI data is encouraged.

To date, there have been few LCI databases developed specifically for Canadian data. A provincial LCI database was developed for Quebec as part of the much largerecoinvent commercial LCI database [LS16]; at a national level the Canadian Raw Materials Database (CRMD) provides LCI data for the aluminum, glass, plastics, steel, and lumber commodity industries, with data collection up to 1998 [oW]. This means that much of the Canadian

agri-food sector outside of Quebec does not have a data repository to access or to contribute to, even as new LCI datasets are being created. The Food Systems Priority Research for Integrated Sustainability Management (PRISM) Lab at the University of British Columbia Okanagan campus has proposed to fill this gap through the development of a new LCI database, the Canadian Agri-food Life Cycle Data Centre (CALDC). The proposed database would provide a public repository for Canadian agri-food LCI data, allowing for the development of accurate life cycle models to support sustainability measurement and management initiatives in the Canadian agri-food industry [Pel].

This thesis describes the goals, methods, and implementation of the new CALDC database, as well as general observations of the current state of LCI databases and recommendations for the development of new, interoperable LCI data sources. It also suggests some features, now common in certain aspects of commercial web data applications and analysis, that may be implemented in future LCI databases to further promote good data management and sharing practices. The contributions of this thesis include a study of the major interoperability issues existent between major data formats; the development, testing, and initial release of the new CALDC database; and the development of a set of recommendations for further interoperability in LCI data sources. Chapter two provides a background on the field of Life Cycle Assessment and its data sources, the interoperability and usability concerns that the LCI field currently faces, and the objectives of CALDC development. Chapter three discusses the methodology used for the development of the CALDC and LCI database recommendations. Chapters four and five cover the technical findings and chosen technical implementation for the CALDC, in addition to a walkthrough of workflow and features implemented. Chapter six discusses current trends in LCI database development, and further recommendations for LCI databases.

Chapter 2

Background

In developing a new LCI database, it is necessary to consider that the methodology and standards used in LCA practice can vary widely. It is also necessary to understand the different datasets that make up a typical LCA model, as this affects which necessary data and metadata must be accommodated within the LCI database. This chapter covers the background of LCA methodology and LCI datasets, as well as covering the interoperability challenges that have been identified in the LCI field, and existing LCI database implementations.

2.1 LCA Methodologies

Although the concept of performing a holistic analysis of inputs, outputs, emissions, and wastes for industrial processes that is now known as LCA was developed almost five decades ago, the framework, methodologies, and standards for this analysis remain in development to the present day. The Midwest Research Institute was a primary source of initial LCA studies during the late 1960's and early 1970's, known at the time as Resource and Environmental Profile Analysis (REPA) [GHH⁺11]. However, it has been suggested that these were generally unpublished internal corporate studies of narrow scope, done during a time when LCA/REPA was primarily company driven. By the latter half of the 1970's this corporate drive was being superseded by regulatory and compliance goals for LCA research [MT15]. However, no common methodology, terminology, or framework was developed for LCA until the 1990's, creating a chaotic period between 1970 and 1990 that Guinée et al. suggest prevented the more widespread adoption of LCA, due to unreliable or irreproducible results produced by studies during this period [GHH⁺11].

By the early 1990's, a movement towards the development and adoption of a common set of standards and methodologies was underway. This was led in part by the creation of the Society of Environmental Toxicology and Chemistry (SETAC) in 1989, whose early standards for retrospective analysis formed part of both regulatory policy and the original 1997 ISO

2.1. LCA Methodologies

14040 standards for LCA frameworks [MT15]. Concurrently, the Netherlands sponsored a study on LCA standardization through the National Reuse of Waste Research Programme. This culminated in the 1992 publishing of the Environmental LCA of Products guide by the Centre of Environmental Science at Leiden University [DBG02]. Following further research by SETAC and subsequent revisions of ISO 14040 in 1997, the Handbook on LCA was published in 2002 as a successor to the original 1992 guide [DBG02]. A set of guidelines for LCA methodology were similarly developed in Denmark between 1997 and 2003; these were developed for the Danish Environmental Protection Agency, and were designed to provide greater detail than existing standards [Wei03]. Additional effort was made to improve the existing standards, resulting in the release of a 2006 revision for ISO 14040, and the introduction of the new and more popular ISO 14044 standard. The 14044 standard encompassed all the requirements of previous ISO LCA standards, while improving readability and consistency to create a set of requirements and guidelines for LCA study [FIT⁺06].

While there has been a proliferation of LCA standards, guidelines, and methodologies since the 1990's, there remain unresolved issues that have prevented adoption of a single LCA framework or methodology. Contentious issues include the acceptability of market mechanisms within LCA models, and the value of attributional versus consequential methodologies in LCA. Consequential modeling methodology is aimed at determining the environmental consequences of a given decision, including indirect economic effects and market pressures. This necessitates the use of system expansion and substitution to avoid multi-functionality problems [PAB⁺15]. Although ISO 14044 provides a hierarchy for allocation, the lack of consensus on allocation procedure has been recognized as a problem in LCA, particularly in industries where multi-functional systems are common [KL14]. Similarly, the Dutch and Danish guidelines for LCA are split over the issue of including market data. The use of monetary value of outputs as a functional unit or as a means of allocating emissions based on product values has been contentious; with Guinée et al. arguing that market mechanisms are outside the scope of LCA [DBG02], while Weidema supports the use of market mechanisms in LCA [Wei03].

Beyond the development of LCA frameworks and methodologies, an area of particular interest is the development of LCI databases. Data availability and quality has been classified as one of the most severe problems facing LCA, a problem compounded by a lack of established data quality analysis standards for LCI data [RRDB08a]. Current LCI databases either tend to be generic databases such as ecoinvent or GaBi, or are limited to a specific

sector or product; variances between generic and sector-specific database values such as data localization and aggregation can result in drastically deviating results for the same model [LHPC15]. Agri-food LCA also poses unique challenges, commonly involving multi-functional systems that demand expansion or allocation, as well as requiring geospatially accurate data to account for different agriculture situations within a region [NSA⁺17].

2.2 The Life Cycle Inventory

In terms of time and monetary expense, the development of the LCI is usually the most involved phase within a LCA study. Following the definition of the project goal and scope, the LCI phase includes enumerating and diagramming all unit processes within the system boundaries, the collection and validation of process data, as well as the identification and justification of modelling decisions such as cut-off criteria or allocation methods [dBG02]. However, modelling decisions and data must be approached with caution during this phase, as uncertainties in the data, choice of arbitrary cut-off criteria, or issues with allocation methodology can result in significant errors in final analysis [RRDB08b] [RRDB08a]. ISO 14044 includes requirements for data validation, sensitivity analysis when refining system boundaries, and documentation of anomalies in the data or special cases within the system [ISO06], potentially avoiding errors or clearly and transparently documenting where modelling decisions have been made that could influence the results of the assessment.

The initial step in the LCI phase is the identification of unit processes within the product system(s), including all input, output, waste, and energy flows associated with them, in terms of the functional unit, which is used as the metric for analysis; for example, an output volume in units or weight of the final product. A single unit process may contain several activities, and is linked to other unit processes via intermediate flows [REF⁺04]. A flow diagram is created showing the unit processes that fall within the system boundary and the intermediate flow relationships between them, as well as inputs and outputs that cross the system boundary; this can be done with aggregate processes for simplified LCA, or iteratively developed into a more detailed model [dBG02]. Branching and looping in the flow chart that represents multifunctional or recycling processes should be accounted for either through system expansion or allocation [KG14]. Processes can be defined as either foreground processes that are specific to the system being studied, over which the producer or operator has influence, and background

processes that are not specific to the system and which are not influenced by the producer or operator [JI10]. Foreground and background classification of processes can help determine the appropriate type of data to use. Although the Handbook on LCA suggests avoiding generic datasets, they are often appropriate for background processes involving homogenous market commodities where specific data for a given producer may be unavailable [KG14].

Following the development of a satisfactory flow diagram, the process of data collection can begin. ISO standards require that the data collection process be documented, including referencing any public datasets, data collection times and techniques, and data quality markers [ISO06]. While the standard specifies that metadata such as geographic location, temporal coverage, and sources should be included in data quality requirements, it provides no specific format or nomenclature with which to address these requirements. As a result, several standards for metadata documentation and data transfer have been developed within LCA practice, such as the SPOLD format and the database-oriented SPINE format; ISO 14048 has further itemized appropriate metadata for LCA datasets [REF⁺04]. The data itself (known as computable data) quantifies the process or elementary flows, whether they be in materials, energy, transport, or economic value, while the metadata describes the actual object of the flow and how the computable data was collected or generated [CTSL98]. The data may be classified into types of inputs, such as energy, raw material, or ancillary inputs, outputs such as products and waste, or emissions to environmental compartments [dBG02]. Environmental compartments serve as categorizations for inputs, outputs, and emissions of the product system, although there may be complicated relationships between compartments as emissions travel from one to the other through evaporation, condensation, or leaching; the typical three compartments of air, water, and soil may also be split into more specific sub-compartments such as surface soil or root-zone soil [vZLLR14]. Boundaries may also be drawn between system inputs and emissions between the ecosphere and the technosphere, although these boundaries may be difficult to determine in an agricultural context [vZLLR14]. Data availability and the need for spatially specific data remain among the most severe problems in LCA [RRDB08b].

The data collected during the LCI phase must be validated and assessed for quality and uncertainty. The data should meet the data quality requirements set out during the Goal and Scope Definition phase of the study, and missing data or obvious anomalies should be substituted with justifiable values or alternative data [ISO06]. The Handbook on LCA suggests

both quantitative measures of data quality, such as precision and completeness, alongside qualitative metrics such as consistency, reproducibility, and representativeness [dBG02]. Uncertainties or dispersions in the data sets should be known, and may be quantified through additional metrics such as the NUSAP pedigree approach; these can be propagated in analysis using the Monte Carlo method, Latin Hypercube, fuzzy logic, or other means [HGH⁺14]. However, uncertainty remains a problem, particularly in comparative analysis where some uncertainty may be shared; the use of relative certainties for untraceable commodities and dependent sampling have been suggested to help reduce the uncertainty associated with averaged or point data [HHD⁺15].

Having validated the collected data, the LCI phase can conclude with the actual calculation of the inventory. This is usually done using LCA-specific software for matrix calculations, and scales all processes in the system such that their output is in terms of the functional unit [dBG02]. Software can also be used to determine stable values for infinite process loops that might exist within the system [JI10]. At this point, the study may proceed onto Life Cycle Impact Assessment (LCIA), or it may iterate back into data collection, should the system boundaries be refined or it is determined that more data is required [ISO06]. The scope of the study may require refining from the initial definition; this includes the system boundary and cut-off criteria, which may be iteratively re-determined until a satisfactory result is produced [JI10]. However, it is also pointed out that if data has already been collected that causes the cut-off to be re-evaluated, it may simply be best to keep the data anyways [RRDB08a]. A compromise would be using estimated data for less-relevant processes that might fall outside cut-off boundaries, and saving the time and expense of finding data for more relevant, foreground processes [JI10].

LCI databases play a key role in performing LCA research, providing the base datasets used to model common inputs to more complex product systems. However, the diverse ecosystem of LCI databases, including commercial, national, and regional databases, has led to the adoption of a variety of non-interchangeable data formats and implementations. This has proved to be a major problem in the LCA field [KDRJ16]. Initiatives such as the UNEP Global LCA Data Access Network aim to improve LCI database interoperability and access, allowing greater access and compatibility between LCI data sources [UNE17]. As previously mentioned, data gaps also pose a severe problem in the LCA field [RRDB08b].

2.3 Interoperability in LCI

Due to the lack of cohesive standards for both LCI methodology and LCI formatting, there can be issues of interoperability between datasets produced using different modelling criteria, or produced by different LCA software suites. In developing a new public LCI database resource, it becomes necessary to consider how to maximize the interoperability of the datasets provided so that they can be of maximum utility to the LCA community.

Interoperability between the various LCI databases and LCA software currently in use presents a major field of study in LCA. Interoperability issues in LCI data have been tied to the general problem of data availability, specifically that relevant datasets may exist, but cannot be used due to interoperability issues [IHT⁺15]. The issue of data availability has been considered the most pressing concern facing the LCA field [RRDB08b]. This issue has been identified since at least 1998, when SETAC noted that the technical systems for LCI data exchange required further development [Bre99]. Since then numerous new databases and exchange formats have been introduced, but interoperability remains a problem.

2.3.1 Existing LCI Data Formats

Currently, there are two major data exchange formats that are used by most life cycle inventory databases: the EcoSpold format, further split into two revisions; and the International Reference Life Cycle Data System (ILCD) format. Other older formats exist, such as SPINE, but uptake of these formats has been minimal hence they have not been further considered. Similarly, ISO/TS 14048 provides guidelines for LCI format compliance, but real-world use is limited. Some mappings for less common exchange formats, such as SPINE to ISO 14048 have been published [CEF⁺03].

The EcoSpold format is the successor to the older SPOLD97 and SPOLD99 data exchange formats developed by the Society for the Promotion of Life-cycle Development as a means of standardizing data exchange between databases [Cur04], in compliance with the ISO/TS 14048 specifications for LCI data documentation format [Kel07]. A revised version of the format, EcoSpold2, was developed and introduced, replacing EcoSpold1 as the format for the ecoinvent database in May 2013 [MMBV16].

Development of the ILCD began in 2005 under the European Commission Joint Research Center (JRC), and comprises a collection of technical guidance documents that cover the whole of the LCA process. The ILCD

2.3. Interoperability in LCI

similarly claims to be ISO 14048 compliant [WPC⁺12]. Development of the ILCD was driven by the need for greater consistency and quality in LCI data and the lack of a suitable format conducive to achieving these goals [WDK11]. Wolf et al. (2011) note that data formats such as SPINE and ISO/TS 14048 did not achieve widespread uptake. They also argue that EcoSpold, while popular, lacks unique identifiers, full multi-language support, and has limited documentation abilities for some LCI methods (2011). The project includes documentation for the naming and classification of flows [JI10], in addition to development, formatting, and validation tools provided through the JRC Life Cycle Data Network (LCDN) [JI18].

The EcoSpold and ILCD data exchange formats are both based on the Extensible Markup Language (XML) data encoding, and both have their own editors for creating datasets, the EcoEditor and ILCD Editor respectively. However, it has been noted that these editors can be difficult to use, increasing the time and expense of generating compliant datasets. While moving to a single dataset editing system would help alleviate the problems of interoperability between datasets and reduce the expense of creating datasets using existing editors, this would again require a consensus among the LCA community on using a singular data model and editing workflow; whether this is feasible remains to be seen. The underlying XML encoding was developed to support a wide array of data exchange tasks while being human-readable and easy to process [BPSM⁺08]. This shared data encoding has allowed for conversion between EcoSpold 1, EcoSpold 2, and ILCD data formats [Ope15]. However it has been noted that while much of the relevant data can be mapped between the EcoSpold and ILCD formats, some fields have no equivalent, and metadata may be lost between conversions. For example, the ILCD format has no equivalent fields for the EcoSpold2 mandatory ‘Type’ and ‘Special Type’ fields, so an ILCD-to-EcoSpold2 conversion would result in those fields being lost. This has been improved with EcoSpold 2 but remains a problem [MMBV16], presenting a persistent barrier to interoperability.

2.3.2 Other Interoperability Issues

Even if a singular LCI data exchange format could be adopted, or datasets could be seamlessly converted between formats, other interoperability concerns may prevent datasets from working correctly. In this context, nomenclature provides the references required to use the dataset effectively: names of flows, units of measurement, directionality of flow, impacted environmental compartments, and other key properties of the dataset [EIR⁺17].

Differences in nomenclature can cause interoperability issues, even between datasets with the same syntax. A simplified example of this would be two flows of identical format but differing nomenclature, one requiring an input of acetone, the other producing an output of propan-2-one, the IUPAC nomenclature for acetone. LCA software could not be relied upon to determine that acetone and propan-2-one are semantically identical, and as a result it would fail to link the two flows together.

2.4 Third Party Database Initiatives and Programs

Beyond individual LCI databases, there has been a trend towards the development of distributed LCI data networks and third-party LCI data providers. The consolidation of LCI data into networks and third-party repositories has the effect of ensuring some level of compatibility in format, while also addressing the problem of data availability that has been identified in the LCA field [RRDB08a]. This is being done both at the institutional level, through initiatives such as the Life Cycle Data Network (LCDN) developed by the JRC-EPLCA, and through private and commercial organizations such as OpenLCA and ecoinvent.

The LCDN service uses a node-based approach, with individual databases and providers maintaining their own locally hosted nodes using a J2EE servlet environment and MySQL database, coupled with the Soda4LCA package [FKL16]. Individual nodes form a network, with each node sharing an identical interface for searching and browsing the datasets within the node, and the LCDN maintaining a list of all published datasets and their respective nodes [FKL16]. At present, nodes have been registered for the Agri-footprint and GaBi commercial databases, in addition to the ELCD, Italian National LCI Database, and others. Minimum requirements for entry into the LCDN are ILCD syntax and nomenclature compliance, with plans to introduce data quality, documentation, and review validation requirements in the future [JI12]. While this cleanly solves the interoperability problem in terms of nomenclature and format, it moves the burden of ensuring interoperability from the LCA practitioner to the data provider. This shift could potentially limit the data uptake of the LCDN, as even with automated tools, extensive manual intervention, and thus expense, would be required to convert non-compliant datasets. If the expense is not justifiable, the data may simply be omitted from the database altogether [SLTC16]. However, this impact would be lessened on new LCI databases which have not yet

selected a format or nomenclature, and the availability of automated utilities for the hosting and validation of LCI data in the LCDN make it easily accessible for new databases.

A similar approach has been used in the development of the USDA LCA Commons. While previously the LCA Commons had operated as a centralized database, it has switched to a curated collaborative approach, with individual data providers maintaining their own dataset repositories that are then reviewed and released by the National Agriculture Laboratory (NAL) [LBG18]. This has been implemented using the LCA Collaboration Server developed by GreenDelta, which provides a repository system with access and version control features; access to this repository system is integrated into the OpenLCA software suite, allowing datasets to be committed to the repository directly from OpenLCA [BGC19]. This system has the benefit of enabling collaborative development of datasets by multiple LCA practitioners while maintaining a history of dataset changes and minimizing the risk of conflicting edits or accidental overwrites when working collaboratively [BGC19]. Minimum requirements for mandatory data and interoperability are maintained by NAL, which reviews datasets prior to publication through the LCA Commons website or via the LCA Commons Global LCA Data Network (GLAD) node. Datasets are distributed in both ILCD and JSON-LD formats.

Another approach to data consolidation has been the OpenLCA Nexus, a web application that allows the user to search a variety of both institutional and commercial databases and download databases in the proprietary ZOLCA file format for use in the OpenLCA software suite. Dataset formatting and interoperability is handled by GreenDelta, the consultancy behind the OpenLCA and OpenLCA Nexus projects, with a share of the licensing fees and maintenance fees covering the cost of file conversion [Ope18]. Datasets provided through the OpenLCA Nexus are mapped to OpenLCA reference flows, allowing for interoperability between different databases provided through OpenLCA Nexus [Ope18]. This includes a number of agri-food relevant databases such as ecoinvent, GaBi, Agri-footprint, and ESU World Food commercial databases, in addition to the free USDA, ELCD, and Agribalyse databases. The OpenLCA Nexus has the benefit of not burdening the data provider to ensure interoperability. It was noted during consultation with ecoinvent that they send EcoSpold2 files directly to OpenLCA, who handled the conversion to ZOLCA format and mapping of reference flows. However, the proprietary file format means that the datasets are only useful for users of the OpenLCA software suite, and interoperability may not carry over if datasets are exported and re-imported into another

software suite.

A third option for national and regional LCI databases is inclusion into existing databases, such as ecoinvent. This option was chosen for the Quebec LCI Database Project, which used ecoinvent to host regional data as a National Database Initiative. National database initiatives are responsible for data collection, while ecoinvent provides the infrastructure for validation and publishing of the data in the ecoinvent database. Certain advantages to using an existing database make this option ideal for smaller regional and national databases; it eliminates the costs associated with developing and implementing the technical infrastructure of a new database, and allows immediate interoperability with a large dataset through the rest of the ecoinvent database [LS16]. In addition, copyright and license remains with the original data provider, allowing for publishing publicly or through another third-party vendor [WBH⁺13]. Editing and submission of datasets in EcoSpold2 format is done through the aforementioned ecoEditor interface. While publishing data through the ecoinvent NDI initiative restricts usage to those practitioners who hold an ecoinvent license, the ability to self-publish data or publish through other third parties makes it possible to provide data to non-license holders. This is an important consideration for databases that have a mandate to make datasets freely and publicly available. Inclusion in the ecoinvent database also means propagation of the data through third party data providers such as OpenLCA Nexus, allowing for more widespread distribution and interoperability.

It is abundantly clear that third party data providers and networks play a major role in both the distribution of datasets, as well as the validation and standardization thereof. Many of the major LCA databases, including agri-food databases such as Agri-footprint, ESU World Food, and the USDA LCA Commons, are available through networks such as the LCDN and third party providers such as OpenLCA Nexus. In turn, major databases that have taken on regional LCI initiatives have also become parts of these networks, leading to a cascading consolidation of LCI data into centralized repositories. While this does not entirely solve the issue of data interoperability, particularly between different providers and networks, this does provide the opportunity for wider distribution of data and immediate access to interoperable datasets. This immediate access to additional data has been cited as a motivation for working with third party providers [LS16]. Depending on the format(s) chosen for an LCI database, third party data providers could offer a low-barrier means of solving the issue of data availability that is endemic to the LCA field.

In addition to currently available databases, networks, and third party

2.5. Summary

data providers, there are several global initiatives seeking to increase the interoperability and quality of LCI data, both through the introduction of new standards for metadata and new repositories for LCI data. These include the recently launched Global LCA Data Access Network (GLAD) developed through the United Nations Environmental Program, and the Global Guidance Principles for LCA Databases, also known as the Shonan Guidance Principles. Both were developed under the auspices of the United Nations Environmental Program (UNEP).

The GLAD initiative is a new network of LCI data providers that seeks not just to aggregate data, but also to easily convert between LCI data formats through the use of new metadata descriptors and a global mapping of elementary flow nomenclature [UNE17]. Much like the LCDN, the GLAD network consists of independently operated nodes and a central user interface; nodes are required to meet minimum interoperability requirements through the planned use of newly defined metadata descriptors [UNE17]. These specifications include required metadata fields, such as process name, type, time, and geography, as well as a new metadata descriptor structure; this new structure includes representation goals, actual representation, and a calculated conformance based on the distance between goal and representation [CAC⁺17]. Both ILCD and EcoSpold2 datasets can be used with GLAD once the necessary descriptors are identified; GLAD can also integrate with the LCA Collaboration Server and the Soda4LCA software, allowing datasets to be pushed directly from these applications [UNE17]. In addition, the GLAD network provides a documented RESTful API service for searching and indexing LCI data in the GLAD network [US17], increasing the utility of the GLAD network and facilitating the development of external management tools. The GLAD network is still currently in development, and only a subset of the proposed metadata descriptors are available in the public beta version of the network.

2.5 Summary

While the field of Life Cycle Assessment has grown considerably alongside ecological and economic concerns, differences in practice and between different data formats has resulted in the creation of several competing formats and nomenclatures among the LCA community. This poses a challenge for new LCI databases, as the choice of format and nomenclature can have serious impacts on the ability of users to successfully implement a dataset into their LCA model. While the two most common LCI data formats share

2.5. Summary

an underlying data exchange language (XML), they feature very different organization, hierarchy, and metadata fields. This results in potential data loss during conversion between formats, and necessitates the use of mappings both between formats, and between nomenclatures.

Chapter 3

Methodology

This research primarily focuses on the development of a national public database for agri-food LCI data, with the aim of allowing LCA practitioners to more accurately assess the ecological impacts and outcomes of the Canadian agri-food industry. Regional variation in agricultural techniques, technologies, and regulations must be accounted for. This makes globally aggregated data less useful while emphasizing the importance of national and regional data sources. The CALDC is the intended solution, providing primary source gate-to-gate datasets for Canadian agri-food processes, independent of, but compatible with, existing commercial and international LCI database solutions and LCA software.

With these objectives in mind, research was done to determine the best approach in developing the CALDC, the results of which have been summarized in Chapter 4. From this research, recommendations were developed for both the technical implementation and the interoperability components of the database, the results of which have been included in Chapter 6.

3.1 Initial Objectives

The CALDC was proposed as a publicly accessible database for Canadian agri-food life cycle assessment data. Prior to the beginning of research, some key objectives for the CALDC were determined, based on the needs of the PRISM lab as well as stakeholders such as the Canadian Roundtable for Sustainable Beef and the Department of Agriculture and Agri-Food Canada, who were consulted to ensure the success of the database:

- The database must have provisions to allow public access to published datasets
- There must be a system to allow datasets to be reviewed before publishing to ensure that datasets meet required standards of quality and data reporting

3.2. Technical Considerations

- The database must take interoperability into consideration and provide some means of producing datasets in both major formats and ideally handle nomenclature as well
- The database should follow industry standards within the LCI database community in terms of technical implementation

Based on these initial objectives, areas of study were determined based on a preliminary literature review undertaken in the development of the thesis methods proposal. During this review, more than 60 articles, technical reports, and guidelines were studied, dating from 2002 up until 2018. In addition, LCA professionals and existing databases were engaged for further information on the current state of LCI databases. Areas of inquiry included technical considerations in implementing an LCI database, and an emphasis on producing interoperable datasets that could be readily shared through the CALDC platform with other practitioners and reused with minimal effort. Further discussion of the methodology used to examine each of these areas of study is outlined in the subsequent subchapters.

3.2 Technical Considerations

From the outset, it was considered that in order for the database to fulfill the requirement of being open to the public, it would be necessary to integrate some form of public-facing web interface for the database. This was seen commonly across other LCI database implementations such as the ELCD, ecoInvent and GaBi. Due to the sparsity of information describing the system architectures behind these common LCI database implementations, it was decided that existing database owners and documentation would be solicited for information. The findings of this research would then be used to determine the appropriate approach to developing the CALDC.

In order to determine common technical implementations of LCI databases, several major LCI database providers and providers specific to the agri-food sector were contacted, including the USDA LCA Commons, ecoinvent, GaBi, and Agri-footprint databases. Both commercial and institutional database providers were consulted. The European Life Cycle Database (ELCD) was also solicited for information, however the ELCD project was officially discontinued in June, 2019. The WFLDB and the Global Feed LCA Institute (GFLI) were also contacted, but did not provide a response for this research. Responses to requests included documentation, diagrams, and conference calls with developers and technical staff to discuss their implementations.

3.2. Technical Considerations

Data requested, if available, included the chosen database software(s) in use, choice of web server and backend scripting language or framework, the formats used to store LCI datasets within the database, and general information on the workflow of information from contributors through to the underlying database. A typical web application stack is shown in Figure 1, demonstrating the major components that database providers were asked about. Queries also included both the management of front-end portals and search engines for databases, such as theecoinvent ecoQuery system, in addition to back-end management and storage of data. Where possible, additional documentation and discussion with technical staff was requested, although both commercial concerns and varying degrees of familiarity with the underlying technical systems prohibited data collection from some sources.

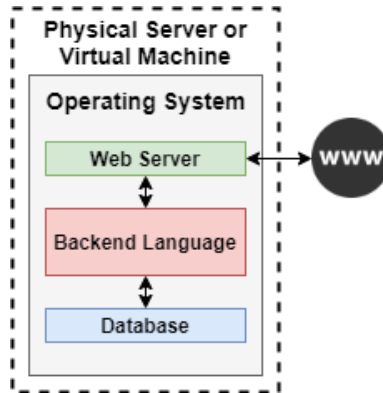


Figure 3.1: A traditional web application software stack.

In order to ensure that the developed database would follow current good practices in web application development, existing literature on database-driven web application development was also consulted. This included developer surveys on popular technologies, as well as documentation on current major software components such as Apache, PHP frameworks, and Flask. This information was used in conjunction with the data solicited from existing LCI databases to help determine the appropriate software, and architecture choices and ensure that the CALDC was developed using up-to-date and appropriate software choices.

3.3 Interoperability Requirements

In terms of developing the CALDC, interoperability was defined as the ability to seamlessly integrate datasets from multiple sources in order to develop cohesive system models while minimizing the need for manual editing. Interoperability further promotes the sharing and re-use of LCI data while potentially reducing the expenditure of time and resources on converting datasets or producing new background process LCI data sets. One of the primary goals of the CALDC is to provide Canadian LCI datasets at the process level that can be seamlessly integrated with background datasets from existing major databases and database providers with minimal effort in order to produce spatially and temporally relevant models for the Canadian agri-food industry.

Considerations for ensuring interoperability between LCI databases were first identified based on a preliminary review of available literature, including peer-reviewed journal articles, conference proceedings, technical reports produced by major institutions, and end-user documentation for existing databases and utilities. On this basis, dataset exchange formats, and nomenclature were identified as core concerns for creating and maintaining interoperability between LCI data sources. An in-depth review was subsequently performed of currently utilized LCI data exchange formats, including both published literature and technical documentation provided by format developers. Data exchange formats represent the primary means by which LCI data is recorded, represented and exchanged. Format origins, major users of each format, and the underlying data encodings were identified, as were the challenges in converting data between formats. The shared encoding systems that underlie each format were also evaluated, providing additional technical information and identifying priority areas for further development of data exchange formats.

In order to facilitate interoperability in format, a direct comparison of EcoSpold2 and ILCDD data formats was performed, based on the technical documentation available for each format. This comparison was done to allow for the development of a cross-compatible data entry or conversion tool. Individual data fields in each format were examined and compared with similar fields to determine potential mappings between similar fields. In addition, all mandatory and optional data fields were enumerated from each format to facilitate identifying the baseline requirements needed to produce a valid dataset.

In addition to format, reviews of previous studies on LCI database interoperability highlighted the importance of nomenclature. Nomenclature

refers to the actual naming of reference flows and units within LCI data sets, as well as classification schemas used to categorize and organize datasets. Even following a strict formatting rule, format conversion only allows for syntactic interoperability – the guarantee that the syntax of the formatted dataset will match the syntax of other datasets of the same format. However, it provides no such guarantee of semantic interoperability. The lack of a standardized nomenclature for LCA presents a greater challenge to interoperability than the lack of a standard data exchange format [Tiv15]. Failure to appropriately translate disparate nomenclatures between datasets can result in linkage failures between processes and flows. Major current nomenclature systems employed in LCI databases were identified, and potential solutions to the problem of disparate nomenclatures reviewed. This included a review of existing tools and mappings available between nomenclatures, such as the OpenLCA/GreenDelta Format Converter tool, as well as proposals for new methods of mapping and aggregating existing nomenclatures. The use of unified nomenclatures by existing LCI databases, such as OpenLCA, was also examined.

3.4 Third Party Providers & Initiatives

In considering interoperability and the technical aspects of implementing a new LCI database, the roles of third-party data providers, networks, and initiatives in distributing and standardizing LCI data were also considered. The emergence of international LCI database initiatives and distributors provides a new avenue for greater distribution and standardization of LCI data, although the benefits and costs of compliance with new standards must be considered. Information regarding third party providers, networks, and initiatives was sourced from published literature, technical reports, end-user data guidelines, and developer documentation. The requirements, reasoning, and impact of these systems and initiatives on the LCI field at the present time was evaluated.

3.5 Development & Testing

The development cycle of the project was managed using a Scrum software development process. Scrum is an agile development process suited to small development teams that uses very short, iterative development cycles known as sprints [RJ00]. The process focuses on producing usable software with each sprint, building the project incrementally towards com-

pletion [RJ00]. The advantages of this process are that prototypes can be quickly produced and feedback elicited at regular intervals, in addition to being able to more easily adapt to changes in requirements than a traditional waterfall model of software development.

The initial features required were identified from both background research, including features available in other major LCI databases, as well as the requirements for the project set out in the initial thesis description provided by Dr. Nathan Pelletier. Some features were considered too large to implement in a single iteration, such as the ability to export both ILCD and EcoSpold2 formatted datasets; these features were subsequently split up to allow for iterative testing of the tool with some implemented features while others remained under development.

As the actual software development was undertaken as a solo effort, not all aspects of agile Scrum process could be used. Traditionally, feature development in each iteration would be split across multiple developers, and each Scrum would be quite short – typically 2-4 weeks between iterations – allowing for new features to be rapidly implemented and then tested, before potentially adding new features or revisions to the queue for the next Scrum iteration. In addition, use of a project/feature management utility such as JIRA or Trello was determined to be unnecessary, as those tools are primarily designed to provide coordination when working with multiple developers. This led to a development approach similar to that seen in the code-and-fix software development model. Prototypes were commonly sent out to the PRISM lab and changes or bug reports solicited, with changelists noted in email announcements. The code-and-fix model is a commonly iterative development process used when project specifications are minimal and allows for rapid development by foregoing the need for documentation and testing, but this exposes projects to additional risk and makes assessing progress difficult [Kne18].

In order to minimize the risk associated with a code-and-fix software development model as well as elicit additional features for the project, emphasis was placed on the iterative testing and feedback cycle used in agile Scrum. Software testing is usually considered a major component of risk management in the development of new software. Three major potential testing groups were identified: internal testing among researchers within the Food Systems Priority Research for Integrated Sustainability Management lab; stakeholder testing, primarily amongst stakeholders who expressed interest in using and releasing data through the CALDC; and public testing, among a global group of life cycle assessment practitioners. Feedback was primarily obtained through email on an individual basis, with internal and

3.5. Development & Testing

external testers submitting bug reports and suggestions. While major features such as the ability to create datasets in both formats had already been determined, user feedback helped identify ambiguities in the user interface in addition to new features. The testing process included both functional and non-functional testing, such as the use of per-function/method unit testing, verification of output data format and content, and usability testing.

It was determined that initial iterations of the software would be distributed and tested internally, with feedback and features to be solicited from researchers familiar with both other LCI databases and with the intended purpose and feature set of the CALDC. Researchers at the PRISM lab were expected to be among the end-users of the software, making their feedback particularly valuable. In addition, internal testing would allow for tighter turnaround times on feedback, as well as the ability to directly demonstrate issues found in the testing. It was decided that initial development would focus on implementation of primary features, such as the ability to create and export a basic dataset, with functionality taking precedence over usability at this stage of development. Once a minimum set of necessary features was developed through internal iteration, a functional software product could then be distributed for stakeholder testing.

Improving usability was a primary focus of the subsequent stakeholder testing process. The external Expert Stakeholder Advisory Committee for the CALDC project included government, academic, and commercial stakeholders, including the department of Agriculture and Agri-Food Canada, the Canadian Roundtable for Sustainable Beef, Group Agéco, and Egg Farmers of Canada. Prior to the beginning of stakeholder testing, materials were shared with the stakeholders to keep them abreast of development, including the thesis methods proposal initially developed at the beginning of the project, and the subsequent journal article that outlined recommendations for the development of the CALDC. While internal testing was done with researchers intimately familiar with both the life cycle analysis process and similar software, the project stakeholders represented a range of familiarity and understanding of LCA practices and terminology. Because of this, testing with external stakeholders focused on improvements to the user interface, and ensuring that users less familiar with LCA processes would be capable of comfortably using the software when complete.

Once the software was thoroughly tested both internally and by the stakeholders, and all necessary features had been identified, implemented, and tested, the software would then be ready for a public announcement and release. It was determined that the ability to give feedback should be built directly into the application, something which has been identified as a

viable means of collecting user feedback [Kne18]. In particular, it has been shown that it must be easy for users to provide feedback, as otherwise many users will simply decline to provide any feedback [Kne18]. This was not considered a problem for earlier stages of testing, as communication via in-person meetings, conference calls, or email could be easily maintained within both the PRISM lab and among the external stakeholders committee, both of whom had a direct interest in providing feedback for the development of the software. In terms of actual release, it was decided that the global life cycle assessment listserve mailing list would be used to announce and disseminate the completed CALDC project.

3.6 Summary

In summary, a thorough review of published literature, technical documentation, and existing LCI data formats and nomenclatures was performed, both to guide the development of the CALDC as well as to provide recommendations for LCI databases moving forward. This included analysis of the technical implementations of LCI databases, identifying major interoperability challenges and analysis of existing data formats and nomenclatures, and consideration of the role of third party initiatives and databases in terms of operability and implementation, and considerations for future maintenance. Besides existing literature, LCI databases were solicited directly for information regarding their own systems to help better inform the research on the current state of LCI database development.

An agile, iterative development cycle was used to continuously implement and test features as they were identified. This testing would be performed internally to identify major features necessary for the success of the project, before moving into testing with external stakeholders to further improve the usability of the tool while continuing to add secondary features. The iterative development cycle and regular testing placed an emphasis on producing working software prototypes that could then be incrementally refined with further feedback. Once all major features were identified, implemented, and refined through the testing process, public release would then be possible, with the ability to further collect feedback during the public release phase and thus continue incremental improvements to the CALDC software.

Chapter 4

Implementation

Based on the requirements and the results of the background research, development of the CALDC web application began, in collaboration with other team members from the PRISM lab who provided testing feedback, as well as external stakeholders. The finished application includes a fully-fledged user login and account system. It further provides the ability to create, view, edit, import, and export LCI datasets, and submit these LCI datasets for acceptance into the CALDC, at which point they may be published for public use.

4.1 Technical Findings

4.1.1 Existing LCI Databases

Based on the responses solicited from existing LCI database providers, it was determined that a number of popular enterprise web software products were in use among them, and that the software and architecture chosen differed greatly between different databases. In terms of processing and serving of data, a variety of languages, frameworks, and content management systems are employed. Choice of database management software is similarly varied, although all are using relational databases based on Structured Query Language (SQL). Agri-footprint makes use of the PostgreSQL open-source DBMS software, while ecoinvent uses the commercially licensed Microsoft SQL Server software. In surveys of both professional and non-professional software developers, MySQL, MSSQL, and PostgreSQL represent the top three most popular DBMS packages in use; in the case of MySQL and MSSQL, this popularity spans multiple years [Exc18]. The popularity of these systems ensures long-term support will exist, and their choice suggests that LCI database providers are following the prevailing trends in database and web application development.

In terms of how LCI datasets were actually stored within the database, a more varied approach was seen among existing databases. Datasets are stored primarily as JSON-LD formatted files in the LCA Commons database,

4.1. Technical Findings

with datasets available in both ILCD and JSON-LD formats. In theecoinvent database, XML data are stored in the relational tables as Binary Large Objects and then converted back into EcoSpold2 format by the web-services layer for use in the ecoEditor. A copy of the database is prepared after each release of the ecoinvent database with additional EcoSpold2 data; this is then translated into readable HTML and served to users via the ecoQuery system.

It is also important to note the maturity and management of LCI databases, and how this has impacted the choice of architecture and software. The ecoinvent database represents a mature LCI database, beginning in 2000 as a joint effort between Swiss research offices and the Swiss federal government [FR05]. Maintenance of the database and web services is not handled by ecoinvent internally, but rather by IFU Hamburg GmbH, a German software consultancy firm that specializes in sustainability software. This could explain the use of IIS and MSSQL in the ecoinvent technology stack; both are popular in enterprise-level web development. In contrast, some databases have moved away from traditional web and database technologies, and instead have adopted LCI-specific systems for the management of datasets. An example of this is the USDA LCA Commons, which has moved away from a custom-built Drupal, Apache, and DKAN software stack to a system based on the GreenDelta LCA Collaboration Server.

Given the variety of different software stacks in use in existing databases, it was determined that there was only minimal consensus on technical implementation. While all the databases who responded used some form of SQL-based database, choice varied between databases in terms of particular DBMS software choice, web server software, and scripting languages. The use of common web application technologies such as Apache and Drupal were also noted, in addition to backend scripting languages such as Python (in use for data management at ecoinvent) and Grails (formerly in use at the USDA LCA prior to the move to LCA Collaboration server). The results of these exchanges suggest that LCI database providers are using traditional enterprise server and database technologies, with some changes in software choice and architecture coming with increased maturity and growth of the database, in addition to the development of specialized LCI applications. The lack of consensus among LCI database technical implementations, and the variety of enterprise web technologies in use, suggested that a standard web application software stack would be an appropriate choice for the development of the CALDC.

A summary of the formats, nomenclatures, and underlying technical implementations of three major LCI databases has been provided below in

4.1. Technical Findings

Table 4.1. It is also worth noting that most of the databases are related to one or more third-party providers, such as via OpenLCA Nexus or via SimaPro.

Table 4.1: Three major LCI databases compared.

	ecoInvent		Agri-Footprint		USDA LCA Commons
Output Format	ecoSpold2 XML		OpenLCA ZOLCA format, SimaPro		ILCD XML, JSON-LD
Input Format	ecoSpold2 via ecoEditor		Variable		Variable, via OpenLCA software suite
Nomenclature	ecoInvent, variable versions available through third parties		ecoInvent 2.2		Combination of ILCD and OpenLCA nomenclatures
Underlying Technical Implementation	Centralized MSSQL database handling eco-Query and datasets, maintained by third party		Provided via third-party providers (SimaPro, OpenLCA Nexus)		Central database hosting multiple repositories using the OpenLCA Collaboration Server

4.1.2 Formats & Nomenclatures

From the initial background research, it became clear that the most common and recognized formats in use among major LCI databases presently are the EcoSpold2 format, and the ILCD format. EcoSpold is in use with the eponymous ecoinvent database [FR05], the National Renewable Energy Laboratory (NREL) US LCI Database [LILS04], and the French Agribalyse database [CAM⁺15], among others. ILCD in comparison is in use with the Life Cycle Data Network (LCDN), which has consolidated data in the ILCD format from both Italian and Brazilian national databases, in addition to the commercial Agri-footprint and Thinkstep AG GaBi databases. The ILCD format has also been adopted by the World Food LCA Database (WFLDB), using the ecoinvent 3.0 naming conventions [NRL⁺14].

4.1. Technical Findings

Based on the popularity of these two formats, it was considered mandatory that the CALDC should support both, such that LCA practitioners could freely use CALDC-distributed datasets in their models regardless of which software and other LCI data sources were in use. A study of the format documentation for each format revealed that although they both use the same underlying XML format, the data field formats for both are different. As some data fields only exist in one format or the other, it was determined that a conversion between the two formats could lead to data loss, or result in empty data fields where the necessary data simply did not exist in the origin format. As a result, a study was conducted by researchers of the PRISM Lab at the University of British Columbia, Okanagan Campus to develop a suitable data reporting template capable of producing both ILCD and EcoSpold2-compliant datasets [TSAP20].

Turner et al. identified 65 distinct fields that would be required to create a dataset that would be compatible with both the ILCD Process format, and EcoSpold2 Activities. They further found that a number of fields unique to each format could be appropriately mapped to fields in the other format, reducing the duplication of effort by having the software perform those mappings automatically [TSAP20]. This eliminates the need to keep duplicate fields from each format, reducing the total amount of data the user is required to enter to produce a compatible dataset.

In addition to the fields identified in the ILCD Processes, the ILCD format documentation identified another nine mandatory fields for Contact datasets, seven mandatory fields for Flow Property datasets, thirteen mandatory fields for Flow datasets, and seven mandatory fields for Source datasets [TSAP20]. It was decided to use only the mandatory fields across ILCD and EcoSpold2, ensuring that datasets are compliant with both formats while requiring only the minimum amount of data entry. An exception was made for eight non-mandatory ILCD fields, including the synonym, Chemical Abstracts Service (CAS) number, and formula fields. These fields provide alternative identification for their datasets, i.e. the CAS number of a chemical flow, or common synonyms for a particular process. These fields are optional, but may contain additional useful data for identifying a dataset that may otherwise use an unconventional naming schema.

Beyond providing datasets in both formats, the literature referenced noted that nomenclature also plays a major role in interoperability between datasets. To this end, the decision was made to use an established nomenclature as the basis for the CALDC, to prevent further fragmentation of LCI data standards and ensure compatibility with existing datasets. The elementary reference flows and units provided by the ELCD were chosen as

4.1. Technical Findings

the basis for nomenclature within the web application, as the datasets are freely available and can be redistributed without concern for licensing or infringement of intellectual property. However, this does not directly resolve the issue of incompatibility between EcoSpold2 and ILCD nomenclatures.

Although a mapping could be used to identify similar data fields within the ILCD and EcoSpold2 formats, this did not necessarily mean that both fields used similar formats or values for their inputs. In this respect, secondary mappings were required to convert between ILCD nomenclature, which was used by default, and EcoSpold2. For example, it was determined that the geographical location categories used by ILCD differed slightly from those used in EcoSpold2; this required a secondary mapping to be developed to convert ILCD location values and UUIDs into EcoSpold2 compatible location values and UUIDs. A similar process is required for the mapping of units between ILCD and EcoSpold2. The mapping produced by OpenLCA/GreenDelta as part of their Format Converter tool [Ope15] was found to be the most complete ILCD-EcoSpold2 mapping available, including complete or near-complete maps of locations and units across the two formats. However, the elementary reference flow mappings included with the tool were found to be incomplete, encompassing only 2,802 of the 41,675 ILCD elementary reference flows. This allows the use of a subset of ILCD-standardized elementary flows, which can then be converted to appropriate references for EcoSpold2-formatted elementary flows.

Ultimately, no clear solution to this problem was found. Use of the Format Converter mappings necessarily limits the user's selection of flows to those that have been properly mapped if fully compatible EcoSpold2 datasets are desired. At this time, no other compatible flow mappings between ILCD and EcoSpold2 were found to be available. Several potential approaches to this problem have been set forth. New structured ontologies for LCA nomenclature have been suggested [IHT⁺15], however the adoption of a single standard nomenclature model is considered a long-term goal [Tiv15]. Alternative approaches such as semantically-linked nomenclature catalogues or dictionaries have been proposed; Kuczenski et al. have developed a semantic catalogue based on major LCI databases, including the ELCD,ecoinvent, the US LCI database, and the GaBi 2016 database [KDRJ16]. While a new ontology may eventually develop, or a semantic catalogue may be completed in the future, it was determined that the existing partial mapping of ILCD to EcoInvent2 reference flows was the most feasible to implement at this time.

4.2 Software & Architecture Choice

4.2.1 Web Application

The primary means of interacting with and using the CALDC is through a public secure web application. The CALDC application allows users to search and browse currently available datasets, view their associated metadata, and download them for use in a manner similar to the system currently used for the European Life Cycle Database (ELCD) [JE17]. In addition, the application also allows users to submit new datasets for addition in the CALDC, pending a curation process that will verify the completeness of the submitted data. This is a manual task and subject of a concurrent study as part of the broader research project.

Initially, the web application was also going to be the user interface for a conversion utility, allowing the user to select a preferred data exchange format and potentially nomenclature when downloading a dataset. When it was determined that this conversion would potentially be very difficult to perform without data loss, it was determined that the web application would instead provide the ability to create new datasets directly in both EcoSpold2 and ILCD formats. Emphasis was placed on creating easily navigable web-pages that clearly display relevant metadata and background information for datasets, allowing users to quickly find and identify appropriate data. The requirements for the UX design (derived from Human-Computer Interaction principles) also take into account the W3C Web Content Accessibility Guidelines, insofar as the CALDC is a public web service and should provide at least some support for screen readers and similar systems [W3C08].

The decision to integrate a new dataset creation tool into the CALDC was made both to avoid the aforementioned data loss during the conversion process, as well as to provide a more user-friendly interface than was available through existing LCI data creation tools. This was of concern because many stakeholders in the agri-food industry may not necessarily be familiar with LCI formats and nomenclature, and LCA practitioners may not be familiar with all of the required metadata fields in an LCI dataset. Thus, it was considered important that the tool be easy to use; not necessarily for the layman, but for LCA practitioners and those at least familiar with LCA terminology and practices who were considered our user group.

Based on the software currently in use in the LCI database field, it was determined that many LCI databases were using a variety of conventional enterprise web application technologies. This suggests that the details of technical implementation were not necessarily important in terms

of interoperability, allowing the use of a typical web application stack such as PHP/Laravel or Python/Flask over a standard Apache or NGINX web server. Subsequently, the decision was made to use a conventional web application architecture. The initial choice of language was PHP, using the Laravel framework, which had previously been used in the development of the Digital Archives Database Project (DADP), a similarly database-driven web application developed under Dr. Ramon Lawrence at the University of British Columbia, Okanagan Campus [MSO17]. However, the decision was ultimately made to go with a Python environment using the Flask framework. Flask was initially released in 2010 and has gained popularity as a lightweight and easily portable framework. Flask development techniques have been well documented in published literature [GM18].

In terms of advantages, Flask required fewer dependencies and provided a clearer organizational structure than the Laravel framework, while also providing a built-in webserver that facilitated testing prior to the actual launch of the web application. Flask also provides the benefit of being Python-based, making the code base more maintainable through the use of a common, high-level programming language with long-term support. In addition, Flask provides a number of built-in security functions, such as the ability to store user-specific session data using encrypted client-side cookies. In developing the web application, functionality was split across multiple Python modules imported into the Flask web routing application, allowing for greater organization and maintainability of the code base through modularization. In addition, documentation of function methods and purposes was included within the code, and the code itself makes use of descriptive variable and function names throughout to make familiarization by future maintainers easier.

The Flask application was hosted using the Apache web server, a popular freeware web server that has the highest market share (>25%) among modern web servers [Ref20]. The web server itself is run on CentOS, a community-maintained variant of the popular RedHat enterprise Linux distribution. These software choices are typical of modern web application software stacks and are based primarily on the reliability and longevity of the software, as both Apache and CentOS offer long-term support for their software.

4.2.2 Database

Similar to the choice of web server and backend software, existing LCI databases were found to use a variety of different SQL-oriented databases. In

addition, there was no consensus on how data was stored within the database itself. As a result, the choice of the database was only limited by the decision to use an SQL-oriented database, and was primarily informed by enterprise choices and cost. As Python supports database drivers for many different databases, it did not impose any limitations on the database choice. The use of parameterized queries within Python’s SQLite/MySQL drivers prevents SQL injection into the database by automatically escaping parameter inputs. The initial client-side implementation of the CALDC used a combination of storing data in complete XML-encoded files, while maintaining a listing of these completed datasets in a SQLite database. SQLite is a lightweight, serverless database with native support in Python, designed for embedded client-side usage in web and mobile applications [Owe06]. While this allowed for quick development of the software, limitations of the SQLite engine such as lack of concurrent operations and slow read/write times when handling large tables required that an alternative database be used.

MariaDB, a community-developed fork of the popular MySQL database was chosen as a replacement. It enjoys broad commercial support among database providers while also being a free and open-source database, removing the need for a licensing fee. To further speed up the application, datasets were moved from standalone XML files to dedicated tables, removing the need to parse XML files to retrieve metadata attributes when displaying this information through the web application.

4.2.3 Stack Architecture

The CALDC uses a traditional web application stack architecture, based on an Apache web server running on the CentOS operating system. On top of this, the web application, known as SimPLCItY, runs as a service, connected to the underlying Apache server using a Web Server Gateway Interface (WSGI), which facilitates the passing of information between Apache and the Python application. It is served via Secure Hyper-Text Transfer Protocol (HTTPS) using a TLS certificate. For storing of datasets and user data, the MariaDB SQL database is used. MariaDB can be interfaced directly using a standard mysql-connector module in Python, allowing the application to communicate directly with the database. A diagram representing this architecture is provided in Figure 4.1 below.

The application is run on a Digital Ocean “droplet” virtual machine, which provides four virtual CPUs, 8GB of RAM, and 160GB of solid-state storage. Initial deployment was done on a smaller virtual machine, but was resized due to issues with poor database performance on lower-powered

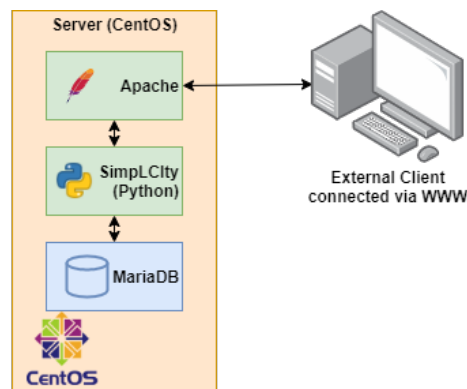


Figure 4.1: Software stack used in the development of the CALDC.

machines. The caldc.ca domain name was acquired and serves as the primary domain for the application.

4.2.4 Database Design

Although initial versions of the SimplCity application used an SQLite database and stored datasets as files, performance was found to decrease heavily when the application required loading of a large number of datasets. An example of this can be seen when selecting a flow for a Process input/output exchange; when using ILCD elementary reference flows, such a query may return several thousand potential flow datasets for a given environmental compartment. This was found to be too much load for the application, due to the requirement that it parse data out of the XML-formatted files for each possible flow. The decision was hence made to switch to MariaDB, a community-developed fork of the MySQL database, and move towards the storage of data directly in the database. The database itself consists of twenty-two tables, organized around six major purposes. Figure 4.2 shows the major organization of the subsequent database:

User-created datasets are stored directly in the database in the appropriate table, organized by ILCD dataset type. Each table dedicated to a type of user-created dataset includes as a column each of the previously discussed mandatory data fields, as well as any additional optional data fields such as synonyms. Each row within these tables represents a dataset that has been created of that type, which is owned by a specific user. These tables

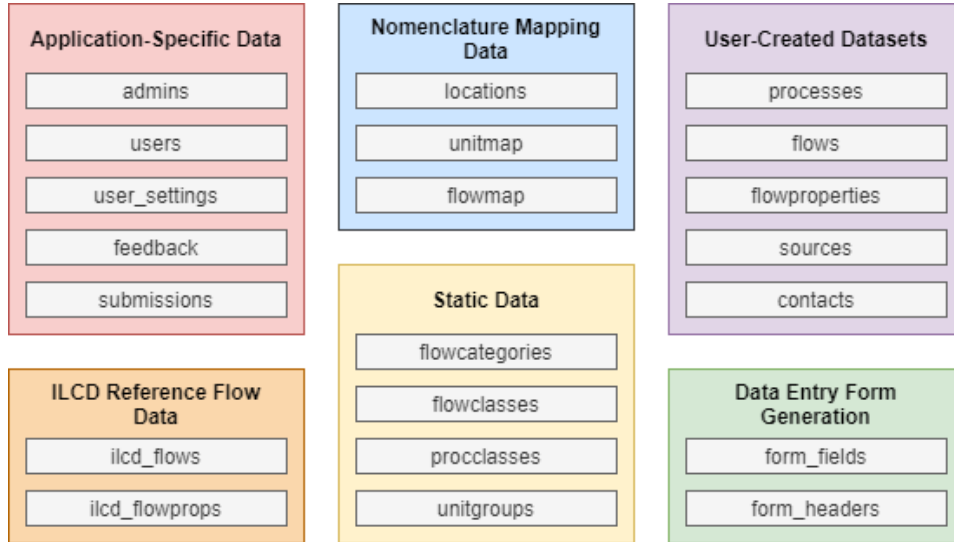


Figure 4.2: Classification of tables in the CALDC database by purpose.

are pulled when rendering a user-created dataset in both SimpLCity and the public-facing CALDC service, as well as for final dataset export. Each dataset also contains a reference to the specific user in the ‘users’ table that created the dataset; this is internally referenced when an attempt to read, edit, or export a dataset is made to ensure that a logged-in user can only perform such actions on their own datasets.

The internal column names used to denote each field in a given user-created dataset table are stored in the form_fields table, which is used to handle dynamic data entry form generation. This table maps internal column names to the proper external names used in the ILCD and EcoSpold2 format documentation, while also applying a step, type, and maximum size to each attribute. The absolute maximum storage size of each type of dataset, as well as the average size as of July 2020 is provided in Table 4.2 below, as reported by the MariaDB database.

Maximum lengths greatly exceed average lengths for all dataset types, in particular as a 65,535B text-type column is used to store the flow property and flow exchange tables. As there is no prescribed maximum length within the specifications for the number of flows or flow properties that can be listed in these fields, they were provided additional length to accommodate large exchange/property tables if necessary. This allows forms to be

Table 4.2: Average and Maximum Sizes of Datasets by Type.

Type	Average Size (B)	Absolute Maximum Size (B)
Contact	372	6380
Source	682	7740
Flow property	1092	5660
Flow	1195	73945
Process	8366	126495

built on a step-by-step basis, with each “step” constituting a number of related data fields that will be prompted from the user on the same page. Specifying a type for each attribute allows for automatic generation in the backend (Flask) of appropriate HTML form fields; for example, a numeric field will be rendered as an HTML5 number-type input, while a dropdown selection will be rendered with the appropriate options for that dropdown. Additional fields denote whether a field is required (and apply an HTML required keyword to the generated form input), as well as allowing maximum lengths and default options to be specified. The `form_headers` table simply provides appropriate titles for each generated data entry step page (i.e. Step 1: Modelling and Validation, Step 2: Exchanges, etc).

The static data and reference flow data tables act as repositories for pre-defined data. This includes data for the ILCD elementary reference flows, and the standard ILCD reference flow properties. The `ilcd_flows` table represents the largest table within the database, consisting of 41675 rows, each representing one of the standard ILCD elementary reference flows. This data was previously stored in file format on the web server and referenced in the database, but was transitioned into the database due to performance concerns. Additional ILCD reference datasets, such as default contacts and sources are stored outside the database in XML file format, as they do not require regular parsing and are merely included with exported datasets as a necessary dependency. Similarly, the static data tables store the classification and categorization groups for flows and classes, as well as the default ILCD unit groups. Additional static data and/or reference flows may be added as necessary to these tables should the underlying ILCD standards change.

The nomenclature mapping tables, `location`, `unitmap`, and `flowmap`, contain mappings between ILCD and EcoSpold2 nomenclature. They represent the in-database version of the mappings laid out in the Excel files that come with the OpenLCA Format Converter [Ope15]. These tables are primarily

used during the export process, during which the original ILCD units, location, and flow(s) are converted to their equivalent EcoSpold2 mapping for export as a .spold file. One exception to this is the flowmap table, which is used during dataset creation to selectively show only the mapped subset of ILCD flows. This feature can be turned off in the user settings, at which point the flowmap table is only referenced during export.

Finally, a group of application-specific tables handle web application data unrelated to LCI datasets. This includes user and admin login information and site settings, as well as feedback provided through the built-in feedback system. The submissions table is responsible for the management of datasets submitted for public release to the CALDC. It does not directly include the submitted data set, but instead a reference to the dataset's unique UUID and the type of dataset it represents (Contact, Process, etc). While most of these tables are used primarily to handle user logins and permissions, the submissions table is also used in the admin console for the application, as well the public-facing CALDC.

4.3 Security & Maintenance Considerations

As the CALDC was designed as a long-term web application, security and maintenance also needed to be considered. Since a full-stack software architecture was used, security has been implemented across multiple levels of the software stack. In addition, a basic risk assessment has been provided in Appendix A, based on the Open Web Application Security Project (OWASP) Top Ten web application security risks awareness document[OWA17].

The underlying CentOS 8 server uses the iptables Linux packet filter firewall by default. This provides only basic filtering ability, but allows all unnecessary ports to be closed on the server machine, leaving only Secure Shell (SSH), Secure File Transfer Protocol (SFTP), HTTP, and HTTPS protocols enabled on their respective ports. In addition, CentOS comes preconfigured with the Security-Enhanced Linux module (SELinux), which provides access control policy and audit logging for processes[MMC06]. SELinux access control prevents the underlying Apache web server process from accessing or modifying files outside the scope of the web application folder, and exists on top of the existing Linux user/group ownership and permissions system. Communication with the web server is done via Hypertext Transfer Protocol Secure (HTTPS), which uses a third-party issued TLS certificate to verify ownership of the web server and encrypt user traffic to and from the server.

4.3. Security & Maintenance Considerations

All standard non-secured HTTP traffic is redirected to use HTTPS.

The MariaDB database is run internally on the CentOS server, and has no external ports. An application-specific user role was created in the database that allows access to read and modify data where necessary, while removing unneeded permissions such as the ability to modify static tables, modify the overall database architecture, or perform administrative functions. All database connections made by the web application use this role, reducing potential for abuse through SQL injection. As the primary purpose of the CALDC is to provide publicly-available datasets it was considered unnecessary to encrypt the contents of the database. However, passwords are stored in the form of a salted SHA-256 hash. This prevents the use of common precomputed lookup or “rainbow” tables to reverse the hashes should the database ever be compromised, requiring the attacker to generate new lookup tables specific to this application. While the system prevents passwords from being stored in plaintext, the SHA-256 standard is now considered deprecated, and should be replaced with an appropriate key-stretching function such as PBKDF2 or bcrypt to provide additional protection against attacks as recommended by the IETF [Mor17].

The Flask and Python back-end of the application includes a number of built-in security features to protect against such attacks, in addition to common attacks such as Cross-Site Scripting (XSS). The Jinja templating engine provides automatic escaping of data passed between the web application and HTML templates, preventing cross-site scripting (XSS) attacks based on malicious user input. In addition, prepared SQL statements are used throughout the application to prevent SQL injection attacks when user input is handled.

In terms of data validation, requirements are low. All fields except for auto-increment identifiers are cast to strings and stored as text, and most fields allow for free-form text entry with minimal formatting requirements. Where an input must follow a specific format or data type, client-side validation is performed using the built-in form validation in HTML5. If client-side validation is disabled or fails, internal try-except catches will return the user to the previous editing page with an error message and prompt them to retry data entry. This system only provides basic syntactic validation of input data. Validation of datasets for completeness, quality, and suitability for the CALDC is a manual task performed by CALDC administrators (see section 5.1). The requirements for manual validation are the subject of additional study within the PRISM lab and fall outside the scope of this thesis.

In terms of maintenance, the application was designed to require minimal

4.3. Security & Maintenance Considerations

day-to-day maintenance beyond CALDC administrators needing to approve or reject datasets in the CALDC submission queue. A systems administrator with shell access is required for periodic updating of software packages, and also if manual access to the database is necessary, such as for a password reset. The application was designed to allow changes to the underlying data templates and input fields without requiring additional changes to the application code. New fields may be added directly via the `form_fields` table and subsequently added to the output XML template file. Currently shell access is only available to the developer; however moving forward additional shell access will be passed off to the PRISM lab should maintenance or access be required in the future. While requiring minimal maintenance, the application still requires an ongoing systems administrator.

One major security feature not yet implemented in the CALDC is the ability for users to recover their password without requiring manual intervention by the system administrator. This is due to the underlying server not being configured for the sending of secure email. Mail would be sent via unencrypted Simple Mail Transfer Protocol (SMTP) in plaintext, potentially allowing an eavesdropper to see the email in-flight and steal login details or reset links. This could be remedied through the use of a secure external mail provider, or by configuring the server for Simple Mail Transfer Protocol Secure (SMTPS), using the existing TLS certificate used for HTTPS.

To protect against catastrophic failure or intrusion, a basic backup system is in place. Complete images of the virtual server, including the operating system configuration, database, and web application software, are taken on a daily basis. These images are stored by DigitalOcean and are performed externally to the machine. Access to the backups requires access to the DigitalOcean account responsible for hosting the CALDC web application. Should the virtual machine be unrecoverable due to failure or compromise, the backup images can also be used to create a new virtual machine, although some post-configuration of DNS records and TLS certificate would be required. Images are kept for a period of thirty days after they are taken, allowing the server to be restored to any day in the past month. A weakness in this backup system is the reliance on DigitalOcean; should they suffer catastrophic data loss it is possible backup images would become unavailable. Offline storage of a cold backup on a weekly or monthly basis for longer periods would provide additional recoverability in such an event and could be achieved with a simple automated script if necessary. OWASP states that average time to discover a breach in a web server was 191 days, suggesting that the current 30-day backup window is too short to adequately

protect against a malicious attack of a clandestine nature[OWA17].

4.4 Development Timeline

After the major requirements of the CALDC were established and background research was completed, development of the CALDC application began in 2018, culminating in the initial prototype release in December 2018. This prototype used the Flask framework’s built-in web server to locally host the web interface for the software on the user’s machine, while using SQLite as a local database. This version of the software was distributed internally within the PRISM lab for testing and feedback, although at this stage many features had not yet been implemented, including ecospold support. Several drawbacks to the client-side approach were found: it necessitated an updating service of some sort to keep the software updated as changes rolled out, took up large amounts of space due to the packaged WinPython installation, and would still require a separate web interface to facilitate sharing and publishing of LCI data. The client-side version of the software continued with development until the release of version 1.5, with version 1.6 switching to a traditional web application without the need for a separate application on the user’s machine.

Initial feedback on the user interface and the ease-of-use of the software was taken from internal testing in the PRISM lab, with subsequent versions of the client-side software provided to select stakeholders for further testing. In addition, version 1.3 of the client-side only software was presented at the 2019 HOLOS Conference on Sustainability of Canadian Agriculture and feedback was solicited from those in attendance, which included users with varying levels of familiarity with Life Cycle Assessment data. Feedback from these sessions was then implemented in subsequent client-only versions (1.4 and 1.5), in addition to the web-only version of the application, which was released in September of 2019 as the 1.0 beta version.

The beta 1.0 version included additional features requested in the feedback, including the ability to import datasets, the ability to export ecoSpold2 formatted datasets, a revamping of the support documentation, and adjustments to the user interface to make it more user-friendly. The online version of the software was subsequently released for public testing in October of 2019, and included an online feedback form to solicit additional feedback. This version was provided to stakeholders for the creation of new datasets, as well as advertised on Life Cycle Assessment listserves to get additional exposure for the software, in addition to feedback. Throughout this pe-

4.4. *Development Timeline*

riod additional features and modifications were added, and bug testing was performed in conjunction with internal testing in the lab and by external stakeholders. The final major patch was released in March of 2020, including a number of small fixes and features, in preparation for a training presentation at the 2020 Conference on Sustainability of Canadian Agriculture that month [Aru20].

Chapter 5

Features

5.1 Application Workflow

The CALDC website is primarily split into two major interfaces: The public-facing CALDC database, and the private SimpLCItY application. Datasets may be developed privately within the SimpLCItY application, and then submitted to the CALDC for public release. Once approved, these datasets then become available to all users on the public-facing database, allowing datasets to be browsed, downloaded, and used in life cycle assessments by the general public. While the two interfaces are linked to allow created datasets to be submitted, they serve separate purposes within the CALDC.

When a user visits the CALDC website (<https://caldc.ca>), they first encounter a splash page that provides portals to both the public and private interfaces. The user may choose to log in or register a new account to access the SimpLCItY application, or may choose to navigate to the public CALDC database. Access to the public database does not require that the user log in or have an account. The public database shares the same basic user interface as the SimpLCItY application, and allows the user to browse, search, preview, and download public datasets. This interface is discussed in subsections 5.2.15 through 5.2.17.

If the user intends to create or edit an LCI dataset, they can instead log in or register a new account on the splash page. Account registration is free of charge and requires only a unique user email address, and a user-provided password. Logging in will redirect the user to the SimpLCItY application home page. Within the SimpLCItY application, entered datasets are kept privately; users only have access to those datasets that they themselves have created, or are part of the standard reference datasets i.e. the ILCD elementary reference flows. Within the SimpLCItY application, users may create new datasets, edit or import existing datasets, browse or view existing datasets, and export completed datasets. This is done primarily through a unified, step-wise editing system that provides the same interface for creating datasets as well as for editing an existing or imported dataset. Use of

5.1. Application Workflow

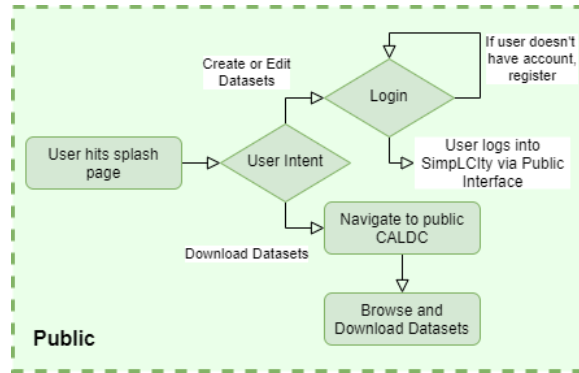


Figure 5.1: Public-facing CALDC website user flow.

the editing system is described in subsection 5.2.8. Once the editing or creation process has been completed the user receives confirmation that their change(s) have been made, and may then return to the home page to begin the workflow process again. A summary of the major features and their permission requirements is provided in Table 5.1 below.

It should be noted that the Administrator role within the CALDC is not analogous to a system or software administrator. The Administrator role is identical to that of a normal user, but an Administrator may also approve or reject datasets (see subsection 5.2.14), and grant the Administrator permission to other users.

The user's logged-in status is maintained even if the user chooses to leave the SimpLCity application. The user may switch from SimpLCity to the public-facing database and then return to SimpLCity without requiring the user to log in again. However, if a user has not yet logged in, or has manually logged out, they will be required to log in again before they can move from the public database into SimpLCity.

The "User intent" decision allows users to pick their flow: Users can create, edit, or import datasets, export their datasets, or submit them to the CALDC for public release. Users are under no obligation to release datasets publicly, and datasets that have not been submitted remain private and viewable only by the logged-in user who created them. The export option, detailed in subsection 5.2.9, allows users to privately download a copy of their dataset in the ILCD/EcoSpold2 format(s) without submitting the dataset for public release. Datasets may only be exported or submitted to the CALDC if they are considered 'complete', meaning that all required

5.2. Feature Walkthrough

Table 5.1: Summary of Features and Permissions.

Feature	Section	Required Permission
Login	5.2.1	Public
User Home Page	5.2.2	Logged-in User
User Options	5.2.3	Logged-in User
Browsing Datasets	5.2.4	Logged-in User
Viewing Datasets	5.2.5	Logged-in User
Managing Datasets	5.2.6	Logged-in User
Creating a New Dataset	5.2.7	Logged-in User
Editing a Dataset	5.2.8	Logged-in User
Exporting a Dataset	5.2.9	Logged-in User
Importing a Dataset	5.2.10	Logged-in User
Support Page	5.2.11	Logged-in User
User Settings	5.2.12	Logged-in User
Submitting User Feedback	5.2.13	Logged-in User
Using the Admin Console	5.2.14	Administrator
Public CALDC Homepage	5.2.15	Public
Navigating the Public CALDC	5.2.16	Public
Viewing and Downloading Public Datasets	5.2.17	Public

fields in the dataset have been filled. Once submitted, the dataset requires the approval of an administrator, after which it becomes immediately available in exported form on the public-facing database. Administrators are PRISM lab members or stakeholders tasked with ensuring that submitted datasets meet the necessary completeness requirements to be publicly released through the CALDC.

5.2 Feature Walkthrough

The CALDC provides a number of features, both for creating, editing, and exporting datasets, as well as managing and viewing existing datasets. Features can be broadly classified as being public features, accessible to any user; user-specific features, which require the user be logged in; and admin-specific features, which are only available to a logged-in user with administrator permissions.

5.2. Feature Walkthrough

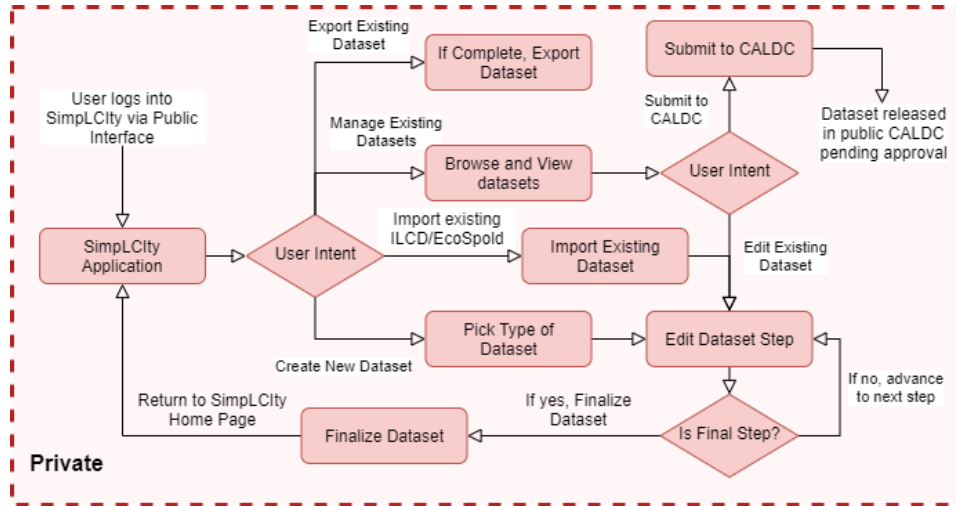


Figure 5.2: Private-facing SimplCity application dataset user flow.

5.2.1 Logging In

When first visiting the web application, the user is provided with a splash page that includes a login/registration form. From this page, the user may log in using their chosen email address and password, or register a new account with the email and password of their choice. In addition, the splash page includes some information about the SimplCity application, and links to both the public-facing CALDC interface (“View Public Datasets”), as well as an external link to the PRISM lab web page. If at any point an unlogged-in user tries to access a page that requires the user to be logged in, they are automatically redirected to the splash page. Attempting to visit the splash page while logged in will redirect the user immediately to the user’s home page.

5.2.2 User Home Page

When a user first logs into the SimplCity application, they are directed to a home page (/home). This home page includes recent SimplCity/CALDC news, as well as a table of the most recent datasets the user has created or edited. Clicking on a recent dataset will take the user directly to the viewing page for that dataset, facilitating quickly accessing recently worked-on datasets.

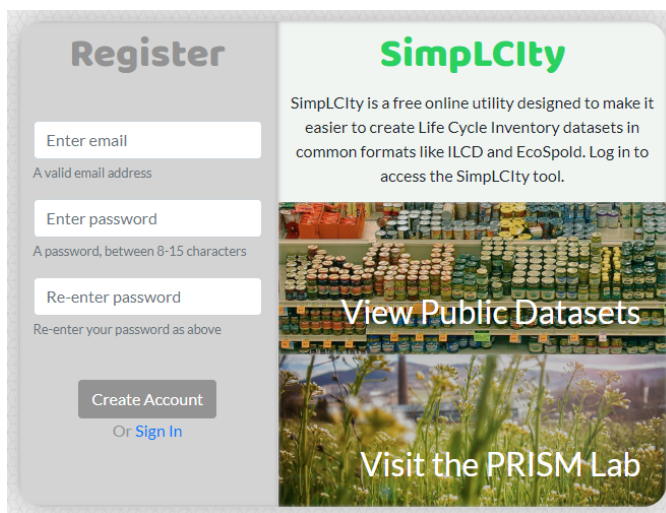


Figure 5.3: CALDC Splash page.

On the left-hand side of the page, a vertical navigation menu allows the user to navigate through the SimplCity application, or to move over to the public-facing CALDC to download public datasets.

5.2.3 Additional User Options

When logged in, the user’s email is displayed in the top navigation bar of the page. Clicking on the user’s email will open a drop down menu, providing the option to log out. If selected, the user will be immediately logged out and redirected to the CALDC splash page. In addition, if the logged-in user has administrator privileges, a second link in the drop-down menu will be available, labelled “Admin Console”, which takes the user to the administrator’s control panel.

5.2.4 Browsing Datasets

The user can browse datasets that they have created or imported by clicking the “Browse Data” link in the left-hand navigation menu. This will take the user to the Browse Datasets page (`/browse`). This page provides a paginated table of datasets belonging to the currently logged-in user. The table lists the name, version, type, UUID, and last-modified date for each dataset, and provides a basic search box that allows users to filter datasets

5.2. Feature Walkthrough

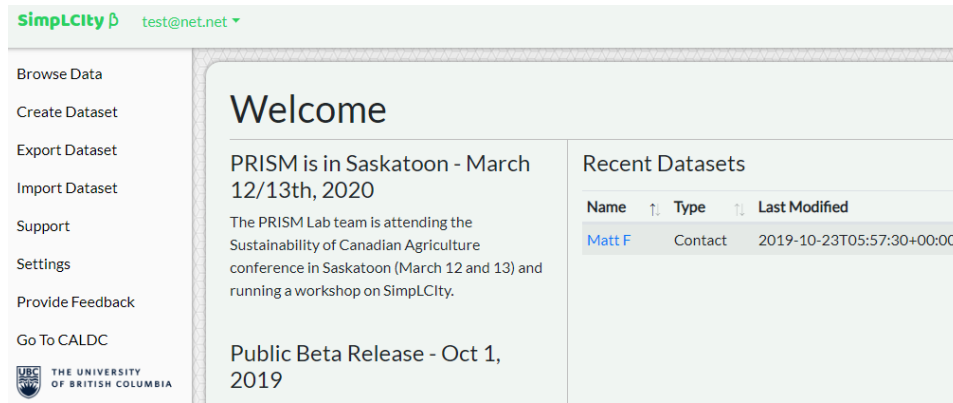


Figure 5.4: SimplCity Application Homepage.

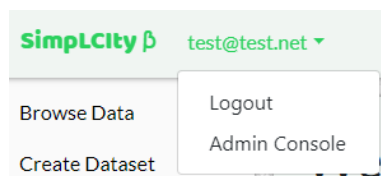


Figure 5.5: Dropdown User Menu. Note Admin Console option, denoting a user with administrator permissions.

based on search term. Clicking the name of the dataset (shown in blue, to denote a link) will immediately take the user to the viewing page for that dataset.

5.2.5 Viewing Datasets

Clicking the name-link for a dataset on your recent datasets (homepage) or on the Browse Datasets page will result in the user being navigated to the dataset viewing page. These pages have the URL format of (`/browse/<type of dataset>/<UUID>`). On this page, you will be able to see the values of each field in the dataset.

The name of the dataset being viewed is provided at the top of the page, and may use either the `shortName` or `baseName` field, depending on the type of dataset. Also at the top of the page will be a banner informing the user whether the dataset they are viewing is considered complete or not;

5.2. Feature Walkthrough

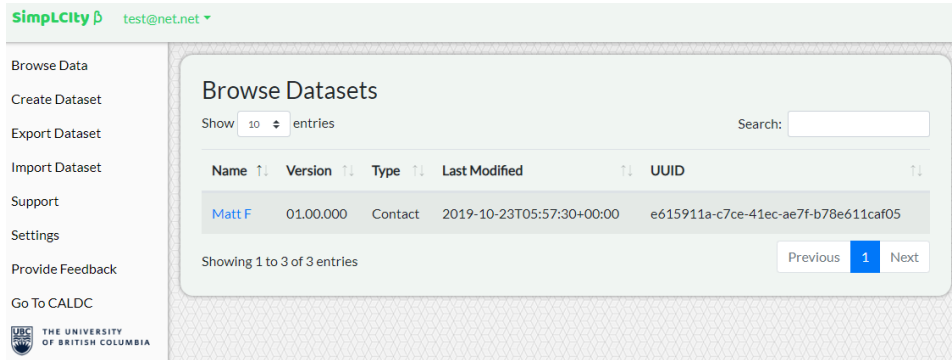


Figure 5.6: Browsing datasets page. Results can be filtered based on the search term provided.

incomplete datasets may not be exported as they lack required data fields.

Where a dataset references another dataset (such as a process referencing a flow), SimpLCity will provide a link to view the referenced dataset. This allows users to quickly follow the dependencies of a given dataset. If a field is missing a value, it will instead be highlighted in red and will display “Empty” in red italics. Note that because some fields are considered optional, it is possible to have a dataset that is considered complete, but has an empty field.

It is also possible for a user to view datasets that may not belong to them. Specifically, a user may choose to view an ILCD reference dataset that is used in one of their own datasets. Because the ILCD reference datasets are primarily stored as files and are static, the contents of the ILCD reference flow is simply rendered as XML in a standard text window. A warning message is displayed to notify the user that they are viewing an un-editable, static dataset.

5.2.6 Managing Datasets

If the user is currently viewing an editable dataset, a “Select Action” button will appear on the upper-right of the page. Clicking it will provide two possible options: “Edit Dataset”, which will immediately take the user to the editing view for this dataset, or “Submit to CALDC”, which will submit the current dataset to the CALDC for review and public release.

Field	Data
Base Name	Test Flow Basename
Treatments, Standards and Routes	Test Flow Technical Info
Mix and Location	Test Flow Production Mix
General Properties	Test Flow additional info
Name Synonyms	TEST FLOW, test flow, test
Category	Emissions to fresh water (1.1.1)
Classification	Metals and semimetals (6.2)

Figure 5.7: Viewing a dataset in SimpLCity.

This dataset appears to have all required fields completed.

Figure 5.8: Message informing the user this dataset is complete.

5.2.7 Creating a New Dataset

The user can create a new LCI dataset by selecting the “Create Dataset” link from the left-hand navigation bar, which will navigate them to the first dataset creation page (/create-dataset). This page prompts the user to select what kind of dataset they wish to create. Due to dependency ordering in the ILCD dataset format, a user will first be required to create a Contact dataset and then a Source dataset, before they are able to create Flows, Processes, or Flow Properties.

A warning will appear if this applies to the user, informing them they must first create a Contact and a Source.

Once the user has selected the type of dataset they wish to create, clicking the “Next Step” button will navigate the user into the dataset editing interface. Before clicking the Next Step button, the user may leave the page, cancelling the dataset creation.

Figure 5.9: An example of a dataset with missing fields in the dataset viewer.

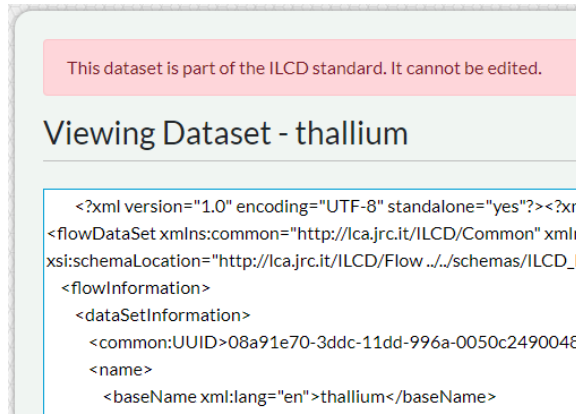


Figure 5.10: User viewing a static, non-editable ILCD reference flow.

5.2.8 Editing a Dataset

When creating a new dataset or editing an existing dataset, the interface used is the same. During the editing process, the user is presented with a number of steps. Each step includes attributes for a given topic or section of the output data format, i.e. ‘Contact Information’ or ‘Administrative Information’. The user can advance to the next step using the ‘Next Step’ button, or return to the previous step using the ‘Previous Step’ button. User-submitted data is saved between steps, making it possible for a user to leave part way through editing and later come back to continue editing. The step number, in the form of a progress bar, the title, and a brief description of the step is provided at the top of the page.

Each dataset field generates a form field in the editing page, with the type of form fields including text, email, numeric, select (with both static and dynamically-generated entries, depending on context), and textarea types. All inputs have a short descriptive label provided underneath, as well as additional information available on mouseover to help identify the necessary context and content for a given field. In addition, placeholder values are used in inputs, providing an example to the user of content and format.

Limits on input length and required fields are automatically generated

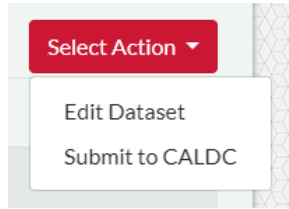


Figure 5.11: Options for managing an existing dataset.

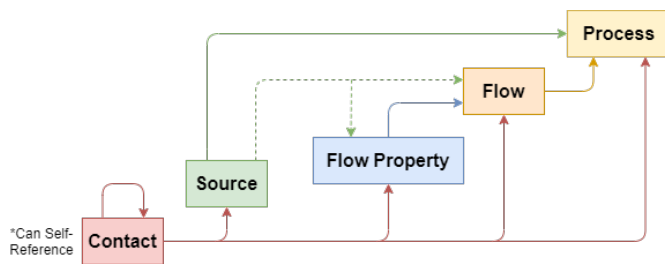
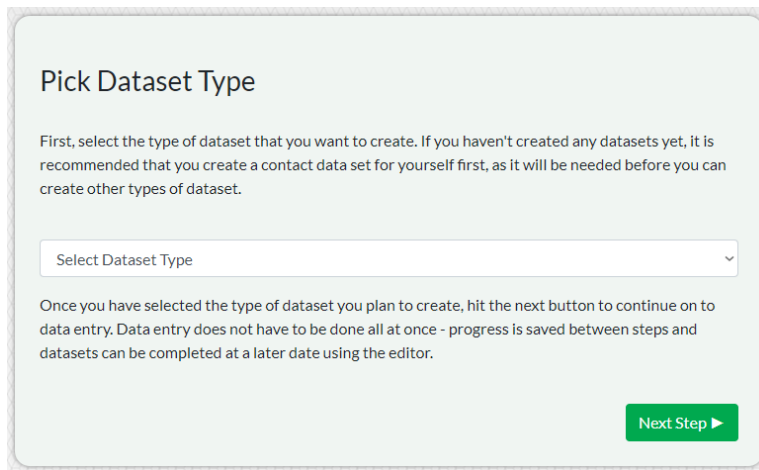


Figure 5.12: Graph of dataset dependencies. Solid arrows indicate a required dependency, while dashed arrows represent an optional dependency.

as part of the form elements. The forms use standard HTML5 client-side form validation, which will prevent the user from moving to the next step in the editing process until all required form fields have been filled. This also highlights the required empty fields for the user.

Some types of data provide a unique interface, for example the Input/Output Exchanges for a Process dataset. In this case, the user is presented with a table for entering flow data. The user may add or remove rows as needed for the required number of flows using the ‘+’ and ‘-’ buttons. The first flow (Flow #0) is considered the default reference flow, and is highlighted in green. In each row, the user can select a flow compartment from a dropdown. Choosing a flow compartment will then populate the ‘Flow Reference’ dropdown for that row with appropriate User-created and ILCD flows that match the specified compartment. The flow reference also states the default unit of the flow, i.e. kilograms, grams, liters, etc. The user can then choose the flow direction (Input or Output), and the mean and resulting amounts for that flow.

The final editing step for each dataset type replaces the ‘Next Step’ button with a ‘Finalize Dataset’ button. Once clicked, this will take the



Pick Dataset Type

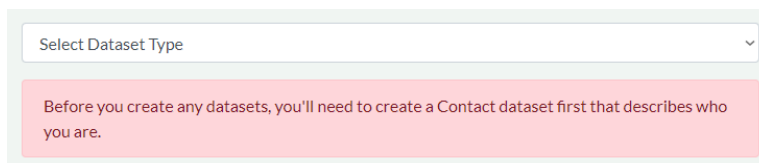
First, select the type of dataset that you want to create. If you haven't created any datasets yet, it is recommended that you create a contact data set for yourself first, as it will be needed before you can create other types of dataset.

Select Dataset Type

Once you have selected the type of dataset you plan to create, hit the next button to continue on to data entry. Data entry does not have to be done all at once - progress is saved between steps and datasets can be completed at a later date using the editor.

[Next Step ▶](#)

Figure 5.13: Picking dataset type to create.



Select Dataset Type

Before you create any datasets, you'll need to create a Contact dataset first that describes who you are.

Figure 5.14: Warning message informing the user they must first create a Contact dataset.

user to a confirmation page, informing the user that data entry is complete. It includes a direct link to the dataset viewing page via ‘My Dataset’, as well as a ‘Return Home’ button that will return the user to the home page.

5.2.9 Exporting a Dataset

The user can export a completed LCI dataset using the “Export Dataset” link from the left-hand navigation bar, which will take them to the export dataset interface (`/export-dataset`). The export dataset interface provides the user with a dropdown selection input, listing the type and name of each completed dataset belonging to the user. Datasets may only be exported if they include all required fields and are thus ‘complete’ – see subsection 5.2.5 for the interface shown on complete datasets in viewing mode. Once the

5.2. Feature Walkthrough

The screenshot shows a web form titled "Contact Information" with a progress bar at the top indicating "Step 1". The form contains several input fields and a dropdown menu. The first field is labeled "CALDC" with a subtext "A short name used when displaying this dataset." The second field is labeled "Canadian Agri-Food Lifecycle Data Center" with a subtext "The full name of the contact person or organization." The third field is a dropdown menu labeled "Persons" with a subtext "The type of contact that this contact dataset represents." The fourth field is labeled "fritter@mail.ubc.ca" with a subtext "Email address of the contact." The fifth field is labeled "https://prismlab.weebly.com/" with a subtext "Web address of the person or organization described by this dataset." The sixth field is labeled "matt@frit.me" with a subtext "A central contact point for an organization or person, should other means of contact fail. May be a telephone number, email address, web address, etc." A green "Next Step" button is located at the bottom right of the form.

Figure 5.15: Typical dataset editing interface, showing step 1 of creating a Contact dataset.

user has selected a dataset from the dropdown, clicking the ‘Export Dataset’ button will result in a download prompt for the compressed dataset.

During export, SimpLCity will automatically find all required dependencies for the dataset and package them together into a ZIP archive using the ILCD format, named ‘ILCD’, allowing import directly into OpenLCA. The dependency-finding process will include both all necessary user-created datasets, as well as supporting ILCD reference datasets. If the exported dataset is of the Process type, it will also include a .SPOLD file, which is the ecoSpold2-formatted Activity with ecoInvent nomenclature. Both the ‘ILCD’ ZIP file and the SPOLD activity profile are compressed into a single ZIP, which is then downloaded by the user.

Once downloaded, the user only needs to decompress the outer ZIP file. From there, the user may choose to use either the SPOLD formatted export, or the ILCD formatted export. Choice of which dataset format is used may depend on the user’s preferred LCA software suite, and the nomenclature and format used by other datasets.

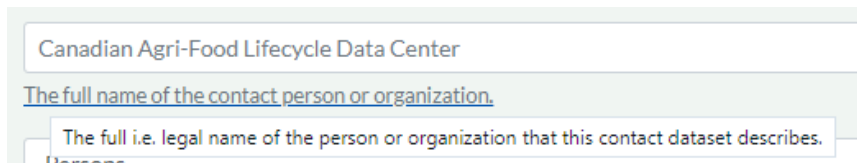


Figure 5.16: Example of placeholder text, label, and mouseover tooltip.

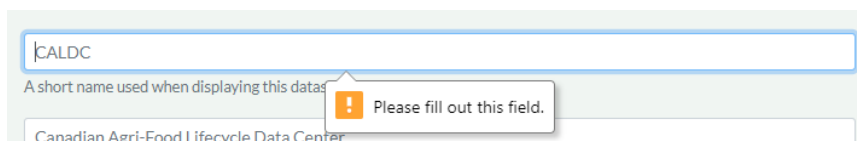


Figure 5.17: Example of form validation, highlighting the missing required field.

5.2.10 Importing a Dataset

SimpLCity also provides an interface for importing existing datasets, which can be reached by selecting ‘Import Dataset’ in the left-hand navigation menu, and takes the user to the import page (/import-dataset). Datasets may be imported as single XML files, either as an ILCD-formatted dataset (Contact, Source, Flow, Process, etc), or as an ecoSpold2 Activity, which will be imported as a Process. The user may upload a file by dragging and dropping it onto the ‘Upload a file’ button, or by clicking the button and finding the file through a navigator.

Once the user has selected a file for upload, they can select the ‘Import Dataset’ button to import. This will immediately take the user to an editing page with an interface identical to that used when editing an existing dataset (see subsection 5.2.8). The editing interface will fill out all fields identified from the imported dataset, while allowing the user to enter any missing data. Datasets that are ‘complete’ and have all required fields when imported will not require additional data entry, while incomplete datasets will. Note that if datasets are imported, they should be imported in order of dependency, such that datasets that are dependent on other datasets are imported afterwards, to preserve the links between datasets.

5.2. Feature Walkthrough

Flow #	Flow Compartment	Flow Reference	Flow Direction	Mean Amount	Resulting Amount
0	Emissions to soil, unspecifi	arsenic trioxide (kg)	Output	123.0	123.0
1	Non-renewable energy re:	brown coal; 11.9 MJ/kg (t	Input	1.1	1.1
2	Emissions to sea water	1,1-di-n-butylhydrazine (Input	1.0	0.0

+

-

- User Datasets
- ILCD Datasets
- 1,1-di-n-butylhydrazine (kg)
- 1,1-diallylhydrazine (kg)

Figure 5.18: An example of the Input/Output exchange interface, showing dropdown flow reference options.

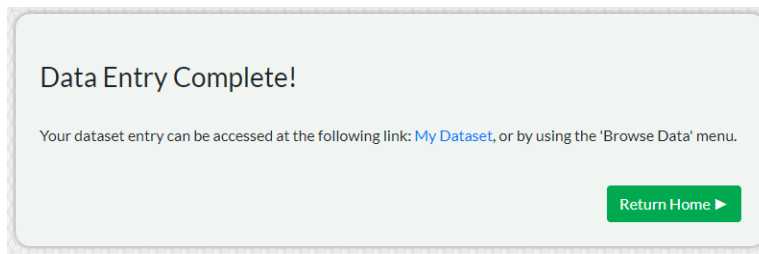


Figure 5.19: The dataset confirmation page shown after editing a dataset.

5.2.11 Support Page

In order to be more user-friendly, SimpLCity includes a basic support page that provides answers to questions users may have about using the service. This includes information about what the SimpLCity application does, how to use it, error descriptions, and general information regarding usage of the application. This page can be navigated to using the 'Support' link in the left-hand navigation menu, which takes you to the support page (/support).

The support page includes a topic list of links, which will automatically scroll the user to the corresponding topic on the support page.

5.2.12 Managing User Settings

User settings are available under the 'Settings' link in the left-hand navigation menu, which will take you to the settings page (/settings). The

5.2. Feature Walkthrough

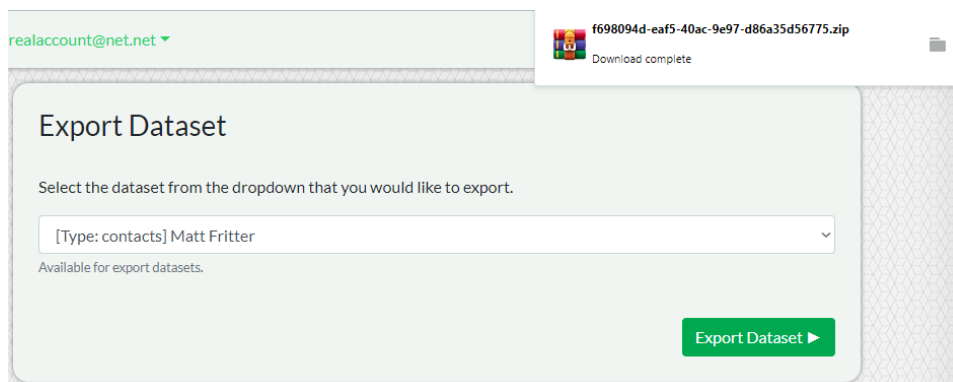


Figure 5.20: Export dataset interface, including download of the ZIP-compressed dataset.

user settings page allows configuration of user-specific settings, which are saved in the database and persist between user sessions. Currently, the only user setting available is the choice to ‘Use Mapped Flows Only’. When set to ‘True’, choice of ILCD reference flows in the Input/Output exchange interface (see subsection 5.2.8) will be limited to those flows that have an available ILCD-to-ecoInvent mapping, as described in the flowmap table (see subsection 4.2.4). This setting is ‘True’ by default.

If the user selects ‘False’, all ILCD reference flows will be available when creating an Input/Output exchange. However, use of unmapped ILCD reference flows will result in interoperability issues when exporting a dataset, as not all flows will be mapped to ecoInvent nomenclature. Once a user has selected their preference, they can click the ‘Save Settings’ button, which will save the current settings into the database and refresh the page.

5.2.13 Submitting User Feedback

SimpLCity includes a built-in feedback function, allowing users to anonymously submit feedback. The feedback page (/feedback) is available by selecting ‘Provide Feedback’ from the left-hand navigation menu. On this page, the user is presented with several questions to gauge user interest in SimpLCity and the CALDC, as well as provide a rating for the SimpLCity application. The user is also provided with free-text inputs to suggest features, describe problems or bugs with the application or its datasets, as well as provide any other comment on the software.

5.2. Feature Walkthrough

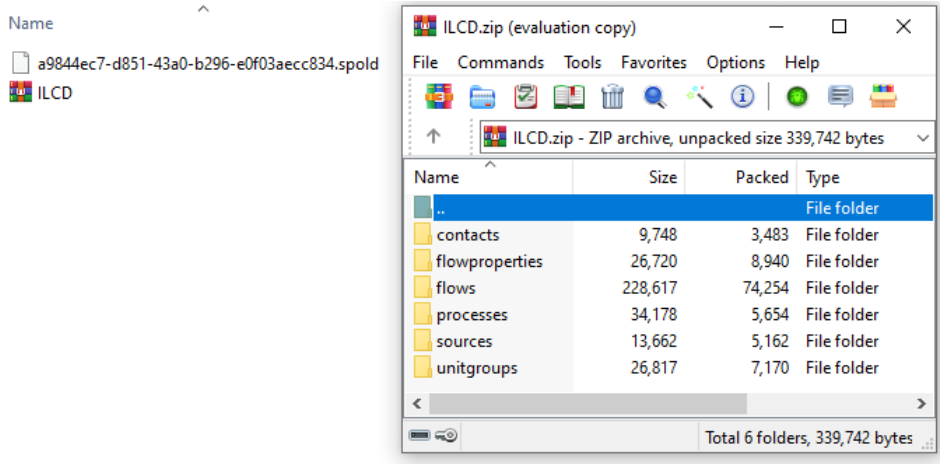


Figure 5.21: Contents of an exported Process ZIP file, showing SPOLD file and ILCD ZIP with hierarchy.

Once a user has provided feedback, they can click the ‘Submit Feedback’ button to submit. There is no requirement that the user completely fill out the feedback form. When the feedback has been successfully submitted, a notification is provided to the user informing them that it has been successfully saved. User feedback is stored in the feedback table in the form of a JSON-encoded object. The feedback table does not store any information on the submitting user, ensuring that feedback is collected anonymously and cannot be linked back to a user account.

5.2.14 Using the Admin Console

Clicking the ‘Admin Console’ link from the drop-down user menu described in subsection 5.2.2 will take the user the administration page (/admin). Only users that have the administrator permissions may access this page. Here, the user may view and approve datasets submitted to the CALDC for public release, view approved datasets, and add new administrators.

The first table presented to the user shows datasets that users have submitted using the ‘Submit to CALDC’ dropdown menu item on the viewing page. The table includes the UUID, type of dataset, dataset name, the submitting user’s email, and the time that the dataset was last submitted. The table can be paginated, sorted, and searched by the user.

5.2. Feature Walkthrough

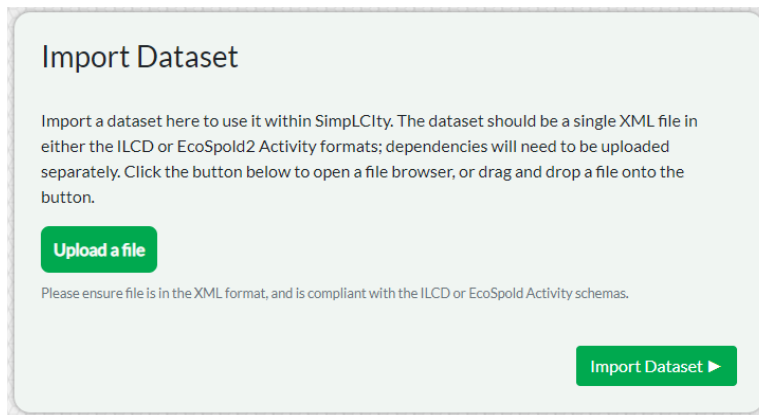


Figure 5.22: The Import dataset interface.

Clicking the ‘UUID’ link will download an exported copy of the submitted dataset with all dependencies, identical to that produced by the Export Dataset interface. This allows an administrator to download the submitted dataset for review in LCA software of their choice. Clicking the ‘Submitted By’ email link will automatically open up the administrator’s default mail utility to send an email to the user, should the administrator need to request additional information from the submitting user. Once an administrator has decided whether the dataset meets the necessary completeness criteria, they can then select either ‘Approve Dataset’ or ‘Decline Dataset’ from the dropdown ‘Actions’ menu.

If a dataset is declined, it will be removed from the submissions table. The submitting user will then need to edit or revise their dataset and re-submit it for review. If a dataset is approved, the dataset will be removed from the submissions table, and will instead appear in the ‘Approved Datasets’ table underneath. The ‘Approved Datasets’ table can also be paginated, searched, and sorted, and allows administrators to view datasets that have already been accepted. Once a dataset has been approved, it will be immediately listed on the public-facing CALDC database.

Finally, the Admin Console also allows an administrator to add additional administrators. A table of all administrators, listed by email address, is displayed at the bottom of the page. A dropdown input lists all non-administrator users by email. To give administrator privileges to a user, the administrator can select the user from the dropdown menu, and then click the ‘Add Administrator’ button. This will refresh the page, and the new

5.2. Feature Walkthrough

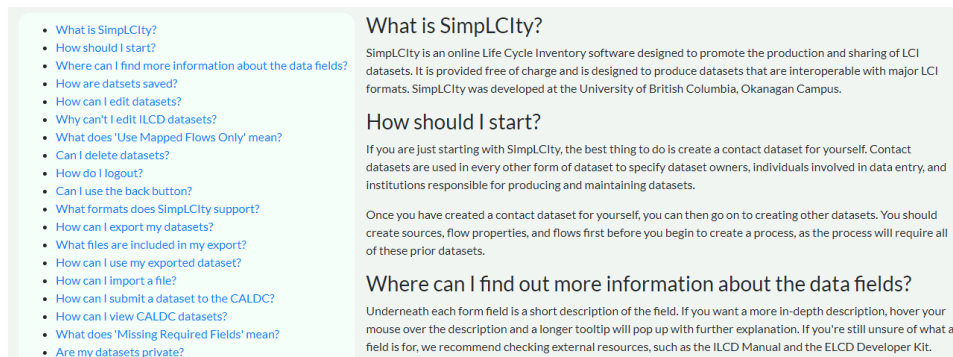


Figure 5.23: The Support page, showing the topic listing and initial topics.

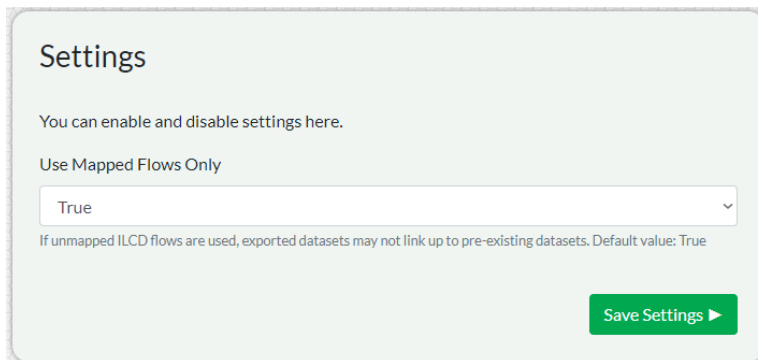


Figure 5.24: The Settings page, showing the 'Use Mapped Flows Only' setting.

administrator will now appear in the 'Current Admins' table. All administrators may approve or decline datasets, as well as add new administrators; it is therefore recommended that only those users actively involved in dataset review are given administrator privileges.

5.2.15 Public CALDC Homepage

The public CALDC database provides access to approved datasets for all users, without requiring a login. There are two ways of accessing the public CALDC: if the user is already logged into SimplCity, they can select the 'Go To CALDC' option at the bottom of the left-hand navigation menu.

5.2. Feature Walkthrough

Provide Feedback

Constructive feedback is always appreciated and helps identify areas of the software that could use further development. Please fill out the following survey to help us improve our software, and let us know what you think! All feedback is anonymous.

I would be interested in using Simplicity and the CALDC for life cycle analysis work:

Agree Neutral Disagree

I would be interested in seeing new features in Simplicity:

Agree Neutral Disagree

I consider LCA dataset format compatibility to be important for my work:

Agree Neutral Disagree

I intend to submit datasets for publication in the Canadian Agri-food Lifecycle Database Center:

Agree Neutral Disagree

Figure 5.25: Feedback page, showing Agree/Neutral/Disagree interface for feedback.

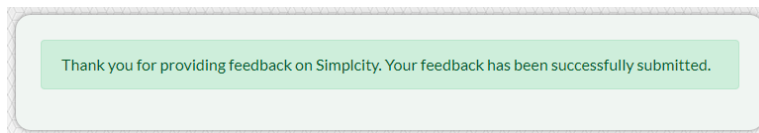
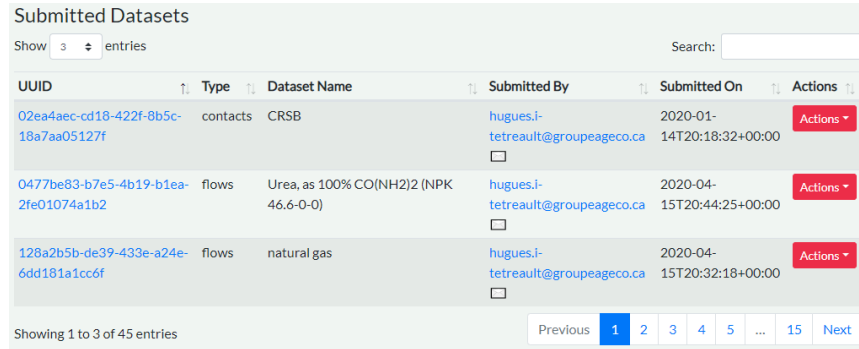


Figure 5.26: Success message upon submitted user feedback.

Alternatively, the user can click the ‘View Public Datasets’ button on the splash page, without registering or logging in. In both cases, the user will be redirected to the public CALDC home page at `/public/home`.

The public CALDC pages maintain the same format and style as those used in the SimpLCItY application. The home page provides some general summary statistics, such as total number of datasets being hosted, the total number of each type of dataset, and when the last new dataset was approved for each type. It also provides a brief explanation on what the CALDC is, and how it relates to the SimpLCItY application. The user can navigate back to this page using the ‘About’ link in the left-hand navigation menu.

5.2. Feature Walkthrough



The screenshot shows a table titled "Submitted Datasets" with columns for UUID, Type, Dataset Name, Submitted By, Submitted On, and Actions. There are three rows of data. The first row has UUID 02ea4aec-cd18-422f-8b5c-18a7aa05127f, Type contacts, Dataset Name CRSB, Submitted By hugues.i-tetreault@groupeageco.ca, Submitted On 2020-01-14T20:18:32+00:00, and an Actions dropdown menu. The second row has UUID 0477be83-b7e5-4b19-b1ea-2fe01074a1b2, Type flows, Dataset Name Urea, as 100% CO(NH2)2 (NPK 46.6-0-0), Submitted By hugues.i-tetreault@groupeageco.ca, Submitted On 2020-04-15T20:44:25+00:00, and an Actions dropdown menu. The third row has UUID 128a2b5b-de39-433e-a24e-6dd181a1cc6f, Type flows, Dataset Name natural gas, Submitted By hugues.i-tetreault@groupeageco.ca, Submitted On 2020-04-15T20:32:18+00:00, and an Actions dropdown menu. The table is paginated, showing 1 to 3 of 45 entries, with page numbers 1, 2, 3, 4, 5, ..., 15, and Next.

UUID	Type	Dataset Name	Submitted By	Submitted On	Actions
02ea4aec-cd18-422f-8b5c-18a7aa05127f	contacts	CRSB	hugues.i-tetreault@groupeageco.ca	2020-01-14T20:18:32+00:00	Actions
0477be83-b7e5-4b19-b1ea-2fe01074a1b2	flows	Urea, as 100% CO(NH2)2 (NPK 46.6-0-0)	hugues.i-tetreault@groupeageco.ca	2020-04-15T20:44:25+00:00	Actions
128a2b5b-de39-433e-a24e-6dd181a1cc6f	flows	natural gas	hugues.i-tetreault@groupeageco.ca	2020-04-15T20:32:18+00:00	Actions

Figure 5.27: The submissions table in the Admin Console interface.

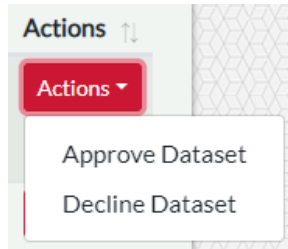


Figure 5.28: The action menu allows an administrator to approve or decline a submitted dataset.

5.2.16 Navigating the Public CALDC

The public CALDC allows users to browse datasets based on type of dataset. The left-hand navigation menu provides links based on dataset type. Selecting a type will take the user to a listing of all publicly available datasets of that type in the CALDC. These listings include the name of the dataset, the dataset UUID, and the date that it was submitted to the CALDC. These listings can be paginated, searched, and sorted by the user – for example, the user may sort based on submission time to get the newest datasets, or search for a specific UUID or dataset name. Clicking the ‘Name’ link for a specific dataset will redirect the user to the public viewing page for that dataset.

Each of these pages can be described using the URL format `/public/data/<type>`, where type describes an ILCD dataset type which will be listed on that page. In addition, the ‘Go to SimpLCItY’ link at the bot-

5.2. Feature Walkthrough

UUID	Type	Dataset Name	Submitted By
5afc3a8c-8d67-4a9b-b301-e98290ce1fe0	flows	harvested hay	hugues.i-tetreault@groupeageco.ca
6aaef363-cc8a-47ce-a649-3aab6714855b	processes	Stored hay from Alberta	hugues.i-tetreault@groupeageco.ca
a9844ec7-d851-43a0-b296-e0f03aecc834	processes	Hay from Alberta	hugues.i-tetreault@groupeageco.ca

Figure 5.29: The 'Approved Datasets' table, showing datasets that have been publicly released in the CALDC.

Admin Email Address
nathan.pelletier@ubc.ca
test@test.net
vivek.arulnathan@ubc.ca

Add New Administrator

test1@net.net

Select the user from the list you wish to promote to administrator.

Add Administrator

Figure 5.30: The administrator management interface, showing existing administrators and allowing new administrators to be specified.

tom of the left-hand navigation menu will allow a user to navigate back to the SimpLCity application. If the user is already logged in, they will be redirected to the SimpLCity home page, otherwise they will be returned to the splash page to either log in, register an account, or navigate elsewhere.

5.2.17 Viewing and Downloading Public Datasets

When a user clicks a 'Name' link in the Public Dataset tables, they will be redirected to the viewing page for that dataset. This viewing page is similar to viewing a dataset in SimpLCity, but provides the user with different options. Dataset views have the following URL format: `/public/data/detail/<type>/<UUID>`. On this page, the user is presented with a table consisting of data field names and values. This table can be paginated, sorted, and searched for specific data. Dependent or linked datasets such as Contacts, Flows, or Sources are rendered as links which will redirect

5.2. Feature Walkthrough

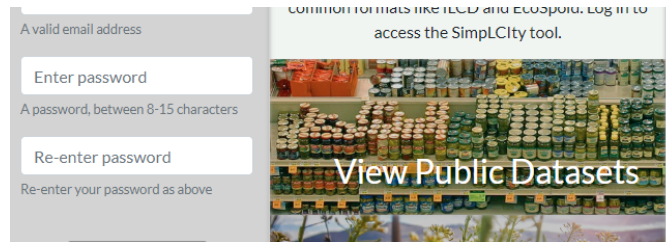


Figure 5.31: The 'View Public Datasets' button available on the splash page will take the user to the public CALDC.

The Canadian Agri-food Lifecycle Database Centre

The CALDC is providing public access to 89 LCA datasets for the Canadian Agri-food industry.

The Canadian Agri-food Lifecycle Database Centre is dedicated to providing public access to high quality, geographically and temporally relevant data for agri-food industries within Canada. These datasets are offered free-of-charge and are produced by industry stakeholders, LCA researchers, and LCA practitioners.

Datasets can be created and submitted directly for inclusion to the CALDC using the SimplCity tool. This tool was developed to provide a simple web interface for creating and viewing Life Cycle Inventory datasets. You can click [here](#) to go to the login page to access the SimplCity tool.

Dataset Summary		
Type	Count	Newest
contacts	5	2020-03-12
flows	63	2020-05-28
processes	20	2020-

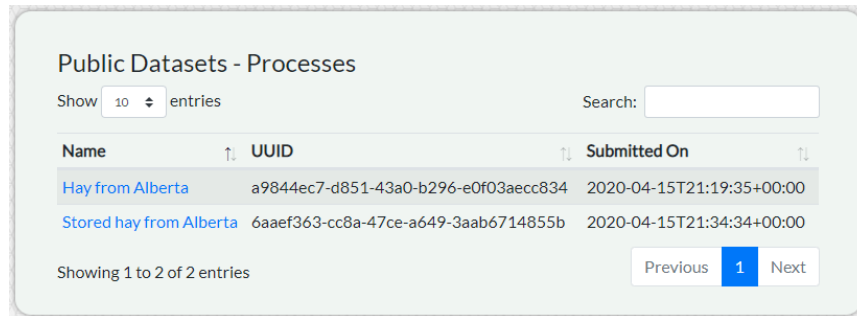
Figure 5.32: The CALDC public home page.

the user to the viewing page for the dependent dataset.

At the top of the page, a 'Download Now' button is provided. Clicking this button will prompt the user to download the complete ZIP version of the dataset, identical to that produced through the export function as described in subsection 5.2.9. In addition, a message will be shown specifying what type of dataset is available; if the dataset is a Process, it will also include the ecoSpold2 equivalent Activity in the export and this will be noted in the message.

When navigating to other datasets listed as dependencies, behaviour may change depending on whether the dataset has been made public, or is part of the ILCD standard datasets, such as ILCD elementary flows or unit groups. If a dataset has been included in a published dataset, but it has not yet

5.2. Feature Walkthrough



Name	↑↓ UUID	↑↓ Submitted On
Hay from Alberta	a9844ec7-d851-43a0-b296-e0f03aecc834	2020-04-15T21:19:35+00:00
Stored hay from Alberta	6aaef363-cc8a-47ce-a649-3aab6714855b	2020-04-15T21:34:34+00:00

Showing 1 to 2 of 2 entries

Previous 1 Next

Figure 5.33: Public listing of processes available in the CALDC.

Field	↑↓ Data
Approval Source	SCAC 2020
Base Name	Hay from Alberta
Classification	Agricultural production means (2.10)
Commissioner	CRSB
Completeness	Very good
Data Cutoff/Completeness Principles	Hay left in the fields was out of scope. CO2 removal from soil included (see the Canadian GHG inventory part 2 for more in
Data Entry	Groupe AGÉCO

Figure 5.34: Example of a public viewing page for a process dataset, showing the field values it contains.

been submitted or approved, the dataset will be included as a dependency when the published dataset is exported. However, the unpublished dataset will not be available through the public viewer, instead returning an error message stating that the dataset is not yet publicly available. If the dataset the user wishes to view is part of the static ILCD reference files, the dataset will be shown, but in a plain XML format. In addition, it will include a warning informing the user that the dataset is part of the ILCD standard reference datasets.

5.2. Feature Walkthrough

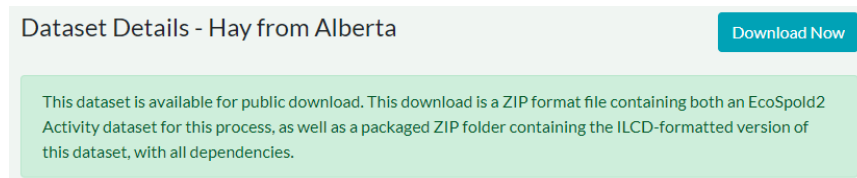


Figure 5.35: 'Download Now' button and message informing the user the dataset will contain both ILCD and EcoSpold2 formatted datasets.

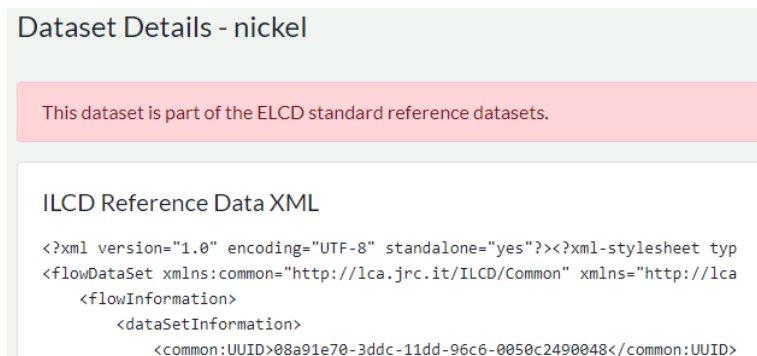


Figure 5.36: Example of viewing an ILCD reference flow in the public CALDC.

Chapter 6

Discussion and Recommendations

As part of the development of the CALDC, a set of recommendations were developed, designed to promote interoperability, good technical practice, and the use of third-party database distributors. These recommendations were subsequently presented and published at the LCA Food 2018 Conference as well as in the International Journal of Life Cycle Assessment. The recommendations are designed to be general guidelines for new LCI database initiatives, with focus being primarily on ensuring that datasets produced for these initiatives are interoperable with other data sources.

Not all of the recommendations made were implemented within the CALDC, or were only partially implemented. This was due to both time constraints in development, as well as a lack of implementable solutions for some issues, such as the mapping of EcoSpold2 and ILCD flows. Where the recommendations have not been implemented represents areas for potential future updates to the CALDC database system, as well as areas of LCA interoperability that require additional research. While the CALDC presents a new approach to LCI database development in promoting interoperability and the inclusion of the ability to create and edit datasets within the database application itself, there remain unresolved issues in the LCI database field that will require further consensus among the LCA community to solve.

With the public release of the CALDC, external users have begun to make use of the application. User uptake is summarized in Table 6.1 below. Due to the relatively niche application of the CALDC and the short time since release, current user count remains low. In addition, very little user feedback has been provided, consisting of only nine user feedback reports. Much of this feedback was generated during the internal and stakeholder testing phases, and represents bugs and/or features that have since been patched prior to public release. It is our hope that we will continue to receive feedback on the web application as more users have a chance to use the CALDC and SimpLCity.

Table 6.1: Total Counts as of July 2020.

Metric	Total Count
Registered Users	130
User-Created Flows	143
User-Created Contacts	44
User-Created Processes	57
Datasets Approved	89

6.1 Interoperability Recommendations

Reviewing the present state of interoperability in the LCI database field, it becomes apparent that developers of a new LCI database must weigh the benefits and costs of creating an interoperability system. While the benefits of some interoperability measures are abundantly clear, others are potentially too new, or too costly, to merit inclusion in the initial release of a new LCI database. However, the potential costs in terms of time and effort converting data to be included within these services at a later date, among either dataset creators or database administrators must also be considered. In this regard, the changing landscape of the LCI database field, particularly in terms of newer semantic data schemes and third-party data providers, requires consideration. The summary of these recommendations is presented in Table 6.2.

It is clear that the two most accepted formats for data exchange in the LCI field are the EcoSpold2 and ILCD formats. For this reason, both were implemented in the CALDC. Other new LCI database initiatives should similarly support both formats from the outset to ensure maximum database interoperability, as well as compatibility with current LCA software suites. The development of data collection/provision templates that contain all fields required for both formats is therefore strongly recommended. This may result in a more resource-intensive LCI data compilation and reporting process, but is ultimately necessary for interoperability across databases using these formats. It is essential that loss of data during conversion from one format to the other is avoided for those fields which do not have an equivalent across the two formats. This system of using an intermediate data collection format has been implemented in the CALDC, showing the ability to perform lossless conversion into both EcoSpold2 and ILCD at the cost of slightly more intensive data entry.

In addition to EcoSpold2 and ILCD, other dataset formats exist, such

6.1. Interoperability Recommendations

Table 6.2: Recommendations for LCI Database Initiatives.

Consideration	Recommendation
Format	Provide, at a minimum, datasets in both the ILCD and EcoSpold 2 formats in XML encoding, consider providing datasets in JSON format.
Nomenclature	Enforce use of a subset of common nomenclatures, particularly the ILCD and EcoSpold nomenclatures, with nomenclature choice dependent on popularity and usage in other databases
Third-Party Providers	Networks & Integrate datasets with third-party providers where possible, particularly those with low barrier to entry such asecoinvent and OpenLCA. Assess cost/benefit ratio of third-party network integration such as LCDN and GLAD, integrate early to minimize necessary dataset conversions.
Third-Party Initiatives	Follow recommended practices in the UNEP/SETAC Global Guidance Principles for data documentation, review, and quality.
Technical Implementation	Implement the LCI database using proven database and web development technologies, such as SQL, Microsoft IIS, Apache/NGINX, etc. Implement APIs where possible to expose underlying database information to developers and users.

as the proprietary CSV format used by the SimaPro LCA software. While LCA software suites usually support either ILCD or EcoSpold2 import, or both, it may be beneficial in the future to consider the ability to also export datasets in these formats. This would ease the integration of data from an LCI database into a practitioner’s LCA software, but potentially creates additional interoperability issues with regard to nomenclature.

XML is currently the standard encoding. However, also providing data in JSON-LD is recommended so as to facilitate interoperability with databases that have moved to the JSON-LD format, such as the USDA LCA Commons. This would also allow for easier parsing and semantic data work. JSON-LD data would also be ideal for a public API, as the lower overhead would reduce bandwidth requirements and be easily parsed with standard JSON parsers. While the JSON-LD encoding does not share the same popularity as XML encoding, it would be in the interest of LCI database initia-

6.1. Interoperability Recommendations

tives to support it as a means of future-proofing should JSON-LD become a more predominant encoding for LCA data. While the CALDC does not currently support JSON-LD encoding, it would be relatively trivial to implement. JSON enjoys good, built-in support in the Python environment that underlies the CALDC, and data is primarily stored in a neutral format within the database, making creation of a JSON-LD encoded object fairly straightforward. Implementation of this feature in the future would help ensure that datasets developed and distributed through the CALDC can also be distributed alongside other JSON-LD datasets through third-party initiatives that have adopted this encoding.

In terms of nomenclature, it is highly desirable that all new LCI databases, including the CALDC, enforce the use of a specified set of common nomenclatures from the outset, with mapping between each to allow for conversion via semantic cataloging or simple equivalence matrices. Ideally, LCI data should be downloadable in multiple nomenclatures to ensure interoperability with LCI databases sourced from third parties and their effective integration using common LCA software. However, this would require further advancement of existing LCI nomenclature mappings, or the widespread adoption of a single nomenclature standard across the LCA field. Until such mappings are made available or a singular format is adopted, the best strategy in terms of nomenclature is to use one that is already popular (such as the ILCD orecoinvent nomenclatures), and use the mappings that do exist to provide conversion where possible.

Nomenclature within the CALDC is currently restricted to the use of ILCD elementary reference flows, and a subset of EcoSpold2 reference flows that have been mapped. This mapping is not ideal, as many ILCD reference flows remain unmapped and thus unusable when creating EcoSpold2-compatible datasets. The mapping of LCA nomenclatures is an active topic of research within the LCA community, and it may be necessary in the future to update the mappings within the CALDC should a better mapping become available. In addition, alternative approaches, such as the use of semantic cataloging or the development of a ‘universal’ nomenclature, may become feasible in the future. It will be necessary for the CALDC to remain up-to-date on existing mappings and nomenclature solutions to ensure that datasets remain interoperable.

6.2 Technical Recommendations

The choice of actual software for web architecture as well as back-end software is unimportant for interoperability. As such, choice of database, web framework, and web server should be predicated on the stability and longevity of the software. Apache, Microsoft IIS, MySQL and MSSQL are all mature software packages with a long history of use in web database applications and would be suitable for development of new LCI databases such as the CALDC. Providers may also consider the LCA Collaboration Server, LCDN, and GLAD networks as alternatives to developing a standalone LCI database system; this would reduce the time and expense of development, allowing those expenditures to be directed towards dataset development. This also has the effect of making it easier for smaller data providers to enter the LCI database field. New LCI databases should, however, include an extensive API that allows full access to the datasets and metadata in both XML and JSON-LD formats. APIs have become commonplace in web development as a means of facilitating third party support. The ability for third parties to develop applications that pull data directly from the database would promote the creation of new LCA utilities at no further expense to the maintainers of the database. In addition, this would further promote open, public distribution of LCI data, a core concept for the CALDC and any other public LCI database.

In terms of technical implementation, the CALDC is typical of a modern, database-driven web application. Use of a common language and web framework, in the form of Python/Flask, helps ensure that code will be maintainable in the future; while the use of enterprise and open-source software such as the Apache web server and MariaDB SQL database ensure continuing software support and updates. These choices of software and back-end scripting language do not directly impact the interoperability of the CALDC, but instead ensure the longevity and stability of the LCI database application, which should be of concern to any LCI database developer.

While a public API has not been implemented within the CALDC, the web application largely relies on an underlying routing schema that could easily accommodate the development of a documented, public API. As the CALDC continues to grow, the development of a public API would allow third parties as well as practitioners to programmatically fetch datasets released through the CALDC, potentially easing the integration of CALDC datasets into third-party databases and initiatives.

6.3 Third Party Data Providers and Initiatives Recommendations

At the external level, new LCI database initiatives should work with third party data providers to further increase the reach of their data and ensure interoperability with existing datasets. As both ILCDC and EcoSpold 2 formats should already be supported, there are minimal costs to submitting the data to theecoinvent and OpenLCA Nexus providers, assuming that nomenclatures are chosen that will function properly with these providers. The ability to have additional datasets immediately available to populate background processes is a convincing argument for their inclusion as interoperability measures, particularly in the case of OpenLCA Nexus, where much of the data conversion and packaging work is handled by a third party. LCI initiatives should also strongly consider participation in a network-based LCI database such as the LCDN, GLAD, or the LCA Commons. In particular, the ability to submit datasets directly through tools such as the Soda4LCA application or the LCA Collaboration Server means that datasets could potentially be shared across multiple networks with relatively little effort.

Presently, the CALDC has only been in public release for a short period of time, and only a small selection of datasets have been publicly released. As a result, the CALDC has not yet integrated its datasets with any third-party databases or initiatives. The lack of an appropriate mapping between EcoSpold2 and ILCDC in particular limits the ability to immediately submit datasets to theecoinvent database, as ILCDC reference flows used within the CALDC may not be automatically mapped to an appropriate EcoSpold2 flow, requiring manual intervention to ensure interoperability. Until a better mapping is available, this will necessarily restrict the ability to submit datasets to alternative providers to those datasets that use only properly-mapped flows.

As the number of datasets available through the CALDC continues to grow, it is recommended that third parties be approached to ensure that datasets enjoy a wide distribution through other databases, where LCA practitioners who are not aware of the CALDC may find them. Of particular interest is the LCA2 Initiative, which seeks to create a Canadian LCI database for all materials used and produced in Canada. This is a much larger scope than the CALDC, and it may be possible to integrate most if not all of the datasets produced through the CALDC into the LCA2 Initiative, providing users with a more centralized repository for Canadian LCI data.

Chapter 7

Conclusion

As ecological concerns associated with industrial activity continue to grow the further expansion of the Life Cycle Assessment field seems inevitable. As new database providers and initiatives continue to come online, LCA practitioners are able to call upon a growing library of datasets. Despite the growing amount of data being made available, the research suggests that barriers remain to the effective sharing and re-use of Life Cycle Inventory datasets between practitioners. This research identified key areas of interoperability that negatively impact the ability to effectively integrate datasets into existing models. Where possible, these issues were addressed in the development of the Canadian Agri-food Life Cycle Data Centre, which serves as a demonstration of potential solutions to interoperability problems in the LCA domain.

This thesis serves as both documentation for the development and usage of the CALDC, as well as a set of recommendations for developing new LCI database applications. Based on background research, prototype development, and consultation with other LCA database providers, it was determined that the underlying application software was not of concern with regards to interoperability. Rather, it was determined that differences in both file format, between databases and LCA software, as well as the use of multiple nomenclatures posed the most significant barriers to the re-use of LCI data sets. To this end, an application called SimpLCity was developed to allow the use of standardized ILCD nomenclature and format, with mapping allowing for the simultaneous creation of EcoSpold2 datasets. This application was integrated directly into the CALDC, allowing datasets to be created, edited, viewed, and downloaded through a single online interface.

In terms of computer science, the novelty of this research stems from the research domain itself. The field of Life Cycle Assessment is heavily based on analytical calculation and estimation; it is an extremely data-driven process, often requiring very large inventories of data. The lack of a single standardized format or nomenclature introduces difficulties in developing an interoperable system, requiring extensive research of existing LCI data formats and the development of new mappings between formats.

In addition, due to the frequent need to create many datasets for an LCA study, the idea of making data entry more efficient and user-friendly also has merit. At this time, SimpLCity is the only fully-online application for the development of ILCD or EcoSpold2 datasets, providing a simple, step-by-step interface to create and modify datasets. While not a complete solution to the interoperability challenges that LCA faces, the CALDC and the SimpLCity application represent a positive step forward towards greater format support and dataset sharing.

In conclusion, a data entry web application and database were developed for the creation and storage of LCA datasets. In doing so, it was necessary to develop solutions to interoperability problems that would prevent the easy integration of CALDC data into existing LCA models. While the CALDC implements a basic mapping of ILCD flows to EcoSpold2 flows for cross-compatibility, it remains incomplete. In developing the CALDC, the application alleviates many interoperability issues through the use of format and nomenclature mapping, but it cannot be said that these issues have been entirely resolved. Further effort in mapping existing nomenclatures would be necessary for the CALDC to become truly interoperable. Nevertheless, the application serves as a new and novel approach to the development of an LCA database and data entry application while addressing interoperability concerns.

Bibliography

- [Aru20] Vivek Arulnathan. Simplicity training. In *Sustainability of Canadian Agriculture Conference*, March 12-13, 2020 March 12-13, 2020. → pages 38
- [BGC19] Jonas Bunsen, Sebastian Greve, and Andreas Citroth. Lca collaboration server user manual and introduction. Technical report, GreenDelta GmbH, January 2019. → pages 11
- [BPSM⁺08] Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, and François Yergeau. Extensible markup language (xml) 1.0 (fifth edition). Technical report, World Wide Web Consortium (W3C), November 2008. → pages 9
- [Bre99] Rolf Bretz. Setac lca workgroup: Data availability and data quality. *The International Journal of Life Cycle Assessment*, 4(1):15, Jan 1999. → pages 8
- [CAC⁺17] Andreas Citroth, Peter Arbuckle, Edivan Cherubini, Cassia Ugaya, and Ashley Edelen. . Technical report, UNEP, June 2017. → pages 13
- [CAM⁺15] Vincent Colomb, Samy Ait Amar, Claudine Basset Mens, Armelle Gac, Gérard Gaillard, Peter Koch, Jerome Mousset, Thibault Salou, Aurélie Tailleur, and Hays M.G van der Werf. Agribalyse®[®], the french lci database for agricultural products: high quality data for producers and environmental labelling. *OCL*, 22(1):D104, Jan 2015. → pages 25
- [CEF⁺03] Raul Carlson, Markus Erlandsson, Karolina Flemström, Ann-Christin Pålsson, and Johan Tivander. Data format mapping between spine and iso/ts 14048. Technical report, Chalmers University of Technology, Göteborg, 2003. → pages 8

- [CF06] Joyce Smith Cooper and James A. Fava. Life-cycle assessment practitioner survey: Summary of results. *Journal of Industrial Ecology*, 10(4):12–14, Oct 2006. → pages 1
- [CTSL98] Raul Carlson, Anne-Marie Tillman, Bengt Steen, and Göran Löfgren. Lci data modelling and a database design. *The International Journal of Life Cycle Assessment*, 3(2):106–113, Mar 1998. → pages 6
- [Cur04] M. Curran. The status of the life-cycle assessment as an environmental management tool. *Environmental progress*, 23(4):277–283, 2004. → pages 8
- [DBG02] Hans P. de Bruijn and Jerome B. Guinée. *Handbook on life cycle assessment*, volume 7. Kluwer Acad. Publ, Dordrecht, 2002. → pages 4, 5, 6, 7
- [EIR⁺17] Ashley Edelen, Wesley W. Ingwersen, Cristina Rodr'iguez, Rodrigo A. F. Alvarenga, Artur Ribeiro de Almeida, and Gregor Wernet. Critical review of elementary flows in lca data. *The International Journal of Life Cycle Assessment*, Jul 2017. → pages 9
- [Exc18] Stack Exchange. Stack overflow developer survey results, 2018. → pages 23
- [FIT⁺06] Matthias Finkbeiner, Atsushi Inaba, Reginald Tan, Kim Christiansen, and Hans-Jürgen Klüppel. The new international standards for life cycle assessment: Iso 14040 and iso 14044. *The International Journal of Life Cycle Assessment*, 11(2):80–85, Mar 2006. → pages 4
- [FKL16] Simone Fazio, Oliver Kusche, and Zampori Luca. Life cycle data network - handbook for data developers and providers. Technical report, Publications Office of the European Union, 2016. → pages 10
- [FR05] Rolf Frischknecht and Gerald Rebitzer. The ecoinvent database system: a comprehensive web-based lca database. *Journal of Cleaner Production*, 13(13):1337–1343, 2005. → pages 24, 25
- [GHH⁺11] Jeroen B. Guinée, Reinout Heijungs, Gjalt Huppes, Alessandra Zamagni, Paolo Masoni, Roberto Buonamici, Tomas Ekvall,

- and Tomas Rydberg. Life cycle assessment: past, present, and future. *Environmental science & technology*, 45(1):90–96, Jan 1, 2011. → pages 3
- [GM18] Grinberg and Miguel. *Flask Web Development*. O’Reilly, Sebastopol, 2 edition, Mar 5, 2018. → pages 29
- [HBOM17] Michael Z. Hauschild, Anders Bjørn, Mikolaj Owsianiak, and Christine Molin. *Life Cycle Assessment*. Springer International Publishing AG, Cham, 2017. → pages 1
- [HGH⁺14] Patrik Henriksson, Jeroen Guinée, Reinout Heijungs, Arjan de Koning, and Darren Green. A protocol for horizontal averaging of unit process data—including estimates for uncertainty. *The International Journal of Life Cycle Assessment*, 19(2):429–436, Feb 2014. → pages 7
- [HHD⁺15] Patrik J. G. Henriksson, Reinout Heijungs, Hai M. Dao, Lam T. Phan, Geert R. de Snoo, and Jeroen B. Guinée. Product carbon footprints and their uncertainties in comparative decision contexts. *PloS one*, 10(3):e0121221, 2015. → pages 7
- [IHT⁺15] Wesley Ingwersen, Troy Hawkins, Thomas Transue, David Meyer, Gary Moore, Ezra Kahn, Peter Arbuckle, Heidi Paulsen, and Gregory Norris. A new data architecture for advancing life cycle assessment. *The International Journal of Life Cycle Assessment*, 20(4):520–526, Apr 2015. → pages 8, 27
- [ISO06] ISO. Environmental management, life cycle assessment, requirements and guidelines, 2006. → pages 5, 6, 7
- [JE17] JRC-EPLCA. European life cycle database, 2017. → pages 28
- [JI10] JRC-IES. General guide for life cycle assessment : Detailed guidance, 2010. → pages 6, 7, 9
- [JI12] JRC-IES. International reference life cycle data system (ilcd) data network compliance rules and entry-level requirements version 1.1. Technical report, Publications Office of the European Union, 2012. → pages 10
- [JI18] JRC-IES. European life cycle database, June 2018. → pages 9

- [KDRJ16] Brandon Kuczenski, Christopher B. Davis, Beatriz Rivela, and Krzysztof Janowicz. Semantic catalogs for life cycle assessment data. *Journal of Cleaner Production*, 137:1109–1117, Nov 2016. → pages 7, 27
- [Kel07] Daniel Kellenberger. Modelling principles for the collection of consistent and comprehensive lci data, November 2007. → pages 8
- [KG14] Walter Klöpffer and Birgit Grahl. *Life Cycle Assessment (LCA) : A Guide to Best Practice*. Wiley-VCH, Weinheim, 1 edition, Mar 21, 2014. → pages 5, 6
- [KL14] Walter Klöpffer. *Background and future prospects in life cycle assessment*. Springer, Dordrecht, 2014. → pages 4
- [Kne18] Ralf Kneuper. *Software Processes and Life Cycle Models*. Springer, Cham, 1st ed. 2018 edition, 2018. → pages 20, 22
- [LBG18] Shaobo Liang, Richard Bergman, and Hongmei Gu. Workflow for publishing forestry lci data through the lca commons a case study. Technical report, United States Department of Agriculture, October 2018. → pages 11
- [LHPC15] Sébastien Lasvaux, Guillaume Habert, Bruno Peuportier, and Jacques Chevalier. Comparison of generic and product-specific life cycle assessment databases: application to construction materials used in building lca studies. *The International Journal of Life Cycle Assessment*, 20(11):1473–1490, Nov 2015. → pages 5
- [LILS04] National Renewable Energy Laboratory, Athena Sustainable Materials Institute, Franklin Associates Ltd., and Sylvatica. U.s. lci database project - user’s guide. Technical report, National Renewable Energy Laboratory, February 2004. → pages 25
- [LS16] Pascal Lesage and Réjean Samson. The quebec life cycle inventory database project. *The International Journal of Life Cycle Assessment*, 21(9):1282–1289, Sep 2016. → pages 1, 12
- [MMBV16] Ingo Meinshausen, Peter Müller-Beilschmidt, and Tobias Viere. The ecospold 2 format—why a new format? *The International*

- Journal of Life Cycle Assessment*, 21(9):1231–1235, Sep 2016.
→ pages 8, 9
- [MMC06] Karl MacMillan, Frank Mayer, and David Caplan. *Selinux by example: using security enhanced linux*. Technical report, Prentice Hall, 2006. → pages 34
- [Mor17] K. Moriarty. *Pkcs #5: Password-based cryptography specification*. RFC 8018, IETF, 1 2017. → pages 35
- [MSO17] Brenda Macdougall and Nicole St-Onge. *Digital archive database project summary*, 2017. → pages 29
- [MT15] Marcelle C. McManus and Caroline M. Taylor. The changing nature of life cycle assessment. *Biomass and Bioenergy*, 82:13–26, Nov 2015. → pages 3, 4
- [NRL⁺14] Thomas Nemecek, Vincent Rossi, Jens Lansche, Sebastien Humbert, and Patrik Mouron. *World food lca database methodological guidelines for the life cycle inventory of agricultural products*. Technical report, Quantis and Agroscope, July 2014. → pages 25
- [NSA⁺17] Bruno Notarnicola, Serenella Sala, Assumpció Anton, Sarah J. McLaren, Erwan Saouter, and Ulf Sonesson. The role of life cycle assessment in supporting sustainable agri-food systems: A review of the challenges. *Journal of Cleaner Production*, 140:399–409, Jan 2017. → pages 5
- [Ope15] OpenLCA/GreenDelta. *Openlca format converter*, 2015. → pages 9, 27, 33
- [Ope18] OpenLCA/GreenDelta. *Openlca nexus*, 2018. → pages 11
- [oW] University of Waterloo. *Canadian raw materials database*. → pages 1
- [OWA17] OWASP. *Owasp top ten*, 2017. → pages 34, 37
- [Owe06] Michael Owens. *The Definitive Guide to SQLite*. Apress, Berkeley, CA, 2006 edition, 2006. → pages 30
- [PAB⁺15] Nathan Pelletier, Fulvio Ardente, Miguel Brandão, Camillo De Camillis, and David Pennington. *Rationales for and limitations*

- of preferred solutions for multi-functionality problems in lca: is increased consistency possible? *The International Journal of Life Cycle Assessment*, 20(1):74–86, Jan 2015. → pages 4
- [Pel] Nathan Pelletier. Prism lab. → pages 2
- [RBF⁺15] Marco Recchioni, Gian Blengini, Simone Fazio, Fabrice Mathieux, and David Pennington. Challenges and opportunities for web-shared publication of quality-assured life cycle data: the contributions of the life cycle data network. *The International Journal of Life Cycle Assessment*, 20(7):895–902, Jul 2015. → pages 1
- [REF⁺04] G. Rebitzer, Tomas Ekvall, R. Frischknecht, D. Hunkeler, G. Norris, T. Rydberg, W-P Schmidt, S. Suh, B. P. Weidema, and D. W. Pennington. Life cycle assessment - part 1: Framework, goal & scope definition, inventory analysis, and applications. 2004. → pages 5, 6
- [Ref20] Netcraft - january 2020 web server survey, January 2020. → pages 29
- [RJ00] L. Rising and N. S. Janoff. The scrum software development process for small teams. *IEEE Software*, 17(4):26–32, 2000. → pages 19, 20
- [RRDB08a] John Reap, Felipe Roman, Scott Duncan, and Bert Bras. A survey of unresolved problems in life cycle assessment: part 1. *The International Journal of Life Cycle Assessment*, 13(4):290–300, Jun 2008. → pages 4, 5, 7, 10
- [RRDB08b] John Reap, Felipe Roman, Scott Duncan, and Bert Bras. A survey of unresolved problems in life cycle assessment: part 2. *The International Journal of Life Cycle Assessment*, 13(5):374–388, Aug 2008. → pages 5, 6, 7, 8
- [SLTC16] Sangwon Suh, Matthew Leighton, Shivira Tomar, and Christine Chen. Interoperability betweenecoinvent ver. 3 and us lci database: a case study. *The International Journal of Life Cycle Assessment*, 21(9):1290–1298, Sep 2016. → pages 10
- [Tiv15] Johan Tivander. A minimal ontology pattern for life cycle assessment data. volume 1461, 2015. → pages 19, 27

- [TSAP20] Ian Turner, Alyssa Smart, Emily Adams, and Nathan Pelletier. Building an ilcd/ecospold2-compliant data-reporting template with application to canadian agri-food lci data. *The International Journal of Life Cycle Assessment*, Mar 18, 2020. → pages 26
- [UNE17] UNEP/SETAC. Glad network, 2017. → pages 7, 13
- [US17] UNEP-SETAC. Glad network api/sdk documentation, 2017. → pages 13
- [vZLLR14] R. van Zelm, P. Larrey-Lassalle, and P. Roux. . *Chemosphere*, 100:175–181, 2014. → pages 6
- [W3C08] W3C. Web content accessibility guidelines 2.0. Technical report, W3C, 2008. → pages 28
- [WBH⁺13] Bo Pedersen Weidema, Christian Bauer, Roland Hischier, Chris Mutel, Thomas Nemecek, Juergen Reinhard, Carl Vadenbo, and G. Wernet. Overview and methodology: Data quality guideline for theecoinvent database version 3. Technical report, ecoInvent Center, May 2013. → pages 12
- [WDK11] Marc-Andree Wolf, Clemens Döpmeier, and Oliver Kusche. The international reference life cycle data system (ilcd) format – basic concepts and implementation of life cycle impact assessment (lci) method data sets. In *25th International Conference on Informatics for Environmental Protection, EnviroInfo 2011*, Aachen, October 5-7, 2011 2011. Shaker Verlag. → pages 9
- [Wei03] Bo Weidema. Market information in life cycle assessment. Technical report, 2003. → pages 4
- [WPC⁺12] Marc-Andree Wolf, Rana Pant, Kirana Chomkhamri, Serenella Sala, and David Pennington. The international reference life cycle data system. Technical report, Publications Office of the European Union, 2012. → pages 9

Appendix

Appendix A

Risk Assessment

This risk assessment is based on the OWASP Top Ten application security risks. Each risk is assigned an exploitability, weakness prevalence, weakness detectability, and technical impact score out of three in accordance with the OWASP Risk Rating Methodology. Three indicates a major risk in a given category, while one indicates a minor risk. Mitigating factors are listed for each risk.

A.1 Injection

Table A.1: Injection Risks.

Metric	Risk (1-3)
Exploitability	3
Prevalence	2
Detectability	3
Technical	3

Risk of malicious code injection is highly mitigated. The use of prepared SQL statements without direct concatenation prevents SQL injection, while automatic escaping prevents Cross-Site Scripting. Checking of dataset ownership prevents injection of UUIDs for datasets that do not belong to the user. Some risk may be posed by injection via uploaded XML file, in particular to 'billion laugh' (XML bomb) type attacks.

A.2 Broken Authentication

Risks of broken authentication are partially mitigated. User sessions are stored in encrypted, expiring cookies to minimize the chance of cookie theft. User passwords have an enforced minimum input length of eight characters, but are not compared against known bad passwords or required to have a

A.3. Sensitive Data Exposure

Table A.2: Broken Authentication Risks.

Metric	Risk (1-3)
Exploitability	3
Prevalence	2
Detectability	2
Technical	3

minimal complexity. Lack of password recovery system means no attack vector for falsely recovering a password, but does potentially allow for a social engineering attack against the system administrator. Use of a salted SHA-256 password column provides some security against lookup or rainbow table attacks, but could be improved through the use of a key-stretching function.

A.3 Sensitive Data Exposure

Table A.3: Sensitive Data Exposure Risks.

Metric	Risk (1-3)
Exploitability	2
Prevalence	3
Detectability	2
Technical	3

Risk of sensitive data exposure is largely mitigated. Most data stored within the CALDC is of low sensitivity and presents minimal risk if leaked due to the public nature of the CALDC mission. User passwords present the major sensitive information stored within the application. As mentioned, the security of user password storage could be improved through the use of alternative hashing/salting arrangements. All standard communication with the server is done via encrypted HTTPS, SSH, or SFTP protocols, preventing data from being exposed while in-transit.

A.4 XML External Entities

Risk of XML external entities is completely mitigated. The Python3 eTree XML parser implementation does not allow external entity expansion

A.5. Broken Access Control

Table A.4: XML External Entities.

Metric	Risk (1-3)
Exploitability	2
Prevalence	2
Detectability	3
Technical	3

or document type definition retrieval, and returns a parse error if it encounters either. Only a problem when using an older XML parser, or a parser that enables external entities and document type definitions.

A.5 Broken Access Control

Table A.5: Broken Access Control.

Metric	Risk (1-3)
Exploitability	2
Prevalence	2
Detectability	2
Technical	3

Risk of broken access control is largely mitigated. Access control is handled server-side via database checks against an encrypted cookie generated at login on a per-page basis. SELinux and default Linux permissions are configured to minimize privileges of the web server application to only those needed to directly run the CALDC. Use of an .htaccess file prevents users from accessing directories and source code via HTTP/HTTPS, restricting user access to those files that are directly served to the user via application logic.

A.6 Security Misconfiguration

Risk of security misconfiguration is largely mitigated through previously enumerated security features. CentOS operating system comes partially pre-configured and pre-hardened against attack, but additional security hardening could be used. Use of an additional stateful firewall, or active security

A.7. Cross-Site Scripting

Table A.6: Security Misconfiguration.

Metric	Risk (1-3)
Exploitability	3
Prevalence	3
Detectability	3
Technical	2

module such as the Fail2Ban or mod_security Apache modules could provide additional protection against common automated attacks.

A.7 Cross-Site Scripting

Table A.7: Cross-Site Scripting.

Metric	Risk (1-3)
Exploitability	3
Prevalence	3
Detectability	3
Technical	2

Risk of cross-site scripting is mostly mitigated. Use of dynamically generated web pages without GET parameters prevents the use of typical URL-based reflected XSS attacks. Use of Flask’s built-in escaping prevents stored XSS scripting attacks by escaping sensitive characters in user input prior to the page being served. There may be potential for DOM XSS attacks targeting a user’s browser environment for session theft or redirection via insecure browser extensions. A Content Security Policy (CSP) should be used to protect users from such XSS attacks that manage to avoid the built-in escaping.

A.8 Insecure Deserialization

Risk of insecure de-serialization is completely mitigated. All serialization of data is done on the server-side by the application. All user inputs are in unserialized form. For example, all data collected in a flow exchange table is sent as individual integers and strings, and is serialized to JSON for storage on the server-side after being appropriately cast to necessary data types.

Table A.8: Insecure Deserialization.

Metric	Risk (1-3)
Exploitability	1
Prevalence	2
Detectability	2
Technical	3

A.9 Using Components with Known Vulnerabilities

Table A.9: Using Components with Known Vulnerabilities.

Metric	Risk (1-3)
Exploitability	2
Prevalence	3
Detectability	2
Technical	2

Risk of using components with known vulnerabilities is somewhat mitigated. The use of well-maintained and frequently updated software packages with long term support (LTS) largely prevents vulnerabilities, or patches them quickly after they are made public. Zero-day exploits remain a risk even with updates and LTS. Automation of patching and updating would ensure the most timely application of security-critical patches, but with the potential for issues if package updates contain any bugs or changes to underlying functionality.

A.10 Insufficient Logging & Monitoring

Risk of insufficient logging and monitoring is somewhat mitigated. SELinux and Apache provide good logging at the external web server and internal audit levels. Long average time until intrusion is identified suggests that current 30-day backup cycle is not long enough. Backup cycle could be strengthened through the use of longer-term offline backups in addition to daily 'hot' backups. In addition, the use of additional modules such as Fail2Ban would allow better identification and logging of malicious activity.

A.10. *Insufficient Logging & Monitoring*

Table A.10: Insufficient Logging & Monitoring.

<u>Metric</u>	<u>Risk (1-3)</u>
Exploitability	2
Prevalence	3
Detectability	1
Technical	2