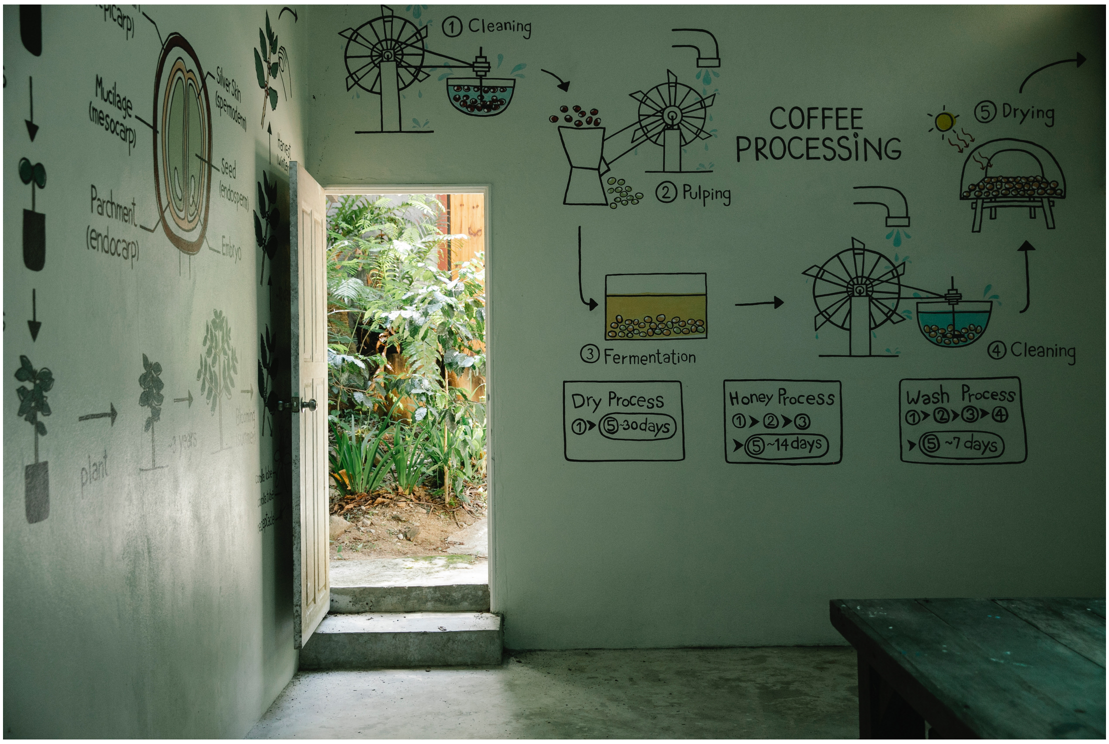
UNIVERSITY OF COPENHAGEN
FACULTY OF SCIENCE



Master Thesis

TRAZE

Total Resource Accounting for Zero-Emission

Camilla Krogh Dalsgaard & Sarah Maria Hyatt

Supervisor: Professor Fritz Henglein
Co-Supervisor: Christina Singh

Submitted: December 20, 2022

This thesis has been submitted to the Graduate School of the Faculty of Science, University of Copenhagen

Abstract

As the economy continues to grow, the demand for food, energy and materials follows suit. Climate change, biodiversity loss and pollution are globally pressing matters that require a fundamental transformation of resource-producing, consuming and recovering systems, to which sustainable supply chain management (SCM) and Carbon Credits (CCs) have gained momentum.

This thesis introduces TRAZE, a generic prototypic framework for total resource accounting of supply chain networks (SCNs), particularly addressing carbon accounting and CCs. TRAZE implements REALISTIC, an event-based modelling framework by Bager, Düdder, Henglein, Hébert, & Wu, 2022, and extends the algebraic resource accounting framework by Hébert, 2020 and the prototype by Alstrup & Borgert, 2022.

We demonstrate that it is formally possible to model and compute the carbon effect based on event data, where event data comprises assertions that something has happened or is measured. We provide a tool for interpreting data and deriving information related to SCNs and carbon emissions, and we address and analyse common attack vectors and challenges related to CCs. To prove our hypothesis, we implement a model and compute the carbon effect applied to a simple yet illustrative use case based on the Columbian coffee SCN. In conclusion, we argue that TRAZE is a flexible framework for fine-grained tracking of production and transportation of physical resources and reliably modelling their direct and indirect carbon effects across mutually distrusting actors in SCNs.

Acknowledgements

We express our deepest gratitude to our supervisor, Professor Fritz Henglein, for his invaluable guidance, feedback and expertise.

We also express our deepest gratitude to our co-supervisor, Christina Singh, for sharing her insight into the world of Carbon Credits, her feedback and shared scientific material.

Participating in this research project has been inspiring and enlightening. We thank our supervisor Professor Fritz Henglein for the opportunity and thesis idea.

Last, we would be remiss in not mentioning our gratitude for the support from our partners, close friends, families, and not to mention each other.

Contents

1 Introduction

The world faces a triple planetary crisis on climate, biodiversity and pollution where each issue has its causes and effects. Climate change is today's most pressing issue (Hetemäki, Kangas, & Peltola, 2022). The global temperature has continuously risen since the Industrial Revolution due to human activities, causing the release of large amounts of carbon dioxide (CO2) and other greenhouse gases (GHGs) into the atmosphere. The consequences are revealed through increased severity of rising sea levels, droughts, wildfires, water scarcity, flooding, storms, declining biodiversity and others[1].

On December 12 2015, 196 Parties agreed on a legally binding international treaty on climate change, known as the Paris Agreement (PA)[2]. This treaty aims to limit global warming to at most 2 degrees Celsius above pre-industrial levels and achieve a climate-neutral world by 2050. The PA calls for a fundamental transformation of resource-producing, consuming and recovering systems because the world must (1) reduce and avoid GHG emissions, (2) remove GHGs from the atmosphere and (3) reduce the extraction and processing of natural resources, which currently cause half of the global emissions and over 90% of biodiversity loss[3].

According to the United Nations (UN), biodiversity is the answer to several sustainable development challenges and is directly correlated to providing ecosystem services that are crucial to human life and well-being[4]. During the past 50 years, the global economy has grown nearly fourfold, following suit by increasing demand for food, energy and materials (Hetemäki et al., 2022).

Many solutions, therefore, look towards sustainable supply chain management (SCM) and supporting technologies. Blockchain[5] has emerged as one such, with several studies analysing the benefits and related challenges to its application to supply chain networks (SCNs)[6] and SCM. In the recent paper by Bager, Düdder, et al., 2022, the authors introduce a novel event-based modelling framework, REALISTIC, and utilise blockchain as a demonstration of its application.

REALISTIC extends the Resources Events Agents (REA) accounting model by McCarthy, 1982, by providing a systematic methodology for registering and tracking resources, ownership and location of resources, as well as events such as transfers, transports and transformations to provide tamper-proof recording and accounting, where agents (authenticated humans or robotic Internet of Things (IoT)) digitally and cryptographically

[1] Triple Planetary Crisis [online] `https://unfccc.int/blog/what-is-the-triple-planetary-crisis` retrieved 02-10-2022.

[2] Paris Agreement [online](2015) `https://unfccc.int/files/essential_background/convention/application/pdf/english_paris_agreement.pdf` retrieved 20-08-2022.

[3] UNFCCC and Circular Economy [online] `https://unfccc.int/news/shifting-to-a-circular-economy-essential-to-achieving-paris-agreement-goals` retrieved 01-10-2022.

[4] UN Biodiversity [online] `https://www.un.org/en/observances/biological-diversity-day` retrieved 02-12-2022.

[5] A blockchain (or distributed ledger) is a decentralised, append-only data structure containing nodes that share a global state and resembles a log of ordered transactions (Dinh et al., 2018).

[6] A SCN is a decentralised network of autonomous entities that collectively produce, transport, trade and finance physical goods, and it describes the flow and exchanges among parties on the network.

sign their events and, thereby, actions. With *secure transformations*, REALISTIC guarantees the exclusion of out-of-nothing production, preventing fraudulent activities, such as *double counting*, a term for counting resources or assets more than once. In other words, REALISTIC accounts for all resources, agents, ownerships and locations to ensure resource preservation across events, meaning everything appears out of something, and nothing disappears into nothingness.

This notion of circularity concerning resource tracking and accounting has been reviewed in related research. Several studies discuss the application of the Circular Economy (CE) as a means to eliminate many of the disadvantages of the traditional Linear Economy (LE) and SCM, as well as transition to a sustainable economy (Casado-Vara et al., 2018; Corona, Shen, Reike, Carreón, & Worrell, 2019; Hetemäki et al., 2022; Yadav, 2022).

In an international attempt to mitigate climate change and transition towards a more sustainable economy, capturing or sequestering carbon from the atmosphere has emerged as a tradeable commodity, namely Carbon Credits (CCs). Pricing carbon capture and establishing markets for trading CCs, so-called carbon markets, is, according to Marke, Mehling, & de Andrade Correa, 2022, one of the most feasible approaches to reducing carbon emissions in the economy. In addition, according to Reichle, 2020, the price represents the cost of the environmental damage of GHG emissions.

While CCs prove promising, challenges and common attack vectors need addressing. In creating CCs, it is required that the captured carbon is additional to what would otherwise have occurred, the carbon must be captured permanently and leakage correctly accounted for (Thamo & Pannell, 2015). Several studies discuss difficulties and proposed aids in ensuring these requirements (Galik, Murray, Mitchell, & Cottle, 2014;Kotsialou, Kuralbayeva, & Laing, 2022;Gustavsson et al., 2000;Thamo & Pannell, 2015). Among common attack vectors are *double counting* and *greenwashing*, a term used for fraudulently claiming to reduce emissions or act environmentally friendly, i.e. buying fake CCs or falsely overestimating the amount of carbon captured. Ultimately, many of these challenges prove hard to solve and, therefore, support the importance of securely tracking direct carbon effects, such as carbon emitted and carbon captured, and tracking products they are associated with across an entire SCN.

This thesis introduces TRAZE, a generic framework for total resource accounting, particularly addressing carbon accounting. TRAZE implements REALISTIC, employs the prototypes by Bager, Düdder, et al., 2022 and Alstrup & Borgert, 2022 and implements and extends the algebraic resource accounting framework by Hébert, 2020. We hypothesise that it is formally possible to model and compute the carbon effect based on event data, where event data comprises assertions that something has happened or is measured. Using models, we provide a tool for interpreting data and deriving information related to SCNs and carbon emissions.

We implement TRAZE as a web application with a local database, where the logic, in essence, builds on vector spaces to model REALISTIC using well-known algebraic operations for combining, calculating and evaluating events. While presenting resources as one or more ideal, virtual or real effects that must be registered (and thereby certified), we model CCs as resources and use geographical location and area detections to monitor CCs and prevent common attacks.

In addition, the TRAZE architecture is event-based and implements *total resource accounting*, meaning the provenance of all resources, including time, owners and locations, is tracked consecutively based on digitally signed asserted events. Thereby, using events of any given SCN, TRAZE acts as a digital twin of which we can compute the according carbon effects of the given SCN, also implying that we can model and calculate the carbon and ecological effects of each resource throughout its entire lifecycle. Built into the algebra is a *global invariant* to ensure resource preservation across events, guaranteeing certified resources cannot be transferred, transformed or transported from uncertified resources, and the amount of certified outgoing resources cannot be greater than the amount of certified incoming resources. Consequently, the demand for environmentally certified resources along SCNs is incentivised.

The main contributions of this thesis are the following:

- We present TRAZE, a generic framework based on REALISTIC by Bager, Düdder, et al., to model arbitrary SCNs, encompass CCs for modelling and computing carbon effects of any SCN and prevent or encumber common attack vectors.
- We extend the algebraic resource accounting framework by Hébert to include transports and locations.
- In proving our hypothesis, we implement a model for computing the carbon effect applied to a simple yet illustrative use case based on the Columbian coffee SCN.
- We include a background, analysis and discussion on the topic of CCs and the paradoxical aspects in the field.
- Secure transformations, transfers and transports that guarantee all transformations, transfers and transports are conducted with certified resources of the rightful owners and locations.

The remainder of this thesis is organised as follows. Section 2 describes the background and evolution of CCs with accompanying concerns (such as common attack vectors), a background of REALISTIC and how we build upon this framework, including the algebraic theory of the framework. Last, we include a glossary containing often-used terms and our nomenclator. Section 3 presents the design and implementation details. Section 4 evaluates the framework and presents the results. Section 5 discusses the research and paradoxical aspects related to CCs and blockchain application in SCM, including future work. In section 6, we conclude the work and results of this thesis.

1.1 Notation

We informally use the term *vector space* (where the underlying structure is a field) even when it formally is a *module* (where the underlying structure is a ring, meaning it is a generalisation of a field not requiring multiplicative commutativity and inverses) because it does not have an impact on the implementation or in terms of modelling.

The TRAZE prototype is available on the public GitHub link:
`https://github.com/smhyatt/TRAZE`

2 Background

According to Hetemäki, Kangas and Peltola, the LE closely couples with climate change and environmental degradation because extraction, processing and consumption of natural resources comprise taking, making and wasting (Hetemäki et al., 2022). Subsequently, this causes ecological harm because the production of goods comes at the cost of the productivity of ecosystems, and this excessive pressure compromises essential ecosystem services, such as water, air and soil cleaning (Michelinia, Moraesa, Cunhab, Costaa, & Omettoa, 2017).

The CE, opposite the LE, seeks to reduce, reuse and recycle. Many definitions exist of what CE entails (Kirchherr, Reike, & Hekkert, 2017); however, in essence, everything we use, spend or consume is transformed, transferred or transported into something, to someone or to somewhere else. Nothing appears or disappears, which is the circular part of the concept (Pearce & Turner, 1990).

Many perceive the CE as a sustainable economic system where economic growth decouples from resource use and, therefore, considers it a necessary step toward climate change mitigation (Corona et al., 2019). However, studies have shown a potential trade-off between the CE and sustainability, as the CE can sometimes be more resource-intense (and thereby emission-intense) than the LE (Corona et al., 2019; Saidani et al., 2022). A solution might, therefore, be tracking and tracing value chains to understand how carbon emissions relate to value chains, i.e. at which steps in value chains do emissions occur, how much emits and how much carbon remains in the final product.

In 2001, the Greenhouse Gas Protocol introduced the concept of Scopes 1-3 of carbon emissions (*The Greenhouse Gas Protocol*, 2004). These Scopes are often used to measure the environmental aspect of the Environmental, Social, and Governance (ESG) criteria, formulated in a landmark study from 2005 entitled *Who Cares Wins*. These measures have, among others, been used as an investment strategy and an encouragement for companies to act responsibly (*Who Cares Wins*, 2005).

- *Scope 1* refers to a company's *direct emissions*, for example, running its boilers and vehicles.

- *Scope 2* refers to the *indirect emissions owned* by the company, for example, purchased electricity, steam, heating, and cooling. These are indirect because the emissions physically occur at the facility where the sources are generated.

- *Scope 3* refers to the *indirect emissions not owned* by the company (and the ones not included in Scope 2). These emissions are consequences of the company's activities from upstream and downstream sources, for instance, business travel, products purchased from suppliers and when customers use products produced or sold by the company (*The Greenhouse Gas Protocol*, 2004). In other words, these emissions include the entire SCN.

To date, most companies manage to account for Scopes 1-2, while Scope 3 remains tricky. In most sectors, Scope 3 emissions make up the majority of a company's inventory, and this lack of direct control and difficulty collecting high-quality data creates a barrier to reducing Scope 3 emissions. Simultaneously, Scope 3 is where the highest emissions occur. For example, in Apple's reported carbon footprint, Scopes 1-2 account for 1% of their emissions, while Scope 3 the remaining 99% (Jackson, 2021). However, it is unlikely that all companies can achieve complete CO2 neutrality, which introduces the next section, Carbon Credits.

2.1 Carbon Credits

On December 11 1997, the Kyoto Protocol[7] (KP) was adopted and entered into force on February 16 2005, with 192 Parties involved. The objective of the KP was to enforce the UN Convention on Climate Change and thereby commit industrialised countries and economies in transition to limit and reduce GHG emissions under agreed individual targets, and while the Parties should primarily meet their targets through national measures; a crucial element of the KP was the establishment of flexible market mechanisms. These mechanisms included International Emissions Trading (IET), Joint Implementation (JI) and Clean Development Mechanism (CDM) and introduced the notion of *compliance markets*, which are markets for trading captured carbon created by the need to comply with a regulatory act.

In short, the targets denoted emission allowances, and if Parties had a surplus of allowances, they could sell the remaining emission levels via IET. The JI mechanism allowed Parties with emission-reduction or emission-limitation commitments (Annex B Parties) to collectively implement emission-reduction or emission-removal projects. The CDM allowed Annex B Parties to implement emission-reduction projects in developing countries based on a model of swapping national debt to protect natural resources. A concept first proposed by Thomas E. Lovejoy in 1984[8].

The idea behind this model was to enable developed countries with high-carbon-based GDPs to assist the growing economies of developing nations, which economically

1. have yet benefitted from high-carbon economic growth,
2. are constrained in GHG emissions and
3. have significant debts to foreign creditors.

Subsequently, carbon became trackable and tradeable like any other commodity, and while the KP ended in December 2020, the mechanisms of the KP became the ancestors of CCs, which have since developed into a cap-and-trade model to reduce carbon emissions. The cap sets a threshold or limit for GHG emissions, which is lowered steadily over time. Putting a price on GHGs moves energy markets toward cheaper renewables and creates an incentive to innovate (Reichle, 2020).

A CC (often called a carbon offset) is a credit for capturing or sequestering GHGs from the atmosphere by a carbon offset project. One CC equals one metric ton of carbon dioxide equivalent gases (CO2-eq) (Reichle, 2020).

[7]UNFCCC: What is the kyoto protocol? [online] `https://unfccc.int/kyoto_protocol` retrieved 01-10-2022.

[8]Aid Debtor Nations' Ecology [online](1984) `https://www.nytimes.com/1984/10/04/opinion/aid-debtor-nations-ecology.html` retrieved 27-09-2022.

2.1.1 Carbon Markets

As briefly mentioned in section 1, carbon markets are markets for trading CCs. The EU Emissions Trading System (EU-ETS) is the first and currently the largest company-based cap-and-trade compliance carbon market, which came into force in 2005. Since the KP, there has been no global replacement for the compliance markets; however, this has given the Voluntary Carbon Market (VCM) momentum, enabling voluntary carbon offset projects where governments, NGOs, businesses and private individuals can buy CCs to compensate for the emissions they generate (Reichle, 2020).

The VCM was first established in 1990 and has doubled since 2017. The VCM operates in parallel outside the compliance markets. Unlike regulated compliance markets, there are no established rules and regulations for the VCM, creating the freedom to experiment, innovate and test new procedures, methodologies and technologies. However, the lack of quality control has caused dubious carbon offset projects and CCs—described more in section 2.1.3.

2.1.2 Carbon Offset Projects

Carbon offset projects can occur worldwide because GHG reduction can contribute to climate change mitigation regardless of location. Therefore, most carbon offset projects have been developed in middle-income countries, such as China, India and Brazil, and less than 1% of projects are in the least developed countries. The projects can include, for example, tree planting, urban forestry and avoided deforestation projects, capturing industrial gases and renewable energy projects such as wind or solar-power (Acampora, Ruini, Pratesi, & Lucchetti, 2022). Appendix 1.1 lists additional carbon offset project types.

Carbon offset projects require approved methodologies, standards and monitoring systems that prove and measure emissions capturing to allow trading CCs. A methodology defines specifications and operations required, such as how to handle processes, for calculating emission capturing during the lifespan of a carbon offset project.

Carbon offset standards are used to certify and issue CCs within a set of defined principles and procedures. Different standards exist, such as Gold Standard, CDM, Voluntary Carbon Standard (Verra) and Climate Action Reserve, which each provide their procedures for certifying CCs for non-governmental carbon offset projects, some more rigorous than others (Acampora et al., 2022).

Carbon offset standards and certifications are necessary because many scams exist for greenwashing. By certifying CCs, companies buying them can ensure their existence and value (Acampora et al., 2022; *What are carbon markets?*, 2009). However, there are some requirements and concerns related to CCs:

- *Permanence* addresses whether the CC investment remains intact long enough to account for the carbon capture.
- *Non-permanence*, or *reversals*, concerns the possibility that the captured carbon can be reversed through, for instance, fires or pest outbreaks, which can be inten-

tional and unintentional by human activities or natural disasters.

- *Leakage* addresses any increased emissions or decreased removals not accounted for within and outside the scope of a project.

- *Additionality* addresses the originality of the carbon offset project, meaning the captured carbon must be carbon that would not have been captured otherwise.

- *Double counting* (or *double spending*) concerns multiple parties claiming the same CC; for instance, a trader sells the same CC to multiple buyers or overlapping land used to account for the same carbon capture.

While the requirements of permanence, additionality and leakage provide insurance for the CC investment, they are challenging to guarantee. Different standards have different risk mitigation measures to avoid socio-economic or environmental trade-offs, for instance, resettlement, adverse impacts on ecosystems/biodiversity and inadequate benefit sharing (COWI & IEEP, 2020).

2.1.3 Cases of Carbon Credit Misconduct

Greenwashing is a known issue with many large fossil fuel companies. They do so by advertising their ongoing contribution to the climate crisis. Saudi Arabia's Aramco is one of them, claiming to conduct business that addresses climate change. However, they are the world's largest GHG emitter and plan to continue doing so. Within Scopes 1-2, Aramco abides by the Greenhouse Gas Protocol guidelines. However, according to files published by environmental lawyers ClientEarth[9], Aramco contributes to 4.38% of the world's historical carbon emissions, and they are estimated to produce 27 billion tonnes of CO2 between 2018 and 2030.

In another case of greenwashing, a Bloomberg Green analysis[10] using more than 215,000 CC transactions from public datasets over the past decade reveals that many global brands (airlines, online retailers, industrial firms and energy producers) rely on cheap and questionable CCs as a claim of their carbon neutrality.

Many REDD+ projects have questionable legal additionality. REDD+ is a framework that *reduces emissions from deforestation and degradation*, created by the UNFCCC Conference of the Parties (COP) as a guide for the forest sector to reduce emissions, deforestation and forest degradation and to increase sustainability, reforestation and conservation[11]. In a 2015 French Research Centre report, Simonet et al. (2015) analysed 410 REDD+ projects discovering that 37% spatially overlapped either partially or totally with protected lands such as national parks.

Studies have identified and analysed challenges in meeting leakage requirements; an analysis from 2009 estimated up to 90% leakage in some carbon offset projects (Montserrat

[9] The Greenwashing Files [online] `https://www.clientearth.org/projects/the-greenwashing-files/` retrieved 11-11-2022.

[10] Junk Carbon Offsets [online](2022) `https://www.bloomberg.com/graphics/2022-carbon-offsets-renewable-energy/` retrieved 03-12-2022.

[11] What is REDD+? UNFCCC [online] `https://unfccc.int/topics/land-use/workstreams/redd/what-is-redd` retrieved 23-10-2022.

& Sohngen, 2009). According to Foss (2018), Norway is no exception. In a report from 2018 by Norway's Office of the Auditor General, results showed that despite Norway's collaboration with REDD+ and efforts to reduce logging, the stop of logging in one area led to an increase in logging elsewhere.

Indonesia has endured multiple issues with CC reliability. One such is double counting, where they have addressed illegal self-declared carbon offset projects of national parks, such as the Sebangau National Park[12]. These projects were cancelled because they caused double counting of Indonesia's NDC target[13], proving the necessity of strict carbon certification regulations (Friess, Howard, Huxham, Macreadie, & Ross, 2022).

Another issue is value estimation and issuance of CCs where, according to Nikkei in 2017[14], one of the largest forest preservation projects, the Katingan Mentaya Project, issued credits up to three times more than the amount of CO2 it is likely to absorb. The project estimates to reduce CO2 emissions by an annual average of 7.45 million tons CO2-eq over 60 years. It has since 2017 issued voluntary credits equivalent to 30 million tons of CO2-eq with a profit of $210 million USD. Nikkei discovered the overestimation by analysing satellite data; however, according to a paper by Mertz et al. (2017), estimating carbon capturing is unlikely to be a cost-effective strategy. In their research, they demonstrated, by analysing satellite data, that additionality of REDD+ projects in fast-developing areas is complex.

In the same article, Nikkei also stated that the Indonesian government introduced a moratorium, which caused redundancy of one-third of the project's anti-deforestation efforts. Even so, the CC value estimation was not regulated accordingly.

Providing permanence is challenging, and an example of this is the many forest fires that Indonesia experiences. In 2015, 9000 hectares of Katingan Mentaya burned, and despite many efforts to help prevent forest fires in Katingan Mentaya, 1900 hectares burned in 2019[15]. ECMWF[16] estimated that the forest fires in 2019 caused 708 megatonnes of CO2 emissions and, according to Nurbaya, Efransjah, Murniningtyas, Erwinsyah, and Muttaqin (2022), 857.755 hectares burned, making it the worst fire since 2015. Nevertheless, actions are taken to accommodate permanence. Various studies test and implement mechanisms to mitigate the risk of reversal, such as insurance, buffer accounts, incremental crediting over time and risk management, such as risk diversification (Galik et al., 2014).

[12] Indonesian ministry cancels self-declared carbon projects [online](2021) `https://foresthints.news/indonesian-ministry-cancels-self-declared-carbon-projects-to-avoid-illegalities` retrieved 10-11-2022.

[13] NDCs are where countries set targets for mitigating the GHG emissions that cause climate change and for adapting to climate impacts.

[14] Indonesian carbon credit project appears to betray its purpose [online](2021) `https://asia.nikkei.com/Spotlight/Environment/Climate-Change/Indonesian-carbon-credit-project-appears-to-betray-its-purpose` retrieved 10-11-2022

[15] REDD Monitor [online](2021) `https://redd-monitor.org/2019/12/12/indonesias-katingan-redd-project-sells-carbon-credits-to-shell-but-that-doesnt-mean-the-forest-is-protected-its-threatened-by-land-conflicts-fires-and-a-palm-oil-plantation/` retrieved 13-11-2022.

[16] Copernicus: A year in fire [online](2019) `https://atmosphere.copernicus.eu/copernicus-year-fire` retrieved 13-11-2022

2.2 REALISTIC

As presented in section 1, the recent paper by Bager, Düdder, et al. (2022) introduce REALISTIC, an event-based modelling framework for SCNs, as an extension of McCarthy's REA accounting model. REALISTIC builds on the same concepts of Resources, Events, and Agents, and additionally the concepts of Locations, Information, Strategies, Time, Identities and Contracts.

The concept of REALISTIC is a descriptive model of the CE to account for the events of SCNs. These events include resource-producing, -consuming, -transporting, -transforming and -transferring. The idea is to include the provenance of all resources, which encompasses the complete history and information of resources to facilitate privacy-preserving data forensics across an ecosystem of independent agents. The framework includes *secure transformations*, which ensures that certified resources cannot exist from non-certified resources across SCNs, meaning that nothing can exist or be produced out of nothing, and all resources are certified and tracked.

Their case study is an end-to-end coffee SCN, of which they have implemented a prototype to model the SCN as a proof-of-concept. Value chains are part of the agro-food system, which involves hundreds and thousands of companies and suppliers. The products are mixed, sold, packaged, resold and repackaged many times before reaching consumers. SCNs rely on trust, which is challenging to achieve since the entities are typically distributed across countries, and fraudulent activities are not uncommon. Providing full transparency and guaranteeing provenance would require segregated supply chains, which is why, for instance, the final coffee blend consists of coffee from several countries and hundreds of different producers. At the same time, the agro-food system is the second largest GHG emitter and the primary driver of global biodiversity loss (Bager, Düdder, et al., 2022).

By utilising blockchain and distributed ledger technology (BC/DLT), the authors provide a framework with distributed databases, decentralised governance, tamper-proof recording, high availability and non-copyable digital assets, making it possible to:

1. track and trace full SCNs and
2. guarantee no double spending of resources (and CCs).

All events can be traced back to agents because they have to digitally sign and cryptographically commit to events in the immutable database and are thereby accountable.

The concepts of REALISTIC comprise the following

- *Resources* are one or more ideal, virtual or real effects (money, bonds, apples) that can neither be duplicated nor discarded.
- *Events* are digitally signed and dated statements of virtual or real happenings, often accompanied by evidence of the happening. Some types of events include:
    1. *Resource transfers*: resource ownership transfers among two agents.
    2. *Resource transportations*: altering the location of virtual or real resources.

3. *Resource transformations*: one or more resources transforming into something else or bundling resources together, such as packaging multiple items together.
4. *Information transmissions*: passing information from one agent to another.
5. *Observations*: recording statements of ascertainments made by the agent.

- *Agents* are entities that can register and sign events, such as humans, legal entities, corporate divisions, associations and IoT devices. Agents sign with a key accompanied by a geo- and timestamp.
- *Locations* are resources' virtual or real places of residence or activity.
- *Information* represents data, such as instructions, manuals and invoices.
- *Strategies* are agents' manual or automatic processes that generate or sign events.
- *Time* represents both the geographic location and time of events.
- *Identities* are identifiers for all entities, such as agents, resources and events, helpful in accessing and tracking the entities.
- *Contracts* are agreements between two or more agents that specify future events and what a successful execution constitutes.

2.3 The TRAZE Framework

By extending the work by Bager, Düdder, et al., described in section 2.2, we have implemented a framework consisting of the REALISTIC concepts as a carbon accounting system that tracks all resources and events of SCNs to make it possible to trace carbon emissions and reveal greenwashing and fraudulent activities.

2.3.1 Resources, Agents and Carbon Credits

Resources entail the same concept Bager, Düdder, et al. proposed, meaning carbon is included as a resource. Likewise, agents also entail what Bager, Düdder, et al. proposed. However, in this framework, the atmosphere and soil also count as agents, even though they cannot digitally sign for themselves. Had the atmosphere and soil been able to sign for themselves, the world might not be facing the current climate problems. However, by "forcing" agents to digitally sign their events, it establishes an accounting premise to hold agents accountable for their events.

All resources are of a specific type, and, as mentioned, we define CCs as a type of resource. CCs are issued with carbon credit certificates, and the certificates are registered using a timestamp, and a sophisticated geographic area detector, which means that an issued CC certificate accounts for an estimated carbon capturing of a specific area within a specific timespan (i.e. one year). The CC retires when the amount of carbon it has captured is released into the atmosphere by the CC owner and the owner uses the CC to pay for the emission. After which, the CC is out of scope for more usage. Section 3.3 elaborates on the implementation in more detail.

2.3.2 The Architecture

The event-based architecture, where every event is recorded and digitally signed, enables the accounting premise, which we can use for post hoc forensics analysis, to which we can trace entire SCNs, including the provenance of all resources, track outliers and analyse coherence in the data.

This design ensures that even if agents commit fraud by, for instance, not registering a transfer of carbon to the atmosphere, the system will not force the transfer to occur. However, it will keep track of the carbon retained by the agent such that all fraudulent or regular activities can always be traced back to the responsible agent.

The global invariant ensures that the amount of outgoing resources is the same as incoming resources, making the framework a descriptive version of the CE, as with REALISTIC. For example, an agent cannot transform 100 kg of apples into 1 kg of pudding. The total mass cannot change; instead, it is transformed into something else, such as 1 kg of pudding and 99 kg of apple waste.

The framework's traceability enables accounting for upstream and downstream activities and carbon emissions, which can help shed light on Scopes 1-3. Seeing that the framework provides provenance, we can implement models for tracking:

1. what carbon is transferred, transformed or transported among agents,
2. what carbon resides captured in resources (such as in products) and
3. what carbon is released or captured into or from the atmosphere.

2.3.3 Attack Vectors

In light of the above concepts and architecture, we identify attack vectors and how the framework prevents such attacks.

Greenwashing

Attack: Fraudulently claiming to reduce emissions or act environmentally friendly.

Example: Chuck's Green Trucks claim to be CO2 neutral because they buy cheap CCs to compensate for their emissions. However, the CCs they buy are "certified" by a non-credible CC certification source. At Chuck's Green Trucks, they know the CCs are questionable; however, they think this will go undetected.

Prevention: TRAZE includes certifications with evidence of third-party verification, ensuring that the CCs are trackable and their value is registered. This transparency makes it easy to detect the legality of CCs.

Double Counting

Attack: Selling the same CC to multiple buyers.

Example: Malicient Incorporated bought a couple of CCs and decided to try and sell the same CCs to multiple buyers to gain more profits.

Prevention: TRAZE tracks ownership of items, ensuring that a transfer of ownership occurs only once from the rightful owner. Transferring ownership of resources not owned by the transferring agent is prohibited. The same applies for transports and transformations.

Attack: Creating CC certificatess in overlapping areas within the same timespan (i.o.w. creating more of either fully or partially the same CC certificate).

Example: Farmer Lucious heard that selling CCs is a profitable business, so he decided to try to create multiple CC certificates on partially and fully overlapping parts of his land to gain more profits from the same areas.

Prevention: When creating CC certificates, TRAZE requires a timespan and specific coordinates to detect the location and area in question. With this data, TRAZE prevents agents from creating temporal and spatial overlapping CC certificates.

Attack: Re-spending retired CCs.

Example: At Sybil's Shipping Service, they have spent all their CCs, and they need an extra CC to complete a vital shipment from China. So Sybil's Shipping Service decided to try and re-spend some of the CCs they have already spent–thinking this could solve their problem.

Prevention: Once the CC certificate is fully used and thereby retired, it is no longer available, making it unattainable and unable to be re-spent.

Leakage

Attack: Increased emissions not accounted for outside the scope of a project.
Example: Farmer Mallory also heard that selling CCs is a profitable business, so Farmer Mallory, who owns 20.000-hectare forest land, decides to create CCs based on parts of her land, intending to log more than usual on the remaining parts of her land.

Attack: Increased emissions not accounted for within the scope of a project.

Example: As part of a nature-preserving carbon offset project, a national park is registered and sold as CC certificates. During the period of the CC certificate, a fire breaks out in the national park, causing tremendous emissions that are not accounted for in the CC certification.

Attack: Decreased removals not accounted for within the scope of a project.

Example: An Annex B country plans to reduce logging as part of their steps towards the PA goals. In doing so, several projects are established and set out to reach these goals. However, one of the projects went wrong, and the reduced logging did not occur as planned. The actor responsible for the project reports that the logging had decreased when it was unchanged.

Attack: Decreased removals not accounted for outside the scope of a project.

Example: As with the leakage in Norway example from section 2.1.3, say an Annex B country plans to reduce logging and successfully does so in areas with much logging. Due to the high demand for timber, the stop of logging in some areas leads to increases in logging elsewhere.

Prevention: The same applies in all four leakage cases: since TRAZE is a framework for total accounting, all land, resources and time are accounted for continuously. Had it not, a gap in either time or an area would open an opportunity for leakage. In total accounting, checking the net effect of all agents is a simple task, and any unchanged or decreased carbon capture will be noticeable. In addition, by providing evidence for third-party verification of certificates, an agent (representing the CC certificate standards as mentioned in section 2.1.2) authorises, measures and estimates a project before and after the project starts and ends, making computing the net effect plausible, and the authorising agent accountable.

Additionality

Attack: The carbon offset project does not capture carbon that would not have been captured otherwise.

Example: A country owning a preserved and protected 1000-year-old national park decides to utilise it and create CCs as part of its preservation.

Prevention: Accounting for leakage helps to account for additionality as well because the total accounting of TRAZE makes the net effect for each agent visible. In this example, if the country sells the carbon capture created by preserving the national park, this carbon capture will no longer count as part of the country's carbon reduction targets.

Fraud

Attack: Selling fake or uncertified resources.

Example: Trudy owns a business selling illegal goods, such as elephant's ivory, coffee and timber. In TRAZE, Trudy registers the items under either fake pseudo names or using fake evidence papers.

Prevention: The event-based and descriptive architecture of TRAZE does not prevent Trudy from storing fake data; it does, however, make it evident that she is the source of these particular illegal goods if someone were to look into it.

Attack: Transforming resources from illegal or uncertified resources.

Example: Craig wants to build a shed for his garden, but certified lumber is expensive, so Craig buys lumber from an illegal and unregistered supplier. When registering the new shed, Craig registers 5 kilograms of certified lumber to TRAZE, although Craig built the shed using 200 kilograms of lumber.

Prevention: When Craig registers the transformation event of building the shed to TRAZE, TRAZE keeps track of an invariant, ensuring that the weight of resources coming in must equal the resources going out. However, Craig could enter a fake total weight of the shed, but by doing so, it will be stored in TRAZE, and forensics could reveal this mischievous event.

2.4 Algebraic Framework

In this section, we describe the algebraic framework for resource accounting. The mathematical framework is clearly presented and proved in J. García Hébert's thesis (Hébert, 2020) which this work is based on. It allows for modelling of *resources, events,* and *ownership states*. Events encompass *transfers, transports,* and *transformations* which can be applied to a state.

2.4.1 Resources

The mathematical framework is built on the theory of infinite vector spaces. To define resources and events within the framework we must first define vector spaces, the coproduct and both their properties.

Definition 1 (Vector space) *A vector space over a field K is a set of vectors V, the addition $(+)$ and multiplication $(\cdot)$ operators, and a designated 0 element such that $\forall a, b \in K$ and $\forall u, v, w \in V$ we have:*

Associative property. $u + (v + w) = (u + v) + w$

Commutative property. $u + v = v + u$

Identity element, $(+)$. $u + 0 = u$

Inverse element. $\forall u \in V$. $\exists$-$u \in V$ *such that* $u + (-u) = 0$

Identity element, $(\cdot)$. $1 \cdot u = u$

Field & scalar mult. $a \cdot (b \cdot u) = (a \cdot b) \cdot u$

Distributive properties. $a \cdot (u + v) = (a \cdot u) + (a \cdot v)$

$$(a + b) \cdot u = (a \cdot u) + (b \cdot u)$$

In this thesis, the only field used is the real numbers, such that $K = \mathbb{R}$.

Definition 2 (Support) *Let X be a set, V be a vector space and f a dependent function such that $f : X \to V_{x\in X}$. Then the support of f is $Supp(f) = \{x \in X \mid f(x) \neq 0_V\}$.*

Definition 3 (Coproduct) *Let X be a set and let $V_{x\in X}$ be a class of vector spaces indexed by X over a field K. Then the dependent functions $f : X \to V_{x\in X}$ with finite support are the coproduct and is denoted $\coprod_{x\in X} V_x$.*

$\forall a \in K, x \in X$ and $f, g \in \coprod_{x\in X} V_x$ we have the following definitions:

***Function application distributes over** $(+)$.* $(f + g)(x) = f(x) + g(x)$

***Function application distributes over** $(\cdot)$.* $(a \cdot f)(x) = a \cdot f(x)$

***Function application distributes over** $(-)$.* $(-f)(x) = -(f(x))$

Zero function. $0(x) = 0$

These functions f of the coproduct are finite maps with a default return value of 0.

Definition 4 (Copower) *If $\forall x, y \in X.\ V_x = V_y$ then instead of $\coprod_{x\in X} V_x$ we write $\coprod_X V$. This is the copower.*

Both coproducts and, the specialised versions, copowers are vector spaces.

Definition 5 (Resources) *Let X be a set. We call it the resource types. Then the set of resources R is the copower of $\mathbb{R}$ by X, $R = \coprod_X \mathbb{R}$. In other words, R is the set of finite maps from X to $\mathbb{R}$.*

Example. Let the resource types of the system be $X = \{USD,\ WoodenPlank,\ Screw\}$ then some example resources are:

$$r_1 := 0$$

$$r_2 := 11 \cdot USD$$

$$r_3 := 20.5 \cdot WoodenPlank - 10 \cdot USD$$

r_1 is the zero resource. r_2 is a simple resource with only one resource type. r_3 is a compound resource. One can ask how many *Screw*s are in the listed resources, and the intuitive answer is zero. A resource, an element of a copower, has some amount of every resource type, but oftentimes this amount is 0, and qua the notation we do not write it out. The number of elements with 0 amount can be infinite. We only write out the amounts of resource types which are in the finite support, like *USD* in r_2 and r_3.

2.4.2 Ownership States and Transfers

With the resources definition in hand, we can define ownership states. Informally, an ownership state shows who owns what, which agent owns what resources, at a specific time point. Formally, it is defined as follows:

Definition 6 (Ownership states) *Let A be a set. We call it the agents. Let R be a vector space. Then the set of ownership states O is the copower of R by A, $O = \coprod_A R$. O is the set of finite maps from A to R.*

Example. Let the resource types be $X = \{USD,\ WoodenPlank,\ Screw\}$ and let the agents be $A = \{Alice, Bank, Bob\}$. Some example ownership states are:

$$o_1 := \{\} = 0$$

$$o_2 := \{Alice \mapsto 11 \cdot USD\}$$

$$o_3 := \{Bob \mapsto 20.5 \cdot WoodenPlank - 10 \cdot USD,\ Alice \mapsto 5 \cdot Screw\}$$

$$o_4 := \{Bank \mapsto -500 \cdot USD,\ Alice \mapsto 350 \cdot USD,\ Bob \mapsto 150 \cdot USD\}$$

To update an ownership state a transfer is applied. A transfer moves ownership of resources between agents. Because we in our definition of ownership states allow negative amounts of resources we can define transfers T as a subspace of ownership states O, meaning $T \subseteq O$ and that T is closed under $0, (+), (\cdot)$, and $(-)$. To formally define transfers we need to recall linear maps and kernels.

Definition 7 (Linear maps) *Let V and W be vector spaces. Functions $f : V \to W$ distributing over addition and scalar multiplication are called linear maps and are written as $f : V \to_1 W$.*

Definition 8 (Image & kernel) *Let V and R be vector spaces over $\mathbb{R}$, and function f a linear map, $f : V \to_1 W$. The image of f is $\mathtt{im}\ f = \{\, f(v) \mid v \in V\,\}$, and the kernel of f is $\mathtt{ker}\ f = \{\, v \in V \mid f(v) = 0\,\}$.*

Definition 9 (Transfers) *Let A be a set. We call it the agents. Let the linear map $\mathtt{sum}$ from ownership states O to resources R be defined such that $\forall o \in O$:*

$$\mathtt{sum}(o) = \sum_{a \in Supp(o)} o(a) \tag{2.1}$$

Then the set of transfers T is the kernel of $\mathtt{sum}$. That is, all the ownership states $o \in O$ where $\mathtt{sum}$ applied to them returns 0:

$$T = \mathtt{ker}\ \mathtt{sum} = \{o \mid \mathtt{sum}(o) = 0\} \tag{2.2}$$

This definition makes the sum total of transfers 0, meaning the algebra ensures we have *resource preservation* built into the model. Resource preservation ensured by transfers means the application of a transfer to an ownership state results in a new ownership state containing the same amount of resources. The resources cannot disappear.

Example. Let the resource types be $X = \{USD,\ WoodenPlank\}$ and let the agents be $A = \{Alice,\ Bob,\ Charlie\}$, then some example transfers are:

$$t_1 := \{\} = 0$$

$$t_2 := \{Alice \mapsto 11 \cdot USD,\ Bob \mapsto -11 \cdot USD\}$$

$$t_3 := \{Bob \mapsto 2 \cdot WoodenPlank - 10 \cdot USD,\ Alice \mapsto -2 \cdot WoodenPlank + 10 \cdot USD\}$$

$$t_4 := \{Alice \mapsto 10 \cdot USD,\ Bob \mapsto -5 \cdot USD,\ Charlie \mapsto -5 \cdot USD\}$$

t_1 is the zero transfer. t_2 is a 2-party transfer from Bob to Alice. t_3 is a 2-party transfer where resources are transferred both from Bob to Alice and from Alice to Bob. t_4 is a multi-party transfer.

We only want a transfer to be applied to an ownership state if the agents own the resources. To handle this we need a resource manager.

2.4.3 Resource Managers

A transfer is only applied to an ownership state when the agents are ensured to obey their *credit limit*.

Credit limit. Each agent must obey its credit limit by not transferring more resources than the limit allows. An agent simply cannot transfer resources they do not have ownership status of.

The credit limit is a function $c : A \to X \to \mathbb{R}$, and the *credit limit policy* is a predicate P on ownership states classifying them as valid or invalid. Only agents with a credit limit, denoted A', must respect the policy. So we have that $\forall a \in A' \subseteq A$ and $\forall x \in X$:

$$P(o) = o(a)(x) \geq c(a)(x) \tag{2.3}$$

A *resource manager* is a service with an internal ownership state o which satisfies P and when given a transfer t the responsibility to update o with a new value such that $o \leftarrow o + t$ and return `Success` if $P(o + t)$ is valid and if invalid return `Failure`.

Example. If a resource manager, m, implements the credit limit of 0 for all resource types and agents, meaning no agents are allowed negative amounts of resources. If its initial ownership state is o_1 as shown below and the transfers t_1 and t_2 are received, then $P(o_1 + t_1)$ will return `Failure`, because Alice does not own 100 screws (only 20), and the new state is invalid. $P(o_1 + t_2)$ is valid and the resulting state is o_2.

$$o_1 := \{Alice \mapsto 3 \cdot USD + 20 \cdot Screw,\ Bob \mapsto 100 \cdot USD\}$$

$$t_1 := \{Alice \mapsto -100 \cdot Screw,\ Bob \mapsto 100 \cdot Screw\}$$

$$t_2 := \{Alice \mapsto 5 \cdot USD - 11 \cdot Screw,\ Bob \mapsto -5 \cdot USD + 11 \cdot Screw\}$$

$$o_2 := \{Alice \mapsto 8 \cdot USD + 9 \cdot Screw,\ Bob \mapsto 95 \cdot USD + 11 \cdot Screw\}$$

Zero-balance ownership states. Hébert shows how an ownership state is decomposed into a tuple of a resource balance and a transfer, (r_b, t). This means, without loss of generality, the ownership state can be maintained as a transfer in the resource manager. We need to extend the set of agents A with a designated virtual agent, lets call it *bank* such that $A = A \cup \{\, bank \,\}$, and let *bank* be the owner of the resource balance and the transferor of t, such that t is a transfer from *bank* to the other agents. This means *bank* holds the negative sum of the resources that all the original agents own, and is not expected to obey credit limits. The resource manager is "born" with the balance which is invariant. If a resource manager is born with the total amount of resources $2000 \cdot USD + 20 \cdot Screw$, then o_1, from the example above, can be decomposed into the tuple (r_b, t) as follows:

$$o_1 := (2000 \cdot USD + 20 \cdot Screw, \\ \{bank \mapsto -103 \cdot USD - 20 \cdot Screw,\ Alice \mapsto 3 \cdot USD + 20 \cdot Screw,\ Bob \mapsto 100 \cdot USD\})$$

This means the balance r_b can be saved in a central resource manager, m_c, and that all resource managers' internal state is a transfer only. We call these states *zero-balance ownership states.*

For example, now m has zero-balance ownership state t and m_c has t_c:

$$t := \{\, bank \mapsto -103 \cdot USD - 20 \cdot Screw,\ Alice \mapsto 3 \cdot USD + 20 \cdot Screw,\ Bob \mapsto 100 \cdot USD \,\}$$
$$t_c := \{\, bank_c \mapsto -2000 \cdot USD - 20 \cdot Screw,\ bank \mapsto 2000 \cdot USD + 20 \cdot Screw \,\}$$

2.4.4 Location States and Transports

A location is specified by its coordinate set of longitude and latitude. What a location is interpreted to be in the real world is a physical location or a virtual location, i.e. in storage or on a server. A transport moves resources from one location to another. Thus, a transport can be applied to a location state to calculate a new location state, like a transfer can be applied to an ownership state to gain a new ownership state. The definitions are alike, as seen below.

Definition 10 (Location states) *Let L be a set. We call it locations. Let R be the resources vector space. Then the set of location states Z is the copower of R by L, $Z = \coprod_L R$. In other words Z is the set of finite maps from L to R.*

Definition 11 (Transport) *Let L be a set. We call it locations. Let the linear map* `sumLoc` *from location states Z to resources R be define such that $\forall z \in Z$:*

$$\texttt{sumLoc}(z) = \sum_{l \in Supp(z)} z(l) \tag{2.4}$$

Then the set of transports D is the kernel of `sumLoc`*. That is, all the location states $z \in Z$ where* `sumLoc` *applied to them returns 0:*

$$D = \texttt{ker}\ \texttt{sumLoc} = \{\, z \mid \texttt{sumLoc}(z) = 0 \,\} \tag{2.5}$$

Resource preservation is ensured by transports, because the sum total of transports is 0. A resource manager can maintain an internal location state and apply transports to it. The state can be represented as a zero-balance ownership state, but being a transport instead of a transfer. This is possible by extending L by an abstract location, a virtual designated location, that allows negative amounts of resources placed there.

2.4.5 Transformations

Resource preservation is a cornerstone of the framework; the transformation of resources r_1 into resources r_2 guarantees *nothing disappears*; it gets turned into *something else.* Because of this, the sum total of resources is not guaranteed to be 0, but resource preservation across transformations is guaranteed nonetheless. To show this, we recall the definition of valuation maps.

Definition 12 (Valuation maps) *Let V be a vector space over $\mathbb{R}$. Let w be a linear map from resources R to V, being $w : R \to_1 V$. w is a valuation map over R into V.*

Definition 13 (Transformations) *Let V be a vector space over $\mathbb{R}$ and let $w : R \to_1 V$ be a valuation map over R into V. The set of transformations P_w with respect to w is well-defined as it is the set of resources $r \in R$ where w applied to them returns 0:*

$$P_w = \{\, r \mid w(r) = 0 \,\} \tag{2.6}$$

So a transformation under valuation map w is an element of the kernel space of w, and thus a vector space. The valuations can, for example, map from resources to a vector space over $\mathbb{R}$ where the real can represent mass, energy, or another suitable unit.

Example. If we have the following valuation map w, then r is a transformation p_w. This is clear since $w(r)$ is the zero vector.

$$\begin{aligned}
w := \{\, & 1 \cdot Screw \to 10\,g + 0.01\ CO_2\text{-}eq, \\
& 1 \cdot WoodenPlank \to 2000\,g + 0.1\ CO_2\text{-}eq, \\
& 1 \cdot TableLeg \to 1000\,g + 0.05\ CO_2\text{-}eq, \\
& 1 \cdot Table \to 16200\,g + 1\ CO_2\text{-}eq \,\}
\end{aligned}$$

$$p_w := r = -20 \cdot Screw\ - 6 \cdot WoodenPlank\ - 4 \cdot TableLeg\ + 1 \cdot Table$$

$$\begin{aligned}
w(r) :=\ & -20 * (10\,g + 0.01\ CO_2\text{-}eq) \\
& - 6 * (2000\,g + 0.1\ CO_2\text{-}eq) \\
& - 4 * (1000\,g + 0.05\ CO_2\text{-}eq) \\
& + 1 * (16200\,g + 1\ CO_2\text{-}eq) \\
=\ & -200\,g - 0.2\ CO_2\text{-}eq \\
& - 12000\,g - 0.6\ CO_2\text{-}eq \\
& - 4000\,g - 0.2\ CO_2\text{-}eq \\
& + 16200\,g + 1\ CO_2\text{-}eq \\
=\ & 0\,g + 0\ CO_2\text{-}eq = 0
\end{aligned}$$

2.5 Coffee Supply Chain Network

To demonstrate how TRAZE can be used in practice, we apply it to an example SCN, being the Columbian coffee SCN described in detail in Bager, Düdder, et al., 2022 and Hébert, 2020, through the lens of the algebraic framework (2.4). Note that the numbers regarding CO_2-eq and conversion between grams and CO_2-eq are made up for the sake of example.

2.5.1 Coffee SCN Described in the Algebraic Framework

The following definitions and sets are required:

- The necessary **agents** A are *Farmer, Cooperative, Almacafé* and *Roaster*, and since this thesis' focus is on the use of CCs, the following two agents are also included: *Atmosphere* and a CC certificate issuer called *CCCIssuer*. An agent in the agents set is identified by its name as a string.

Example. The following are all agents: "FarmerA"; "FarmerB"; "Atmosphere".

- To define the resource types we need the **product types** being the set $K = C \cup G$ where

$$
\begin{aligned}
C = \{ & \text{ 1 g of Coffee cherry,} \\
& \text{1 g of Wet parchment,} \\
& \text{1 g of Dry parchment,} \\
& \text{1 g of Green coffee,} \\
& \text{1 g of Roasted coffee,} \\
& \text{1 g of GHG,} \\
& \text{1 g of Unaccounted } \} \\
G = \{ & \text{ 10 CO2-eq Carbon Credit Certificate } \}
\end{aligned}
$$

  The *Carbon Credit Certificate* product type, lets call it g, is a specific nonfungible certificate, i.e. from a carbon offset project in Minas Gerais, that can be used to "pay" for emitting 10 CO2-eq. *Unaccounted* is for lossy transformations, and encompas everything unknown[17]. Note there is a total order on C limiting the set of valid transformations:

$$
\begin{aligned}
&\textit{Coffee cherry} < \textit{Wet parchment} < \textit{Dry parchment} < \textit{Green coffee} < \textit{Roasted coffee} \\
&\qquad < \textit{GHG} < \textit{Unaccounted}
\end{aligned}
$$

- The set of **resource types** is $X = Y \cup G$ where $Y = C * \mathbb{N}$. The first entry of Y is the product's type and the second is an id uniquely identifying a specific product. We do not have multiple *Unaccounted* resource types, so the id for *Unaccouted* is always 1.

- The space of **resources** R is then defined as $R = \coprod_X \mathbb{R}$.

- The set of **locations** L are the coordinates of the farms, purchasing points, dry millers, warehouses and roasters.

The following functions are defined:

- A credit limit function $c : A \to X \to \mathbb{R}$ returning the credit limit of a resource type for an agent.

- A predicate $p_{cherry} : A \to \{1, 0\}$ on agents to decide if they can produce coffee cherries "out of nothing" (harvesting). Some resource manager holds this credit limit policy.

$$
p_{cherry} = \begin{cases} 1 & \text{if agent } a \text{ is farmer,} \\ 0 & \text{otherwise.} \end{cases} \tag{2.7}
$$

[17]i.e. the transformation of 200 g water and 5 g tea leaves into 180 g tea, where 25 g is unaccounted for. It might be steam or spilled but we do not exactly know.

- Ownership state predicate $p : O \rightarrow \{1, 0\}$:

$$p(o) = \begin{cases} 1 & \text{if } \forall a \in A. \forall x \in X \text{ we have } o(a)(x) \geq c(a)(x) \lor ((x \text{ is Coffee cherry}) \land p_{cherry}(a)), \\ 0 & \text{otherwise.} \end{cases} \tag{2.8}$$

- Valuation function $w : R \rightarrow_1 \mathbb{R}^2$ which returns a 2D vector with the first entry being the total mass of the resource, and the second the total CO_2-eq of the resource. We have $\forall r \in R$:

$$w(r) = \{ \sum_{y \in Y} r(y), \ \sum_{y \in Y} w_{coffee}(r, y) \ \}$$

Where we use the function $w_{coffee} : R * Y \rightarrow \mathbb{R}$ to get the CO_2-eq from the mass of the resource. We are aware that 1 g of GHG's worth in CO2-eq depends on which GHGs it is comprised of, but for the sake of example we choose *some* number.

$$w_{coffee}(r, y) = \begin{cases} r(y) * 0.00012 \, \frac{CO_2\text{-}eq}{g} & \text{if } y \in Y \setminus \{(\text{Roasted coffee}, \_), (\text{GHG}, \_), (\text{Unaccounted}, 1)\}, \\ r(y) * 0.00010 \, \frac{CO_2\text{-}eq}{g} & \text{if } y = (\text{Roasted coffee}, \_), \\ r(y) * 0.00110 \, \frac{CO_2\text{-}eq}{g} & \text{if } y \in \{(\text{GHG}, \_), (\text{Unaccounted}, 1)\} \end{cases}$$

- There exists only some valid **transformations** as listed below (2.5.1). All of them have a potential loss, which is omitted from the listing, but would be captured using the *Unaccounted* resource type. A transformation predicate $q_p : P_w \rightarrow \{0, 1\}$ maps to 1 if P_w is one of the valid transformations. To make it simple the first three listed transformations maps one gram of a resource type to one gram of another resource type. The roasting of green coffee transforms it into roasted coffee and GHGs, such that 98% is transformed into roasted coffee and 2% into GHGs.

  Valid transformations:

Wet milling. Coffee cherry $\rightarrow$ Wet parchment

Drying & fermenting. Wet parchment $\rightarrow$ Dry parchment

Dry milling. Dry parchment $\rightarrow$ Green coffee

Roasting. Green coffee $\rightarrow$ 98% Roasted coffee, 2% GHG

Example. An example transformation is:

$$Roasting_w := -\,500 \cdot \text{Green coffee} + 490 \cdot \text{Roasted coffee} + 10 \cdot \text{GHG}$$

$$\begin{aligned} w(Roasting_w) = \{ & -500g + 490g + 10g = \ 0g = 0, \\ & -500g * 0.00012 \frac{CO_2\text{-}eq}{g} + 490g * 0.00010 \frac{CO_2\text{-}eq}{g} + 10g * 0.0011 \frac{CO_2\text{-}eq}{g} \\ & = -0.06 \ CO_2\text{-}eq + 0.049 \ CO_2\text{-}eq + 0.011 \ CO_2\text{-}eq \\ & = \ 0 \ CO_2\text{-}eq = 0\} \end{aligned}$$

- A transfer predicate $q_t : T \to \{0, 1\}$ that returns 0 if *Unaccounted* is part of the transfer.

With these definitions the farmers have an infinite credit limit with respect to *Coffee cherries* where each farmer normally has a quota saying how much cherries they are allowed to produce from harvest.

2.6 Glossary

Circular Economy (CE)	The Circular Economy seeks to reduce, reuse and recycle.
Linear Economy (LE)	The traditional Linear Economy consists of taking, making and wasting.
Greenhouse Gases (GHGs)	Gases like carbon dioxide (CO2) and methan; reflects parts of the Earth's heat radiation back.
CO2-eq (CO2 equivalent)	A metric used to compare emissions from various greenhouse gases.
UNFCCC/Paris Agreement (PA)	A legally binding international treaty on climate change, signed December 12, 2015.
Kyoto Protocol	International treaty committing Parties to reduce GHG emissions based on individual targets.
Cap-and-Trade System	A government regulatory program designed to limit, or cap, the total level of emissions of certain chemicals.
Carbon Credit (CC)	Capturing or sequestering carbon from the atmosphere represented as a tradeable commodity.
Carbon Credit Certificate	A certification from a formal standard relating to the quality of a carbon project.
Carbon Markets	Markets for trading CCs.
Greenwashing	A term used for fraudulently claiming to reduce emissions or act environmentally friendly.
Double Counting/Spending	A term for counting resources or assets more than once.
Leakage	Addresses increased emissions or decreased removals not accounted for within and outside the scope of a project.
Permanence	Addresses if a CC investment remains intact long enough to account for the carbon capture.
Additionality	States captured carbon must be carbon that would not have been captured otherwise.
Provenance	The lineage/history of resources, being movement and ownership over time.
Supply Chain Management (SCM)	Management of the entire production flow of goods or services, i.e. from coffee farm to cup.
Supply Chain Network (SCN)	A supply chain network shows how agents are linked together in a network of information or material distribution and sharing.

Table 2.1: Glossary and terminology.

Blockchain (or Distributed Ledger)	A decentralised, append-only data structure containing nodes that share a global state and resembles a log of ordered transactions.
Resource Events	Producing (transforming), transferring or transporting resources, which have a resource effect, i.e. who owns what.
Events	Significant real-world events that update the state of the world.

Table 2.2: Glossary and terminology continued.

3 Design and Implementation of TRAZE

The following section comprises the technical components of the framework. Sections 3.1, 3.2 and 3.4 describe the business logic and UML class design and implementation; section 3.5 describes the applied software and hardware; section 3.6 outlines how all components interact and describes the database design. Collectively, the components create the backend of a web application for modelling SCNs.

3.1 Vector Spaces

To implement vector spaces as defined in section 2.4.1, we use the design shown in the UML diagram in figure 2.1. The abstract class `Vector` implements the interface `IVector`, meaning a vector space must implement the zero vector, addition and multiplication operators as methods. Furthermore, we use the `Map` interface to implement vector spaces as stateless, immutable maps. The vector methods are implemented as static methods in the `Ownership` and `Resources` classes, with slight differences.

3.2 Ownership, Resources, Transfers, Transports and Transformations

We distinguish the two classes, `Ownership` and `Resources`, theoretically outlined in sections 2.4.2 and 2.4.1. The `Ownership` class implements ownership states, and the `Resources` class implements the resource vector space, mapping resource types to amounts.

The theory in sections 2.4.2 and 2.4.4 describe ownership and location states, where transports are subspaces to location states and transfers are subspaces to ownership states. However, the implementation utilises the `Ownership` class as a superclass for transfers and transports, meaning the `Transfer` and `Transport` classes are subclasses of `Ownership`.

To accommodate this, we have implemented an `IActor` interface consisting of an id and a name, which the `Agent` and `Location` classes implement. Since the `Ownership` class maps `IActor` to `Resources`, the `Ownership` class can represent both the ownership and location states. Figure 2.2 demonstrates the UML diagram of this relationship.

Likewise, the `Transformation` class is implemented as a subclass of `Resources` because transformations are subspaces of resources, as described in section 2.4.5. Upon creating a `Transfer`, `Transport` or `Transformation` object, the framework ensures that the global invariant is respected, as described in section 2.4.3. If it is not, the framework will reject the transfer, transport or transformation.

3.3 Carbon Credit Certificates and Carbon Credits

A CC certificate is implemented as a specialised resource type, extending the resource type with an area, evidence of their existence and validity, valuation and a timespan. The area and timespan define the specific area and duration of the carbon offset project, while the valuation determines the estimated carbon capture for the given area, measured in CO2-eq. Locations are specified by longitude and latitude, as mentioned in

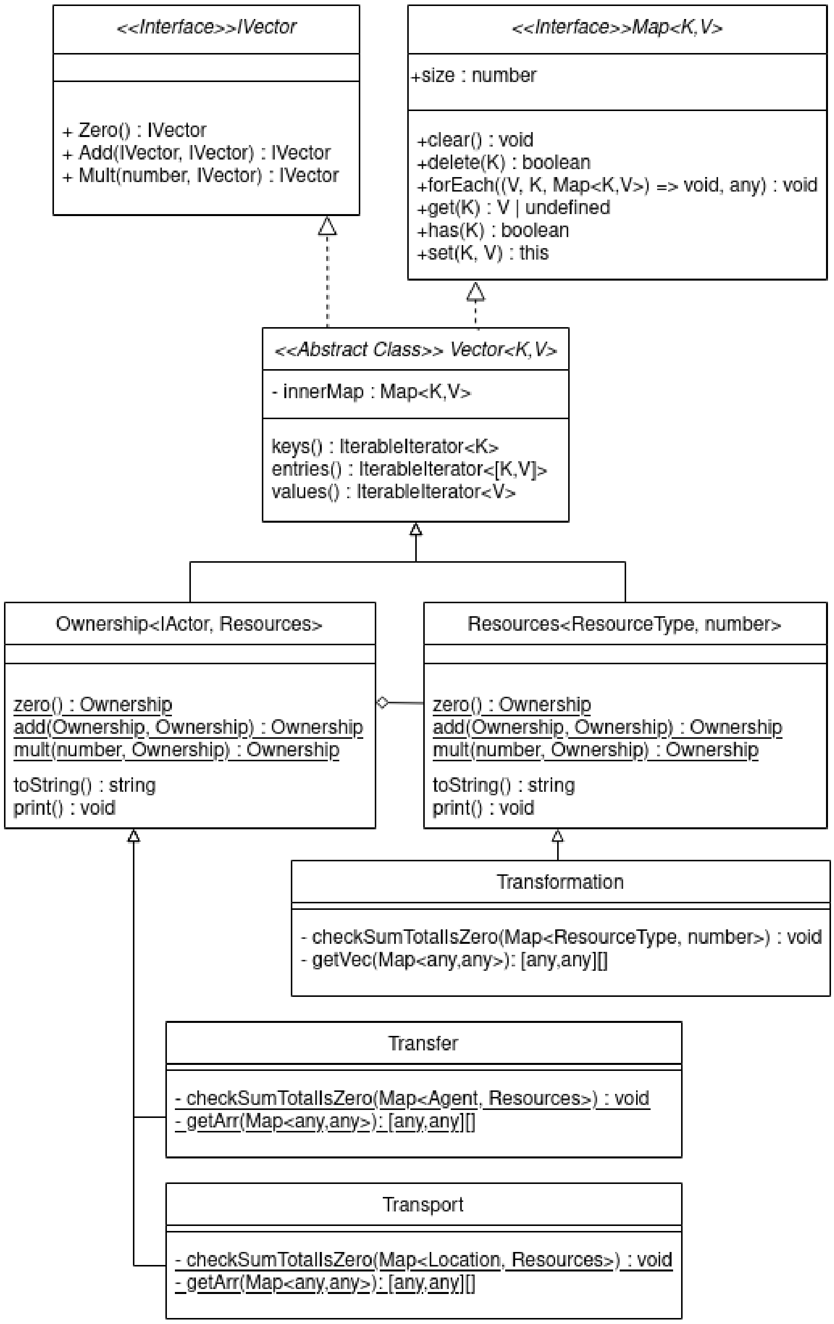


Figure 2.1: A UML diagram demonstrating the vector space design and implementation. The abstract `Vector` class implements interface `IVector` and `Map`, and classes `Ownership` and `Resources` inherits from `Vector`. `Transformation` is a subclass of `Resources`, and `Transfer` and `Transport` of `Ownership`.

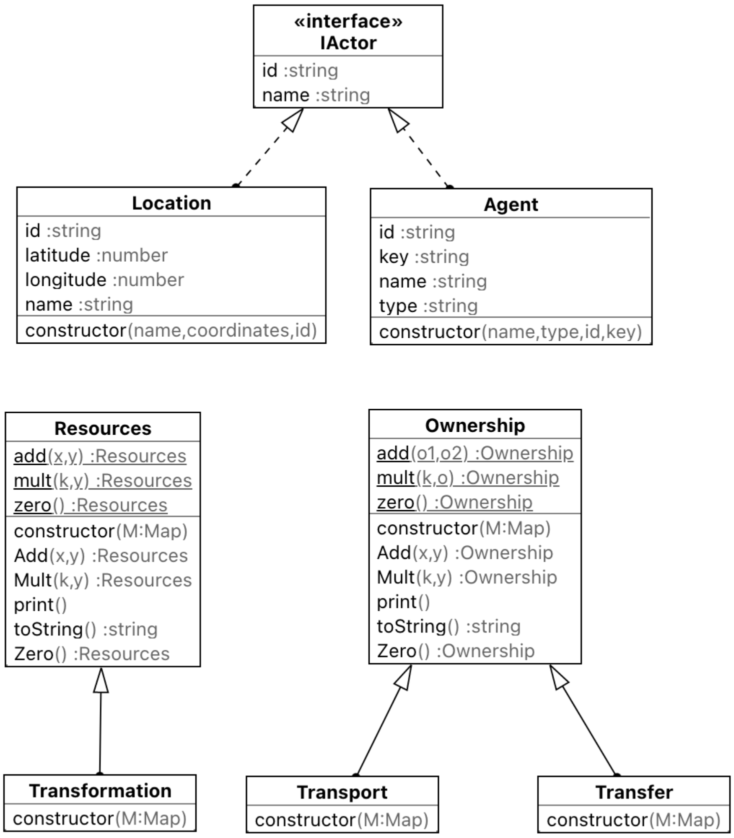


Figure 2.2: A UML class diagram demonstrating the inheritance between the classes `Ownership`, `Transfer` and `Transfer` with `IActor`, `Agent` and `Location`, and the inheritance between the classes `Resources` and `Transformation`.

section 2.4.4. Areas are computed based on polygons constructed from locations.

For areas overlapping in two CC certificates, we distinguish whether they are fully or partially overlapping. If they fully overlap, we check if the timespans are in sequence because if they are, it means the CC certificate is being reissued; if they are not, the CC certificate is rejected because we have a possible attack vector of either leakage or double counting, as described in section 2.3.3. For partially overlapping areas, the CC certificate issuance is immediately rejected.

Since CC certificates are nonfungible and because a CC certificate amounts to a specific number of CCs based on its valuation, for example, a CC certificate with a valuation of 24 metric tonnes of CO2-eq amounts to 24 CCs, an agent can own fractions of CC certificates; to do so, the ownership must be transferred, as with all other resources.

A specialised transfer occurs when a CC is used, in which case, the transferee is the designated agent *Atmosphere*, as described in section 2.3.1. Subsequently, the CC is out of reach for "re-spending".

3.4 Resource Manager

The resource manager class, `ResourceManager`, is a subclass of the superclass `Agent` and is implemented with the algebraic requirements and constraints as defined in section 2.4.

Figure 2.3 visualises the UML diagram of this relationship. As the figure also shows, the resource manager is stateful because through its constructor, it is given an `Ownership` object, defined by the `Ownership` class, and the primary job of the resource manager is to maintain and manage the internal state of ownerships and locations of resources and provide digital resource management for arbitrary resource types.

The resource manager receives transfers, transports or transformations of which it:

1. guarantees resource preservation, meaning no resource is duplicated or lost,
2. ensures that the resources applied in each adhere to the accurate ownerships and respect the zero-sum policy, as described in sections 2.4.3 and 3.2, and
3. enforces credit limits by adhering to the credit limit policy.

Each agent has a credit limit specific to each resource, whereas the resource manager can have infinite negative credits, which is necessary for implementing the strategy of zero-balance ownership states defined in section 2.4.3.

Upon receiving an input event, the resource manager evaluates the input and updates the internal states accordingly. Ownership states and location states are implemented as zero-balance ownership states, meaning the first is implemented as a transfer and the latter as a transport.

A transformation of resources applied to the the ownership state results in a new ownership state because the input and output resources of the transformation are not necessarily the same. However, resources are not required to have a location, and therefore, the location state is only modified if locations are present for the resources to transform. To transport or transform resources, they must have the same location, be that a location or no location in terms of transformations.

3.5 Technology Stack

3.5.1 Programming Language

Desired Properties: As the implementation goal of this thesis is to construct a back-end prototype of an easily extendable web application that implements the algebraic framework described in section 2.4, including the requirements described in section 2.3, it would be beneficial to use a programming language that

- is capable of creating a scalable web application,
- reasonably common and up-to-date with extensive libraries,
- works well with relational database systems,
- includes an advanced testing framework and

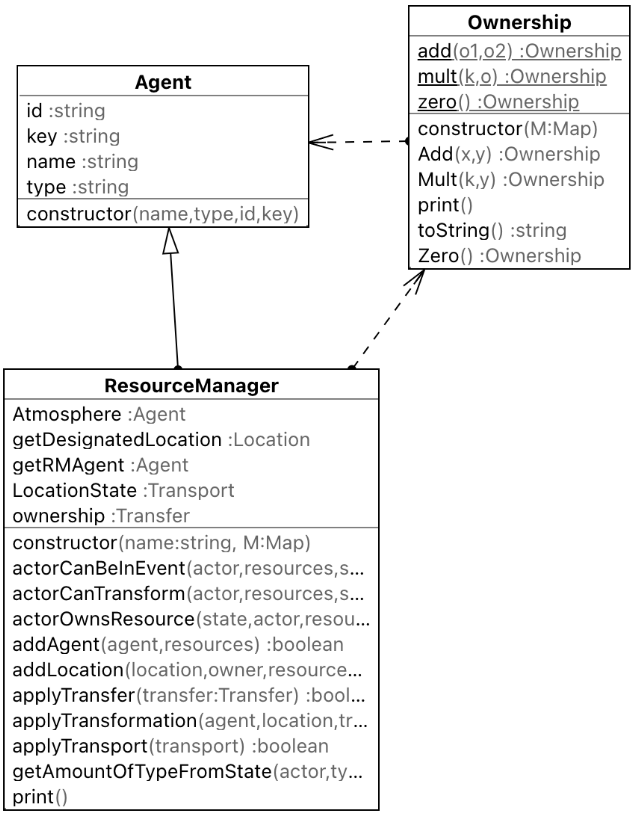


Figure 2.3: A UML Class Diagram of the relationship between the classes `ResourceManager`, `Agent` and `Ownership`.

- is strongly typed for fewer bugs and more security.

In addition, since this thesis extends the work by Bager, Düdder, et al., Hébert and Alstrup & Borgert, it would be natural to build upon the algebraic theory and existing prototypes implemented in JavaScript and Java.

Chosen Programming Language: By combining the knowledge of these prototypes and the desired properties of the framework, we deemed it beneficial to implement the framework using TypeScript and PostgreSQL. TypeScript is an Object-Oriented strict superset of JavaScript, meaning the TypeScript code is compiled into JavaScript code. The benefits of using TypeScript for implementing the framework are the following:

- It is designed to create scalable web applications easily.
- It uses static type checking, which allows statical detection of programming errors more quickly and reliably, which reduces the time spent debugging.
- It is strongly typed, forcing developers to write type-specific code with proper checks, reducing bugs like null handling, undefined and more.
- It is a well known language, and therefore easy to extend for future work.

- Since it is an Object-Oriented version of JavaScript, we combine knowledge from both existing prototypes by Bager, Düdder, et al. and Alstrup & Borgert.

However, one of the trade-offs endured with TypeScript was its inability to compare objects based on values of object fields, causing extensive needs for loops because the framework is built upon vectors as maps, as described in section 3.1. Although maps have high performance, this inability caused underutilisation of maps resulting in poorer performance.

3.5.2 Applied Tools

The predominant libraries in the framework are NodeJs, Express and Pg-Promise.

NodeJs is a fast open-source, cross-platform runtime environment and library used to run web applications external to browsers. Most commonly used for server-side programming and primarily deployed for non-blocking event-driven servers. However, NodeJs cannot handle requests, HTTP methods or create file servers; this is where Express excels since it is a layer built on top of NodeJs to help manage servers and routes, and it can build singlepage, multipage and hybrid web applications.

Pg-Promise is a PostgreSQL library for NodeJs and, technically, a wrapper for node-postgres, a collection of NodeJs modules for interfacing with PostgreSQL databases. As a wrapper, it converts the callback interface into a promise-based interface, meaning an action can either be completed or rejected and unlike callbacks, promises are chainable.

3.5.3 Available Hardware

The framework is designed to run on all modern computers, although implemented and tested in Visual Studio Code using the following:

- a macOS 2.7 GHz Quad-Core Intel i7 with 16 GB RAM computer and
- a Ubuntu Linux 2.5 GHz Quad-Core Intel i5 with 8 GB RAM computer.

3.6 Application Overview

The design of the web application is based on the Model-View-Controller (MVC) pattern[18], which is designed to separate concerns in the application (Sharan & Späth, 2022). In single-project scenarios, folders separate the concerns, as with this thesis. The default MVC pattern includes separate folders encapsulating the responsibilities of Models, Views, and Controllers and additional folders for Classes, Services and Repositories.

This way, presentation details are limited to the `View` folder and data access details and business logic to the `Classes`, `Interfaces` and `Services` folders, which reside within the `Models` folder. More specifically, Services manage direct interaction with the database through Repositories and database queries from the `Repositories` and `Database` folder, and Classes define the business logic of the entities with no database interaction.

[18]Common web application architectures [online] `https://learn.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/common-web-application-architectures` retrieved 20-11-2022.

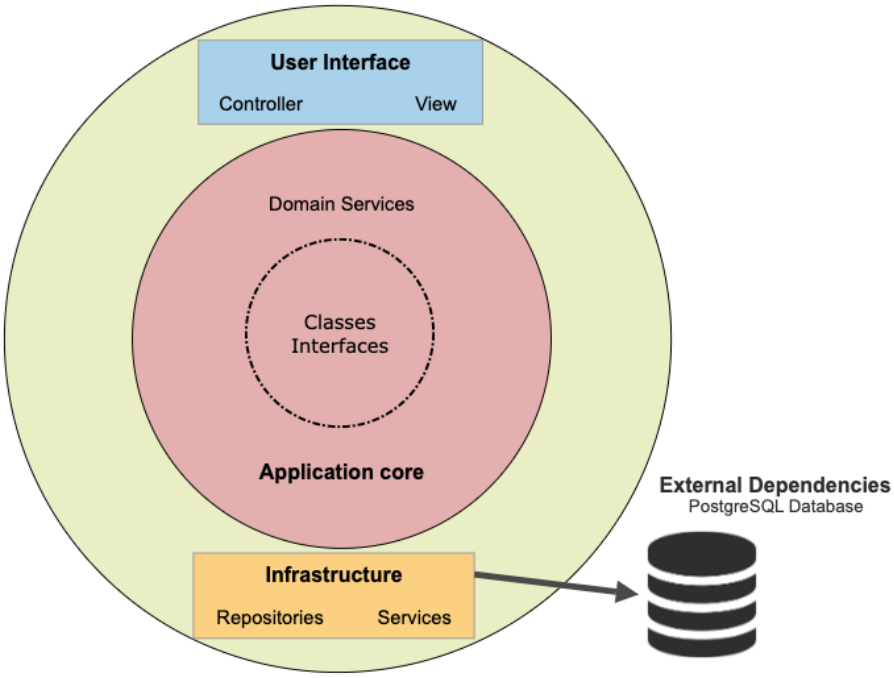


Figure 2.4: A representation of the overall design using the Onion Architecture.

The Controller manages the REST-API with requests and responses from the `Controller` folder. This part acts as the communicator between the user and the system. The Controller retrieves data from the Models based on the request and the presentation from the View based on the response.

Figure 2.4 visualises the framework's architecture using an onion view to demonstrate how the application follows the Dependency Inversion and the Domain-Driven Design principles to achieve a clean architecture. As the figure shows, the dependencies flow toward the innermost circle, where the business logic and application model resides.

By not having business logic dependent on data access or other infrastructure concerns, the dependency is inverted, and as the figure shows, the Application Core has no dependencies on other application layers. The application's entities and interfaces are at the centre.

The Resource Manager, described in section 3.4, operates as a domain service. The domain services reside at the middle layer, which contains business logic and is part of the domain model (as with entities and value objects), meaning the domain services participate in the decision-making process as entities and value objects do. On the outer layer, the User Interface and the Infrastructure layers depend on the Application Core, however, not on one another. The Infrastructure communicates with the database.

3.6.1 Communication Between the Framework and the User

As mentioned, the Controller manages the REST-API and communication between the user and the system. The API endpoints are, therefore, the events that a user can employ. More specifically, the user should be able to and can:

- sign up as an agent,
- register resources on sign up,
- create resource types,
- transfer ownership of resources to other agents,
- transform resources into other resources,
- transport resources from one location to another,
- create CC certificates,
- spend CCs and
- retrieve data regarding all the user's events.

Figures 2.5 and 2.6 show simple sequence diagrams demonstrating the communication between agent Alice and the framework. In the examples, Alice wants to transfer 10 DKK to Bob, and the request is forwarded and, in one case, verified and executed and in the second case, rejected since Alice in this example does not own sufficient resources to transfer to Bob.

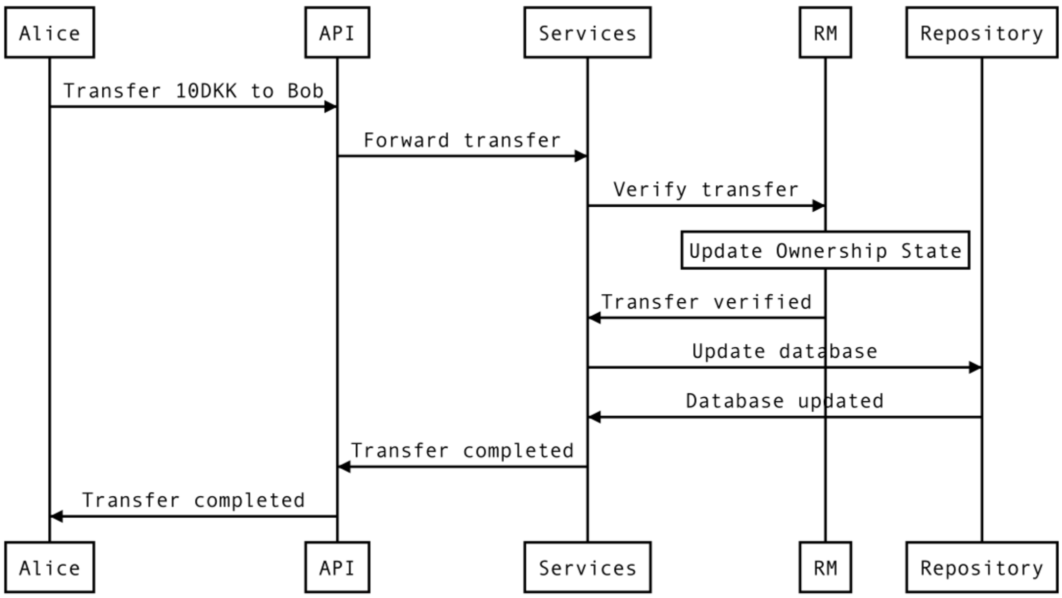


Figure 2.5: A simple sequence diagram to demonstrate the communication between agent Alice and the framework when requesting a valid transfer.

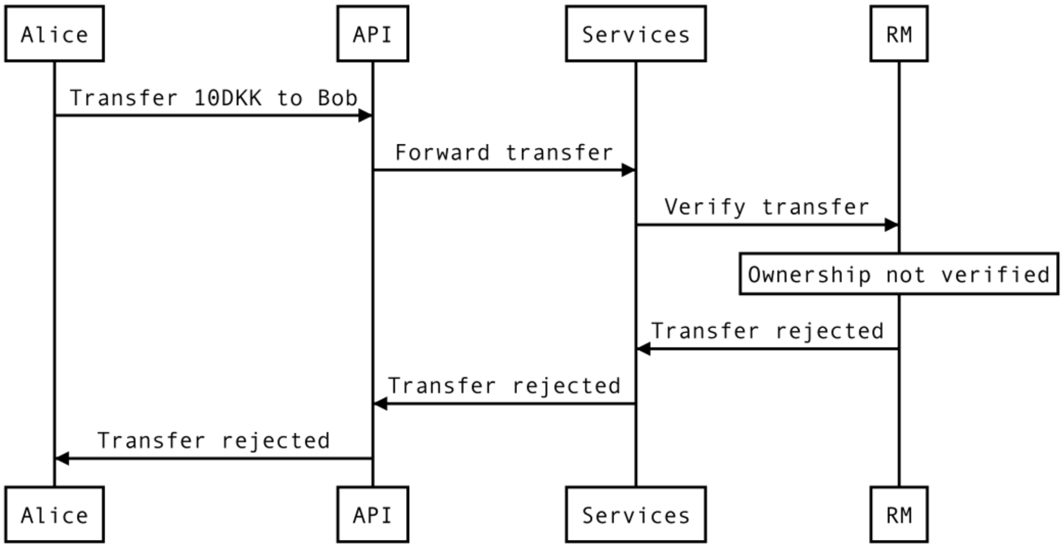


Figure 2.6: A simple sequence diagram to demonstrate the communication between agent Alice and the framework when requesting a non-valid transfer.

3.6.2 Database Design

To create the conceptual and logical design, we analysed the requirements presented in sections 2.4 and 2.3 and the API calls defined in section 3.6.1. The data types were modelled and implemented in a standard relational database system (RDBMS), visible in figure 2.7, illustrating the database design's Entity Relation (ER) Diagram.

The event-based design of the framework means that most tables are event tables and can, therefore, only insert and view events. The exceptions are the derived tables `Ownership` and `Location_State`, which can, therefore, both insert and update. These two tables maintain the system's current state, and while both could be replaced by materialised indexed views, the trade-off might be poorer performance since both tables are updated at every transfer, transport and transformation. The framework updates both tables during the same query execution at each transfer, transport and transformation to avoid inconsistencies.

Using a persistent RDBMS as PostgreSQL, we store persistent data in the form of objects in tables, which ensures durability when changing devices and software. However, in terms of process persistence, the framework does not currently store data to non-volatile memory. Nevertheless, the framework's design ensures that an event is only included in the current state if the event has been stored in the database.

For example, during a transfer, if the resource manager approves, but the system (excluding the database) crashes before or while updating the database. Once the system runs again, the transfer is not included and requires redoing because the transfer and ownership tables are updated simultaneously, and the resource manager defines the current ownership state based on the ownership table. Had the crash occurred after the database update, the resource manager would include the transfer to the current state, which would not need redoing. The same concept applies to all features of the framework.

`Resource_Type`, from figure 2.7, is based on the resource definition from section 2.2. However, since CC certificates are defined as specialised resource types, their Universally Unique Identifiers (UUIDs) in table `Carbon_Credit_Certificate` are foreign `Resource_Type` id keys.

One of the ways we ensure data integrity is by using composite primary keys. In the `Transfer` and `Transport` tables, transfers and transports can include a compound resource, meaning multiple rows per transfer or transport may occur. In this case, the composite primary keys are included within the same transfer/transport UUID, transferor/owner UUID, locations start UUID (for transports) and transferee/destination UUID, but the resource type UUID differs. This way, we can:

1. identify specific transfers and transports containing compound resources,
2. avoid duplicated data since no duplicated entries of the same transfer or transports containing the same resource type will occur, and
3. avoid conflicting data since many transfers and transports might include the data, but the UUID will be unique for each transfer/transport.

In the `CC_Retire` table, a CC certificate can retire in fractions, meaning the table might have multiple entries of the same transferor and CC certificate UUID. However, the system ensures that a CC certificate cannot be overspent.

As for transformations, we do not need composite primary keys; instead, we include three tables because the transformations are typically a compound resource transforming into a single–or compound resource, meaning there are multiple transformation inputs and outputs for each transformation UUID, which reflects in the design in figure 2.7. This design ensures data integrity because it:

1. avoids conflicting data between transformation inputs and outputs due to the separate tables and
2. avoids duplicated and orphaned data by having a transformation UUID and using it as a foreign key reference in the input and output tables.

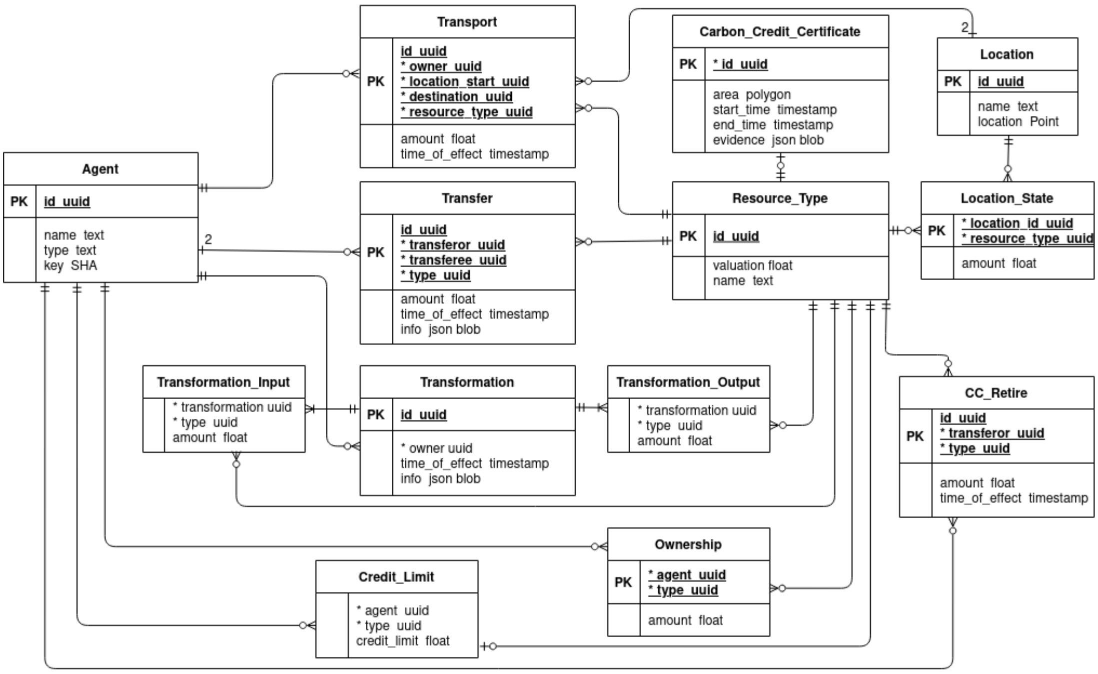


Figure 2.7: The Entity Relation Diagram of the database design applied in the framework.

4 Evaluation

The following section evaluates the framework's technical performance and usefulness. Section 4.1 demonstrates the framework put to use with a specific use case and shows modelling of carbon effects, section 4.2 proves the framework's robustness with its development process, tests and techniques and section 4.3 includes a short guide for running the framework.

4.1 Modelling Carbon Effects in TRAZE

To evaluate the TRAZE web application we have designed and implemented, using REALISTIC constructs, a simple use case based on the Columbian coffee SCN. Further, we prove that TRAZE can be used for modelling carbon effects by making a simple model that fits the use case. The model is a function interpreting the data and calculating the carbon effect.

4.1.1 Use Case

Figure 2.8 shows an overview of the use case (the implementation is available on GitHub). The three **A**gents *Farmer, Cooperative* and *Almacafé* all reside at different **L**ocations, although omitted from the figure, i.e. *Farmer* is related to the location *Sixaraca coffee farm* which has longitude 5.3414399 and latitude −75.7031506. When interpreted to real-world coordinates it locates the farm in Columbia. The **R**esource types of the use case are *Cherry, Wet parchment, Pulp, Dry parchment, Pesos, Green coffee, Pasilla* and

Unaccounted, and the **E**vents are as follows:

Farmer owns and signs four events:

Transformations:

- *Wet milling* of coffee cherries to turn them into wet parchment and pulp.
- *Drying & fermenting* wet parchment to get dry parchment. Some weight is unaccounted for and represented by the Unaccounted resource type.

Transfer of ownership of dry parchment to *Cooperative*.

Transport of dry parchment to the location of *Cooperative*.

Cooperative has three events:

Transfer of pesos to *Farmer*.

Transfer of dry parchment to *Almacafé*.

Transport of dry parchment to the location of *Almacafé*.

Almacafé has two events:

Transfer of pesos to *Cooperative*.

Transformation, being *dry milling*, of dry parchment into green coffee and the lower quality beans called passila. Some weight is unaccounted for and represented by the Unaccounted resource type.

Note. In the figure the transfers between *Farmer* and *Cooperative* are shown as a single transfer:

$$t = \{\mathit{Farmer} \rightarrow \{-167\cdot\text{Wet parchment}+550\cdot\text{Pesos}\}, \mathit{Cooperative} \rightarrow \{+167\cdot\text{Wet parchment}-550\cdot\text{Pesos}\}\}$$

This is also the case for the transfers between *Cooperative* and *Almacafé*.

4.1.2 Model

We model carbon effects from the base data of the use case for (1) a single agent, (2) the full SCN, which in this case, is the three agents, and (3) an agent and its immediate neighbours based on events. We assume transfers do not emit CO2 in the model, while transports and transformations do. The amount emitted by a transport depends on the distance and the weight of the resources transported. The CO2 emissions of a transformation depend on the weight of the resources to be transformed, on a constant that is different for each resource type and on the output resources.

Single agent. The local emissions of an agent are discovered as follows: We extract all transport and transformation data for the agent. Then calculate the carbon effect for the transports, et transformations, by linearly summing all the effects. The total effect is the sum of the carbon effect of the transports and the transformations.

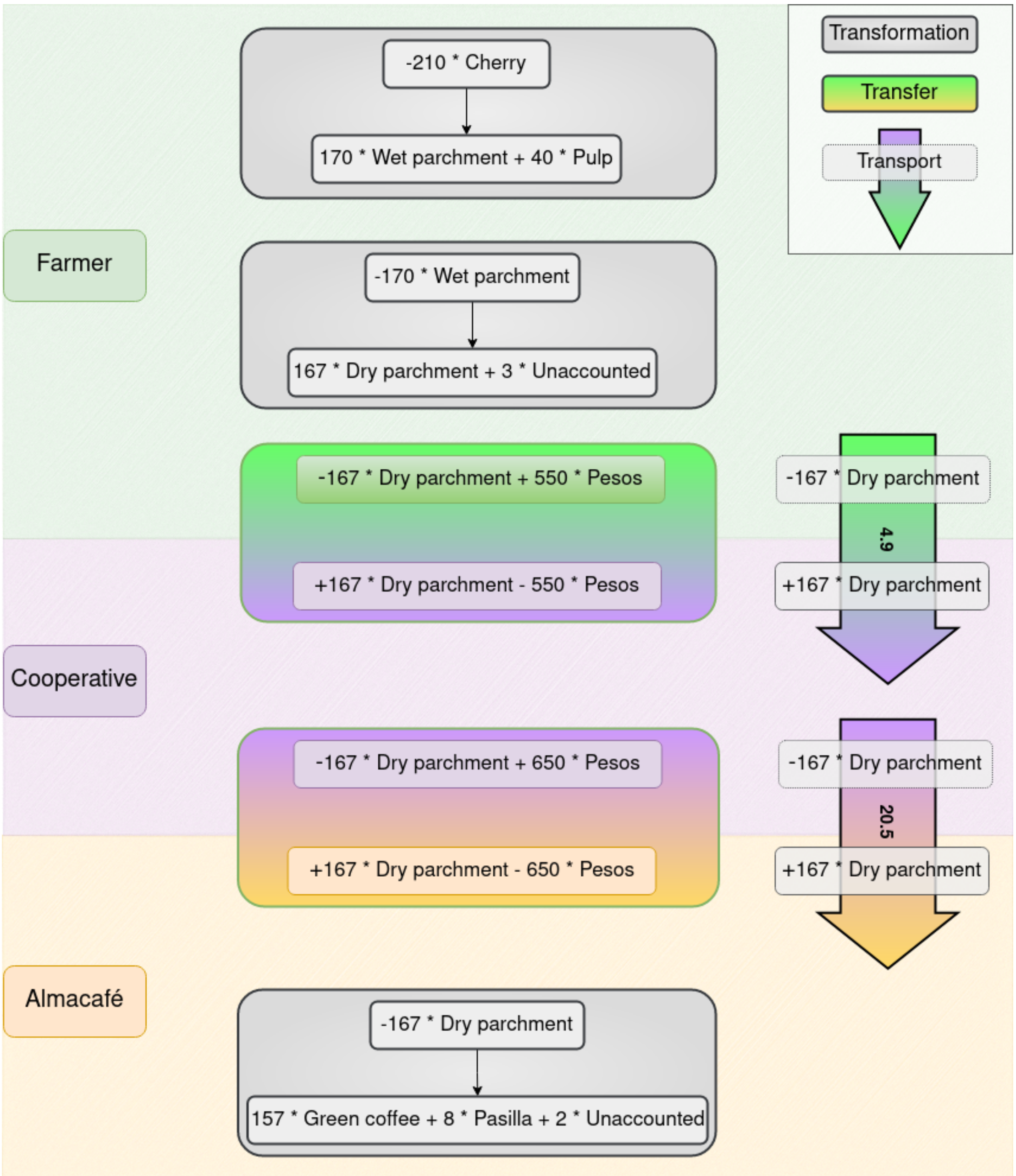


Figure 2.8: Use case with three agents *Farmer, Cooperative* and *Almacafé* that all resides at different locations. The boxes show events where the grey boxes are *transformations*, the graduaded boxes are *transfers* and the arrow boxes are *transports* with the distance shown. The vertical axis depicts time, so the *Farmer*'s events happens before the *Cooperative*'s. The two transformations of the *Farmer* are *wet milling* and *drying and fermenting*. The *Farmer* transfers and transports dry parchment to *Cooperative* that transfers pesos as payment, and afterwards transfers and transports the dry parchment to *Almacafé* whom transfers pesos as payment. Finally *Almacafé* transforms the dry parchment by dry milling.

Full SCN. The full SCN is calculated as the summation of the carbon effects of *Farmer*, *Cooperative* and *Almacafé*, respectively.

Agent and immediate neighbours. The agent's immediate neighbours are all agents that partake in a transfer with said agent. The carbon effect of the neighbours' events is calculated and summed with the local carbon effect of the agent.

When assumed, a transfer always transfers the total amount of a resource, and not a fraction, the carbon effect of an agent and its immediate neighbours corresponds to the agent's *total carbon emission.* The total emission of an agent is the emission emitted somewhere in the SCN from events on resources that the agent has owned at some point in time.

The relation to Scopes 1-3 (defined in section 2) are as follows: The local emission of an agent corresponds to Scope 1. Scope 2 is not related since we do not include any indirect emissions owned by an agent in the use case (for instance, no electricity or heating). Scope 3 corresponds to an agent's local emission subtracted from an agent's total emission.

Thus, we have shown that the events of the coffee SCN are formally expressive in TRAZE, and we have made a simple model for interpreting the carbon effect of agents of the Columbian coffee SCN. If an expert in carbon effects sees our model and deems it inadequate, they can produce new or multiple models with varying variables and apply them to the event data of the system.

4.2 Testing and Development

The development process has been Test-Driven Development with a combination of solo and pair programming, along with code reviews and refactoring.

We test the framework using both unit and integration tests. As figure 2.9 shows, unit tests are applied directly on the Application Core, which is easy since the Application Core does not depend on the Infrastructure. Figure 2.9 also shows integration tests from the Infrastructure layer to test the system response from all API calls. These tests include all framework layers and test positives, negatives and edge cases.

In addition, the tests conducted prove code robustness and that our framework adheres to the architectural constraints and is resilient towards the attack vectors explained in section 2.3. These tests ensure the following:

- CC certificates cannot overlap in time or geographical area.
- Retired CCs cannot be re-spent, over-spent and can be partially spent.
- Agents digitally sign all events.
- Credit limits cannot be violated, and agents cannot administer resources, including CCs, they do not own.
- The global invariant is always respected.
- The architecture is event-based:

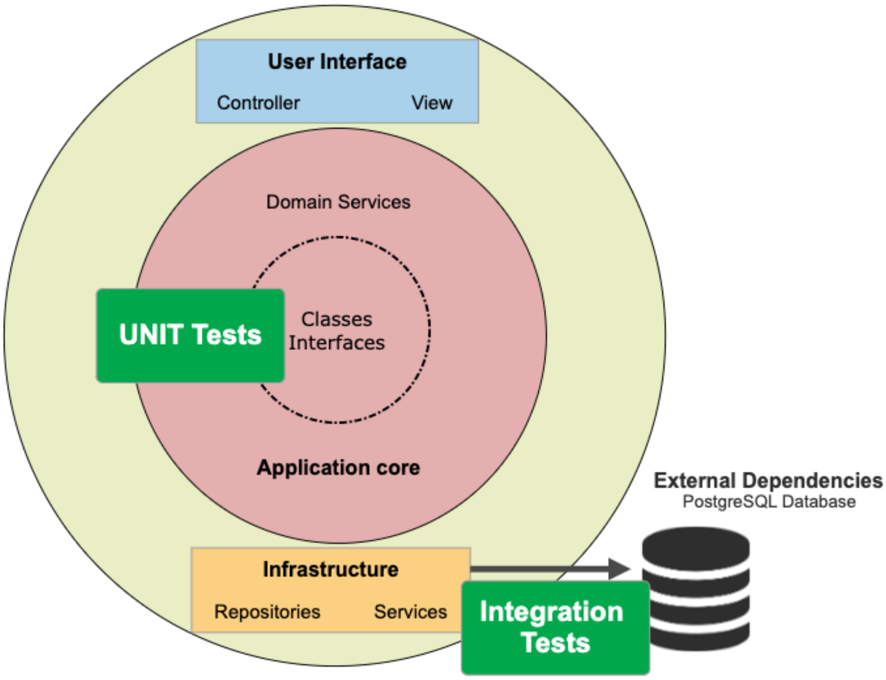


Figure 2.9: Using the same illustration of the overall design from figure 2.4, we demonstrate the test applications.

  - All events are accurately stored in the database.
  - Specific details of events are retrievable.
  - Full SCNs are testable.
  - States are updated accurately and in coherence with the database.

- Leakage of CC certificates is prohibited because time and areas must be continuous and coherent.

4.3 Guide for Running the Framework

The implementation is publicly available on GitHub[19].

4.3.1 Prerequisites and Requisites

The framework requires JavaScript, TypeScript, NodeJs, PostgreSQL and PostGis installed. In addition, the command, `psql`, for running PostgreSQL must work from the command-line. Some setups require creating an alias for the `psql` command.

Once installed, locate the `init.sh` file from the top folder and run it to setup the database and install the NodeJs packages.

[19]GitHub: `https://github.com/smhyatt/TRAZE`

4.3.2 Steps of Execution

The first time building the framework:

```
tsc -build
```

To compile the framework:

```
npm run build
```

To clean the framework:

```
npm run clean
```

To run tests:

```
npm test
```

5 Discussion

In summarising the implementation and design section, the key takeaways are the following:

- Vector spaces are implemented across the entire framework, as described in section 2.4, meaning the business logic for states, ownerships, transfers, transports and transformations is implemented using linear algebra.

- The resource manager maintains the system's current ownership and location states and handles business logic, as described in section 2.4.

- CCs are handled by issuing CC certificates based on a timespan and an area based on latitude and longitude coordinates. Agents own CCs by partitioning CC certificates, and CCs are retired by specialised transfers to the atmosphere.

- The technical backbone of the framework is implemented as a web application using TypeScript and PostgreSQL. The overall design and structure apply the MVC pattern to separate concerns.

- Finally, the database design is designed to achieve persistence and data integrity.

The following section discusses and analyses paradoxical aspects concerning the theory and the implementation of this thesis with reference to related work and assesses the framework in terms of future work.

5.1 Blockchain and Decentralisation

We identify multiple benefits in the application of blockchain and decentralisation. We want the benefits of global collaboration across competition without classical data sharing. A secure, privacy-preserving, immutable and decentralised system, without the control of any centralised and privileged agents (government, technology or industry), can provide this by sharing the results and not the data itself, and this incentivises agents to participate because it creates a liberal market where agents can include valuable data of their business, without losing their competitive advantage.

As briefly mentioned in section 1, many studies consider blockchain valuable for modelling SCNs and SCM (Azzi, Chamoun, & Sokhn, 2019; Dwivedi, Amin, & Vollala, 2020; Casado-Vara et al., 2018). In the paper by Azzi et al., the authors integrate blockchain into supply chain architectures to create reliable, transparent, authentic and secure systems. In the paper by Dwivedi et al., the authors utilise blockchain for secure information sharing to ensure proper authentication among participants, data management and data integrity for SCM in pharmaceuticals. In the paper by Casado-Vara et al., the authors model SCNs using blockchain to enable the concept of CE as a case study. The paper by Bager, Düdder, et al. argues the benefits of using blockchain, particularly its tamper-proof data handling and storage, due to a lack of trust among SCN participants. Also, decentralisation enables more access and control of the data to non-dominant actors (such as small-scale actors).

However, in a recent paper by Bager, Singh, & Persson, 2022, the authors argue that while blockchain for SCNs provides transparency and knowledge on provenance, which can promote the transformation of consumer behaviour, their pilot implementation belied blockchain for agro-food supply chain sustainability. The authors identify a need to understand and minimise supply chain barriers because most barriers exist in the real world, not digitally. Therefore, they deem it necessary before the benefits of digitalisation and decentralisation can be reaped concerning blockchain for SCNs.

Some barriers may include challenges concerning internet connectivity and smart devices, as described in the papers by Bager, Düdder, et al., 2022 and Mehrabi et al., 2021. Many small-scale farmers have limited or momentary internet connectivity, and the cost of data remains steep, posing challenges to always-online solutions and, thereby, possibly to blockchain solutions.

As this framework builds on the work by Bager, Düdder, et al., decentralisation follows as a natural extension, and while the framework prototype is not currently decentralised, several measures make the framework extensible to decentralisation:

- The event-based nature of the framework is well suited for decentralisation and blockchain because we process and store a sequence of certified events and the framework utilises provenance.

- In addition, the digitally signed events also motivate blockchain and decentralisation.

- While only one local database is attached to the prototype, the web application allows a simple extension to let any user access TRAZE having a local database and internal procedures.

- Currently, the system uses one resource manager; however, extending to distributed resource managers is doable because the logic of the resource manager is isolated to its class, and in doing so, the framework would be more scaleable for more users.

5.2 Carbon Credits, the Silver Bullet?

While CCs have emerged as an aid for mitigating climate change, challenges follow. Among the challenges, several studies identify problems related to ensuring permanence and additionality (Galik et al., 2014;Kotsialou et al., 2022;Gustavsson et al., 2000).

Gustavsson et al., 2000 discuss the challenges of additionality as an inherently uncertain concept because it depends on an unobservable counterfactual scenario, where buyers and sellers share a common motive to exploit the ambiguity and overestimate the CCs. As additionality requires offsetting that would not have occurred otherwise, one could argue that it might remove the incentive for maintaining and protecting pre-existing carbon-capturing resources, as only "new" projects are rewarded.

Thamo & Pannell, 2015 argue that permanence is commonly estimated for 100 years of sequestration, and re-release following suit may cause atmospheric carbon levels to rise above the initial levels. In the paper by Galik et al., 2014, the authors propose alternative approaches to alleviate non-permanence. One is to create permanent CCs that are incrementally rewarded over time, another is to incorporate the incidence of non-permanence automatically into the crediting system, and a third is to address the potential loss of permanence through commercial insurance, to name a few.

The latter has likewise been proposed in the recent paper by Friess et al., 2022, where the authors argue that blue carbon projects[20] experience challenges due to the cost and burden of verification compared to CCs in other ecosystems. Therefore, they propose other supplementary financial instruments beyond CCs, such as bonds and ecosystem service insurance, suggesting that a portfolio of financial instruments will generate substantial and reliable funding streams to promote blue carbon as a mitigation solution.

Another study by Kotsialou et al., 2022, looks towards applying blockchain technology in REDD+ projects to address challenges such as additionality, permanence and leakage. They discovered the potential in blockchain technology to improve verifiability, reduce transaction costs and, to some degree, alleviate challenges with additionality and permanence.

Ultimately, the challenges of additionality, leakage and permanence are matters that, to this date, cannot be entirely solved, which promotes the need for a system where post hoc forensics makes it possible to monitor the matters to which the three challenges occur. Concerning additionality, post hoc forensics would show and measure differences in carbon emissions. Concerning leakage and permanence, post hoc forensics would show the actual effect of a carbon offset project over a given period and enable data analysis, which in turn could provide more accurate estimations of carbon effects. Consequently, supporting the need for TRAZE.

[20]Blue carbon projects are carbon offset projects that capture carbon by the world's ocean and coastal ecosystems.

6 Conclusions

As CCs take their place in the economy as an aid towards climate change mitigation, solutions for tackling their related challenges are emerging.

This thesis introduces TRAZE, building upon the event-based modelling framework, REALISTIC, for total resource accounting, where CCs and carbon accounting are central components. TRAZE is prototyped as an open-source web application, to which we implement CC certificates as specialised resource types, using area detection and timestamps. We analyse common attack vectors and other challenges related to CCs, such as greenwashing, leakage and ensuring additionality and permanence—how related research addresses these challenges, including their consequences.

Ultimately, we identify the need for a system that supports privacy, reliability and transparency in SCNs while facilitating carbon accounting and enabling forensics to help prevent attacks and fraud. We argue that TRAZE is a flexible framework for fine-grained tracking of production and transportation of physical resources and reliably modelling their direct and indirect carbon effects across mutually distrusting actors in SCNs.

In proving our hypothesis, we have modelled and computed the carbon effect based on event data of a simple yet illustrative use case of the Columbian coffee SCN, and the following section presents recommendations for future work.

6.1 Future Work

We have identified natural extensions to TRAZE, which we recommend for future work.

- A frontend to the TRAZE web application, such that users can navigate a user interface and not only engage through the REST-API.
- User verification, i.e. some third-party verification, such that users can rely on transfers and transports reaching the rightful agents.
- Cryptographic signatures for signing in as users (agents) and signing events, such as using SSH keys, to provide accountability of events. Currently, agents have a key, but others can extract it.
- Privacy protection and confidential information are not objectives of this thesis, although these would be natural extensions.
- It should be possible to group agents, such that agents from the same organisation or group shares resources and possibly location. Currently, agents are single identities.
- Confirmations of transfers and transports to avoid the possibility of transfers and transports occurring to unwilling agents.
- The contractual part of REALISTIC should implement contracts to allow future events between multiple agents. Additionally, netting of events because netting of

transfers can allow resulting compound transfers to be valid, which would otherwise be invalid when applied sequentially (Hébert, 2020).

- Attaching PDFs and other documents to events as evidence. Currently, users can only attach JSON objects.

- CC certificates should allow overlapping areas as long as one is no longer used for CC certification and the timespans are always sequential. Further, one way of solving location and area detections could be partitioning land areas.

- Enforcing resources to have a location at all times. In cases where resources, i.e. bicycles, are owed and thereby not produced yet, they can have a designated virtual location until they are produced.

- Decentralising TRAZE, for instance, using blockchain. Suck that multiple resource managers can be instantiated and distributed to improve system scalability, as discussed in section 5.1.

1 Appendix

1.1 Types of Carbon Offset Projects

Carbon Offset Project Types and Categories (2018)*

Project Category	Project Type
Agriculture	Fertilizer - N20
	Grassland/rangeland management
	Livestock methane
	No-till/low-till agriculture
	Rice cultivation/management
	Sustainable agricultural land management
	Other - Agriculture
Chemical Processes/Industrial Manufacturing	Nitric Acid
	Ozone-depleting substances (Article 5)
	Ozone-depleting substances (US based)
	Carbon capture and storage
	Coal mine methane
	Other - Chemical Processes/Industrial Manufacturing
Energy Efficiency/Fuel Switching	Energy efficiency - community-focused (targeting individuals, communities, etc.)
	Energy efficiency - industrial-focused (targeting corporations)
	Fuel switching
	Waste heat recovery
	Other - Energy Efficiency/Fuel Switching
Forestry and Land Use	Afforestation/reforestation
	Agro-forestry
	Avoided conversion
	Improved forest management
	REDD - Avoided planned deforestation
	REDD - Avoided unplanned deforestation
	Soil carbon
	Urban forestry
	Wetland restoration/management
	Other - Forestry and land use
Household Devices	Clean cookstove distribution
	Water purification device distribution
	Other - Household Devices
Renewable Energy	Biogas
	Biomass/biochar
	Geothermal
	Large hydro
	Run-of-river hydro
	Solar
	Wind
	Other - Renewable Energy
Transportation	Transportation - private (cars/trucks)
	Transportation - public (bikes/public transit)
	Other - Transportation
Waste Disposal	Landfill methane
	Waste water methane
	Other - Waste Disposal

*As defined by Ecosystem Marketplace. Please note that we modified our categorization of project types and categories between 2017 and 2018.

Figure .10: A table of carbon offset projects from page 84 in the book Carbon Neutrality in the Agri-food Sector by Acampora et al., 2022.

References

Acampora, A., Ruini, L., Pratesi, C. A., & Lucchetti, M. C. (2022). *Carbon neutrality in the agri-food sector - challenges and opportunities in a dynamic setting.* Gewerbestrasse 11, 6330 Cham, Switzerland: Springer Nature Switzerland AG. doi: https://doi.org/10.1007/978-3-030-88048-4

Alstrup, D. S., & Borgert, A. S. H. (2022). *Total resource accounting in a circular economy.* Unpublished master's thesis, University of Copenhagen.

Azzi, R., Chamoun, R. K., & Sokhn, M. (2019, June). The power of a blockchain-based supply chain. *Computers and Industrial Engineering*, *135*, 582-592. doi: 10.1016/j.cie.2019.06.042

Bager, S. L., Düdder, B., Henglein, F., Hébert, J. M., & Wu, H. (2022, June). Event-based supply chain network modeling: Blockchain for good coffee. *Frontiers in Blockchain*, *5*. doi: 10.3389/fbloc.2022.846783

Bager, S. L., Singh, C., & Persson, M. (2022, May). Blockchain is not a silver bullet for agro-food supply chain sustainability: Insights from a coffee case study. *Current Research in Environmental Sustainability*, *4*(100163). doi: 10.1016/j.crsust.2022.100163

Casado-Vara, R., Prieto, J., la Prieta, F. D., Corchado, J. M., Yasar, A., & Shakshuki, E. (2018). How blockchain improves the supply chain: case study alimentary supply chain. *Procedia Computer Science*, *134*, 393–398. doi: 10.1016/j.procs.2018.07.193

Corona, B., Shen, L., Reike, D., Carreón, J. R., & Worrell, E. (2019, September). Towards sustainable development through the circular economy—a review and critical assessment on current circularity metrics. *Resources, Conservation and Recycling*, *151*. doi: 10.1016/j.resconrec.2019.104498

COWI, & IEEP, E. I. . (2020, July). *Analytical support for the operationalisation of an eu carbon farming initiative: Lessons learned from existing result-based carbon farming schemes and barriers and solutions for implementation within the eu.* Report to the European Commission. Retrieved 2022-09-25, from https://climate.ec.europa.eu/system/files/2022-01/policy_forest_carbon_report_en.pdf

Dinh, T. T. A., Liu, R., Zhang, M., Chen, G., Ooi, B. C., & Wang, J. (2018, July). Untangling blockchain: A data processing view of blockchain systems. *IEEE Transactions on Knowledge and Data Engineering*, *30*(7), 1041-4347. doi: 10.1109/TKDE.2017.2781227

Dwivedi, S. K., Amin, R., & Vollala, S. (2020). Blockchain based secured information sharing protocol in supply chain management system with key distribution mechanism. *Journal of Information Security and Applications*, *54*(102554). doi: 10.1016/j.jisa.2020.102554

Foss, P.-K. (2018, May). *The office of the auditor general of norway's investigation of norway's international climate and forest initiative* (Vol. 3).

Friess, D. A., Howard, J., Huxham, M., Macreadie, P. I., & Ross, F. (2022, August). Capitalizing on the global financial interest in blue carbon. *PLOS CLIMATE*, *61*, 16. doi: 10.1371/journal.pclm.0000061

Galik, C. S., Murray, B. C., Mitchell, S., & Cottle, P. (2014, May). Alternative approaches for addressing non-permanence in carbon projects: an application to afforestation and reforestation under the clean development mechanism. *Mitig Adapt Strateg Glob Change*, *21*, 101–118. doi: 10.1007/s11027-014-9573-4

The greenhouse gas protocol. (2004, March). World Business Council for Sustainable Development and World Resources Institute.

Gustavsson, L., Karjalainen, T., Marland, G., Savolainen, I., Schlamadinger, B., & Apps, M. (2000). Project-based greenhouse-gas accounting: guiding principles with a focus on baselines and additionality. *Energy Policy*, *28*, 935-946. doi: S0301-4215(00)00079-3

Hetemäki, L., Kangas, J., & Peltola, H. (2022). *Forest bioeconomy and climate change.* Cham: Springer Nature. doi: 10.1007/978-3-030-99206-4

Hébert, J. M. (2020). *Algebraic resource accounting for transfers and transformations.* Unpublished master's thesis, University of Copenhagen.

Jackson, L. (2021). *Environmental progress report.* Apple Inc. Retrieved 2022-10-9, from https://www.apple.com/environment/pdf/Apple_Environmental_Progress_Report_2021.pdf

Kirchherr, J., Reike, D., & Hekkert, M. (2017, September). Conceptualizing the circular economy: An analysis of 114 definitions. *Resources, Conservation and Recycling*, *127*, 221–232. doi: 10.1016/j.resconrec.2017.09.005

Kotsialou, G., Kuralbayeva, K., & Laing, T. (2022, May). Blockchain's potential in forest offsets, the voluntary carbon markets and redd+. *Environmental Conservation*, *49*, 137–145. doi: 10.1017/S0376892922000157

Marke, A., Mehling, M. A., & de Andrade Correa, F. (2022). *Governing carbon markets with distributed ledger technology.* University Printing House, Cambridge cb2 8bs, United Kingdom: Cambridge University Press. doi: 10.1017/9781108919166.003

McCarthy, W. E. (1982, July). The rea accounting model: A generalized framework for accounting systems in a shared data environment. *The Accounting Review*, *3*, 554-578.

Mehrabi, Z., McDowell, M. J., Ricciardi, V., Levers, C., Martinez, J. D., Mehrabi, N., ... Jarvis, A. (2021, February). The global divide in data-driven farming. *Nature Sustainability*, *4*, 154–160. doi: 10.1038/s41893-020-00631-0

Mertz, O., Grogan, K., Pflugmacher, D., Lestrelin, G., Castella, J. C., Vongvisouk, T., ... Müller, D. (2017, November). Uncertainty in establishing forest reference levels and predicting future forest-based carbon stocks for redd+. *Journal of Land Use Science*, *13*, 1-15. doi: 10.1080/1747423X.2017.1410242

Michelinia, G., Moraesa, R. N., Cunhab, R. N., Costaa, J. M. H., & Omettoa, A. R. (2017). From linear to circular economy: Pss conducting the transition. *The 9th CIRP IPSS Conference: Circular Perspectives on Product/Service-Systems*, *9*, 2-6. doi: 10.1016/j.procir.2017.03.012

Montserrat, A., & Sohngen, B. (2009). How big is leakage from forestry carbon credits? estimates from a global model. *IOP Conf. Ser.: Earth Environ. Sci. 052011*, *6*. doi: 10.1088/1755-1307/6/5/052011

Nurbaya, S., Efransjah, Murniningtyas, S., Erwinsyah, & Muttaqin, M. Z. (2022, May). *The state of indonesia's forests 2022.* Ministry of Environment and Forestry, Republic of Indonesia. Retrieved 2022-11-9, from https://www.menlhk.go.id/uploads/site/post/1664894502.pdf

Pearce, D. W., & Turner, R. K. (1990). *Economic of natural resources and the environment.* Hertfordshire HP2 4RG: Harvester Wheatsheaf.

Reichle, D. E. (2020). *The global carbon cycle and climate change.* Oak Ridge National Laboratory: Elsevier. doi: 10.1016/B978-0-12-820244-9.00001-9

Saidani, M., Pochat, S. L., Monteil, A., Yannou, B., Garcia, J., & Osset, P. (2022).

Benchmark of circularity indicators and links with life cycle assessment. *E3S Web of Conferences*, *349*. doi: 10.1051/e3sconf/202234901004

Sharan, K., & Späth, P. (2022). *Model-view-controller pattern.* Berkeley, CA: Apress LLC. doi: 10.1007/978-1-4842-7848-2_11

Simonet, G., Karsenty, A., Newton, P., de Perthuis, C., Schaap, B., & Seyller, C. (2015, July). *Redd+ projects in 2014: an overview based on a new database and typology* (Vol. 32).

Thamo, T., & Pannell, D. J. (2015). Challenges in developing effective policy for soil carbon sequestration: perspectives on additionality, leakage, and permanence. *Climate Policy*, *16*(8), 973–992. doi: 10.1080/14693062.2015.1075372

What are carbon markets? (2009). Overseas Development Institute.

Who cares wins. (2005, October). onValues Investment Strategies and Research Ltd.

Yadav, A. (2022, January). *Fighting the triple planetary crisis of our times.* Athena Information Solutions Pvt. Ltd. Retrieved 2022-12-06, from https://www.proquest.com/magazines/fighting-triple-planetary-crisis-our-times/docview/2620099609/se-2?accountid=13607