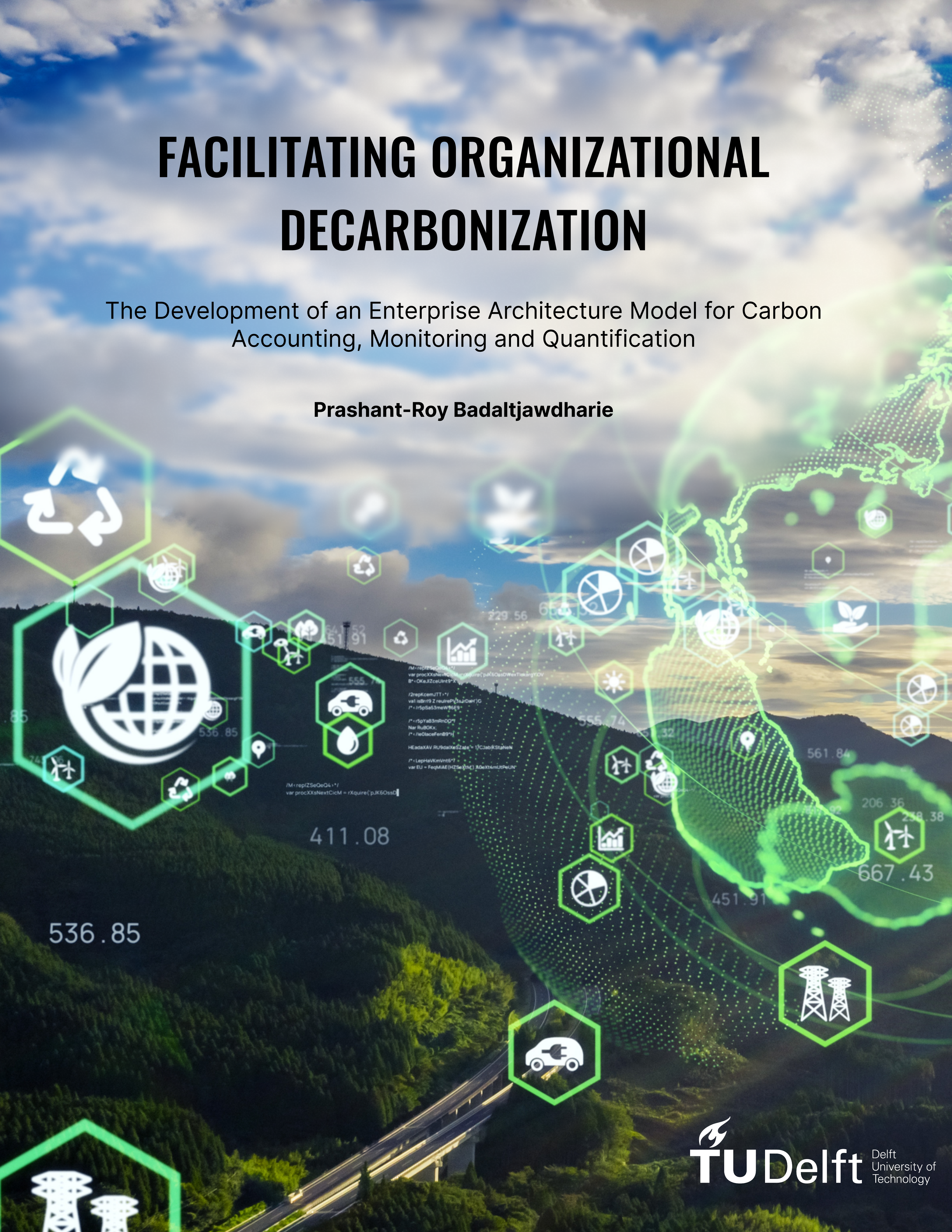


FACILITATING ORGANIZATIONAL DECARBONIZATION

The Development of an Enterprise Architecture Model for Carbon Accounting, Monitoring and Quantification

Prashant-Roy Badaltjawdharie

TU Delft
Delft University of Technology

This page is intentionally left blank.

Facilitating Organizational Decarbonization: The Development of an Enterprise Architecture Model for Carbon Accounting, Monitoring and Quantification

Master Thesis submitted to Delft University of Technology

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

In **Management of Technology**

Faculty of Technology, Policy and Management

by

Prashant Badaltjawdharie

Student number: 4304136

To be defended in public on July 20, 2022

Graduation Committee

Chair: Prof. dr. K. (Kornelis) Blok	Group: E&I (TPM)
First Supervisor: dr. J. (Jolien) Ubacht	Group: ICT (TPM)
Advisor: dr. I. (Ivan) Ligardo Herrera	Group: E&I (TPM)
First External Supervisor: D. (Dennis) Coll	CGI
Second External Supervisor: R. (Ramon) Bouter	CGI

This page is intentionally left blank.

Acknowledgements

Dear reader,

This master thesis marks the end of my time as a student at the Delft University of Technology. I sincerely thank the graduation committee for the academic guidance given during the research. Jolien and Ivan took the time for weekly meetings, even during the pandemic. Without their continuous feedback and support I could not have finished the thesis. Jolien gave valuable feedback regarding the ICT aspects of the thesis, and Ivan gave valuable feedback regarding energy related aspects of the thesis. Furthermore, Professor Blok's feedback on energy related aspects was crucial in producing this thesis; from giving feedback on updating global warming potential values to explaining policies with regards to greenhouse gas reporting and emission targets, all of it was valuable. I want to thank Professor Blok for taking the role of chair and for managing the greenlight meeting and the master thesis defense.

I sincerely want to thank Dennis, my supervisor from CGI. Dennis was always full of energy and helped discussing thesis topics in the beginning stages of the research. Furthermore, discussing technological developments with Dennis was always enjoyable. Dennis helped organizing 13 expert interviews, which was a crucial part of requirements elicitation. I want to thank Ramon for organizing interviews and introducing me to CGI members as well. The ICT-architects of CGI helped me with developing, evaluating, improving and demonstrating of the enterprise architecture models. I sincerely thank Wouter, Marc, Harold, Eltjo, Michiel and 'Al' from Canada. Without their input I could not have finished the thesis.

Finally, I want to thank my parents and friends for their support.

I sincerely hope this thesis is an interesting read.

Prashant Badaltjawdharie,
Capelle aan den IJssel, July 2022

Executive Summary

Industries, businesses and even the entire population are experiencing the adverse effects of climate change, more specifically, global warming. Both national and international policies aim to inhibit growth of global warming. The Dutch parliament closed a coalition agreement. Climate and energy is an important part of the coalition agreement, making up 5 pages out of the 50-page document. The ambition is to reduce emissions up from 49% to 55% compared to 1990 levels - in line with the strengthened ambitions of the European Union (EU). The coalition partners decided to implement policies targeting 60% emission reduction. Looking forward, for the period beyond 2030 emission reduction ambitions have been formulated: 70% in 2035, 80% in 2040. In the coalition agreement it is stated that for the top 10 - 20 largest industrial emitters binding emission reduction agreements will be made. The Netherlands furthermore signed the Paris Climate Agreement. The Paris Climate Agreement aims to limit global warming to well below 2.0°C and pursues efforts to limit it to 1.5°C, since a higher temperature rise will have severe impacts on the planet and human life. The EU introduced emissions trading in 2005, targeting energy-intensive facilities to register CO_2-emissions in annual financial statements. The integration of aspects of climate change mitigation into accounting is called carbon accounting. KPMG identified that a well defined reporting strategy is a success factor valid for any business. Aforementioned success factor ensures that results can be proven, and embeds achievements in external narratives, as climate impact has become an increasingly important component of a brand. Understanding and planning for the likely profound implications of decarbonization on business is crucial as the business response will meet increasing pressure from governments, consumers, employees, investors and lenders.

The aformentioned policies, targets and potential adverse effects on an organization's image, form an incentive for organizations to review decarbonization strategies. The initial step is to develop a carbon-accounting strategy on organizational level. The organization focused on in this research is CGI; an Information and Communications Technology (ICT)- and business consulting organization. CGI aims to develop a carbon-accounting solution, which is based on the reference architecture of the Open Source Data Universe (OSDU)-platform. Developing a Carbon Accounting Solution (CAS) on the OSDU-platform, allows CGI to: reap benefits of platform-based applications, establish their own carbon-accounting strategy, but also to offer the CAS as a Software as a Service (SaaS)-solution for clients. At the heart of this thesis is researching how ICT can enable automatic carbon-accounting on an organizational level in emission scope 2. Scope 2 emissions come from generation of acquired and consumed electricity, steam, heating or cooling (collectively referred to as 'electricity') and are considered an indirect emission source. Scope 2 emissions represent one of the largest sources of Greenhouse Gas (GHG)-emissions globally (1/3 of global GHG-emissions). In this thesis an Enterprise Architecture Model (EAM) will be developed, which displays how ICT-architecture facilitates the business process of carbon-accounting. Enterprise Architecture (EA) is a coherent set of principles, methods, and models to design and realize an enterprise's organizational structure, business processes, information systems, and infrastructure. The modelling language ArchiMate will be used to develop the EAM, which is aligned with the Open Group's Architecture Framework (TOGAF). The research objective is explicated by the main research question: *"Which Enterprise Architecture Model (EAM) is needed for a Carbon Accounting Solution (CAS) to facilitate carbon accounting, monitoring and quantification for organizations?"*.

The main research question is addressed by following activities of the Design Science Research Methodology (DSRM): problem identification and motivation, definition of objectives for the solution, design and development, demonstration, evaluation and communication. The output of the DSRM's activities is an artifact, which in this thesis is the EAM. The research started with a literature review, which showed that information requirements for an EAM for carbon-accounting are missing. The requirements for an EAM in this thesis were elicited through 13 expert interviews and 7 ICT-architecture sessions with 5 ICT-architects. The requirements served as input for the development of an EAM. The knowledge gap was bridged by eliciting requirements for CAS, using academic literature to strengthen requirements and developing the final EAM. In this case, the knowledge gap consisted of the lack of requirements for Carbon Accounting Solutions and lack of Enterprise Architecture Models. The EAM shows how the business process of carbon accounting is supported by a carbon accounting application, which aggregates data that is loaded in CAS by the corporate social responsibility (CSR)-manager via a carbon accounting User Interface (UI). The carbon accounting UI uses and is used by the carbon

accounting & building usage Application Programming Interface (API) which in turn used and is used by two services; carbon accounting & building usage monitoring. Building usage refers to the electricity consumption of a building, resulting in indirect CO_2 emissions. The services are realized by the respective carbon accounting component and building usage monitoring component, where the former generates carbon accounting data and the latter building usage data. The CAS runs on a cloud server (front-end), together with the Cron scheduler and the Database Management System (DBMS). Cron ensures monthly generation of carbon accounting data in JavaScript Object Notation (JSON)-format, the DBMS provides CAS with data and facilitates data storage. Via a private network, there is communication between the carbon accounting & building usage API and the carbon accounting UI (back-end), where the latter serves the CSR-manager. The EAM is modifiable, since the carbon accounting & building usage API can be connected to new services, which in turn aggregate different data. Furthermore, the EAM is adaptable, since CAS can run on any cloud server and can use multiple cloud services. The client using CAS can be outside CGI's organization, if the client is connected to the private network which connects to CAS. Both the modifiability and adaptability properties do not affect the working of the EAM. Modifiability is important because the Carbon Accounting Solution can be effectively and efficiently modified without introducing defects or degrading existing product quality. Adaptability is important because it allows the Carbon Accounting Solution to use different cloud servers. Furthermore, adaptability allows multiple users to use Carbon Accounting Solution if there is a connection with the private network via the carbon accounting UI and carbon accounting & building usage API.

The thesis adds to the academic body of knowledge because it is the first Enterprise Architecture Model which shows how carbon accounting can be facilitated on an organization level. Furthermore, key business processes, policies, actors, application architecture and the architecture of physical infrastructure were identified in the thesis. The thesis provides a comprehensive overview and shows relationships between aforementioned. The elicited requirements for developing a Carbon Accounting Solution may serve as input for more advanced development of carbon accounting systems.

The societal contribution of the thesis consists of indirect environmental benefits the Enterprise Architecture Model will initiate and facilitate. When (energy-intensive) organizations use the Enterprise Architecture Model to adopt a Carbon Accounting Solution, more insight in current emissions will be generated through carbon accounts. Organizations can act upon insights which are data-driven by adopting mitigating efforts to halt indirect emissions. Finally, CAS enables monitoring the impacts organizational interventions have on the reduction of CO_2-emissions.

Contents

List of Figures

List of Tables

1 Introduction

This chapter introduces the master thesis topic. In section 1.1 a short description of the background and societal relevance of the research is given. Furthermore, it is explained why businesses are subjugated to increasing pressure from internal and external stakeholders to reduce CO_2-emissions. In section 1.2 the problem statement from an academic perspective is given. More specifically, it is made clear that the academic body of knowledge regarding carbon accounting with the use of solutions built on platforms is incomplete. Section 1.3 states the design objective of the thesis, which will result in the deliverable of an Enterprise Architecture Model (EAM). The research approach is described in section 1.4, which furthermore includes the research questions and sub-deliverables. The Research Flow Diagram (RFD) in section 1.5 summarizes the research strategy, deliverables, and shows the structure of the thesis.

1.1 Background

Both national and international policies are forcing industries and businesses to reduce Greenhouse Gas (GHG) emissions. In December 2021, four parties in Dutch parliament closed a coalition agreement. Climate and energy is an important part of the coalition agreement, making up 5 pages out of the 50-page document [1]. The ambition is to reduce emissions up from 49% to 55% compared to 1990 levels - in line with the strengthened ambitions of the European Union (EU) [1]. According to Blok [1], the coalition partners decided to implement policies targeting 60% emission reduction. Looking forward, for the period beyond 2030 emission reduction ambitions have been formulated: 70% in 2035, 80% in 2040 [1]. In the coalition agreement it is stated that for the top 10 - 20 largest industrial emitters binding emission reduction agreements will be made [1]. The aforementioned is the responsibility of the minister of Climate and Energy. The Netherlands furthermore signed the Paris Climate Agreement. The Paris Climate Agreement aims to limit global warming to well below $2.0°C$ and pursues efforts to limit it to $1.5°C$ [2], since a higher temperature rise will have severe impacts on the planet and human life [3]. According to the United Nations Framework Convention on Climate Change (UNFCCC), the Paris Climate Agreement envisions to fully realize technology development and transfer for both improving resilience to climate change and reducing GHG emissions [2].

To comply with the environmental, social and governance (ESG) targets and policies, industries and organizations need to review decarbonization strategies to reduce CO_2-emissions. To add to the complexity, the EU introduced emissions trading in 2005. The implementation of emissions trading forces power generation plants and energy-intensive facilities to enter CO_2-emissions in annual financial statements [4]. The integration of aspects of climate change mitigation into accounting is called carbon accounting [5]. This ties in with organizations that are reviewing decarbonization strategies to reduce GHG emissions in order to achieve net-zero targets. Decarbonization strategies vary across industries and organizations. However, according to KPMG [6] a key success factor of the decarbonization strategy for any organization is a well defined reporting strategy for both internal and external stakeholders. KPMG [6] furthermore states that (a) reliable reports are necessary to prove results, as words and pledges are no longer enough and (b) achievements need to be embedded in external narratives as climate impact has become an increasingly important component of a brand. No organization wants to weaken a strong position in the market due to loss of image. Understanding and planning for the likely profound implications of decarbonization on organizations is crucial as the organizational response will meet increasing pressure from governments, consumers, employees, investors and lenders [6].

1.2 Problem statement

Subsection 1.1 elaborated on the increased pressure on industries and organizations to review decarbonization strategies. More specifically, the carbon accounting and reporting practices need to be more sophisticated to foster organizational decarbonization. To make the problem more challenging, net-zero carbon targets are forcing organizations to accelerate carbon accounting practices. In this master thesis, the research focuses on Information and Communications Technology (ICT) as a facilitator for carbon accounting, monitoring, and quantification on an organizational level. Where the facilitator of the aforementioned are Software as a Service (SaaS) solutions based on platform technology, more specifically the OSDU-Platform's reference architecture. In subsection 1.1, a general definition of carbon accounting was given; the integration of aspects of climate change mitigation into accounting.

A more specific definition of carbon accounting, which furthermore is in line with the research at hand, is given by Hespenheide, Pavlovsky, and McElroy [5]. Carbon accounting refers to "to the activity of measuring [direct and indirect] carbon emissions and removals and retaining an ongoing inventory of operations-based emissions" [5, p. 57], whereas the measurement, monitoring, and reporting can be voluntary or mandatory. Carbon accounting according to this definition can be the foundation for emission reductions, cost savings, and trading of emissions allowances and offset credits [5]. With regards to decarbonization, a solution based on the OSDU-Platform's reference architecture could serve the goal of carbon accounting.

The concept of organizational decarbonization is relatively unexplored in the academic literature. More specifically, carbon accounting, monitoring and quantification with the use of solutions based on data platforms to foster carbon emission reduction is barely explored. However, the International Energy Agency (IEA) concludes that: "digitally interconnected systems could fundamentally transform the current energy industry" [7]. The IEA [8] furthermore estimates that 40% of global CO_2-emissions could be reduced through energy efficiency improvement. Although the IEA's conclusion is geared towards the energy industry, it doesn't mean that the conclusion does not hold value for organizations. In Section 2 a literature review is presented regarding data platforms and carbon accounting.

1.3 Research objective and deliverable

The goal of this master thesis is to perform exploratory research in the form of a case study regarding the ICT-architecture of a SaaS-solution based on the reference architecture of the OSDU's data platform to facilitate carbon accounting, monitoring, and quantification in emission scope 2 on an organizational level. The goal can be further explicated. If platform technology is used for carbon accounting, goals for corporate sustainability can be set and hopefully achieved. Organizations can furthermore initiate carbon removal efforts based on data driven decision making, and prove the efficacy of these efforts through the reporting strategy. The organization Conseillers en Gestion et Informatique (CGI) is the problem owner. CGI is a Canadian business consulting company in the field of ICT. Data is an important building block that when used well could contribute to CGI's goal to become a net-zero carbon organization by 2030, by means of data driven decision making through carbon accounting and monitoring. How value can be created by using a Carbon Accounting Solution (CAS) is at the heart of this study. More specifically, CO_2-emissions in emission scope 2 will be targeted in the thesis. Emissions in scope 2 come from the generation of acquired and consumed electricity, steam, heat or cooling (collectively referred to as "electricity") and are considered as an indirect emission source [9]. The reasoning for the focus on scope 2 CO_2 emissions is as follows. According to KPMG [6], a key success factor for every organization/industry with respect to the decarbonization strategy is to pursue partnerships and engage supply chain partners in a common decarbonization target. CGI could therefore enrich their Carbon Accounting Solution (CAS) with data from the energy suppliers. This is in line with the success factor for decarbonization strategies applicable to all industries and organizations. More importantly, Scope 2 represents one of the largest sources of GHG emissions globally: the generation of electricity and heat now accounts for at least a third of global GHG emissions [9]. Focusing on emission scope 2 would therefore be a logical decision. Other emission scopes include scope 1 and scope 3. Scope 1 emissions are direct emissions that a site directly causes and controls, and scope 3 emissions are indirect emissions that a site causes to occur but where it does not control the asset [10], [11].

The deliverable of this case study will be an Enterprise Architecture Model (EAM). The EAM will be a blueprint that shows the ICT-Architecture of a CAS, which facilitates carbon accounting, monitoring and quantification in scope 2 on an organizational level. Carbon accounting reports are valuable information sources for internal- and external stakeholders. Ultimately, through a consistent, accurate and reliable reporting strategy, carbon removal initiatives can be initiated and effects of these initiatives can be monitored. The research results furthermore serve as input for further development efforts of CGI's Pivot, as academic material for the scientific body of knowledge, and as input for further research.

1.4 Research approach

Chapter 1.4 is structured as follows. Section 1.4.1 introduces CGI's Software as a Service (SaaS)-solution Pivot developed on the Open Source Data Universe (OSDU)-Platform and the main research question. The research strategy, methodology, activities, sub-research-questions and deliverables are specified in section 1.4.2.

1.4.1 Main research question

The research in this thesis will explore how CGI can develop a Carbon Accounting Solution based on platform technology to facilitate carbon accounting, monitoring and quantification for organizational decarbonization purposes in scope 2. CGI is affiliated with the OSDU-Platform, which is an open exploration and production data platform to accelerate innovation and transformation in the industry [12]. The OSDU-Platform initially targeted the oil- and gas industry (OSDU is an abbreviation for Open Subsurface Data Universe), however currently businesses from other industries are able to join the OSDU-Platform. According to CGI [13] and Feder [14], oil- and gas businesses create enormous data volumes which are stored in (application)-data silos. Data silos hamper the ability to fully exploit data in terms of value creation for businesses. Furthermore, data silos limit the ability of enabling technologies such as Artificial Intelligence (AI) and Machine Learning (ML), which foster digital transformation. The OSDU-Platform aims to eliminate data silos and vendor lock-in effects.

CGI developed the Platform as a Service (PaaS) solution Pivot, which is implemented as a SaaS implementation to capitalize on the benefits of the OSDU-Platform. See Fig. 1 for an illustration of the platform ecosystem.

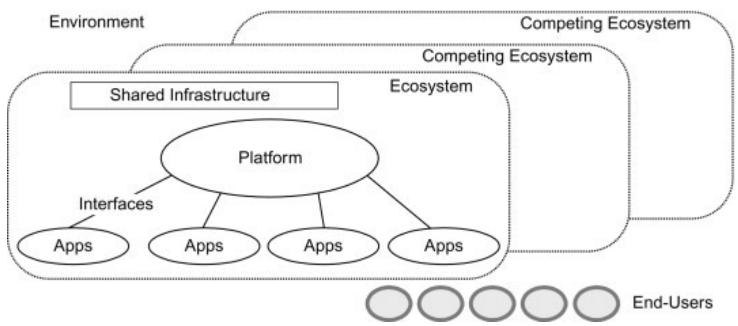


Fig. 1: The platform ecosystem [15], Platform: OSDU (PaaS), Apps: Pivot (Saas), End-Users: CGI.

Pivot offers oil- and gas businesses, academic institutions, renewable energy organizations, and other members of the subsurface community a starting point to explore and migrate to the OSDU Data Platform [12]. The OSDU Data Platform and Pivot are primarily used for data exploration of oil- and gas businesses to optimize business processes. However, the research objective of the thesis is to identify the EAM of a Carbon Accounting Solution developed on the OSDU-Platform / based on the OSDU's reference architecture, which facilitates carbon accounting, monitoring and quantification to foster organizational decarbonization in scope 2. The aforementioned CAS is not yet developed by CGI. Since a success factor for decarbonization strategies involves pursuing partnerships and engagement of supply chain partners, the relevant actors and data sources need to be identified. Scope 2 entails the indirect CO_2-emissions through imported utilities [10]. See table 1 for identified actors and data sources.

Table 1: Actors in scope 2 and data sources

Actors in scope 2	**Data sources**
Electricity suppliers	Emission data, smart- and real-time data, invoices
Suppliers of heating and cooling	Emission data, smart-and real-time data, invoices

Not only CGI has lack of knowledge regarding carbon accounting solutions based upon platform technology, the research domain regarding this topic is also underdeveloped. The study conducted in the thesis will result in the initial step of developing an Enterprise Architecture Model (EAM) of a CAS which shows the ICT-Architecture that facilitates carbon accounting, monitoring and quantification in emission scope 2. To capture the research objective, the following main research question is formulated:

"Which Enterprise Architecture Model (EAM) is needed for a Carbon Accounting Solution (CAS) to facilitate carbon accounting, monitoring and quantification for organizations?"

The main research question will be answered through Design Science Research (DSR), see subsequent sections. The first requirements for the EAM will be identified through literature research and expert interviews. The requirements serve as input for an initial EAM. The initial EAM will be demonstrated, evaluated and improved through feedback sessions with CGI's ICT-Architects.

1.4.2 Design Science Research

The research objective of developing an EA-model of a Carbon Accounting Solution, is in line with the general objective of Design Science Research. According to Johannesson and Perjons [16, p. 7], *design science* is the scientific study and creation of artifacts as they are developed and used by people with the goal of solving practical problems of general interest. More specifically, Hevner, March, Park, *et al.* [17, p. 77] define Information Technology (IT) artifacts as *constructs* (vocabulary and symbols), *models* (abstractions and representations), *methods* (algorithms and practices), and *instantiations* (implemented and prototype systems). In this case, the developed IT artifact is an EA-model of a Carbon Accounting Solution (CAS), which should be input for the development of a SaaS based upon the OSDU's reference architecture.

Design science aims to make the world better by solving/mitigating practical problems through the design of artifacts. However, it is paramount to understand that the knowledge produced while designing an artifact is of equal importance. Research outcomes for design science are artifacts and knowledge [16, p. 7]. Hevner, March, Park, *et al.* [17, p. 78] state that to understand and appreciate design science, the dichotomy of design as both a process (set of activities) and a product (artifact) must be faced. The design process is a sequence of expert activities that produces an innovative product (i.e., the design artifact). The evaluation of the artifact then provides feedback information and a better understanding of the problem in order to improve both the quality of the product and the design process [17, p. 78].

Design science research projects need to fulfill three requirements to develop a working solution. Johannesson and Perjons [16, p. 8] mention the following: (1) rigorous research methods are required to produce new knowledge of general interest, (2) the knowledge produced has to be related to an already existing knowledge base, in order to ensure that proposed results are both well founded and original, and (3) the new results should be communicated to both practitioners and researchers.

1.4.3 The three-cycle view

Hevner [18] extended the design science research framework with the three-cycle view in order to increase research quality in Information Systems (IS). The relevance cycle bridges the contextual environment of the research project with the design science activities [18]. The Rigor Cycle connects the design science activities with the knowledge base of scientific foundations, experience, and expertise that informs the research project [18]. The central Design Cycle iterates between the core activities of building and evaluating the design artifacts and processes of the research [18]. See Fig. 2 for an illustration of the design science research cycles, and how the cycles relate to the thesis.

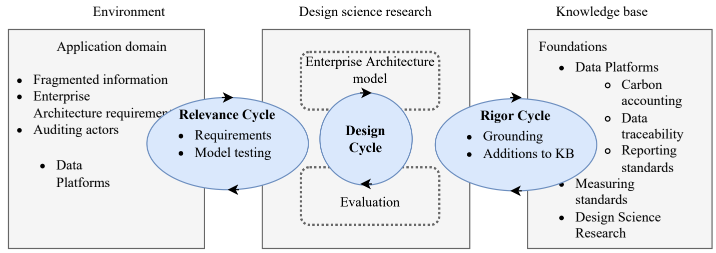


Fig. 2: Design Science Research Cycles, derived from Hevner [18, p. 88]

1.4.4 Design Science Research Methodology

To carry out research based on design science principles, Peffers, Tuunanen, Rothenberger, *et al.* [19] developed a Design Science Research Methodology (DSRM) based on a consensus-building approach to produce the design. The result is a methodology which fulfils the design science requirements as mentioned in section 1.4.2 and [16, p. 8]. The DSRM recognizes six activities which need to be completed in an iterative fashion; (1) problem identification and motivation, (2) define the objectives for a solution, (3) design and development, (4) demonstration, (5) evaluation, and (6) communication [19].

The DSRM is a flexible model, since Peffers, Tuunanen, Rothenberger, *et al.* [19] mention that depending on the approach (problem-centered, objective-centered, design- and development-centered) researchers can start with different activities. Furthermore, design science projects are always carried out in an iterative way, moving back and forth between different activities [19, p. 76-77]. The activities are logically related through input-output relationships; every activity can receive input and produce output for any other activity [19, p. 76-77]. See Fig. 3 for an illustration of the activities and input-output relations.

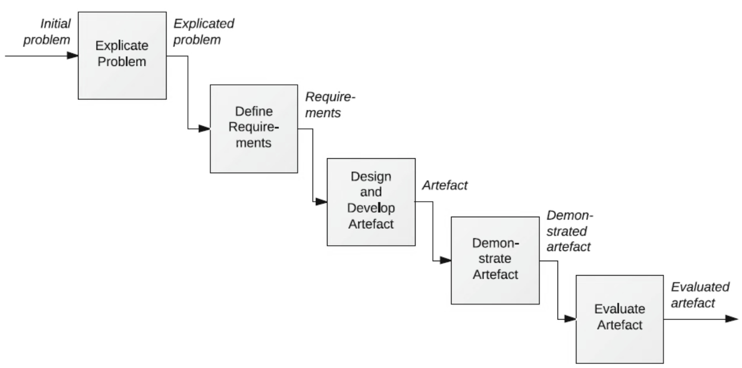


Fig. 3: Overview of method for the Design Science Research Methodology (DSRM) [19, copied from p. 77].

In order to answer the main research question, the activities need to be executed. If the aforementioned activities are executed, the design cycles as in Fig. 2 will be fulfilled and the environment, design science research, and knowledge base will be connected.

Activity one: Problem identification and motivation

In this activity the research problem is introduced and substantiated. It should furthermore be clear that the research problem is significant in both general and global practices [16]. The value of the artifact needs to be captured in order to (1) motivate the researcher and audience to pursue the solution and (2) for result acceptance and understanding of the researchers' perception of the problem [19].

This activity is addressed by section 1 of the chapter. This section highlights the research problem on a business, industrial, and global level. Furthermore, the academic and societal relevance of the research is demonstrated. Activity one is completed by answering sub-research question one (SRQ_1), which furthermore contributes to the knowledge base in chapter 2 by means of a literature review:

$SRQ_{1:}$ *What is the knowledge base on which the development of an Enterprise Architecture Model (EAM) for a Carbon Accounting Solution (CAS) should rely?*

The literature review will produce fundamental information regarding platform technology architecture and carbon accounting.

Activity two: Define the objectives for the solution

The objectives of the solution (artifact) need to be specified in activity two. According to Johannesson and Perjons [16] the requirements for the solution to the problem need to be identified. The requirements furthermore are the input for structure and environment, besides functionality [16].

The objectives of the solution can be of quantitative and/or qualitative nature [19]. The former shows a difference between current inefficient states of the world versus the ideal situation. The latter could be descriptions wherein new artifacts support solutions to problems, such as the EA-model. Resources required for this include knowledge of the state of problems and current solutions, if any, and their efficacy [19]. Activity two will be fulfilled by answering sub-research question two (SRQ_2):

SRQ_2: *What are the requirements for a Carbon Accounting Solution (CAS) to develop an Enterprise Architecture Model (EAM)?*

SRQ_2 will be addressed by literature review and expert interviews. Organizational decarbonization in scope 2 entails a common decarbonization strategy with the energy providers. This is in line with the key success factor for enterprise decarbonization as mentioned in [6], which is; a well defined reporting strategy for both internal and external stakeholders. However, this means that data types and further (data) requirements need to be identified which serve as input for the CAS. Energy suppliers could potentially enrich the CAS with (smart)-data from cloud and/or Internet of Things (IoT) devices, such as smart-meters. Invoices from electricity suppliers contain valuable information, such as monthly and annual kilowatt-hour (kWh) usage. Besides data source and data type identification, and data requirements elicitation, the data configuration also needs to be determined. The results of SRQ_2 serve as input for design and development of an initial EAM.

Activity three: Design and development

The Design and Development activity creates an artifact that addresses the explicated problem and fulfills the defined requirements. Designing an artifact includes determining its functionality as well as its structure [16]. The artifact in this case will be an EAM of a Carbon Accounting Solution (CAS). The EAM should showcase the ICT-Architecture of a CAS to facilitate carbon accounting. Activity three will be fulfilled by answering sub-research question three (SRQ_3).

SRQ_3: *How can the identified requirements for a Carbon Accounting Solution (CAS) serve as a building block for the development of CAS's Enterprise Architecture Model (EAM)?*

SRQ_3 uses the output of SRQ_2. An initial EA-model will be developed by answering SRQ_3 through literature review and expert interviews.

Activity four: Demonstration and evaluation

The Demonstrate and evaluate activity uses the developed artifact in an illustrative or real-life case, sometimes called a "proof of concept", thereby proving the feasibility of the artifact. The demonstration will show that the artifact actually can solve an instance of the problem [16]. Sub-research question four (SRQ_4) will address this activity:

SRQ_4: *"To what extent does the Carbon Accounting Solution (CAS)'s initial Enterprise Architecture Model (EAM) comply with the requirements?"*

Demonstration, evaluation and improvement of the EAM was achieved through feedback received from CGI's ICT-Architecture experts.

Activity five: Communicate artifact

This activity uses the output of activity four to finalize the thesis by answering the main research question. Furthermore, the contribution of the research on an academic and organizational level is elaborated upon. Recommendations for further research will be given. A reflection on the research is part of activity five.

An overview of the research strategy and deliverables is presented in Section 1.5, Fig. 4.

1.5 Research flow Diagram

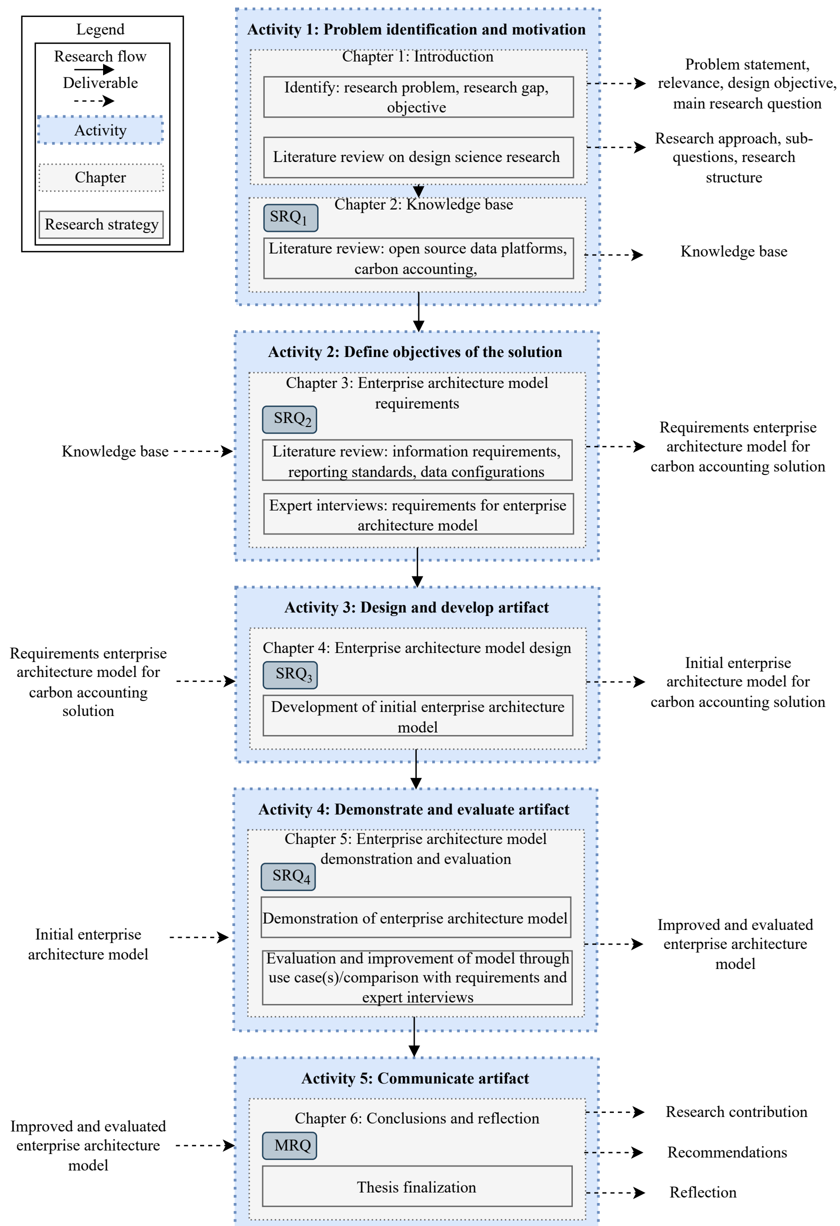


Fig. 4: Research Flow Diagram.

2 Knowledge base

This chapter addresses the first activity in the DSRM and gives an overview of key concepts that are relevant for the thesis. In the first activity, sub-research-question one (SRQ_1) is addressed:

$SRQ_{1:}$ *What is the knowledge base on which the development of an Enterprise Architecture Model (EAM) for a Carbon Accounting Solution (CAS) should rely?*

The research problem is further substantiated in chapter 2. By reviewing state of the art literature, information is collected regarding open source data platform ecosystems in section 2.1 and carbon accounting and footprinting in section 2.2. The information regarding open source data platform ecosystems clarifies the ICT-architecture of such ecosystems, and which ICT-architecture to develop for the CAS. The chapter is concluded with section 2.3, where the knowledge base is summarized. The summary of the knowledge base serves as input for the next activity in the DSRM, which is the development of an initial EA-model.

2.1 Open source data platforms and data platform ecosystems

According to Tiwana [20], data platform ecosystems are becoming the dominant model for the software industry and digital services. The utility of platforms is increasingly shaped by the ecosystem that surrounds it. For example Apple's iOS platform that includes the iPhone, iPod, and iPad. The iOS' value to 365 million users comes largely from the 800,000 complementary apps over which Apple has little ownership [20]. Tiwana [20, p. 5] states that "platforms are designed to leverage the expertise of a diverse developer community—with ingenuity, hunger, skills, and an appreciation of user needs that platform owners might not possess". The goal of such platforms is to accelerate the development of new capabilities and foster innovations which the platform's original designers did not foresee initially [20]. Table 2 illustrates core elements of a platform ecosystem.

Table 2: Core elements of a platform ecosystem, derived from Tiwana [20]

Element	**Definition**	**Example**
Platform	The extensible codebase of a software-based system that provides core functionality shared by apps that interoperate with it, and the interfaces through which they interoperate	iOS, Android, Dropbox, Twitter, AWS, Firefox, Chrome
App	An add-on software subsystem or service that connects to the platform to add functionality to it. Also referred to as a module, extension, plug-in, or add-on	Apps, extensions
Ecosystem	The collection of the platform and the apps specific to it	-
Interfaces	Specifications that describe how the platform and apps interact and exchange information	APIs, protocols
Architecture	A conceptual blueprint that describes how the ecosystem is partitioned into a relatively stable platform and a complementary set of apps that are encouraged to vary, and the design rules binding on both	-

Tiwana [20] identified five drivers why traditional industries are reconfigured along the lines of software-centric platforms abundant in ecosystem-creation opportunities. According to Tiwana [20], these drivers include (1) deepening specialization within industries; (2) the "packetization" of products, services, business processes, and activities; (3) the baking of routine business activities into software; (4) the emergence of the "Internet of Things"; and (5) the growing ubiquity of mobile Internet protocol-based data networks. Aforementioned drivers are summarized in Fig. 5 and the consequences in table 3.

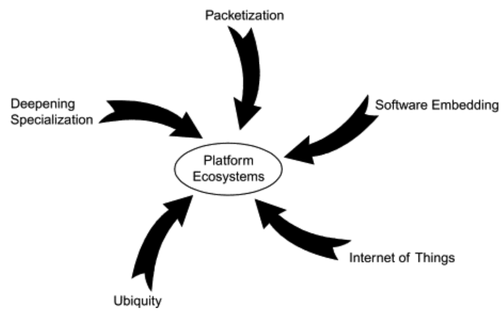


Fig. 5: The five drivers of the migration toward platform-centric business models [20, copied from p. 10].

Table 3: Consequences of the five drivers towards platform-centric business models [20, copied from p. 10].

Driver	**Description**	**Consequences**
Deepening specialization	Increased need for deep expertise due to growing complexity of products and services	• Simultaneously shrinking and expanding firm boundaries • Red Queen effect from clockspeed compression • Increased interdependence among firms
Packetization	Digitization of “something”—an activity, a process, a product, or a service—that was previously not digitized	• Location-independent distribution ability of work • Deepening specialization
Software embedding	Baking a routine business activity into software	• Products-to-services transformation • Morphing physical–digital boundary • Convergence of adjacent industries
Internet of Things	Everyday objects inexpensively gaining the ability to directly talk using an Internet protocol	• Deluge of data streams from networked objects • Context awareness
Ubiquity	The growing omnipresence of cheap and fast wireless Internet data networks	• Loosely coupled networks rival efficiencies of firms • Alters who can participate from where • Alters where services can be delivered • Scale without ownership

Zutshi and Grilo [21] furthermore state that digital platform ecosystems allow for the interaction and exchange of information, in order to provide value to its users. Zutshi and Grilo [21] further list value propositions that platform ecosystems bring to the economic systems: discovery of collaborators, information aggregation and data access, developer tools, price discovery, building of trust, elimination of gatekeepers, and digital delivery of goods, services and value. Fig. 6 illustrates key players and infrastructure components of digital platform ecosystems.

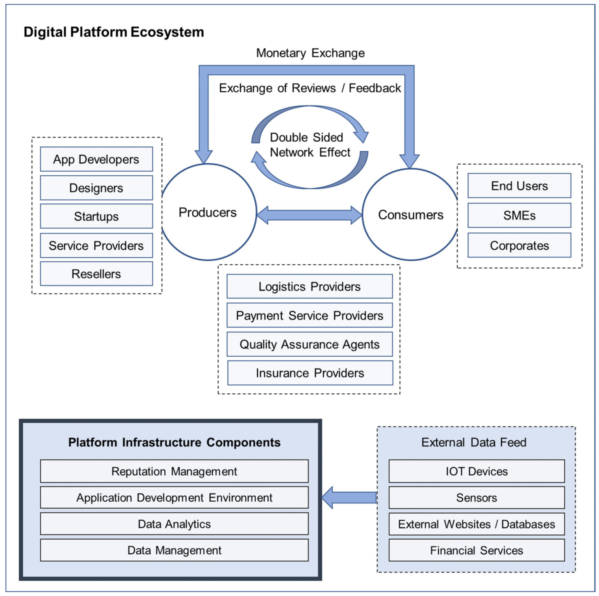


Fig. 6: Key players and infrastructure components of digital platform ecosystems [21].

2.1.1 Ecosystem Architecture

The OSDU platform and the platform implementation Pivot are part of a platform ecosystem. Platform ecosystems are composed of interacting subsystems, and how these subsystems interact is determined by the platform ecosystem's architecture [22, p. 84]. Two subsystems are the platform itself and the portfolio of Apps that augment it [22]. Ecosystem architecture ideally partitions the ecosystem into two types of subsystems: (1) a highly reusable core platform that remains relatively stable and (2) a set of complementary apps that are encouraged to vary [23].

Architecture is a hierarchical concept: Ecosystems can be decomposed to interrelated subsystems such as apps which have architectures [24, p. 413]. Ecosystem architecture can be divided to two levels: (1) the architecture of the platform (platform architecture) and (2) that of an app, the app's microarchitecture [22], see Fig. 7.

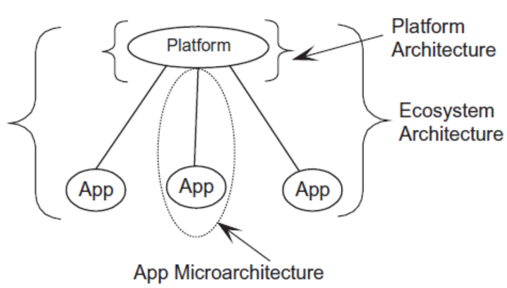


Fig. 7: Ecosystem architecture consists of platform architecture and app microarchitecture [22, copied from p. 85].

Platform architecture includes the core platform and its interfaces [22]. Platform architecture should tell apps both what the platform does and how to use the platform [25]. The latter is a role directly played by the platform's interfaces, which therefore must be treated as an integral part of a platform's architecture [22].

Although the platform has a specific architecture that all apps see, the architecture of individual apps within the same platform can vary from one app to another [22]. Platform architecture imposes constraints on all apps in a platform's ecosystem; therefore many properties of app architectures are correlated with the architecture of the platform [22]. However, the two are rarely identical because there can be considerable variance among apps developed for the same platform [22]. Therefore, an app's "microarchitecture" will define how each individual app interacts, communicates, and interoperates with the platform [22]. An app's internal functionality can be classified in four functional elements as illustrated as in Fig. 8.

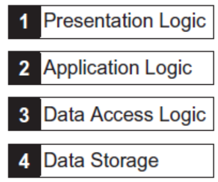


Fig. 8: The four elements of an app's internal functionality [22, copied from p. 86].

For a description of the functional elements which constitute the internal microarchitecture of an app, see Table 4.

Table 4: Four functional elements of the internal microarchitecture of any app, derived from [22, p. 86].

Element	**Description**
1. Presentation logic	An app's presentation logic is where almost all of the interaction with the end-user occurs. It is the part of the application that handles receiving inputs from the end-user and presenting the application's output to the end-user.
2. Application logic	The second function is the core work performed by the application that is distinctive to it. This encompasses the functionality of the app that makes it uniquely valuable to its end-users.
3. Data access logic	The third function is the processing required to access and retrieve data. This often equates with database queries through which user-specified data is retrieved from data storage.
4. Data storage	The last function is data storage. Most apps require data to be stored somewhere in order to be retrieved.

These four elements can be placed on either the client side or the server side in several plausible arrangements [22]. This section shows that developing of solutions on platforms have benefits in terms of accelerating new capability creation, interaction and exchange of information and increased value for its users. Furthermore, it is clear that the platform for this thesis is the OSDU-Platform, which gives a reference architecture. The 'App' will be the Carbon Accounting Solution (CAS), for which an Enterprise Architecture (EA)-model will be developed.

2.2 Carbon accounting and carbon footprinting

The definition of carbon accounting changes according to the scale of analysis: national, project, organizational or product scale [4]. This can create confusion, especially since most definitions are used interchangeably. For example, carbon accounting and carbon footprinting are terms used interchangeably. In section 1.2 the definition of carbon accounting on an organizational scale was defined for the thesis, since the thesis focuses on organizational/business carbon accounting. This definition is as follows: carbon accounting refers to "the activity of measuring [direct and indirect] carbon emissions and removals and retaining an ongoing inventory of operations-based emissions" [5, p. 57], whereas the measurement, monitoring, and reporting can be voluntary or mandatory. Carbon accounting according to this definition can be the foundation for emission reductions, cost savings, and trading of emissions allowances and offset credits [5]. This definition of carbon accounting has an embedded quantification element for carbon emissions. Similarly, the term carbon footprinting on an organizational level, according to Stein and Khare [26, p. 293] entails "the amount of carbon dioxide equivalent Carbon dioxide equivalent (CO_2e) emitted during the operation of a ... plant for one year". Table 5 gives an overview of various carbon accounting and carbon footprinting terms used on various scales. For a complete overview of alternative carbon accounting terms at national, project, organizational or product level, see table A.1 in appendix A. This thesis focuses on the organizational scale.

Table 5: Different scales with alternative definitions for carbon accounting, partially derived from Stechemesser and Guenther [4].

Scale	**Alternatively used terms**
National scale	Account for CO_2 emissions, carbon emissions accounting, (GHG) emission accounting, carbon footprint accounting
Project scale	Project-based greenhouse gas accounting, carbon accounting system
Organizational scale	(Carbon) emissions accounting, accounting for emission rights, GHG project accounting
Product scale	(GHG) emissions accounting, carbon flow accounting, CO_2 accounting

There are several reporting methods which organizations can use for carbon accounting and footprinting. Some are web-based, some are country/area specific, some are designed for industry and some are designed for domestic/consumer use [10]. Most of aforementioned methods are based on the Greenhouse Gas Protocol (GHGP) (www.ghgprotocol.org) which gives a standard format for classifying and reporting emissions. The EN ISO 14064-2:2012 [27] standard gives guidance for the quantification, monitoring and reporting of GHG emissions and removals at organizational scale. The standards for carbon accounting and footprinting provided by the GHG protocol and EN ISO 14064-2:2012 ensure that: (1) organizations can add up emissions across the supply chain without double-counting and (2) ensure that emissions are correctly allocated to the appropriate corporate entity [10].

What is furthermore important for organizations, is the data collection and expression of data in quantification of CO_2. Emissions should be expressed in mass CO_2e-emissions. CO_2e-emissions, or carbon dioxide equivalents are units for comparing the radiative forcing of a GHG to carbon dioxide [10], [27]. In some cases, gases other than CO_2 are emitted, such as methane (CH_4), or nitrous oxide (N_2O) (Kyoto gases). Emissions of these gases are converted to CO_2e, which allows calculations of total CO_2e. This conversion takes place with Global Warming Potential (GWP) values, which are indexes with CO_2 having the index value of 1, and the GWP for all other GHGs is the number of times more warming they cause compared to CO_2 [28]. For an overview of GHGs as described by the Kyoto Protocol with the respective GWPs, see table 6.

Table 6: Kyoto greenhouse gases with global warming potential values, derived from [29]

Nr.	**Greenhouse gas**	**Global warming potential (GWP)**
1	Carbon dioxide (CO_2)	1
2	Methane (CH_4)	27.9
3	Nitrous oxide (N_2O)	273
4	Hydrofluorocarbons (HFCs)	4.84 - 14,600
5	Perfluorocarbons (PFCs)	7,380 – 12,400
6	Sulfur hexafluoride (SF_6)	25,200

2.2.1 Emission scopes

Emissions are grouped by the level of control that a site or organization has over them. Kent [10] states that this gives a 3-part classification as in table 7.

Table 7: Carbon emission classifications, derived from Kent [10] and GHG Protocol [11].

Scope	**Description**
1: direct emissions	Direct emissions cover emissions that a site directly causes and controls.
2: Indirect emissions from imported utilities	All sites import utilities, e.g., electricity, and Scope 2 covers emissions from purchased electricity or other utilities such as imported heat or steam. In this case, the CO_2e is emitted at some distance from the site (organization) by the power station that has generated the electricity, heat or steam.
3: Indirect emissions	Indirect emissions are emissions that a site causes to occur but where it does not control the asset. Common types of indirect emissions include emissions from transport in vehicles owned by other organisations or emissions from outsourced activities or the supply chain.

For an illustration of the emission scopes, see Fig. 9.

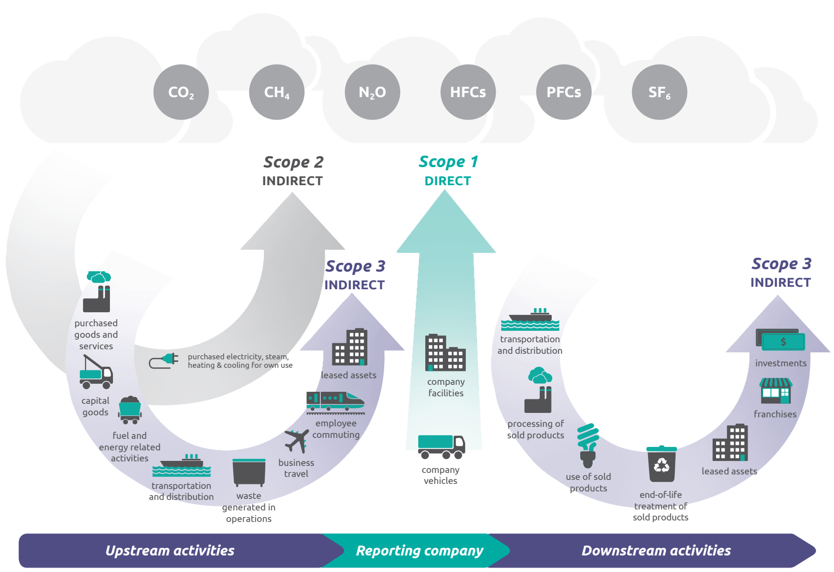


Fig. 9: Emission scopes across the energy value chain [11].

The thesis focuses on carbon accounting, monitoring and quantification in scope 2. On an annual basis, the data required to calculate the carbon footprint is the total number of kWh used in the year and the relevant carbon intensity factor for the supply country. The number of kWh used in the year should be available from the electricity bills and the interval data. The generation of electricity will inevitably result in CO_2e emissions but the size of these emissions depends on the generation method [10]. Nuclear, thermal or hydroelectric generation result in low CO_2e emissions, whereas coal, gas and other fossil fuels result in high CO_2e emissions [10]. According to Kent [10], each country will have a mix of generation for grid electricity that will therefore affect the total emissions. This is also true at the supplier level, i.e., a supplier can have a different emission factor to the country as a whole if their generation profile is significantly different to the country profile [10]. For example, a country generating a large proportion of grid electricity from nuclear sources or renewable energy will have a lower CO_2e emissions factor than a country generating a large proportion of grid electricity from coal or other carbon-based fuels [10]. The factors for any specific country will change with time as the generation pattern changes [10]. Grid electricity is generated at the power station and this results in emissions, but the transmission and distribution of the electricity to the user also results in losses, i.e., transmission and distribution losses [10]. The emission factor at the point of consumption is therefore higher than the emission factor at the point of generation.

2.2.2 Calculating Organizational Carbon Emissions in Scope Two

The previous sections emphasised the importance of the direct activity data, e.g. the amount of electricity used by an organization in kWh. On an organizational level, invoices provided by the electricity supplier should give insight in the monthly and annual direct activity data. Furthermore, the definition of GWP values were given and their significance to convert carbon quantities to Carbon dioxide equivalent (CO_2e) The following steps show how to calculate carbon emissions on an organizational level in scope two. The calculation steps are based on guidelines given by the GHGP in [9].

The first step is the calculation of kilograms of CO_2. See equation 1

$$\sum_{i=1}^{i=n} L_i \times CF_i \tag{1}$$

Where:

L: is the load in [kWh]

CF_i: is the conversion factor in [kg CO_2/kWh]

i: is consumption

The second step is to calculate total Carbon dioxide equivalent (CO_2e) values with the use of Global Warming Potential (GWP) values, see equation 2.

$$kg\ CO_2 \times GWP \tag{2}$$

Where:

$kg\ CO_2$: is the amount of CO_2 in [kg]

GWP: is the Global Warming Potential in [kg CO_2e/kg CO_2]

The third step is to report final scope 2 values in metric tonnes of CO_2e, i.e. tCO_2e.

2.3 Conclusion

Chapter 2 answered SRQ_1 through a literature review: *SRQ_1: "What is the knowledge base on which the development of an Enterprise Architecture Model (EAM) for a Carbon Accounting Solution (CAS) should rely?"*. The knowledge base is the output of SRQ_1 and serves as the foundation for the development of an Enterprise Architecture Model for carbon accounting, monitoring and quantification in emission scope 2 on an organizational level. Concepts from the literature which can support the design of the data architecture model were identified, which should ensure that the results are sound and grounded in the academic literature.

The chapter initially presented platform technology and their ecosystems, and how data platforms can leverage development of new capabilities and innovations. Elements of platform ecosystems were introduced: platform, applications, ecosystems, interfaces and architecture. The drivers for industries to migrate towards software centric platforms were elaborated upon. These include (1) deepening specialization within industries; (2) the "packetization" of products, services, business processes, and activities; (3) the baking of routine business activities into software; (4) the emergence of the "Internet of Things"; and (5) the growing ubiquity of mobile Internet protocol-based data networks.

Subsequently, the ecosystem architecture was described. The ecosystem architecture consists of platform architecture and the application microarchitecture, the latter being the focus of the research. This is called a hierarchical architecture. Platform architecture tells applications what the platform does and how the platform should be used. Architecture of individual applications may vary. Internal functionality of application functionality are classified in (1) presentation logic, (2) application logic, (3) data access logic and (4) data storage.

Carbon accounting as a concept was introduced and important parameters to include in carbon reports and calculations were identified. Carbon accounting refers to “the activity of measuring [direct and indirect] carbon emissions and removals and retaining an ongoing inventory of operations-based emissions” [5, p. 57] The differences between emission scopes were elaborated upon, where scope 1 entails direct emissions, scope 2 entails indirect emissions from imported utilities and scope 3 entails indirect emissions that a site causes to occur but where it does not control the asset. Carbon accounting and reporting standards, such as the greenhouse gas protocol and ISO 14064 were identified. The parameters to calculate Carbon dioxide equivalent were identified, such as GWP values, emission factors and activity data. A method was given to calculate CO_2e. The knowledge base serves as input for the development of an EAM.

3 Enterprise Architecture Model Requirements Analysis

In chapter 3 the second activity of the design science research methodology is addressed. According to Johannesson and Perjons [16], the second activity is defining requirements. The goal is to identify and outline an artifact that can serve as a solution for the problem and to elicit requirements on that specific artifact [16]. In this thesis, the second activity addresses sub-research-question two (SRQ_2):

SRQ_2: *What are the requirements for a Carbon Accounting Solution (CAS) to develop an Enterprise Architecture Model (EAM)?*

According to Johannesson and Perjons [16], the second activity is an extension to the problem explication, where the proposed solution can be used as an outline to examinate the problem. Thus, the question is to be answered by descriptive knowledge that specifies requirements on the artifact [16].

In section 3.1 the fundamentals of requirements engineering are briefly addressed, since they serve as building blocks for the EAM. Section 3.2 gives a rationale for the artefact, which is an EAM. In Section 3.3 a strategy is given to elicit requirements for the EAM. The results of requirements elicitation will be summarized in Section 3.4. Finally, section 3.5 concludes chapter 3 by answering SRQ_2.

3.1 Requirements Engineering

Requirements engineering (RE) is the process of analyzing, specifying, validating, and managing requirements [30]. Johannesson and Perjons [16] state that a requirement is a property of an artefect that is considered as desirable by stakeholders in a practice and that is to be used for guiding the design and development of the artefact. Johannesson and Perjons [16] categorize requirements in types as stated in table 8.

Table 8: Requirement types and descriptions, derived from Johannesson and Perjons [16, p. 103].

Requirement type	**Description**
Functional requirements	Specific requirements: functions of an artefact dependent on the problem to be addressed as well as the needs and wants of stakeholders.
Non-functional requirements (including structural requirements and environmental requirements)	More generic requirements pertaining to design, modularity, availability of information systems

Elmasri and Navathe [31] specify functional requirements as user defined operations (or transactions) that will be applied to databases, including retrievals and updates. Additionally to functional and non-functional requirements, goals on the effects of using an artefact can be formulated [16].

3.2 Outline Artefact

Johannesson and Perjons [16] state that a decision needs to be made to choose which artifact type should be designed to solve the problem, i.e. choosing whether the solution should be a construct, a model, a method, or an instantiation. In this thesis, artifact is the Enterprise Architecture Model (EAM). [16] explains that the next step is requirements elicitation. However, in the development of Information Systems (IS), an initial step, developing design principles, is usually performed first. [32] describe design principles as an organization's basic philosophies that guide the development of the architecture. [32] furthermore explain that design principles have the most far-reaching and significant impact on an organization because they are the most stable element of an architecture. Finally, [32] state that design principles provide guidelines and rationales for examination and re-evaluation of technology plans, and they may serve as a starting point subsequent decisions that affect the architecture. The design principles listed below serve as an initial guideline for the further development of the EAM, and as a step before conducting requirements elicitation.

1. The EAM shows how Carbon dioxide equivalent (CO_2e)-values are calculated automatically on a monthly basis;

2. The EAM shows compliance with the (reporting) standards of The Greenhouse Gas Protocol and EN ISO 14064-1:2012;
3. The EAM shows how carbon accounts are created on a monthly basis which are stored on a database and are retrievable;
4. The EAM shows how data can be loaded and retrieved from the Carbon Accounting Solution (CAS);
5. The EAM shows how a single Carbon Accounting Solution facilitates an organization's carbon accounting, monitoring and quantification in emission scope 2;
6. The EAM shows how the corporate social responsibility (CSR)-manager is facilitated with data regarding emissions in scope 2.

Design principle 5 was developed because current carbon accounting practices occur manually, using a combination of spreadsheets and vendor software. This results in scattered data, inaccurate reports and a lagging reporting strategy. The design principles above give initial guidelines for the development of the EAM.

3.3 Requirements Elicitation and Research Strategy

To elicit requirements for the EAM, primary data was collected through interviews with several CGI members (hereafter 'interviewees'). Primary data collection methods gather data from original sources for the specific purpose of the study [33], and one such method is interviews. Interviews take a direct approach by asking stakeholders about preferred features and explicit requirements in the outlined artifact [16]. In terms of efficiency, interviews allow for identification of a large number of requirements in a short time [16]. Interviewing allows for rich data collection [33]. According to Johannesson and Perjons [16], very structured interviews may become counter-productive for eliciting requirements due to stifling creativity. Therefore, the interviews with CGI interviewees were semi-structured. Semi-structured interviews allow the respondent to take more initiative [16].

The interviewee selection consisted of employees with different job titles and positions in the corporate hierarchy. For an overview of the interviewees that were interviewed, see table B.1, Appendix B. The interviewees had varying experience levels, from medior to senior. The job titles could be divided in the following functions: engineering, management, regulatory compliance and risk. The diverse set of job titles was selected in order to ensure rich and diverse answers. For example, selecting for data scientists and sustainability advisors, ensures that there is a mix of technical and non-technical responses. The interviewees gave their responses to the interview questions, which served as input for the requirements.

The interview protocol for architectural requirements elicitation can be found in Appendix C. The interviews were recorded and transcribed. After transcription, data was cleaned and structured per interview question. The results of the interviews together with scientific literature were used to develop the functional requirements (FRQ) and non-functional requirements (NFRQ) requirements.

The activity of requirements analysis is illustrated in Fig. 10.

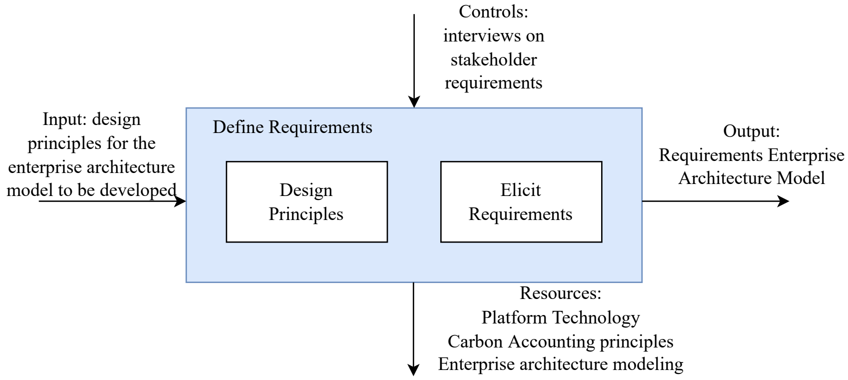


Fig. 10: Requirements analysis activity summarized, derived from Johannesson and Perjons [16, p. 104].

3.4 Requirements Analysis

3.4.1 Enterprise Architecture Model Functional Requirements

By using the interview protocol for eliciting requirements as in Appendix C and academic literature where needed, requirements for the EA-model are formulated. There are several Architecture Topics (AT_x) that the interview protocol elicited requirements for, see Table 9. The topics were chosen as follows. [34] describes that enterprise modeling consists of three concepts; (1) business layer, (2) application layer and (3) technology layer. The business layer provides products and services to external customers realized by business processes performed by business actors [34]. The application layer supports the business layer with application services realized by applications. The technology layer consists of infrastructural services for running applications realized by computer and communication hardware and system software [34]. Topics AT_1 and AT_2 aimed to identify the business processes involved with carbon accounting and the relevant actors. The subsequent topics were aimed to identify the requirements for the Carbon Accounting Solution (CAS) pertaining to data, software and physical infrastructure and networking.

Table 9: Covered topics of the interview protocol for Enterprise Architecture.

Topic Code	**Topic**
AT_1	Carbon accounting data relevancy for internal and external stakeholders
AT_2	Standards for the data and Carbon Accounting Solution
AT_3	Minimum functionalities of the Carbon Accounting Solution
AT_4	Data sources and data types
AT_5	Data accessibility
AT_6	Data storage
AT_7	Data traceability
AT_8	Data format

The interview results of the topics in Table 9 will be presented in the following sections.

AT_1: Carbon Accounting Data Relevancy for Internal and External Stakeholders
The relevant stakeholders who may benefit from carbon accounting data are summarized in Table 10.

Table 10: Internal and external stakeholders for which carbon accounting data is relevant.

Internal	**External**
Sales / finance departments	Investors
Higher management	Lenders
corporate social responsibility (CSR) managers	Clients
Members	Regulatory bodies
	Auditors

The results in table 10 are consistent with the findings of [6]. The interviewees claim that if CGI has a well organized carbon accounting system, the sales and finance departments could use this to their advantage. See quotes below.

"Being able to prove a sustainable way of working through an organized carbon accounting system, increases the value of CGI, since investors have increased trust, banks may give loans quicker, and the sales department has leverage to close deals and contracts with clients [IV_1]."

"Accounting for sustainability may result in a higher level of appreciation of CGI and could strengthen CGI's sustained competitive advantage if CGI gives insight in carbon accounting data to potential clients [IV_2]."

Other findings include the following. If a client that receives services from CGI, wants insight in respective carbon emissions in their own supply chain, a client may ask carbon-accounting data from CGI. With an established carbon accounting, monitoring and quantification system CGI could facilitate the aforementioned.

Within CGI, it is the responsibility of the CSR-manager to provide carbon accounting reports. Although Table 10 highlights the stakeholders for whom the carbon accounting data are relevant, the CSR-manager first and foremost should be facilitated in the proper data to compile the carbon accounting reports for the respective business unit. Based on the interview results and the findings of [6], a functional requirement (FRQ_1) can be formulated.

FRQ_1: *"The EAM shows how the CSR managers are facilitated in carbon accounting data to support the CSR-Reporting."*

AT_2: Standards for the Data and Carbon Accounting Solution

A significant amount of interviewees were not well aware of existing standards pertaining to quantification and reporting of GHG emissions. However, some interviewees mentioned the standards listed below, which are also elaborated upon in subsection 2.2.

- Standards of the Greenhouse Gas Protocol, specifically GHG Protocol Scope 2 Guidance: An Amendment to the GHG Protocol Corporate Standard. See also [9].
- EN ISO 14064-1:2012 [35], which is the ISO standard for quantification and reporting of GHG emissions and removals at organization level.

Both standards are consistent when it comes to the content on quantification and reporting of GHGs, specifically CO_2. However, EN ISO 14064-1:2012 [35] mentions that quantification methodologies should minimize uncertainty and yield accurate, consistent and reproducible results. See the list below for such methodologies recognized by [35]:

(a) One methodology is calculation based as described in 2.2.2 based on GHG activity data multiplied by GHG emission factors.

(b) Measurement of GHG emissions either intermittent or continuous.

(c) A combination of (a) and (b), i.e. combination of calculation and measurement.

Both [9] and [35] elaborate on GHG accounting and reporting principles. Similarly to financial accounting and reporting, generally accepted GHG accounting and reporting principles are intended to underpin and guide GHG accounting and reporting to ensure that the reported information represents a faithful, true, and fair account of a company's GHG emissions. The following GHG accounting and reporting principles are mentioned by [9] and [35]:

(a) Relevance: The GHG inventory reflects the GHG emissions of the company and serves decision-making needs of both internal- and external stakeholders.

(b) Completeness: Inclusion of all relevant GHG emissions and removals. Disclose and justify specific exclusions.

(c) Consistency: Methodologies should be consistent to foster meaningful performance tracking of emissions over time. Transparently document any changes to the data, inventory boundary, methods, or any other relevant factors in the time series.

(d) Transparency: Issues should be mentioned factually and coherently, based on a clear audit trail. Assumptions should be disclosed, and references should be made to accounting and calculation methodologies and used data sources.

(e) Accuracy: Pertains to reduction of bias and uncertainty to a minimum. Sufficient accuracy needs to be achieved to support decision making with reasonable confidence.

The following functional requirements (FRQ_2, FRQ_3) for data and the Carbon Accounting Solution were derived.

FRQ_2: *"The EAM shows the practice of carbon accounting, monitoring and quantification according to The Greenhouse Gas Protocol Scope 2 standard."*

FRQ_3: *"The EAM shows the practice of carbon accounting, monitoring and quantification according to the EN ISO 14064-1:2012 standard for quantification and reporting of GHG emissions and removals at organization level."*

AT_3, AT_4: Minimum Functionalities of the Carbon Accounting Solution and Data sources and Types

Most interviewees were consistent with requiring the following functionalities for a Carbon Accounting Solution. The CAS should give insights into the usage data of electricity used in scope 2, i.e. the quantities and times of electricity usage of objects / facilities should be known. Furthermore, dashboards that show metrics such as (past) usage, maxima and minima values, means, recent emissions, and trend analyses are popular functionalities. What was mentioned less frequently, but is not unimportant, was the functionality to create and monitor environmental, social and governance (ESG) goals and workflows.

According to Honarvar and Sami [36], incorrect utilizations of home apparatuses together with absence of smart energy infrastructure leads to waste of electricity consumption. Although aforementioned statement holds for households, it can be argued that for bigger buildings the same problem exists. Through the development of sensor technology, the power use data of apparatuses can be collected [36]. Smart power meters assist in data collection of appliance usage, and to extract valuable information from this data, there is a need to develop data mining algorithms [36]. The development of data mining algorithms is beyond the scope of this research. Several researchers have focused on the relation between electricity usage and reduction in CO_2 emissions. Conservation of electricity is a tedious task due to the lack of detailed electricity usage [36]. With data and patterns of electricity usage, adjustments can be made to conserve electricity effectively [37]. In 1.4.4, smart-data was introduced, and smart-meters were also called IoT devices. In 2.2 the data sources and types were identified through academic literature. The data sources for calculating monthly tCO_2e include: invoices (from electricity supplier's utility bills), emission factors and GWP values from the IPCC. The invoice includes the activity data in [kWh], the emission factors are in [kg CO_2/kWh] and the GWP values are in [kg CO_2e/kg CO_2].

Based on the information above, the following functional requirements (FRQ_4, FRQ_5, FRQ_6) can be formulated:

FRQ_4: *"The EAM captures how data from IoT-devices can be used to facilitate insight in a building's electricity usage."*

FRQ_5: *"The EAM captures how invoice data can be transformed to tonnes of CO_2e per month."*

Tracking ESG progress and targets is the final minimum aspect that should be covered in the EA-model. One such target is the Science-Based Target (SBT) under the Science Based Target Initiative (SBTi). According to SBTi [38, p. .]: *"Science-based targets provide a clearly-defined pathway for companies to reduce GHG emissions, helping prevent the worst impacts of climate change and future-proof business growth."*. This is relevant since CGI is a member of the SBTi, and has the following SBT: reducing GHG emissions in absolute terms by 46% for own operations (scope 1 and 2), and by 46% for business travel (scope 3) by 2026 from a 2019 base year [39]. Recording SBTs and monitoring the progress towards the SBTs with the use of data as described above should preferably be facilitated by the Carbon Accounting Solution, therefore FRQ_7 is:

FRQ_6: *"The EAM captures the recording and monitoring of SBTs."*

AT_5 and AT_6: Data Access and Storage

The CAS will contain large volumes of data with varying relevance for the respective CAS users. There is a general consensus among the interviewees that a control mechanism needs to be in place to ensure that unauthorized access to information in the database needs to be restricted. For example, financial data from the utility bill can be considered confidential, and only authorized users are allowed to access such data. Some users are only allowed to retrieve data, whereas others are allowed to retrieve and update [31]. The principle used in Information Systems (IS) for restricted access to data is called the need-to-know principle. The need-to-know principle entails access should only be given to users who are required to carry out the subject's responsibilities [40]. In database systems, the following method is used. A Database Management System (DBMS) should provide a security and authorization subsystem, which the Database Administrator (DBA) uses to create accounts and to specify account restrictions [31]. Then, the DBMS should enforce these restrictions automatically [31]. The following functional requirement (FRQ_7) can be formulated:

FRQ_7: *"The EAM captures user's access to the CAS through a security and authorization subsystem."*

The various data sources and potential large volumes of data need to be stored. All interviewees mentioned cloud storage as a solution for CAS. Cloud storage is also popular in scientific literature. There are hundreds of different cloud storage systems; ranging from systems with a specific focus to systems available to store all forms of digital data [41]. The cloud storage system architecture primarily consists of a storage layer, basic management layer, application interface layer and access layer [42]. Some advantages of cloud storage include; location independent data access when Internet access is available, no need for carrying physical storage devices, the ability to share data with external users to foster collaborative efforts [41]. The scenarios in Table 11 of cloud server storage / cloud computing provide more advantages compared to storage on local devices.

Table 11: Scenarios of cloud server storage and advantages.

Cloud Scenarios and benefits	
Infrastructure as a Service (IaaS)	Through virtualization, storing and processing capacity can be split, assigned and dynamically resized to build ad-hoc systems [41].
Platform as a Service (PaaS)	Providing software platform where systems run on [41].
Storage as a Service (StaaS)	Facilitates cloud applications to scale beyond their limited servers and allows users to store data at remote disks and access them anytime from any place [41].
Software as a Service (SaaS)	An alternative to locally run applications [41]

The following functional requirement (FRQ_8) can be formulated:

FRQ_8: *"The EAM captures how carbon accounting data and smart-data is stored through cloud data storage."*

AT_7: Data Traceability

Data traceability for data audit trails was a requirement mentioned by [IV8]. The traceability of data is embedded in the reference architecture of the platform owner, i.e. OSDU. The OSDU uses data lineage for data traceability. Data lineage, or provenance, describes where data came from, how it was derived, and how it was updated over time [43]. Consider that input data sets $[I_1, ..., I_k]$ are fed into a graph of transformations $[T_1, ..., T_n]$ to produce output data sets $[O_1, ..., O_m]$ [43], with the transformation graph as in Fig. 11.

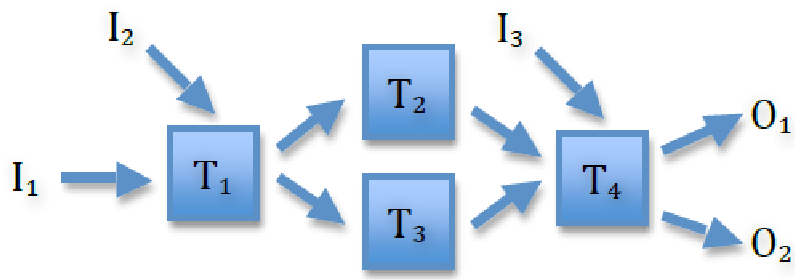


Fig. 11: Example transformation graph [43].

According to Ikeda and Widom [43], given the transformation graph as in Fig. 11, the following questions are asked:

(a) Given some output, which inputs did the output come from?

(b) Given some output, how were the inputs manipulated to produce the output?

The questions above delineate two types of lineage: where-lineage (1) and how-lineage (2) [43]. According to [43] Each type of lineage has two granularities:

1. Schema-level (coarse-grained)
2. Instance-level (fine-grained)

Schema-level where-lineage (a1) answers questions such as which data sets were used to produce a given output data set, while schema-level how-lineage (a2) answers questions such as which transformations were used to produce a given output data set [43]. In contrast, instance-level lineage treats individual items within a data set separately, so more fine grained questions are asked such as which tuples from a set of base tables are responsible for the existence of a given tuple in a derived table (where-lineage) [43].

The following functional requirement (FRQ_9) can be formulated:

FRQ_9: *"The EAM captures data lineage to ensure data traceability in the CAS implementation."*

AT_9: Data formats

Data in the CAS needs to be stored and transported. [IV8] mentioned thatJavaScript Object Notation (JSON) is the preferred data format to show carbon accounting data. JSON is a data format based on the data types of the JavaScript programming language [44]. JSON has gained popularity among web developers and has become the main format for exchanging information over the web [44]. Software executing functions requested by remote machines must establish a precise protocol for receiving and answering requests, which is called an Application Programming Interface (API) [44]. Given that JSON is a language easily understood by both developers and machines, it has become the most popular format to send API requests and responses over the HTTP protocol [44].

The following example based on [44] illustrates how an application which contains information about weather conditions around the world uses an API to allow software to access this information. A call to this API could be a request containing the following JSON file:

```
{"Country": "Chile", "City": "Santiago"},
```

By which a client is requesting the current weather conditions in Santiago, Chile. The API would reply with an HTTP response containing the following JSON file:

```
{"timestamp": "14/10/2015 11:59:07",
"temperature": 25, "Country": "Chile",
"City": "Santiagio", "description": "Sunny"},
```

Indicating that the temperature is 25 degrees and the weather is sunny overall. The previous example illustrates the simplicity and readability of JSON, which partially explains its fast adoption.

Further properties of JSON include the following: (a) JSON is lightweight data-interchange format, (b) JSON is plain text written in JavaScript object notation, (c) JSON is used to send data between computers [45]. The JSON format's syntax is similar to the code for creating JavaScript objects, therefore a JavaScript program can easily convert JSON data into JavaScript objects [45]. The format is text only, which allows easy exchange of JSON data between computers and any programming language can be used [45]. According to [45] JavaScript has a built in function for converting JSON strings into JavaScript objects:

```
JSON.parse()
```

Furthermore, JavaScript has a built in function for converting an object into a JSON string:

```
JSON.stringify()
```

Although eXtensible Markup Language (XML) also is a software- and hardware-independent tool for storing and transporting data, JSON has some advantages over XML. [46] list the following:

- JSON is shorter;
- JSON is quicker to read and write;
- JSON can use arrays;
- JSON can be parsed by a standard JavaScript function, while XML has to be parsed with an XML parser.

Based on the previous information, functional requirements (FRQ_{10} and FRQ_{11}) are formulated:

FRQ_{10}: *"The EAM captures how an API handles requests between the database and the CAS through exchanging JSON data".*

FRQ_{11}: *"The EAM captures how JSON is used to transport and store data within the CAS".*

3.4.2 Architectural Non-Functional Requirements

In 1.4 non-functional requirements ($NFRQ_x$) were described as generic requirements pertaining to design, modularity and availability of Information Systems. For the CAS, the following $NFRQ_x$'s can be formulated:

$NFRQ_1$: *"The EAM captures how monthly carbon accounting reports are generated in an efficient manner".*

$NFRQ_2$: *"The EAM captures the use of CAS in different time intervals".*

3.5 Conclusion

To conclude chapter 3, both the functional and non-functional requirements are summarized in Table 19. The requirements in Table 19 form the foundation of the EAM, however during the modelling it is possible that there will be a change in requirements.

Table 12: Functional and non-functional requirements for the Enterprise Architecture Model (EAM) of the Carbon Accounting Solution (CAS).

Code	**Description**
FRQ_1	The EAM illustrates how the CSR managers are facilitated in data to support the CSR-Reporting.
FRQ_2	The EAM illustrates the practice of carbon accounting, monitoring and quantification according to The Greenhouse Gas Protocol Scope 2 standard.
FRQ_3	The EAM illustrates the practice of carbon accounting, monitoring and quantification according to the EN ISO 14064-1:2012 standard for quantification and reporting of GHG emissions and removals at organization level.
FRQ_4	The EAM illustrates how data from IoT-devices can be used to facilitate insight in a building's electricity usage.
FRQ_5	The EAM illustrates how invoice data can be transformed to tonnes of CO_2 per month.
FRQ_6	The EAM illustrates the recording and monitoring of SBTs.
FRQ_7	The EAM illustrates user's access to the CAS through a security and authorization subsystem.
FRQ_8	The EAM illustrates how carbon accounting data and smart-data is stored through cloud data storage.
FRQ_9	The EAM illustrates data lineage to ensure data traceability in the CAS implementation.
FRQ_{10}	The EAM captures how an API handles requests between the database and the CAS through exchanging JSON data
FRQ_{11}	The EAM illustrates how JSON is used to transport and store data within CAS.
$NFRQ_1$	The EAM captures how monthly carbon accounting reports are generated in an efficient manner.
$NFRQ_2$	The EAM captures the use of CAS in different time intervals.

4 Initial Enterprise Architecture Model Design

In chapter 4 the third activity of the design science research methodology is addressed. According to Johannesson and Perjons [16], the third activity entails design and development of an artifact, which fulfills the requirements from the previous activity. [16] express the third activity as follows:

"Create an artefact that addresses the explicated problem and fulfills the defined requirements."

SRQ_3 is ought to be answered in the third activity:

SRQ_3: *How can the identified requirements for a Carbon Accounting Solution (CAS) serve as a building block for the development of CAS's Enterprise Architecture Model (EAM)?*

To create an initial EAM for CAS the steps as in table 13 will be followed. The steps in table 13 deliver outputs in increasing level of abstraction of CAS's enterprise architecture. Data flow diagrams give insight in data inputs and data outputs for CAS and the relevant actors. The subsequent steps are aimed at developing an EAM by using a modelling language designed for Information Systems. Since an EAM is not designed to visualize JSON-formatted data, this is illustrated separately. The initial EAM was evaluated by ICT-architects. The feedback was input for the next step in the Design Science Research Methodology (DSRM) to improve the initial EAM.

Table 13: Steps followed to create initial Enterprise Architecture Model of CAS.

Step	Subsection	Output
1	4.1	Data Flow Diagrams
2	4.2	Choice of enterprise architecture modelling method and language
3	4.3	Initial enterprise architecture model and description of CAS
4	4.4	Illustration of JSON-formatted data given by CAS
5	4.5	Evaluation of initial enterprise architecture model
6	4.6	Conclusion

The data flows will be illustrated first with the use of a level one and level two data flow diagram (DFD) in 4.1. 4.2 describes the enterprise architecture modeling language and method to model the enterprise architecture of CAS. In 4.3 CAS's initial enterprise architecture model is depicted in several diagrams with the use of ArchiMate. The diagrams with the descriptions translate the requirements from activity two in the DSRM. 4.4 shows the data-format in which the CSR-manager receives carbon-accounting data through an API from CAS. In 4.5 the initial enterprise architecture model of CAS is evaluated by verifying whether the models translate the requirements in terms of enterprise architecture. In 4.6 the enterprise architecture model will be summarized and conclusions will be drawn regarding the translation of requirements into enterprise architecture of CAS.

4.1 Data FLow Diagrams for CAS

The DFD technique is a diagram-based requirements analysis technique [47]–[49]. A DFD depicts a system as a network of functional processes interconnected through data flows. DFDs are often used in the first phases of information analysis to establish a global model of an information system which can be further refined [50]. A DFD can be decomposed into a hierarchy that represents the proposed system at different levels of abstraction [51]. According to Davis and Yen [52], the abstraction levels are in increasing detail:

- Level 0: The context diagram; documents system's boundaries by highlighting sources and destinations, which helps visualizing high-level logical system designs;
- Level 1 DFD: high level logical map of the system showing key relationships, but hides most of the details;
- Level 2 DFDs: Functional decomposition of level 1 diagrams; for each level 1 process a DFD is modelled.

For CAS, a level 0 and level 1 DFD will be drawn. The reasoning for modelling a level 0 and level 1 DFD is as follows. For this research, the primary data flows, processes, data sinks/sources and actors need to be modelled first. Models like a level 0 and level 1 DFD are sufficient for identifying the aforementioned. A level 0 and level 1 DFD are furthermore sufficient for higher management and non-tech stakeholders to understand the data flows. Finally, more detailed data flows, data aggregations, and data visualization will be modelled with the modelling language as described in 4.2. The building blocks of DFDs are denoted in figure 12.

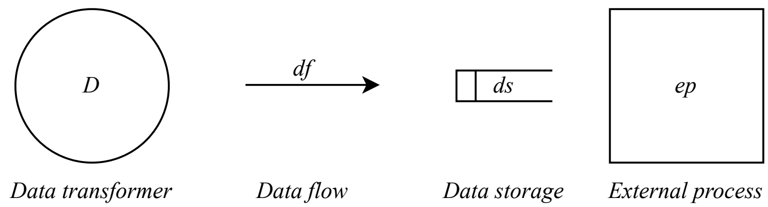


Fig. 12: Elementary building blocks of a DFD.

According to Larsen, Plat, and Toetenel [53]:

- *Data transformers.* Data transformers denote a transformation from (an arbitrary number of) input values to (an arbitrary number of) output values, possibly with side effects.
- *Data flows.* Data flows are represented as arrows, connecting one data transformer to another. They represent a flow of data between the data transformers they connect. The flow of data is unidirectional in the direction of the arrow.
- *Data stores.* Data stores provide for (temporary) storage of data.
- *External processes.* External processes are processes that are not part of the system but belong to the outside world. They are used to show where the input to the system is coming from and where the output of the system is going to.

Data transformers are also known as *processes*, which identify an activity that changes, moves, or otherwise transforms data. External processes are also known as *source* or *destination (sinks)*. Sources and destinations define the system's boundaries; each one represents a person, organization, or other system that supplies data to the system, gets data from the system, or both.

4.1.1 Data Flow Diagrams Applied for CAS

As an initial step towards decarbonization, a reporting strategy for carbon accounting and removals needs to be established. An organization, such as CGI, therefore could partner-up with electricity suppliers to exchange data. How the flow of data is organized, and how data should be transformed to gain insights in carbon emissions (footprints) with data platform technology, requires illustration.

The context diagram shows the system's boundaries and usually consists of one process which represents the entire system. The inputs and outputs from the system are shown to and from various external entities. See Figure 13 for the context diagram of CAS. The Carbon Accounting Solution is the data transformer, the data flows are depicted in rectangular components (business objects) with flow relationships and the data sinks/sources are depicted with actor components containing a stick figure.

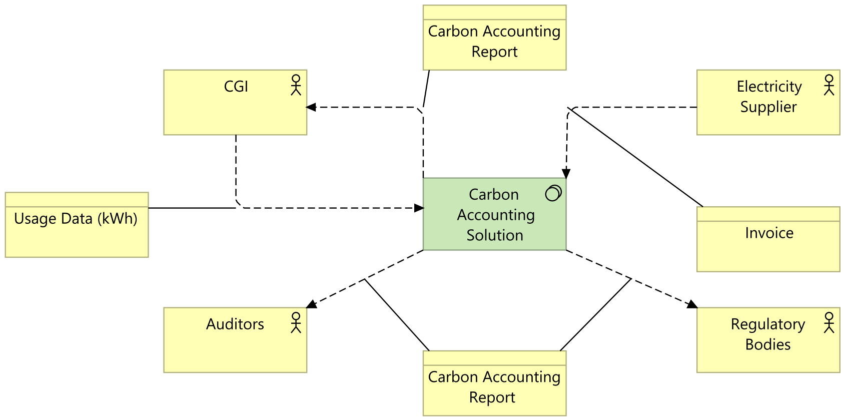


Fig. 13: Context diagram of the Carbon Accounting Solution (CAS).

An example of calculating and reporting carbon emissions in scope two would be as follows. For scope two emissions, the following steps need to be followed to report carbon equivalent CO_2e values in [kg] of CO_2:

1. Multiply activity data [kWh, mWh] from each operation by the emission factor/intensity factor [kg CO_2/kWh] for that activity for each applicable GHG (CO_2 in this case). This step gives [kg] of CO_2
2. Multiply GWP values by the GHG emissions in step one, to calculate total CO_2e-emissions
3. Report final scope 2 in metric tons of each GHG and in metric tons of CO_2e [tCO_2e]

The steps above may look as depicted in Figure 14 in a level 1 DFD diagram.

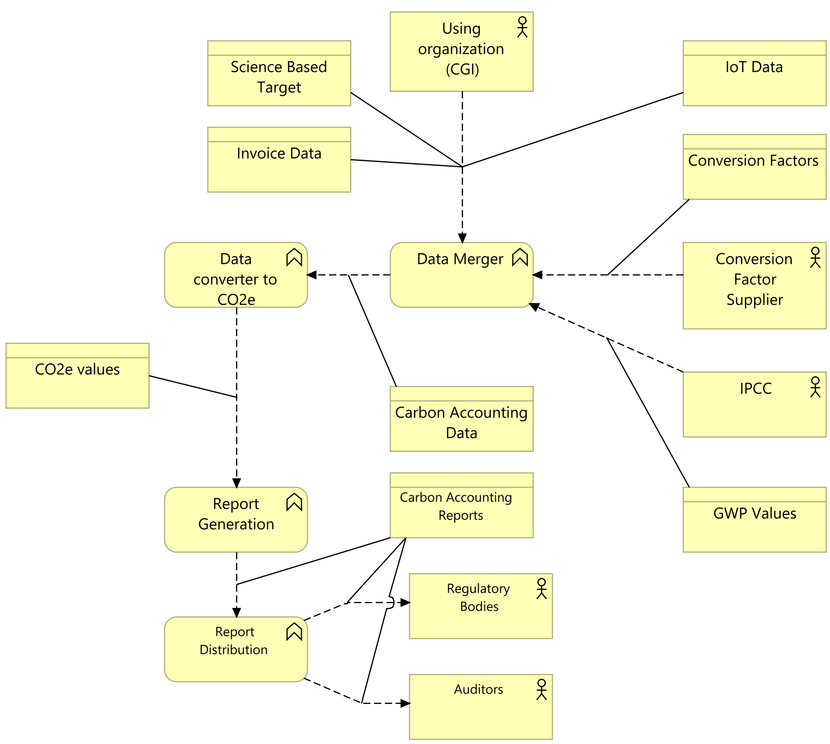


Fig. 14: Level 1 DFD diagram of the Carbon Accounting Solution (CAS).

4.2 Architecture Modeling Method

For the effective application of Information Technology a company is required to have a clear, integrated vision on the relation between its business and IT [34]. When such a vision is lacking, the IT infrastructure will never adequately support the business, and the business will not optimally profit from IT developments. Current business practices require an integrated approach to business and IT, however such integrated views are still far from reality [34].

According to Lankhorst, Proper, and Jonkers [34], managing complexity of any large system, be it an enterprise, an organisation, an information system or a software system, an architectural approach is needed. At an organizational level different core structures should be illustrated, such as business processes, applications and infrastructures, as well as the relationship between these aspects [34]. One modelling language to facilitate enterprise architecture is ArchiMate. Archimate is aligned with the the Open Group's Architecture Framework (TOGAF) enterprise architecture framework. [34] mention the following primary components of ArchiMate:

1. A framework: a conceptual framework consisting of rows (layers) and columns (aspects), which facilitates classification of architectural phenomena.
2. Modeling concepts: a set of modeling concepts allowing for the description of relevant aspects of enterprises at the enterprise level. This set underlies from the abstract syntax, focusing on the concepts and their meaning, separate from the language constructs in which they are used.
3. Abstract syntax: this component contains the formal definition of the language in terms of a metamodel.
4. Language semantics: defines the meaning of each language construct and relation type.
5. A concrete syntax in terms of graphical notation: the concrete syntax defines how the language constructs defined in the metamodel are represented graphically.
6. Viewpoint mechanism: entails the creation of different views for different stakeholders.

The first four components constitute the core of the ArchiMate language, and components 5 and 6 are crucial in making the standard usable in practice [34]. The ArchiMate metamodel defines the characteristics of each language construct, and its relationships to other language constructs [34]. Furthermore, the metamodel positions different language constructs in the cells of the Archimate framework and specificies relationships between constructs and cells [34]. This feature distinguishes Archimate from both Unified Modelling Language (UML) and the Zachman framework, since Archimate models dependencies between different layers, domains and views of the enterprise architecture, resulting in a coherent whole instead of a collection of different isolated diagrams [34]. [34] state that Archimate is more flexible than UML as there is not a strict partitioning of constructs into views.

Earlier it was emphasized that the ArchiMate modelling language and TOGAF are aligned. TOGAF is an enterprise architecture framework. The core of TOGAF is a standard process, i.e. the Architecture Development Method (ADM) [54]. [54] specifies components of an enterprise architecture framework, and are listed below and illustrated in Fig. 15.

- A process for creating architectures; may be accompanied by guidelines, techniques, and best practices.
- A set or classification of viewpoints.
- A language for describing architectures.
- The concept of an architecture repository, possibly containing predefined architectural artefacts and reference models.

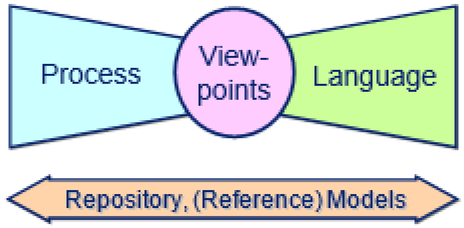


Fig. 15: Components of an enterprise architecture framework [54].

The TOGAF standard identifies the components as in Fig. 15, but does not constitute a formal modeling language. TOGAF and ArchiMate overlap due to the use of viewpoints, and the concept of an underlying common repository of architectural artifacts and models [54]. The two standards are complementary to one another with respect to the definition of an architecture development process and the definition of an enterprise architecture modeling language [54]. The ArchiMate modeling language can be used to model architectures developed using the TOGAF ADM [54], see Fig. 16 for the correspondence of activities between activities of the ADM phases and parts of the ArchiMate language.

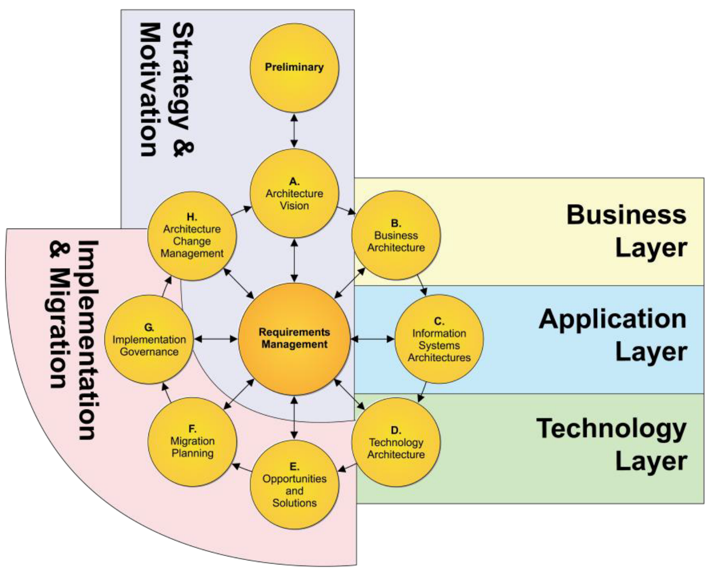


Fig. 16: Correspondence between ADM phases and ArchiMate Language [54]. Circular diagram depicts ADM phases, the layers depict ArchiMate modeling layers.

[54] concludes that the combined use of the TOGAF framework with the Archimate Modeling language can support better communication with internal- and external stakeholders.

4.3 Enterprise Architecture Modeling

The goal of enterprise architecture modeling of CAS is to demonstrate how various data sources are transformed and aggregated to support the goal of automatically creating carbon accounting reports. Furthermore, the data generated through IoT sensors should give insights in the usage patterns of CGI. The enterprise architecture models should translate the requirements for CAS listed in Table 19. The approach of creating enterprise architecture models starts with modeling the technology architecture of CAS. Technology architecture showcases how information technology can be deployed to realize applications and data requirements [54]. ArchiMate has a separate layer for modeling technology architecture wherein technology such as devices, systems software, DBMS, and communications paths can be represented [54]. In Appendix D all ArchiMate elements and relationships are displayed.

In Fig. 32 the devices, system software and communication paths are illustrated to facilitate carbon accounting. Fig. 32 furthermore illustrates the technology architecture for the purpose of creating insights in usage patterns of the CGI head office.

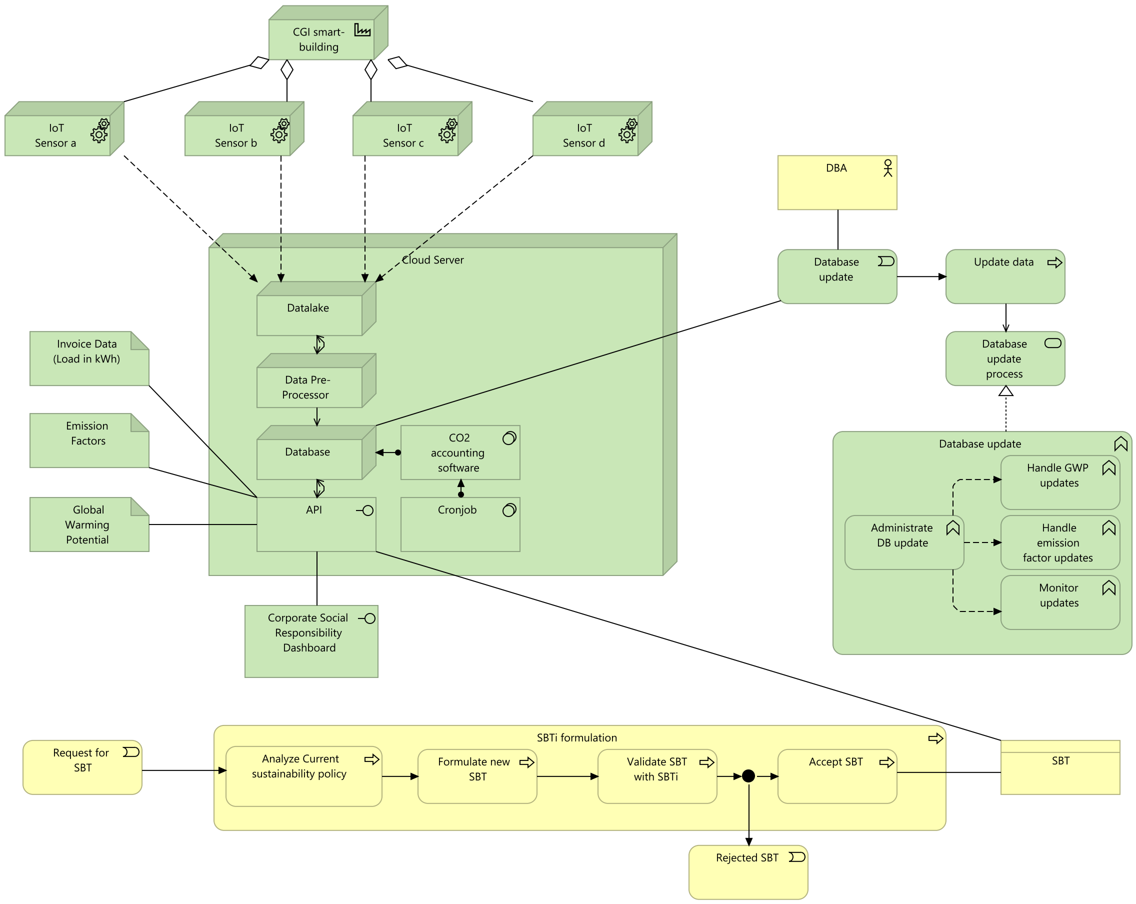


Fig. 17: Technology architecture of the platform implementation.

The technology architecture which supports the data flows, transformations and aggregations is set-up as follows. The top-down analysis of Fig. 32 is as follows. The first element depicts a facility. A facility represents a physical structure or environment [55]. CGI's head office can be considered a smart-building in this case due to the combined use of IoT technology and platform technology. The smart-building aggregates data from IoT-sensors $[IoT - Sensor_a, ..., IoT - Sensor_n]$. The data collected through the IoT-sensors go via a dynamic 'flow relationship' to a data lake. The relationship between the IoT-sensors and the data lake is dynamic, due to the continuous measuring of usage data and generation of associated data. A data lake is a large scalable storage repository that holds a vast amount of raw data in its native format until it is needed plus processing systems that can ingest data without compromising the data structure [56]. Data lakes are designed to handle large and quickly arriving volumes of unstructured data from which further insights are derived [57]. Data lakes use dynamic analytical applications and the data becomes accessible as soon as it is created [57]. Even though the data lake is a sub-component in the IT-infrastructure, it is important to note that preliminary analysis of data in the lake is possible. Data lakes often contain a semantic database, a conceptual model that leverages the same standards and technologies used to create Internet hyperlinks, and add a layer of context over the data that defines the meaning of the data and its interrelationships with other data [57].

The data lake has a 'serving' relationship with the Data pre-processor. Data lakes are large pools wherein historical data is accumulated and new data (structured, unstructured and semi-structured plus binary from sensors and devices) are stored [57]. The data pre-processor is responsible for data pre-processing. Since the input data from the data lake is large in volume and unstructured, operations on the data should result in structured and formatted data suitable for analysis. Data pre-processing is a term used in data mining (DM), and entails the structuring and formatting of data with the use of pre-processing techniques. For the scope of the thesis, data pre-processing will be described generally. Data preparation is usually a mandatory step, wherein prior useless data is converted into new data that fits a DM process [58]. In DM unprepared data will lead to algorithms reporting errors during the runtime and the results of analysis being inaccurate [58]. [58] identify the following data preparation techniques:

- Data cleaning: cleaning up the data;
- Data transformation: to provide accurate data;
- Data integration: incorporating and adjusting data;
- Data normalization: unifying and scaling data;
- Missing data imputation: the handling of missing data;
- Noise identification: the detection and managing of noisy data.

Fig. 18 gives an illustration of the forms of data preparation.

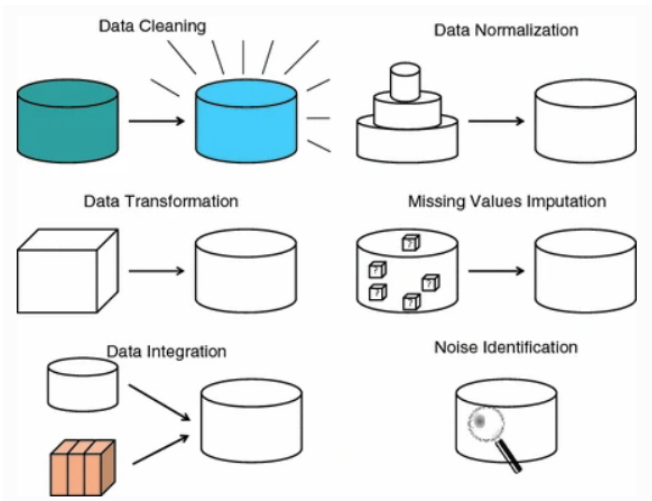


Fig. 18: Forms of data preparation [58].

The next step in data pre-processing is data reduction. Data reductions entail a set of techniques that obtain a reduced representation of original data [58]. [58] describe the following set of data reduction techniques:

- Feature Selection (FS): removing dimensionality of data;
- Instance selection: removing redundant and/or conflictive examples;
- Discretization: simplification the attribute of a domain;
- Feature extraction and/or Instance generation: filling in gaps in data.

Fig. 19 gives an illustration of the forms of data reduction.

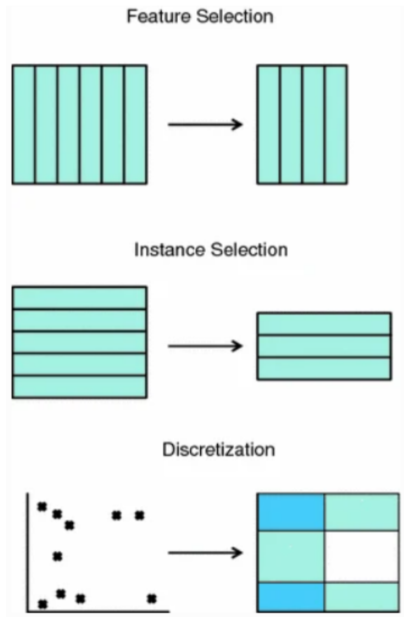


Fig. 19: Forms of data reduction [58].

The processed data is stored in the database, along with the GWP-values, emission factors and loads (kWh). The CO_2-accounting software is triggered by cron. Cron is a system scheduler on UNIX and Linux systems [59]. The purpose of cron is to run commands, series of commands, or scripts on a predetermined schedule [59]. The cron-script triggers the CO_2-accounting software to generate carbon-accounting reports periodically at a fixed time (e.g. monthly). The carbon-accounting reports are stored in the database, which through an API and CSR-dashboard can be retrieved by the CSR-manager.

The database is updated through a set of functions 'Administrate DB update', which realizes a process 'Database update process'. Aforementioned function and database update process are triggered by the event 'database update', which triggers the process 'update data'.

The business process modeled in yellow, depicts the creation and recording of Science-Based Targets. A business event 'Request for SBT' triggers a set of business processes, which either results in an accepted or rejected SBT. An accepted SBT is stored as data in the database.

4.4 Carbon Accounting Data in JSON-format

Below an example of carbon accounting data in JavaScript Object Notation format is displayed. The carbon accounting data can be retrieved by the CSR-manager through the API at any time. Cron will ensure that monthly carbon accounting data in JSON-format is stored in the database. The content covered in the JSON-data below is in line with EN ISO 14064-1 and the standards of the Greenhouse Gas Protocol Scope 2 Standard.

```
{
        "templates": [
        {
                "id": "CGI_TEMP_001",
                "title": "My green title",
                "purpose": "Important purpose",
                "responsible": "A.G. Smit"
        },
        {
                "id": "CGI_TEMP_002",
                "title": "My green title 2",
                "purpose": "Important purpose 2",
                "responsible": "F.C. Cox"
        }
        ],
        "report_contents": [
        {
                "id": "CGI_CONT_001",
                "organization": "CGI",
                "timestamp": "2022-03-13T14:10:49Z",
                "usage_kwh": 1500,
                "emission_factor": 15,
                "gwp": 1,
                "tco2e": 1200,
                "target_id": "CGI_TARGET_001"
        }
        ],
        "targets": [
        {
                "id": "CGI_TARGET_001",
                "target": "tCO2e reduced by x_percentage in e_scope_2",
                "target_met": false
        }
        ]
}
```

4.5 Evaluation of Initial Enterprise Architecture Model

Table 14 shows an overview of the requirements that are covered by the initial enterprise architecture model. The evaluation was done by using both logical reasoning and initial feedback from CGI's ICT-architects. *Important note: $NFRQ_1$ and $NFRQ_2$ in Table 19 were modified. $NFRQ_1$ was too vague and incorrect, $NFRQ_2$ was changed so that the EAM should capture the practice of creating monthly reports, rather than reports in different time intervals. $NFRQ_1$ and $NFRQ_3$ were added after consulting with the ICT-Architects, see Table E.3, session 2. Aforementioned requirements were to be demonstrated and evaluated by the models in the subsequent chapter.*

Table 14: Overview of covered requirements by initial Enterprise Architecture Model (EAM) of CAS.

Code	Description	Covered
FRQ_1	The EAM illustrates how the CSR managers are facilitated in data to support the CSR-Reporting.	✓
FRQ_2	The EAM illustrates the practice of carbon accounting, monitoring and quantification according to The Greenhouse Gas Protocol Scope 2 standard.	✓
FRQ_3	The EAM illustrates the practice of carbon accounting, monitoring and quantification according to the EN ISO 14064-1:2012 standard for quantification and reporting of GHG emissions and removals at organization level.	✓
FRQ_4	The EAM illustrates how data from IoT-devices can be used to facilitate insight in a building's electricity usage.	✓
FRQ_5	The EAM illustrates how invoice data can be transformed to tonnes of CO_2 per month.	✗
FRQ_6	The EAM illustrates the recording and monitoring of SBTs.	✓
FRQ_7	The EAM illustrates user's access to the data platform through a security and authorization subsystem.	✗
FRQ_8	The EAM illustrates how carbon accounting data and smart-data is stored through cloud data storage.	✓
FRQ_9	The EAM illustrates data lineage to ensure data traceability in the data platform implementation.	✗
FRQ_{10}	The EAM captures how an API handles requests between the database and the CAS through exchanging JSON data	✓
FRQ_{11}	The EAM illustrates how JSON is used to transport and store data within CAS.	✗
$NFRQ_1$	The EAM illustrates modifiability of CAS	✗
$NFRQ_2$	The EAM illustrates how monthly carbon accounting reports are generated	✓
$NFRQ_3$	The EAM illustrates adaptability of CAS	✗

FRQ_1 is illustrated in the initial enterprise architecture model as follows. The 'corporate social responsibility Dashboard is used by the CSR-manager to retrieve carbon accounting reports. Via an API which runs on the cloud server, the carbon accounting report is retrieved from the database.

FRQ_2 and FRQ_3 are illustrated by showing the aggregation of GWP-values, emission factors and invoice data (via artefacts) which are processed by CO_2-accounting software into carbon accounting reports. Furthermore, in 4.4 the carbon accounting data in JSON-format is displayed, which shows the mandatory data that should be in a carbon accounting report according to EN ISO 14064-1:2012 and The Greenhouse Gas Protocol Scope 2 Standard.

FRQ_4 is illustrated by showing how IoT data-flows are aggregated within CGI's smart-building. Furthermore, the datalake and data preprocessor form a middle layer wherein IoT data is extracted, transformed and then loaded into the database. Data extraction-transformation-loading (ETL) is usually performed by ETL tools. For the scope of this thesis, an ETL tool won't be modelled. However, the process behind ETL tools (middle layer in the model) is as follows. ETL tools are software which facilitate extraction of data from several sources, their cleansing, customization and insertion into a data warehouse [60]. See Fig. 20 for the environment of an ETL process.

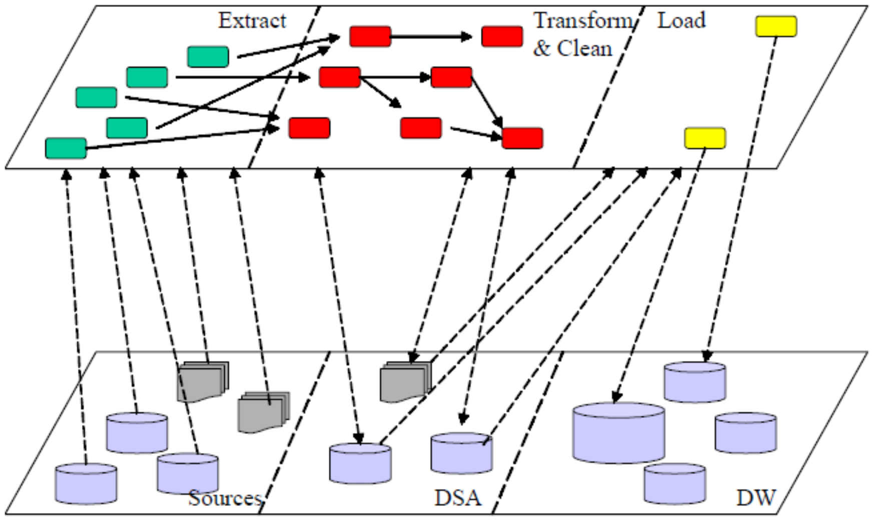


Fig. 20: The environment of ETL processes [60, p. 14]

Data sources are typically relational databases and files [60]. In this case, the data sources are the IoT-sensors which produce data stored in the datalake. Data from these sources are extracted by extraction routines [60], see also top left of Fig. 20. Then, these data are propagated to the Data Staging Area (DSA) where data transformation and cleaning is performed before being loaded into the data warehouse [60]. In this case, The data preprocessor facilitates the data transformation, cleaning, and loading. In this case, the database is wherein the processed data is loaded, instead of a data warehouse.

FRQ_5 is not illustrated by the initial enterprise architecture model of CAS, because it is not clear how the 'CO_2-accounting software' processes data to calculate tCO_2e values.

FRQ_6 is illustrated by the initial enterprise architecture model by showing (in yellow business layer) how SBTs are created and approved by the SBTi. When accepted, the SBT is recorded in the database via an API. In 4.4 the SBT target is included in the JSON-data visualization of the carbon accounting report.

FRQ_7 is not illustrated by the initial enterprise architecture model, since security aspects are not included.

FRQ_8 is illustrated by the initial enterprise architecture model, since a cloud server with all relevant components is modelled. What runs on the cloud server is: datalake, data pre-processor and database (ETL-layer), and the Cronjob, CO_2 accounting software and the API. The API is responsible for loading data in the cloud server provided by the DBA and/or CSR-manager.

FRQ_9 is not illustrated by the initial enterprise architecture model. Data lineage is provided by the reference architecture of the OSDU, according to [IV8].

FRQ_{10} is illustrated by the initial enterprise architecture model and partially by 4.4. It is shown that the API loads data objects into the database and how via an API reports can be retrieved through a CSR-dashboard. The data format of the report is shown in 4.4.

FRQ_{11} is not illustrated by the initial enterprise architecture model. However, it is partially illustrated that carbon accounting reports are stored in JSON-formatted data by 4.4.

$NFRQ_1$ and $NFRQ_3$ are not illustrated by the initial enterprise architecture model.

$NFRQ_2$ is illustrated by the initial enterprise architecture model, by including Cronjob in the model. For a Cronjob description, see 4.3, p.32.

4.6 Conclusion

In chapter 4 activity three of the Design Science Research Methodology (DSRM) was addressed: *"Create an artefact that addresses the explicated problem and fulfills the defined requirements".* SRQ_3 needed to be answered in order to produce an artefact that fulfills the requirements as in Table 19;

SRQ_3: *SRQ_3: How can the identified requirements for a Carbon Accounting Solution (CAS) serve as a building block for the development of CAS's Enterprise Architecture Model (EAM)?*

The initial architecture model of CAS was the result of SRQ_3, which contains the following key elements:

- CGI-office as a smart-building wherein IoT-data is aggregated;
- IoT-sensors;
- A cloud server containing;
  - Datalake (to store IoT-data)
  - Data preprocessor (for data transformation, cleaning, and loading)
  - Database
  - API for loading data (also SBTs) in the database and to deliver carbon accounting reports with the CSR-manager through a CSR-dashboard;
  - CO_2 accounting software (for calculating tCO_2e values);
  - Cronjob for scheduling monthly carbon accounting reports.
- A business layer showing the recording and tracking of SBTs which are stored in the database via the API;
- A database update event.

The illustrated requirements by the initial architecture model of CAS are summarized in Table 15.

Table 15: Overview of covered requirements by initial enterprise architecture model of CAS.

Code	Description	Covered
FRQ_1	The enterprise architecture model illustrates how the CSR managers are facilitated in data to support the CSR-Reporting.	✓
FRQ_2	The enterprise architecture model illustrates the practice of carbon accounting, monitoring and quantification according to The Greenhouse Gas Protocol Scope 2 standard.	✓
FRQ_3	The enterprise architecture model illustrates the practice of carbon accounting, monitoring and quantification according to the EN ISO 14064-1:2012 standard for quantification and reporting of GHG emissions and removals at organization level.	✓
FRQ_4	The enterprise architecture model illustrates how data from IoT-devices can be used to facilitate insight in a building's electricity usage.	✓
FRQ_6	The enterprise architecture model illustrates the recording and monitoring of SBTs.	✓
FRQ_8	The enterprise architecture model illustrates how carbon accounting data and data from IoT-devices is stored through cloud data storage.	✓
FRQ_{10}	The enterprise architecture model illustrates how an API handles requests between the database and CAS by providing JSON-formatted data.	✓
$NFRQ_2$	The enterprise architecture model illustrates how monthly carbon accounting reports are generated	✓

In the next Chapter (5), activity four will be addressed, wherein the enterprise architecture will be demonstrated, evaluated and improved.

5 Enterprise Architecture Model Demonstration, Evaluation and Improvement

In Chapter 5 activity four of the Design Science Research Methodology (DSRM) is addressed, which is demonstration and evaluation of an artifact. Activity four was addressed by answering SRQ_4:

SRQ_4: *"To what extent does the Carbon Accounting Solution (CAS)'s initial Enterprise Architecture Model (EAM) comply with the requirements?"*

SRQ_4 is answered as follows. The initial Enterprise Architecture-model of CAS developed in Chapter 4, p. 52 was evaluated by using the Evaluation Framework as in Appendix E.2. CGI-experts in the field of ICT-Architecture provided feedback on the models in weekly Microsoft Teams sessions. In total, 7 sessions were held with the Dutch ICT-architects. Furthermore, 7 sessions were held with a Canadian ICT-architect. For the list of CGI-experts and their roles, see Appendix E.1, Table E.1. The modeling aspects that were evaluated by the CGI-experts are included in Appendix E.2, Table E.2. The feedback from the CGI-experts was used to: edit the initial EAM, develop new EAMs, requirement evaluation, and requirement coverage/translation by the new EAMs. For an overview of feedback received by CGI-experts, see Appendix E.3, Table E.3.

According to F_{1a} and F_{1b} (in E.3, Table E.3) the EA of any ICT-solution/ICT-system should be modelled in the correct ArchiMate layer in a specific order. Therefore Chapter 5 is split up in section; 5.1 (Enterprise Architecture Model Overview of CAS), 5.2 (Business Layer), 5.3 (Application Layer) and 5.4 (Technology Layer). Every section will start with a rationale explaining the layer choice and what the layer facilitates. The ArchiMate models will be described, and these will be evaluated by verifying whether the requirements are covered and demonstrated in Section 5.5. Chapter 5 is concluded in Section 5.6.

5.1 Enterprise Architecture Model Overview of CAS

Fig. 21 shows an overview of the Enterprise Architecture (EA) of Carbon Accounting Solution (CAS). The (yellow) business layer shows the business services and functions at the heart of CAS. The (blue) application layer shows application services, data objects and applications used to create carbon accounting reports and insights in building usage data. The (green) technology layer shows the infrastructure required to facilitate aforementioned. The models in subsequent sections show detailed EA-models in the three views.

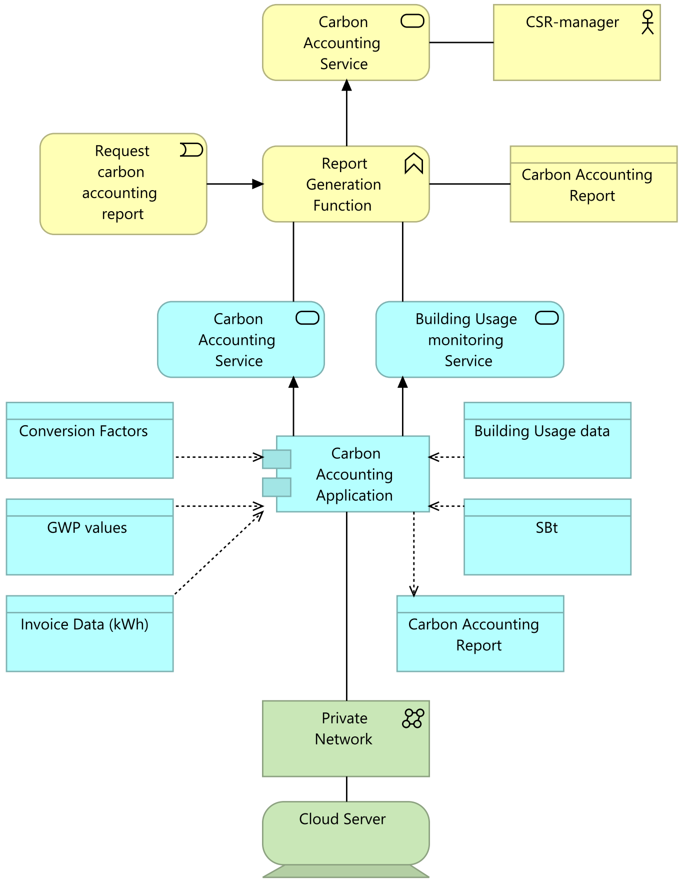


Fig. 21: Enterprise Architecture (EA)-model of CAS.

5.2 Business Layer

The Open Group [61] stated that "Business Layer elements are used to model the operational organization of an enterprise in a technology-independent manner". The Business Layer of ArchiMate is used to model the business processes which are at the heart of what CAS should serve. Furthermore, the models in the Business Layer serve as communication method between architects and management. Finally, models in the Business Layer identify the functionalities, processes, services and actors which are involved in CAS.

5.2.1 Business Process Overview of Carbon Accounting for CAS

The business process of carbon accounting facilitated by CAS is shown in Fig. 22. The business process is grouped to the Data Supplier and CAS. In the Data Supplier group, the Electricity Supplier triggers a business process 'Generate Invoice Process'. 'Generate Invoice Process' triggers the business process 'Mail Invoice Process'. Between the aforementioned business processes is a data (usage data) flow (shown by a dynamic flow relationship). The invoice generation business process 'accesses' the 'Usage data in kWh' business object to establish the data flow.

Between the Data Supplier and CAS groups, the following happens. The 'Mail Invoice Process' has a data (Invoice representation) flow to the actor 'CGI'. CGI sends the following to CAS's 'Data Transformer'; Invoice data, IoT-data and the Science-Based Target (SBT). What furthermore is shown in Fig. 22, is that the SBT is a product of both CGI's input and the CDP's input. For the business process behind the formulation of a SBT, see 5.2.2, Fig. 23.

In the CAS group, the 'Data transformer to tCO_2e' receives through flow relationships the invoice data, IoT-data, SBT, GWP-values from the IPCC and conversion factors from the conversion factor supplier.

The 'Data transformer to tCO_2e' is represented as a business function. According to The Open Group [61], a business function "represents a collection of business behavior based on a chosen set of criteria (typically required business resources and/or competencies), closely aligned to an organization, but not necessarily explicitly governed by the organization". The behavior of the business function is the transformation of aforementioned data into tCO_2es. Between business functions 'Data transformer to tCO_2e', 'Report Generation' and 'Report distribution', the data objects 'tCO_2e values' and 'Carbon accounting report' are transported through 'flow relationships'. From the 'Report distribution' business function, through 'flow relationships' carbon accounting reports flow to Regulatory bodies, Auditors and CGI. The business function 'Data transformer to tCO_2e' is modelled further in 5.2.3, Fig. 24.

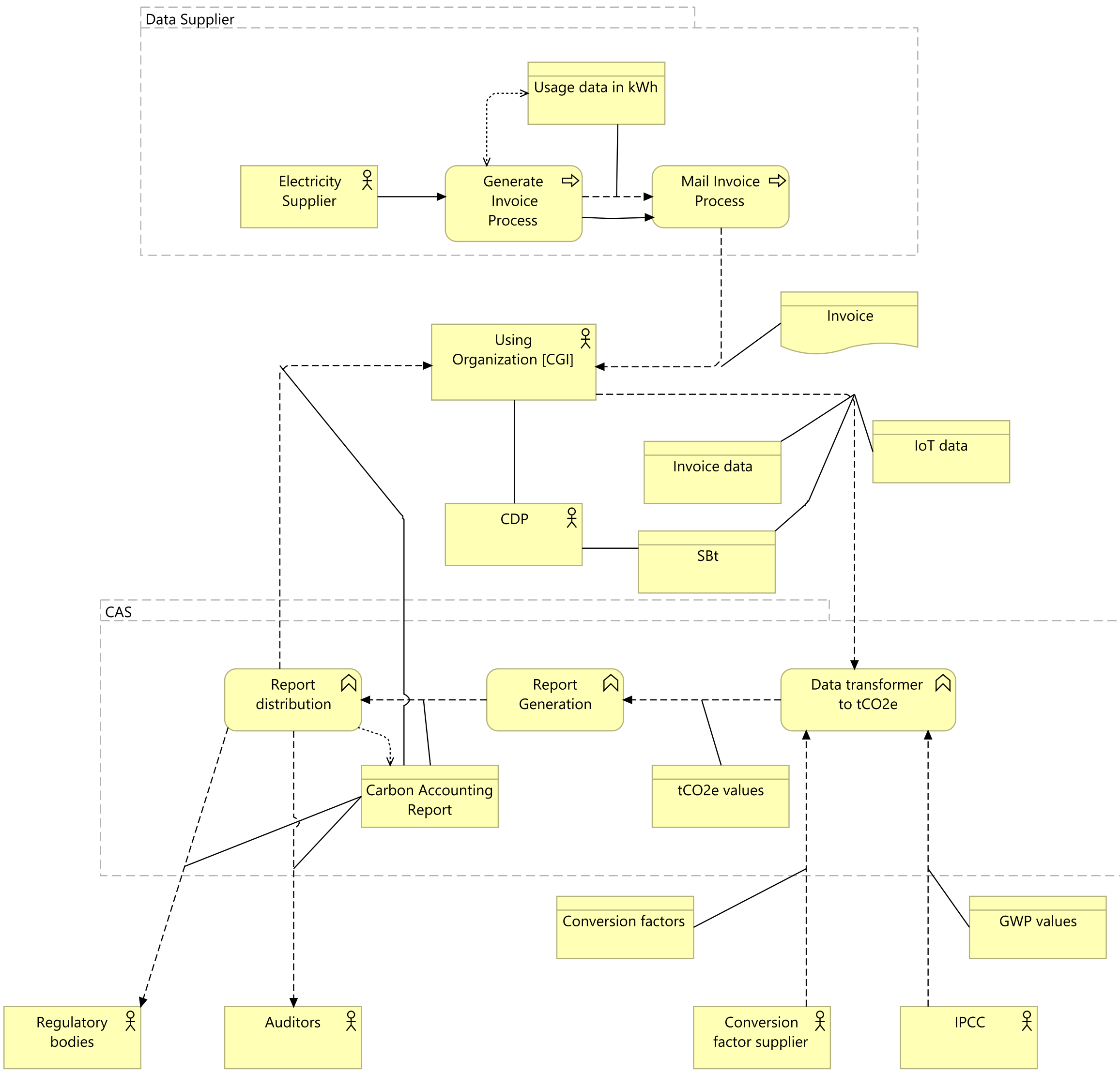


Fig. 22: Business Process of CAS.

5.2.2 Science-Based Target Formulation Business Process

In order to monitor environmental, social and governance (ESG)-progress, organizations need specific targets which serve as a goal to achieve a reduction in Greenhouse Gas (GHG)-emissions. An official target is the Science-Based Target (SBT), which is part of the Science Based Target Initiative (SBTi). A SBT provides clear-defined pathways for organizations in order to reduce GHG-emissions to prevent climate change and to enable future-proof business growth [38].

The formulation of Science-Based Target (SBT) as part of the SBTi is therefore a relevant business process, since SBTs need to be recorded in the database. Furthermore, the carbon accounting reports in conjunction with the SBT serve as input for monitoring the progress of achieving the SBT and ESG-progress. Organizations (CGI) report these targets to the CDP. The CDP is a not-for-profit charity that runs the global disclosure system for investors, companies, cities, states and regions to manage their environmental impacts [62]. For the business process behind formulating a SBT, see Fig. 23

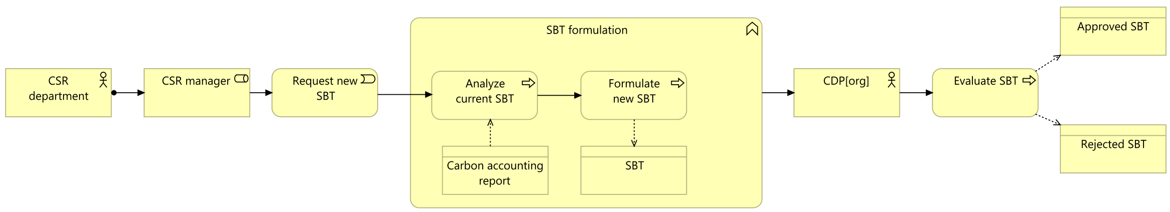


Fig. 23: Business Process of Science-Based Target (SBT) formulation.

The CSR-department 'assigns' the CSR-manager to 'trigger' a business event 'Request new SBT'. The Open Group [61] state that a business event "represents an organizational state change". Aforementioned business process 'triggers' the business function 'SBT formulation', which groups the business processes 'Analyze current SBT and 'Formulate new SBT'. To analyze current SBTs, historical carbon accounting reports are 'accessed'. This is to verify whether CGI is on track with meeting SBTs. Once current SBTs are analyzed, this business process triggers 'Formulate new SBT' which 'writes' a new data object 'SBT'. Once the business processes in the business function (SBT formulation) are completed, the CDP is triggered to evaluate CGI's SBT, which can either be approved or rejected. Once the SBT is approved, it can be stored in the database.

5.2.3 Data transformation to tonnes of Carbon dioxide equivalent (CO_2e)

The business function 'Data transformation to tCO_2e' is modelled more detailed in Fig. 24. The main difference between Fig. 24 and Fig. 22 is the added business function 'Data merger' before calculating tCO_2e values. Since the data comes from variable sources and has various formats, the Data merger aggregates the data in a single dataset, represented by the 'carbon accounting data' business (data) representation. The 'carbon accounting data', which then flows to 'Data transformer to tCO_2e', which calculates tCO_2e-values which are part of carbon accounting reports.

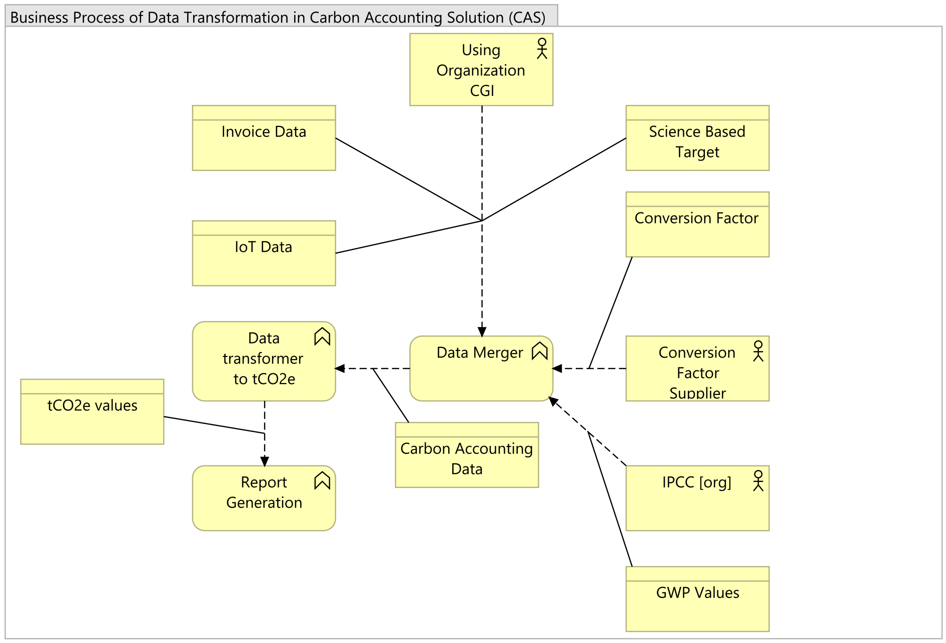


Fig. 24: Business Process of Carbon Accounting Solution (CAS) data transformer.

5.3 Application Layer

The Open Group [63] states that "the Application Layer elements are typically used to model the Application Architecture that describes the structure, behavior, and interaction of the applications of the enterprise". For CAS, the structure of the solution is modelled, as well as the behavior (data reading, writing, visualization, retrieval) and interaction (with an API) is modelled in Fig. 25.

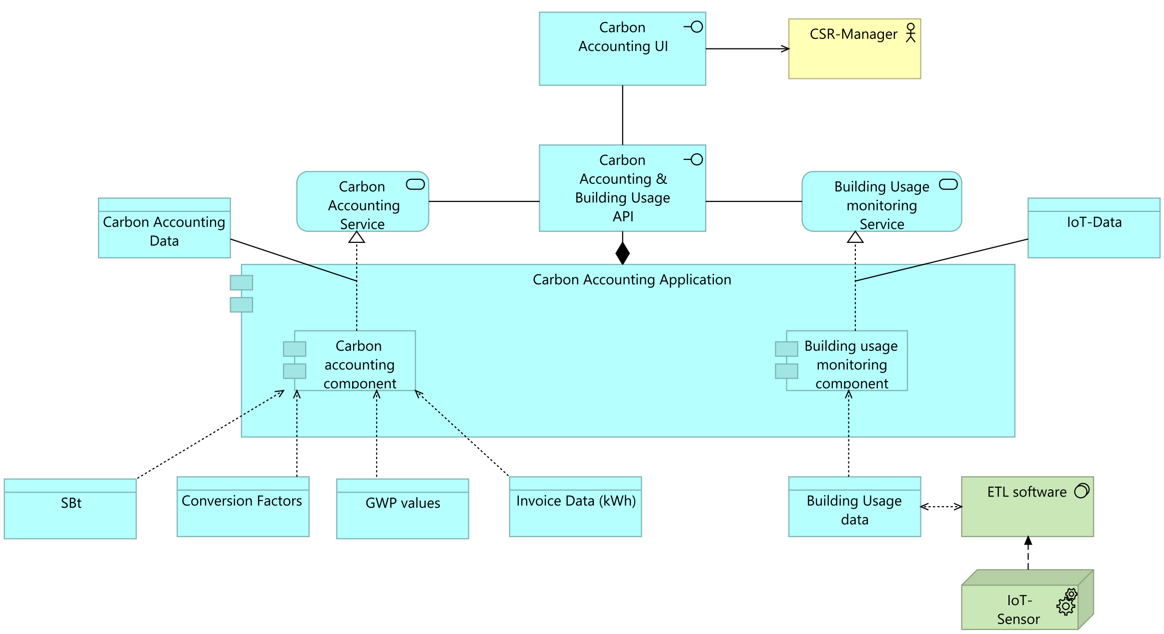


Fig. 25: Application Layer model of Carbon Accounting Solution (CAS).

The 'carbon accounting User Interface (UI)' both uses and is used by the 'Carbon Accounting and Building Usage Application Programming Interface (API)' and serves the CSR-Manager. Via the

Carbon accounting UI data can be loaded into CAS, but also retrieved from CAS via CAS's 'Carbon Accounting and Building Usage Application Programming Interface (API)'. The 'Carbon Accounting and Building Usage Application Programming Interface (API)' uses and is used by two services; 'Carbon Accounting Service' and 'Building Usage monitoring Service', which in turn are 'realized by' the 'Carbon accounting component' and 'Building usage monitoring component'. Aforementioned components are part of the Carbon Accounting Application. The Carbon Accounting Application is 'composed' of the aforementioned components. The 'Carbon accounting component' aggregates data objects; SBT, conversion factors, GWP values and Invoice data (kWh). Behind the 'carbon accounting component', equations 1 and 2 are executed, see below.

$$\sum_{i=1}^{i=n} L_i \times CF_i \tag{1 revisited}$$

Where:

L: is the load in [kWh]

CF_i: is the conversion factor in [kg CO_2/kWh]

i: is consumption

$$kg\ CO_2 \times GWP \tag{2 revisited}$$

Where:

$kg\ CO_2$: is the amount of CO_2 in [kg]

GWP: is the Global Warming Potential in [kg CO_2e/kg CO_2]

The carbon accounting component has as output 'carbon accounting data'. The building usage monitoring component aggregates building usage data, which is accessed from extraction-transformation-loading (ETL)-software (middle layer, see 4.5) which receives data via a flow relationship from IoT-sensor(s). The carbon accounting application is composed of the carbon accounting and building usage API. The IoT-data together with the carbon accounting data serve as input for correlation analysis. For example, the carbon accounting data may show an increase in electricity consumption, which is correlated with the (inefficient) use of Heating, Ventilation and Airconditioning (HVAC) equipment on a specific floor shown by IoT-data.

5.4 Technology Layer

The Open Group [64] states that "the Technology Layer elements are typically used to model the Technology Architecture of the enterprise, describing the structure and behavior of the technology infrastructure of the enterprise". For CAS, the technical infrastructure is shown in Fig. 26.

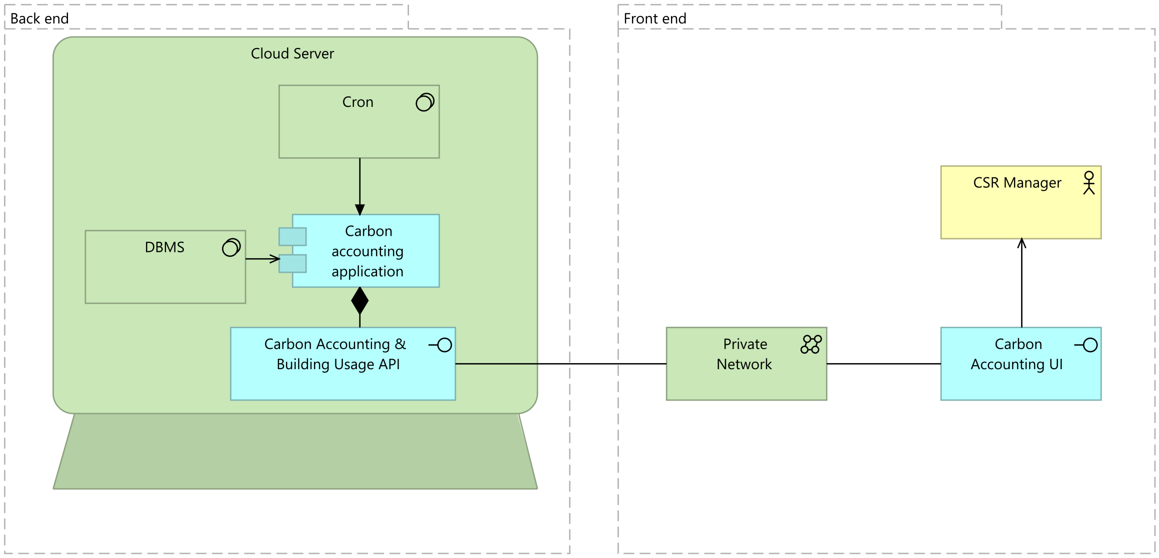


Fig. 26: Technology Layer model of Carbon Accounting Solution (CAS).

The cloud server is part of the back-end. Which cloud service is used (Azure, AWS, ..,) is beyond the scope of this thesis. 'Cron' (see 4.3 for description) triggers the 'Carbon accounting application' which is 'served by' a 'Database Management System (DBMS)'. The DBMS serves the 'Carbon accounting application' by providing carbon accounting reports. The 'carbon accounting application' is composed of the 'carbon accounting and building usage API, which in turn communicates with the 'Carbon Accounting UI' via a private network. The 'Carbon Accounting UI' serves the CSR-manager by providing carbon accounting reports. The private network shows that the CSR-manager can connect to CAS via the 'Carbon Accounting UI' on a browser in a local device. Via a private network the cloud server on which CAS runs is connected. How the cloud server is connected to further networks, is beyond the scope of this thesis (network engineering problem). For example, it may be very well possible that CGI uses multiple cloud servers located in different countries to run CAS. The networking between CGI's servers needs to be established on a Virtual Private Network (VPN). Such networking challenges are beyond the scope of this thesis.

5.5 Enterprise Architecture Demonstration and Evaluation

With the presented EAMs in Section 5.2, 5.3 and 5.4, the demonstration and evaluation of the EAMs was done by using the requirements formulated in Chapter 4. The EAMs were demonstrated and evaluated by verifying whether the requirements are met, which was done using feedback (listen in Table E.3) from the ICT-Architects (listed in Table E.1) by assessing certain components (listed in Table E.2) The requirements are summarized in Table 16 below, which is the same as the Table presented in Chapter 4.

Table 16: Overview of covered requirements byEnterprise Architecture Model (EAM) of Carbon Accounting Solution (CAS).

Code	Description	Demonstrated
FRQ_1	The EAM shows how the CSR managers are facilitated in data to support the CSR-Reporting.	✓
FRQ_2	The EAM shows the practice of carbon accounting, monitoring and quantification according to The Greenhouse Gas Protocol Scope 2 standard.	✓
FRQ_3	The EAM shows the practice of carbon accounting, monitoring and quantification according to the EN ISO 14064-1:2012 standard for quantification and reporting of GHG emissions and removals at organization level.	✓
FRQ_4	The EAM shows how data from IoT-devices can be used to facilitate insight in a building's electricity usage.	✓
FRQ_5	The EAM shows how invoice data can be transformed to tonnes of CO_2 per month.	✓
FRQ_6	The EAM shows the recording and monitoring of SBTs.	✓
FRQ_7	The EAM shows user's access to the data platform through a security and authorization subsystem.	✗
FRQ_8	The EAM shows how carbon accounting data and data from IoT-devices is stored through cloud data storage.	✓
FRQ_9	The EAM shows data lineage to ensure data traceability in the data platform implementation.	✓
FRQ_{10}	The EAM shows how an API handles requests between the database and CAS by providing JSON-formatted data.	✓
FRQ_{11}	The EAM shows how JSON is used to transport and store data within CAS.	✓
$NFRQ_1$	The EAM shows modifiability of CAS	✓
$NFRQ_2$	The EAM shows how monthly carbon accounting reports are generated	✓
$NFRQ_3$	The EAM shows adaptability of CAS	✓
$NFRQ_4$	The EAM shows modularity of the CAS (Additional NFRQ identified after evaluation)	✓

FRQ_1 is demonstrated by Figures; 21, 22, 25 and 26. Fig. 21 shows how the CSR-manager uses a business service 'Carbon Accounting Service' which is triggered by a business function 'Report Generation Function' and business event 'Request carbon accounting report'. The 'Report Generation Function' has as output a carbon accounting report. The 'Report Generation Function' is shown in Fig. 22 with relation to other business functions of CAS and the data flows. Although Fig. 25 doesn't show the CSR-manager as an actor, it is shown how the carbon accounting application aggregates data, and how via an API and carbon accounting UI data can be retrieved and loaded in the application. Via the carbon accounting UI the CSR-manager communicates with the carbon accounting application, demonstrated by Fig. 26.

FRQ_2, FRQ_3 and FRQ_{11} are demonstrated in Section 4.4. Fig. 27 shows how a CAS-user receives carbon accounting data in JSON format.

Fig. 27: Illustration of JSON formatted data wherein user receives carbon accounting data from CAS.

FRQ_4 is demonstrated by Fig. 25. Fig. 25 shows how building usage data is collected from IoT-equipment through ETL-middleware. The building usage data is aggregated by the carbon accounting application, which realizes a building usage monitoring service connected to a dashboard (Carbon accounting and building usage API). This dashboard is connected with the carbon accounting UI, which can be used by the CSR-manager to retrieve data, see Fig. 26.

FRQ_5 is demonstrated by Fig. 24, which shows how CAS uses the business functions 'data merger' and 'data transformer to tCO_2e' to create carbon accounting data and tCO_2e values which are inputs for the 'report generation' business function.

FRQ_6 is demonstrated by Figures 22, 23 and 25. Fig 22 merely shows how CGI and the SBTi collaborate to create an SBT, which is associated to a data flow entering CAS in the 'data transformer to tCO_2e' business function. Fig. 23 shows the entire process of formulating the SBT, for a full description see 5.2.2. Fig. 25 shows the SBT as a data object which is accessed by the 'carbon accounting application' to generate 'carbon accounting data' which is provided by a business service 'carbon accounting service' that is used by a 'carbon accounting and building usage API'.

FRQ_7 is not demonstrated by any EA-model. According to [EA1]: "Security and authorization subsystems can not be properly modelled in ArchiMate". See also F_{7b} in Table E.3.

FRQ_8 and FRQ_{10} are demonstrated by Fig. 26. Fig. 26 shows how the carbon accounting application runs on a cloud server, which furthermore includes the DBMS, Cron and the carbon accounting and building usage API. The carbon accounting UI is used to connect with the API, to store all data on the cloud server. The DBMS facilitates the storage of data in relational databases.

FRQ_9 is not demonstrated by any model. However, according to [IV8]: "the data lineage concept is facilitated by the OSDU-platform's reference architecture". Meaning if CAS is developed on the OSDU platform, the data lineage function will be an integral part of CAS provided by the OSDU's reference architecture. According to [EA1], EA-models are high level, and data lineage can not be modelled as high level. [EA1] furthermore states that the requirement can be covered if the OSDU-platform facilitates it.

$NFRQ_1$; Modifiability is a sub-component of maintainability. According to ISO/IEC [65] maintainability "represents the degree of effectiveness and efficiency with which a product or system can be modified to improve it, correct it or adapt it to changes in environment, and in requirements". ISO/IEC [65] furthermore state regarding modifiability; "Degree to which a product or system can be effectively and efficiently modified without introducing defects or degrading existing product quality". F_{2a} in E.3, Table E.3 mentions $NFRQ_1$, which is shown in pink color in Fig. 28.

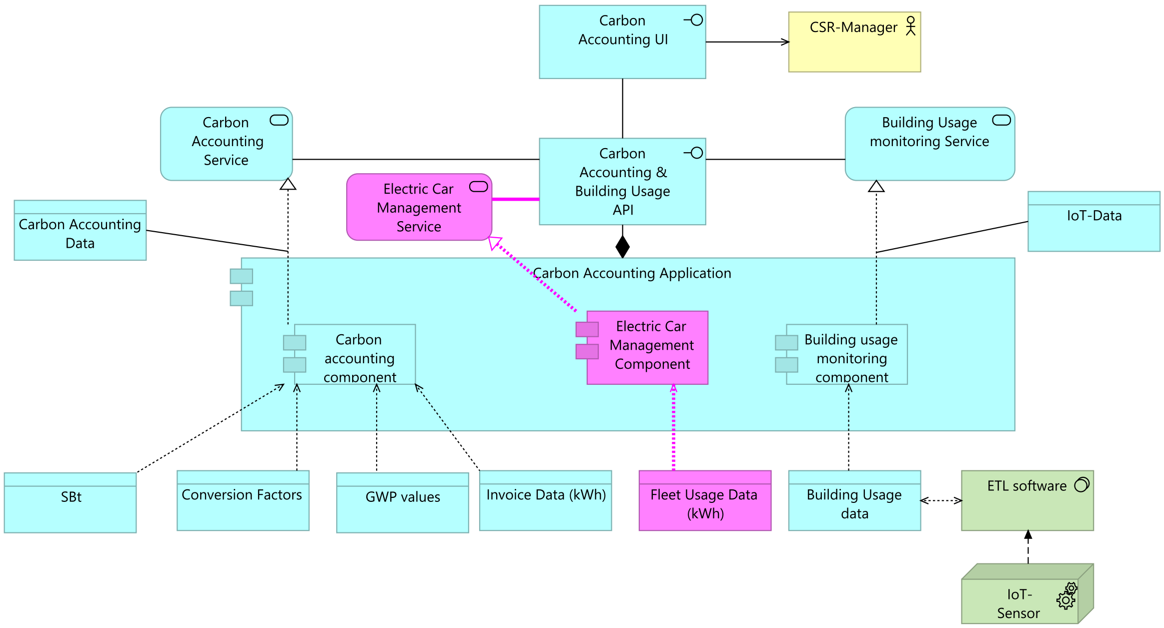


Fig. 28: Modifiability of Carbon Accounting Solution (CAS).

The model can be modified by adding a data object 'fleet usage data' and a new component for electric car management. This component should facilitate a calculation which converts kWh to tCO_2e values. Aforementioned component realizes an electric car management service, which is connected to the API. Modifiability is demonstrated by adding a new data source (Fleet usage data), component (Electric car management component) and application service (Electric car management service) which is connected to the Carbon Accounting and Building Usage API. Adding aforementioned elements will not affect the overall functionality of the EA-model and of the CAS. The relationship between adaptability and modifiability is as follows. The adaptability pertains to how the CAS can be adapted for different usage environments or evolving hardware. Whereas modifiability focuses on the CAS itself, by showing how modifications can be done efficiently while having minimal impact on the CAS's quality.

$NFRQ_2$ is demonstrated by Fig. 26, which shows how Cron triggers the carbon accounting application to generate a monthly report. This report is retrieved via the API and carbon accounting UI.

$NFRQ_3$ mentions adaptability, which according to ISO/IEC [65] is the "degree to which a product or system can effectively and efficiently be adapted for different or evolving hardware, software or other operational or usage environments". $NFRQ_3$ can be demonstrated as follows. The cloud server can use different cloud services (Azure, AWS,..,), which shows a level of adaptability. Furthermore, a client can use CAS only if there is a connection to the back-end (cloud server), see Fig. 29. In pink the front end is now a different client, which uses a 'sustainability manager' to communicate with CAS. [EA1] mentions that adaptability is furthermore demonstrated by using a cloud server, because cloud servers can be Linux servers, Windows servers, IBM servers etc. The cloud server provider which is variable demonstrates adaptability, also because the model in Figures 26 and 29 mention a 'cloud server'. This demonstrates that CAS can run on any server.

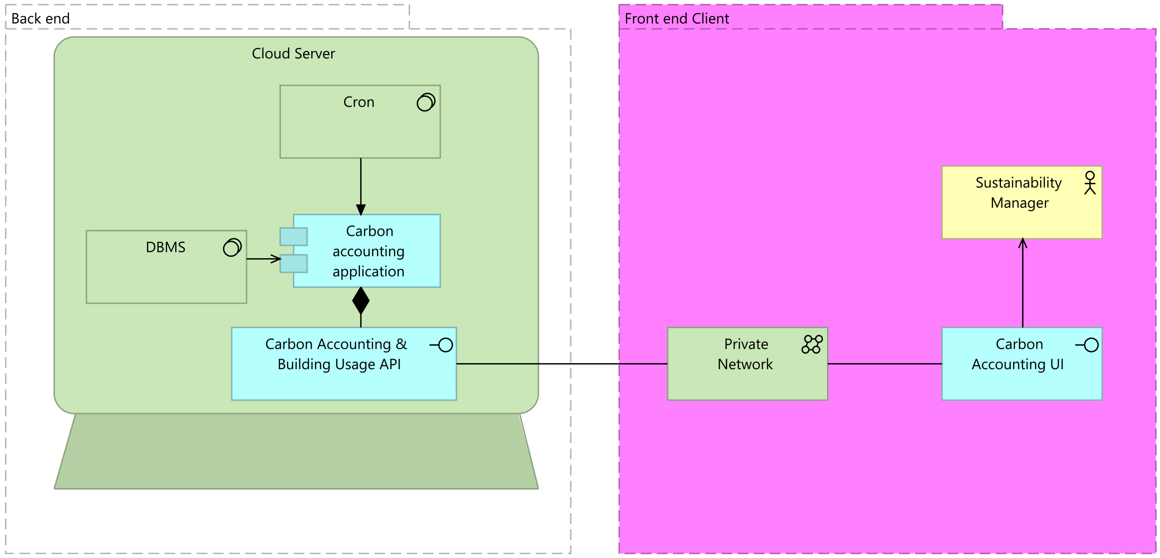


Fig. 29: Adaptability of Carbon Accounting Solution (CAS).

After evaluation, it was identified that one more non-functional requirement is covered by the EAM. [EA1] states that modularity is demonstrated by the EAM, which according to ISO/IEC [65] is the "degree to which a system or computer program is composed of discrete components such that a change to one component has minimal impact on other components". This non-functional requirement which is now called $NFRQ_4$ (The Enterprise Architecture Model (EAM) shows modularity of CAS), is demonstrated by the layered modelling in ArchiMate. Changes in the application layer do not affect the business nor technology layer. Changes in the business layer do not affect the application nor technology layer. Changes in the technology layer do not affect the business nor application layer. Consider Fig. 28 again. By adding: data object 'Fleet usage data', component 'Electric car management component' and service 'Electric car management service', changes are made to the 'Carbon Accounting Application' and the API in this specific layer. However, in the technology layer (Fig. 26) the 'Carbon Accounting Application' and API stay the same, also in terms of functionality and interface usage. There is no impact on the components in the technology layer as a result of changes in the application layer.

5.6 Conclusion

Chapter 5 addressed the fourth activity in the Design Science Research Methodology (DSRM); demonstration and evaluation of an artifact. The artifact being the Enterprise Architecture Model (EAM). To produce the EAM, SRQ_4 was answered:

SRQ_4: *"To what extent does CAS's initial enterprise architecture model comply with the requirements?"*

The input for SRQ_4 was the initial EAM presented in Chapter 4. Through evaluation sessions with the ICT-Architects, the initial EAM was adapted and multiple more detailed EAM were developed. These EAM were presented in Chapter 5 and complied with 93% of the requirements presented in Table 16. FRQ_9 could not be modeled with ArchiMate, however FRQ_9 can be complied with if the CAS is developed on the OSDU-platform. Furthermore, $NFRQ_4$ was added later to the requirements, since $NFRQ_4$ was discovered by [EA1] after evaluation in feedback session 7 (see F_{7a} in Table E.3)

6 Conclusions And Reflection

Chapter 6 covers activity 5 of the DSRM; Communicate artifact. The objective was answering the main research question (MRQ): "*Which Enterprise Architecture Model (EAM) is needed for a Carbon Accounting Solution (CAS) to facilitate carbon accounting, monitoring and quantification for organizations?*". The MRQ will be addressed in Section 6.1. Section 6.2 addresses how CGI can implement a Carbon Accounting Solution (CAS) using the output of the research. The research limitations are furthermore addressed in Section 6.2. Future research directions are elaborated upon in Section 6.3. Finally, the link of the research to the study program Management of Technology (MoT) is addressed in Section 6.4.

6.1 Main Research Question

The research objective was to develop a new ICT-artifact, which is an Enterprise Architecture (EA)-model of a Carbon Accounting Solution (CAS) which shows the ICT-architecture of aforementioned solution to facilitate carbon accounting, monitoring and quantification on an organizational level in emission scope 2. To achieve the research objective, the following MRQ was formulated:

"*Which Enterprise Architecture Model (EAM) is needed for a Carbon Accounting Solution (CAS) to facilitate carbon accounting, monitoring and quantification for organizations?*"

The research was conducted at the ICT- and Business Consulting Company CGI. By researching which EAM was needed for carbon accounting in emission scope 2, a use case was executed for CGI. The use case allowed for accessing empirical data by interviewing CGI-members with different expertise in activities 2 (Requirements Elicitation), 3 (Design and Develop Artifact) and 4 (Demonstrate and Evaluate Artifact). The MRQ was answered by using the Design Science Research Methodology (DSRM). By following the DSRM's activities, sub-research questions were answered which each gave respective research outputs, which can be read in Subsection 6.1.1. The outputs of the DSRM together created the EA-model (artifact) and furthermore answered the MRQ. See Table 17 for an overview of the DSRM activities and in which Subsection the SRQs are covered.

Table 17: DSRM Activities, Sub-research questions (SRQ) and Subsections which address Activities.

	Activity	**SRQ Addressed**	**Subsection**
1	Problem Identification and Motivation	1	6.1.1
2	Define Objectives of the Solution	2	6.1.2
3	Design and Develop Artifact	3	6.1.3
4	Demonstrate and Evaluate Artifact	4	6.1.4
5	Communicate Artifact	MRQ	6.1.5

6.1.1 Problem Identification and Motivation

Activity 1 was addressed by answering SRQ_1:

What is the knowledge base on which the development of an Enterprise Architecture Model (EAM) for a Carbon Accounting Solution (CAS) should rely?

The knowledge base identified in activity 1 elaborated on concepts in order to develop an EAM for a Carbon Accounting Solution. Since CGI wants do develop a CAS using the Open Source Data Universe (OSDU)-Data Platform. Therefore, the first covered concept was data platform technology and ecosystems. It was identified that data platforms can leverage development of new capabilities and innovations. The elements of platform ecosystems were elaborated upon: the platform, applications, ecosystems, interfaces and architecture. The ecosystem architecture consists of platform architecture and the application microarchitecture. In this use case, the platform architecture is the OSDU's reference architecture and the application microarchitecture is the CAS's architecture, which is modeled by an EA-model.

Carbon accounting as a concept was introduced. Carbon accounting refers to "the activity of measuring [direct and indirect] carbon emissions and removals and retaining an ongoing inventory of operations-based emissions" [5, p. 57]. The differences between emission scope 1, 2 and 3 were identified. Scope 1 entails direct emissions, scope 2 entails indirect emissions from imported utilities and scope 3 entails indirect emissions that a site causes to occur but where it does not control the asset. Carbon accounting and reporting standards such as the Greenhouse Gas Protocol and ISO 14064 were identified. Finally, how Carbon dioxide equivalent (CO_2e) values can be calculated using GWP values, conversion factors, and activity data was identified.

6.1.2 Define Objectives of the Solution

Activity 2 was addressed by answering SRQ_2:

What are requirements for a Carbon Accounting Solution (CAS) to develop an Enterprise Architecture Model (EAM)?

The requirements were elicited through 13 expert interviews with CGI-members. Where needed, scientific literature was used to formulate and assess the requirements. This resulted in a list of functional and non-functional requirements (See Table 18), serving as the input for the initial EAM.

Table 18: Functional and non-functional requirements for the Enterprise Architecture Model (EAM) of the Carbon Accounting Solution (CAS).

Code	Description
FRQ_1	The EAM illustrates how the CSR managers are facilitated in data to support the CSR-Reporting.
FRQ_2	The EAM illustrates the practice of carbon accounting, monitoring and quantification according to The Greenhouse Gas Protocol Scope 2 standard.
FRQ_3	The EAM illustrates the practice of carbon accounting, monitoring and quantification according to the EN ISO 14064-1:2012 standard for quantification and reporting of GHG emissions and removals at organization level.
FRQ_4	The EAM illustrates how smart-data can be used to facilitate insight in a building's electricity usage.
FRQ_5	The EAM illustrates how invoice data can be transformed to tonnes of CO_2 per month.
FRQ_6	The EAM illustrates the recording and monitoring of SBTs.
FRQ_7	The EAM illustrates user's access to the CAS through a security and authorization subsystem.
FRQ_8	The EAM illustrates how carbon accounting data and smart-data is stored through cloud data storage.
FRQ_9	The EAM illustrates data lineage to ensure data traceability in the CAS implementation.
FRQ_{10}	The EAM captures how an API handles requests between the database and the CAS through exchanging JSON data
FRQ_{11}	The EAM illustrates how JSON is used to transport and store data within CAS.
$NFRQ_1$	The EAM captures how monthly carbon accounting reports are generated in an efficient manner.
$NFRQ_2$	The EAM captures the use of CAS in different time intervals.

6.1.3 Design and Develop Artifact

Activity 3 was addressed by answering SRQ_3:

How can the identified requirements for a Carbon Accounting Solution (CAS) serve as a building block for the development of CAS's Enterprise Architecture Model (EAM)?

Activity 3 had multiple outputs:

1. Context-diagram;
2. Level 1 data flow diagram (DFD);
3. Initial Enterprise Architecture Model of the Carbon Accounting Solution.

Number 1 and 2 were developed after feedback of ICT-Architects. The context diagram shows the inputs, outputs and actors of CAS, see Fig. 30.

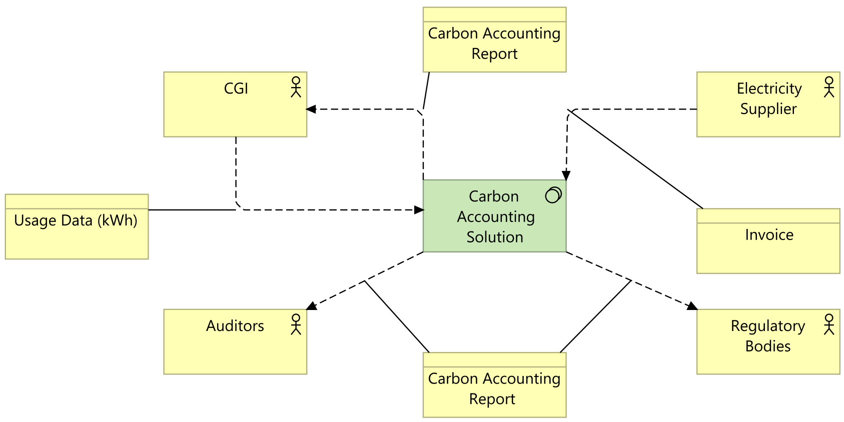


Fig. 30: Context diagram of the Carbon Accounting Solution (CAS).

For level 1 DFD, see Fig. 31.

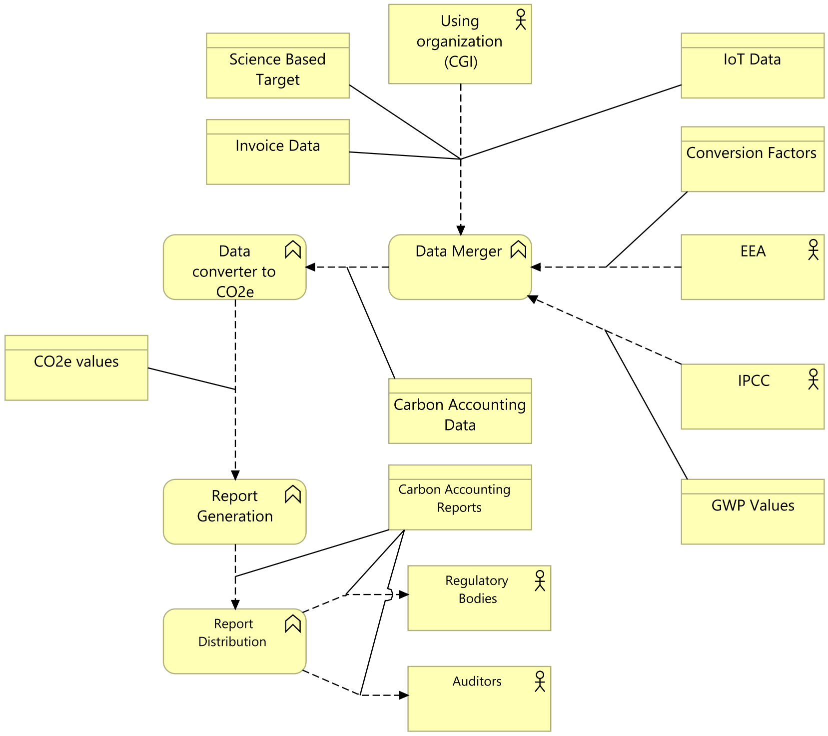


Fig. 31: Level 1 DFD diagram of the Carbon Accounting Solution (CAS).

The initial EA-model of the CAS is shown in Fig. 32

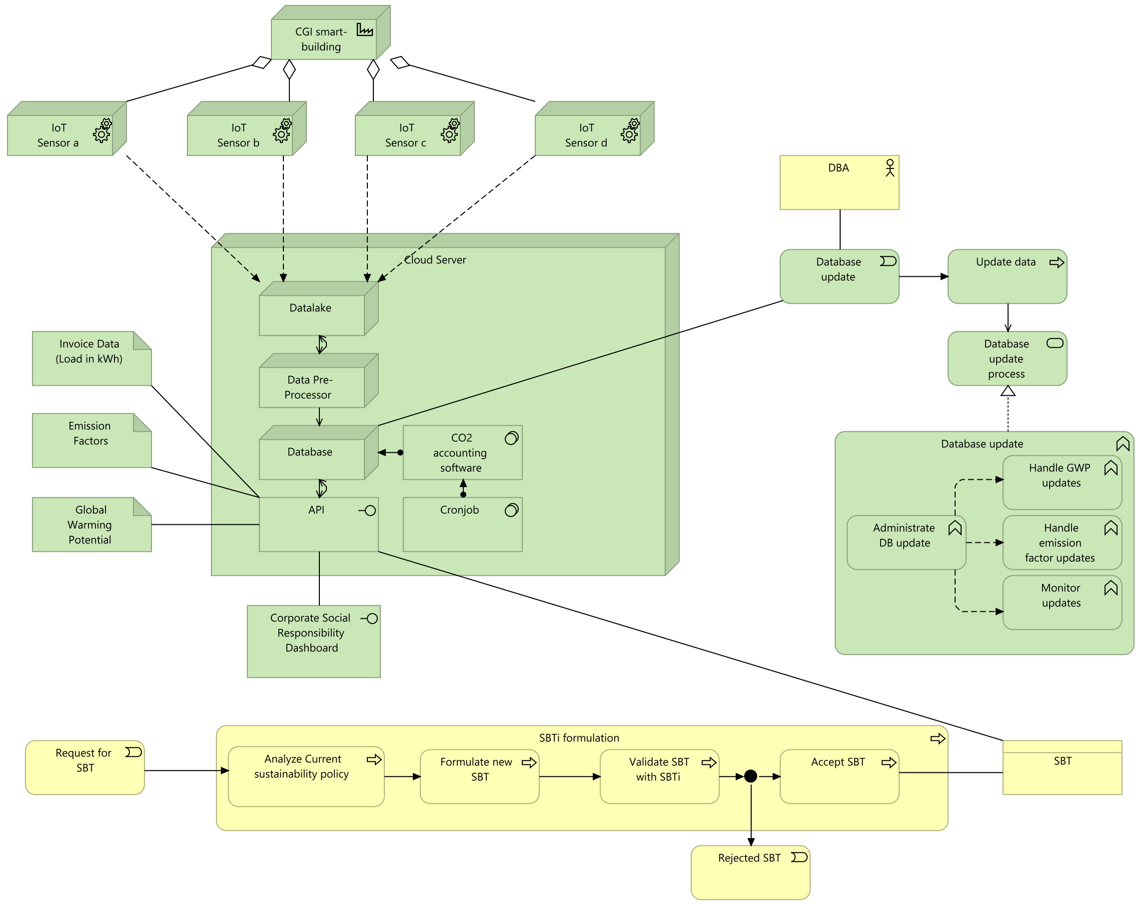


Fig. 32: Initial EA-model of the CAS.

The EA-model as shown in Fig. 32 was split up to multiple EAMs in the next activity with the use of feedback from CGI's ICT-Architects.

6.1.4 Demonstrate and Evaluate Artifact

Activity 4 was addressed by answering SRQ_4:

SRQ_4: *"To what extent does the Carbon Accounting Solution (CAS)'s initial Enterprise Architecture Model (EAM) comply with the requirements?"*

The output of SRQ_4 is a modified list of requirements (see Table 19) which is the input for several Enterprise Architecture-models, which are summarized in Fig. 33. The requirements were evaluated using the feedback of 7 ICT-Architecture sessions with 5 ICT-experts from CGI. The EA-models were evaluated by the ICT-Architecture experts, see Appendix E for the evaluation framework used for developing the EA-models.

Table 19: Functional and non-functional requirements for the enterprise architecture model of the Carbon Accounting Solution.

Code	**Description**
FRQ_1	The enterprise architecture model shows how the CSR managers are facilitated in data to support the CSR-Reporting.
FRQ_2	The enterprise architecture model shows the practice of carbon accounting, monitoring and quantification according to The Greenhouse Gas Protocol Scope 2 standard.
FRQ_3	The enterprise architecture model shows the practice of carbon accounting, monitoring and quantification according to the EN ISO 14064-1:2012 standard for quantification and reporting of GHG emissions and removals at organization level.
FRQ_4	The enterprise architecture model shows how data from IoT-devices can be used to facilitate insight in a building's electricity usage.
FRQ_5	The enterprise architecture model shows how invoice data can be transformed to tonnes of CO_2 per month.
FRQ_6	The enterprise architecture model shows the recording and monitoring of SBTs.
FRQ_7	The enterprise architecture model shows user's access to the data platform through a security and authorization subsystem.
FRQ_8	The enterprise architecture model shows how carbon accounting data and data from IoT-devices is stored through cloud data storage.
FRQ_9	The enterprise architecture model shows data lineage to ensure data traceability in the data platform implementation.
FRQ_{10}	The enterprise architecture model shows how an API handles requests between the database and the CAS by providing JSON-formatted data.
FRQ_{11}	The enterprise architecture model shows how JSON is used to transport and store data within the CAS.
$NFRQ_1$	The enterprise architecture model shows modifiability of the CAS
$NFRQ_2$	The enterprise architecture model shows how the CAS generates monthly carbon accounting reports
$NFRQ_3$	The enterprise architecture model shows adaptability of the CAS
$NFRQ_4$	The enterprise architecture model shows modularity of the CAS (Additional NFRQ identified after evaluation)

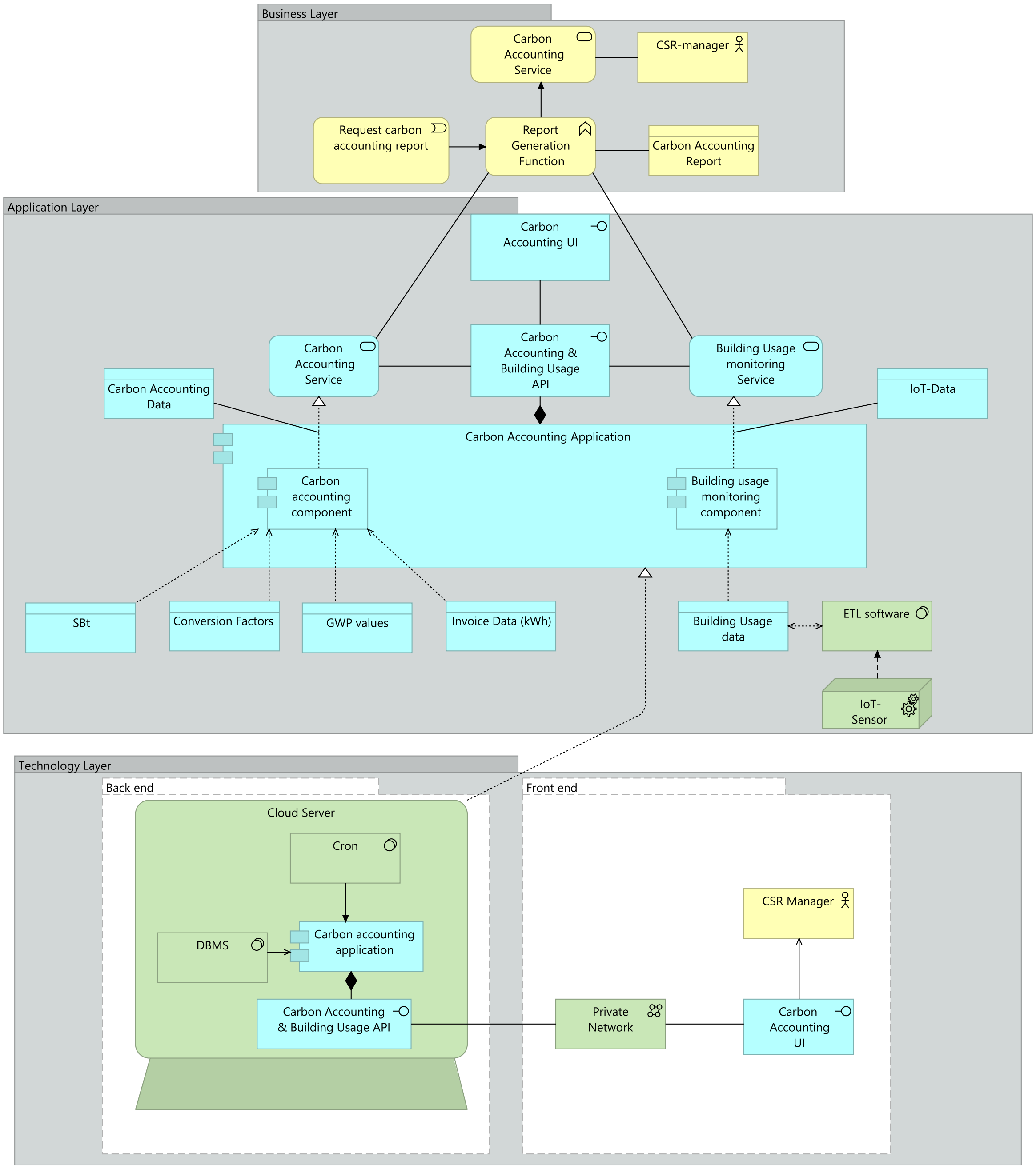


Fig. 33: Enterprise Architecture (EA)-model of CAS.

6.1.5 Communicate Artifact

Activity 5 was addressed by answering the MRQ:

Which Enterprise Architecture Model (EAM) is needed for a Carbon Accounting Solution (CAS) to facilitate carbon accounting, monitoring and quantification for organizations?

The MRQ is answered using Fig. 33. The EAM of CAS in Fig. 33 enables CGI to develop a Software as a Service (SaaS) which facilitates carbon accounting, monitoring and quantification on an organizational level. Fig. 33 shows that the business function 'Report Generation' triggers a business service 'Carbon Accounting Service' which uses and is used by the 'CSR-Manager'.

The 'Report Generation' business function uses the application services 'Carbon Accounting' and

'Building Usage Monitoring', which are realized by the 'Carbon Accounting Application'. The 'Carbon Accounting Application' aggregates data objects; Science-Based Target (SBT), conversion factors, Global Warming Potential (GWP) values and Invoice data to generate 'Carbon Accounting Data' through a 'Carbon Accounting Component'. The 'Building Usage Monitoring Component' aggregates 'Building Usage Data', which is offered by an extraction-transformation-loading (ETL) middleware layer which structures data from the IoT-sensor(s). The aforementioned services use and are used by a 'Carbon Accounting and Building Usage Application Programming Interface (API)', which in turn uses and is used by a 'Carbon Accounting User Interface (UI)'.

Fig. 33 shows that the Carbon Accounting Application runs on a Cloud Server and that the Cloud Server 'realizes' the Carbon Accounting Application. The cloud server furthermore contains; the Database Management System (DBMS) which controls storage of data in relational databases, Cron to trigger monthly creation of carbon accounting reports, and the 'Carbon Accounting Application' and 'Carbon Accounting and Building Usage Application Programming Interface (API)'. Via a private network, the 'Carbon Accounting and Building Usage API' is connected to a 'Carbon Accounting UI', which serves the corporate social responsibility (CSR)-Manager. The cloud server and components form the back-end, the front-end is used by the CSR-manager to communicate with CAS via a local browser.

The EA-model of CAS covers and demonstrates 14/15 of the requirements as formulated in Table 19, excluding FRQ_7 (The enterprise architecture model shows user's access to the data platform through a security and authorization subsystem), due to ArchiMate's limited ability to model such a requirement.

The findings of the research can be generalized to similar organizations such as CGI, i.e. office buildings. The business processes, application architecture, and physical infrastructure for such buildings will not vary much. Especially since CAS can run on a remote cloud server, and the interfaces can be accessed through APIs via private networks. Policy wise, all organizations should report to the CDP and the reporting standards of the Greenhouse Gas Protocol and ISO 14064 should be complied with. The findings of the research can be generalized in lesser degree to energy intensive facilities, because the IoT-component is more prominent in such facilities. The IoT-component in this research is simplified, and serves as a basis for further development efforts of CAS.

Academic and Societal Contribution

The thesis adds to the academic body of knowledge because it is the first Enterprise Architecture Model (EAM) which shows how carbon accounting can be facilitated on an organization level. Furthermore, key business processes, policies, actors, application architecture and the architecture of physical infrastructure were identified in the thesis. The thesis provides a comprehensive overview and shows relationships between aforementioned. The elicited requirements for developing a Carbon Accounting Solution (CAS) may serve as input for more advanced development of carbon accounting systems.

The societal contribution of the thesis consists of indirect environmental benefits the EAM will initiate and facilitate. When (energy-intensive) organizations use the EAM to adopt a Carbon Accounting Solution, more insight in current emissions will be generated through carbon accounts. Organizations can act upon insights which are data-driven by adopting mitigating efforts to halt indirect emissions. Finally, CAS enables monitoring the impacts organizational interventions have on the reduction of CO_2-emissions.

Compliance with Design Science Research Project Requirements

The requirements Design Science Research projects need to fulfill were listed on page 4. Johannesson and Perjons [16, p. 8] mention the following: (1) rigorous research methods are required to produce new knowledge of general interest, (2) the knowledge produced has to be related to an already existing knowledge base, in order to ensure that proposed results are both well founded and original, and (3) the new results should be communicated to both practitioners and researchers.

The rigorous research methods used in the thesis consists of: 13 semi-structured interviews, assessment of interview findings with scientific literature, and the use of scientific literature to strengthen the knowledge base. The knowledge produced (Enterprise Architecture Model) contributes to the existing knowledge base of platform technology, enterprise architecture, and GHG-accounting. The results were communicated with scientific peers and ICT-professionals through the thesis and public defense. Furthermore ICT-architects were involved throughout development of the EAM, which resulted in continuous communication. In short, the results were communicated to researchers (from the TU-Delft) and practitioners (from CGI).

6.2 Implementation Recommendations and Research Limitations

6.2.1 Implementation Recommendations

The following recommendations to implement CAS are presented. CAS is a SaaS developed on the Open Source Data Universe (OSDU)-Platform. This means that CGI's developers need to be mindful of the OSDU's reference architecture. Consider FRQ_9: The EA model shows data lineage to ensure data traceability in CAS. This requirement is not covered in the EA-models, since the OSDU facilitates the data lineage concept provided that the application is developed on the OSDU-Platform. Therefore, implementation of data audit trails is not the concern of CGI's software engineers.

Following the EA-model of CAS as in Fig. 33 will allow CGI to develop a SaaS which is adaptable, modular, and modifiable, due to the feature of adding components to the carbon accounting application, which can realize different services (e.g Electric Car Management Service) that can be connected to the already existing Carbon Accounting and Building Usage API. Modularity is provided by the EA-model due to the layered modeling. This means that CGI's software engineers should follow the EA-model when developing CAS.

The final implementation recommendation pertains to the cloud server storage. The EA-model of CAS does not specify which cloud server and which cloud service has to be used, since the functionality of CAS is independent of the aforementioned. The same holds for the automatic scheduler Cron, which can run on any cloud server using any cloud service. This provides CGI's software developers and clients with freedom regarding which cloud servers and cloud services to use.

6.2.2 Research Limitations

In this specific research, the objective was to develop an Enterprise Architecture Model of a Carbon Accounting Solution which shows the ICT-Architecture needed to facilitate carbon accounting, monitoring and quantification on an organization level. The limitation of this research lies in the latter part of the research objective which specifies the organizational level, the organization being CGI here. One component of the research and EAM is IoT-sensors to gain insight in building usage data. It is debatable whether IoT-Technology is necessary and relevant for office buildings like CGI, since such buildings do not have many energy usage fluctuations compared to energy intensive facilities (oil- and gas facilities, (steel) production facilities). The IoT-data should facilitate correlation analysis for CGI, by showing explanations for increase/decrease of tCO_2e-values (as shown in carbon accounting reports) with the use of IoT-data showing increase/decrease/(in)efficiencies in energy consumption of for example Heating, Ventilation and Airconditioning (HVAC)-equipment. However, such correlation analysis and IoT-data may be more fruitful for energy intensive facilities. There is a possibility that the IoT-sensor data may not result in extra insights for CGI, which would actually mean that the extra energy used for IoT-sensors and software is a waste, and the IoT-component of the EAM becomes obsolete. The reason for including IoT-Technology in the research and EAM, is because this allows CGI to use the EAM to develop a CAS which can be used for their respective clients. The IoT-components in the models show a rather simplified view of what in general are complex systems. The point was to show where data from IoT-devices comes from, how a middleware layer creates uniform data, and how this data in conjunction with the carbon accounts can give insights in the effectiveness of emission reducing interventions.

Another research limitation pertains to the modeling method language and tool (ArchiMate language and ArchiMate tool). ArchiMate is strong in modeling the complete Enterprise Architecture (EA) by making distinction between business, application and technology layers. However, ArchiMate's weakness lies in detailed data modeling. For example, detailed data modeling includes models of relational databases. Database modeling, which includes class-diagrams and entity-relationship diagrams is not facilitated by ArchiMate, however with Unified Modelling Language (UML) this can be facilitated. Therefore, there should be considered whether a combination of ArchiMate and UML should be used to provide more detail. The details that UML can facilitate in terms of database modeling, serves the purpose of giving the software- and database engineers more complete information to develop the CAS.

Finally, conversion factors or emission factors should always be up-to-date, so that the calculations for Carbon dioxide equivalent (CO_2e) are correct. Cloud server storage facilitates storage and usage of continuously changing values (conversion factors), which results in up-to-date carbon accounting reports.

6.3 Future Research Directions

6.3.1 Data Governance

Governance refers to what decisions must be made to ensure effective management and use of IT (decision domains) and who makes the decisions (locus of accountability for decision making) [66]. Data governance is a success "practice" to derive business value from data assets [66]. With the rather new phenomenon of carbon accounting, new data assets are created within and across business domains. Multiple actors need access to the data for their specific business goals. The data is furthermore an asset for strategic decision making. Deciding who makes decision regarding the data and with the data, is part of data governance. The following question can be formulated as basis for further research: *Which data governance strategy with respect to carbon accounting maximizes the business value?*
Opportunities: A sound data governance strategy to utilize data as an organizational asset.

6.3.2 Enterprise Architecture-modeling for Carbon Accounting

In this research, the focus was creating an EA-model for a Carbon Accounting Solution (CAS) in emission scope 2 on organizational level. However, the second most carbon intensive emission scope is scope 3, which include indirect emissions from business travel. How does the EA-model change when all emission scopes are included? Furthermore, in this research the input for calculating tCO_2e-values were kWh-values from the energy-supplier's utility bills (invoice data). There needs to be a Carbon Accounting Solution which calculates tCO_2e-values when other primary data is the input (e.g. data from gas utilities). The EA-model developed in this research may serve as input for a research which covers all emission scopes and focuses on energy intensive facilities. The research question could be: *Which Enterprise Architecture (EA)-model is needed to facilitate carbon accounting in all emission scopes for energy intensive facilities?*
Opportunities: An EA-model which serves as a blueprint to develop a complete solution which facilitates carbon accounting for all emission scopes.

6.4 Link to MoT program

According to TU Delft [67], the following criteria are set for an MoT Thesis:

- "the work reports on a scientific study in a technological context (e.g. technology and strategy, managing knowledge processes, research & product development management, innovation processes, entrepreneurship);
- the work shows an understanding of technology as a corporate resource or is done from a corporate perspective;
- students use scientific methods and techniques to analyze a problem as put forward in the MoT curriculum".

This research shows how platform technology in the form of a Software as a Service (SaaS) (Carbon Accounting Solution (CAS)) can be used as an organizational resource, to facilitate carbon accounting, monitoring and quantification in emission scope 2. When organizations use said technology as a resource, this may result in competitive advantage because a sound carbon accounting reporting strategy increases an organization's credibility, therefore attracting more clients. The Design Science Research Methodology (DSRM) is used to create an artifact (Enterprise Architecture (EA)-model) which shows the ICT-Architecture which facilitates the aforementioned goal. Furthermore, in specific phases of the DSRM, semi-structured expert-interviews were conducted.

It is noticeable that organizations (CGI) and industries are under increased pressure to become more 'green'. Furthermore, there is increased emphasis on being able to prove 'how green/CO_2-neutral' an organization is with the use of data. Therefore the Management of Technology (MoT)-program should put more emphasis on environmental aspects by adopting these in the curriculum.

References

[1] K. Blok, *More climate ambition, fewer options: the new Netherlands' coalition agreement*, 2021. [Online]. Available: https://kornelisblok.home.blog/2021/12/17/more-climate-ambition-fewer-options-the-new-netherlands-coalition-agreement/#respond.

[2] UNFCCC, *The Paris Agreement*, 2016. [Online]. Available: https://unfccc.int/process-and-meetings/the-paris-agreement/the-paris-agreement (visited on 10/12/2021).

[3] IPCC, "Summary for policymakers," World Meteorological Organization, Geneva, Tech. Rep., 2018. [Online]. Available: https://www.ipcc.ch/sr15/chapter/spm/ (visited on 10/12/2021).

[4] K. Stechemesser and E. Guenther, "Carbon accounting: a systematic literature review," *Journal of Cleaner Production*, vol. 36, pp. 17–38, Nov. 2012, ISSN: 09596526. DOI: 10.1016/j.jclepro.2012.02.021.

[5] E. Hespenheide, K. Pavlovsky, and M. B. T. F. E. McElroy, "Accounting for sustainability performance: Organizations that manage and measure sustainability effectively could see benefits to their brand and shareholder engagement and retention as well as to their financial bottom line," English, vol. 26, no. 2, p. 52, Oct. 2010, ISSN: 08954186. [Online]. Available: https://link.gale.com/apps/doc/A223140899/AONE?u=anon$\sim$18530d2b&sid=googleScholar&xid=05fdadd6.

[6] KPMG, *Decarbonization & Greenhouse Gas (GHGs)*, 2021. [Online]. Available: https://home.kpmg/be/en/home/services/sustainability-services/carbon-greenhouse-gas-ghg-emissions-accounting.html (visited on 10/12/2021).

[7] IEA, "Digitalisation and Energy," IEA, Paris, Tech. Rep., 2017. [Online]. Available: https://www.iea.org/reports/digitalisation-and-energy.

[8] IEA, "Energy Efficiency," IEA, Paris, Tech. Rep., 2018. [Online]. Available: https://www.iea.org/reports/energy-efficiency-2018.

[9] World Resources Institute, "GHG Protocol Scope 2 Guidance," Tech. Rep. March, 2014. [Online]. Available: https://ghgprotocol.org/sites/default/files/standards/Scope2Guidance_Final_Sept26.pdf (visited on 12/03/2021).

[10] R. Kent, "Chapter 9 - Carbon footprinting," in R. B. T. E. M. i. P. P. T. E. Kent, Ed., Elsevier, 2018, pp. 387–404, ISBN: 978-0-08-102507-9. DOI: https://doi.org/10.1016/B978-0-08-102507-9.50009-X. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B978008102507950009X.

[11] GHG Protocol, "Technical Guidance for Calculating Scope 3 Emissions," World Resources Institute & World Business Council for Sustainable Development, Tech. Rep., 2013, p. 182. [Online]. Available: https://ghgprotocol.org/sites/default/files/standards/Scope3_Calculation_Guidance_0.pdf.

[12] CGI, *CGI Pivot: Unlocking the OSDU Data Platform*, 2021. [Online]. Available: https://www.cgi.com/en/solution/cgi-pivot (visited on 10/27/2021).

[13] CGI, "CGI Pivot Data Platform as a Service," Tech. Rep., 2021. [Online]. Available: https://www.cgi.com/sites/default/files/2021-03/cgi_pivot_factsheet_nl_.pdf.

[14] J. Feder, "Upstream Digitalization Is Proving Itself in the Real World," *Journal of Petroleum Technology*, vol. 72, no. 04, pp. 26–28, Apr. 2020, ISSN: 0149-2136. DOI: 10.2118/0420-0026-JPT. [Online]. Available: https://doi.org/10.2118/0420-0026-JPT.

[15] A. Tiwana, "Chapter 1 - The Rise of Platform Ecosystems," in A. B. T. P. E. Tiwana, Ed., Boston: Morgan Kaufmann, 2014, pp. 3–21, ISBN: 978-0-12-408066-9. DOI: https://doi.org/10.1016/B978-0-12-408066-9.00001-1. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780124080669000011.

[16] P. Johannesson and E. Perjons, *An Introduction to Design Science*. Cham: Springer International Publishing, 2014, ISBN: 978-3-319-10631-1. DOI: 10.1007/978-3-319-10632-8. [Online]. Available: http://link.springer.com/10.1007/978-3-319-10632-8.

[17] Hevner, March, Park, and Ram, "Design Science in Information Systems Research," *MIS Quarterly*, vol. 28, no. 1, pp. 75–105, 2004, ISSN: 02767783. DOI: 10.2307/25148625. [Online]. Available: https://www.jstor.org/stable/10.2307/25148625.

[18] A. Hevner, "A Three Cycle View of Design Science Research," *Scandinavian Journal of Information Systems*, vol. 19, pp. 87–92, Jan. 2007.

[19] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A Design Science Research Methodology for Information Systems Research," *Journal of Management Information Systems*, vol. 24, no. 3, pp. 45–77, Dec. 2007, ISSN: 0742-1222. DOI: 10.2753/MIS0742-1222240302. [Online]. Available: https://www.tandfonline.com/doi/full/10.2753/MIS0742-1222240302.

[20] A. Tiwana, *Platform Ecosystems: Aligning Architecture, Governance, and Strategy*. Elsevier, 2014, pp. 1–302, ISBN: 9780124080669. DOI: 10.1016/C2012-0-06625-2. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/C20120066252.

[21] A. Zutshi and A. Grilo, "The Emergence of Digital Platforms: A Conceptual Platform Architecture and impact on Industrial Engineering," *Computers and Industrial Engineering*, vol. 136, pp. 546–555, Oct. 2019, ISSN: 03608352. DOI: 10.1016/j.cie.2019.07.027. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0360835219304188.

[22] A. Tiwana, "Chapter 5 - Platform Architecture," in *Platform Ecosystems*, A. Tiwana, Ed., Boston: Morgan Kaufmann, 2014, pp. 73–116, ISBN: 978-0-12-408066-9. DOI: https://doi.org/10.1016/B978-0-12-408066-9.00005-9. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780124080669000059.

[23] C. Y. Baldwin and C. J. Woodard, "The Architecture of Platforms: A Unified View," *SSRN Electronic Journal*, 2008, ISSN: 1556-5068. DOI: 10.2139/ssrn.1265155. [Online]. Available: http://www.ssrn.com/abstract=1265155.

[24] C. Y. Baldwin and K. B. Clark, *Design Rules: The Power of Modularity*, en. The MIT Press, 2000, ISBN: 978-0-262-26764-9. DOI: 10.7551/mitpress/2366.001.0001. [Online]. Available: https://direct.mit.edu/books/book/1856/design-rulesthe-power-of-modularity (visited on 01/10/2022).

[25] D. Parnas, P. Clements, and D. Weiss, "The Modular Structure of Complex Systems," *IEEE Transactions on Software Engineering*, vol. SE-11, no. 3, pp. 259–266, Mar. 1985, ISSN: 0098-5589. DOI: 10.1109/TSE.1985.232209. [Online]. Available: http://ieeexplore.ieee.org/document/1702002/.

[26] M. Stein and A. Khare, "CALCULATING THE CARBON FOOTPRINT OF A CHEMICAL PLANT: A CASE STUDY OF AKZONOBEL," *Journal of Environmental Assessment Policy and Management*, vol. 11, no. 03, pp. 291–310, Sep. 2009, ISSN: 1464-3332. DOI: 10.1142/S1464333209003373. [Online]. Available: https://www.worldscientific.com/doi/abs/10.1142/S1464333209003373.

[27] EN ISO 14064-2:2012, "Greenhouse gases - Part 2: Specification with guidance at the project level for quantification, monitoring and reporting of greenhouse gas emission reductions or removal enhancements (ISO 14064-2:2006)," The British Standards Institution, Tech. Rep., 2012.

[28] M. Brander, *Greenhouse Gases , CO2 , CO2e , and Carbon : What Do All These Terms Mean?* 2012. [Online]. Available: https://ecometrica.com/assets/GHGs-CO2-CO2e-and-Carbon-What-Do-These-Mean-v2.1.pdf (visited on 11/25/2021).

[29] IPCC, *IPCC AR6 WGI Chapter 07 Supplementary Material*, 2021. [Online]. Available: https://report.ipcc.ch/ar6wg1/pdf/IPCC_AR6_WGI_Chapter_07_Supplementary_Material.pdf.

[30] D. Zowghi and Z. Jin, Eds., *Requirements Engineering*, ser. Communications in Computer and Information Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, vol. 432, ISBN: 978-3-662-43609-7. DOI: 10.1007/978-3-662-43610-3. [Online]. Available: http://link.springer.com/10.1007/978-3-662-43610-3.

[31] R. Elmasri and S. B. Navathe, *Fundamentals of Database Systems*, 7th ed. Pearson, 2015.

[32] G. L. Richardson, B. M. Jackson, and G. W. Dickson, "A Principles-Based Enterprise Architecture: Lessons from Texaco and Star Enterprise," *MIS Quarterly*, vol. 14, no. 4, p. 385, 1990, ISSN: 02767783. DOI: 10.2307/249787. [Online]. Available: https://www.jstor.org/stable/249787?origin=crossref.

[33] U. Sekaran and R. Bougie, *Research methods for business*. wiley, 2016.

[34] M. Lankhorst, H. Proper, and H. Jonkers, "The Anatomy of the ArchiMate Language," *International Journal of Information System Modeling and Design*, vol. 1, no. 1, pp. 1–32, 2010, ISSN: 1947-8186. DOI: 10.4018/jismd.2010092301. [Online]. Available: http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/jismd.2010092301.

[35] EN ISO 14064-1:2012, "ISO 14064-1 Greenhouse gases - Specification with guidance at the organizational level for quantification and reporting of greenhouse gas emissions and removals," The British Standards Institution, Tech. Rep., 2012, p. 32.

[36] A. R. Honarvar and A. Sami, "Extracting Usage Patterns from Power Usage Data of Homes' Appliances in Smart Home using Big Data Platform," *International Journal of Information Technology and Web Engineering*, vol. 11, no. 2, pp. 39–50, 2016, ISSN: 1554-1045. DOI: 10.4018/IJITWE.2016040103. [Online]. Available: http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/IJITWE.2016040103.

[37] Y.-C. Chen, W.-C. Peng, J.-L. Huang, and W.-C. Lee, "Significant Correlation Pattern Mining in Smart Homes," *ACM Transactions on Intelligent Systems and Technology*, vol. 6, no. 3, pp. 1–23, 2015, ISSN: 2157-6904. DOI: 10.1145/2700484. [Online]. Available: https://dl.acm.org/doi/10.1145/2700484.

[38] SBTi, *What are Science Based Targets?* 2021. [Online]. Available: https://sciencebasedtargets.org/how-it-works.

[39] CGI, "Corporate Social Responsibility Report 2021: Our commitment to a more inclusive and sustainable world," Tech. Rep., 2022. [Online]. Available: https://www.cgi.com/sites/default/files/2022-02/cgi-2021-csr-report.pdf.

[40] R. Sandhu and P. Samarati, "Access control: principle and practice," *IEEE Communications Magazine*, vol. 32, no. 9, pp. 40–48, 1994, ISSN: 0163-6804. DOI: 10.1109/35.312842. [Online]. Available: http://ieeexplore.ieee.org/document/312842/.

[41] J. Wu, L. Ping, X. Ge, Y. Wang, and J. Fu, "Cloud Storage as the Infrastructure of Cloud Computing," in *2010 International Conference on Intelligent Computing and Cognitive Informatics*, IEEE, 2010, pp. 380–383, ISBN: 978-1-4244-6640-5. DOI: 10.1109/ICICCI.2010.119. [Online]. Available: http://ieeexplore.ieee.org/document/5565955/.

[42] K. Liu and L.-j. Dong, "Research on Cloud Data Storage Technology and Its Architecture Implementation," *Procedia Engineering*, vol. 29, pp. 133–137, 2012, ISSN: 18777058. DOI: 10.1016/j.proeng.2011.12.682. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1877705811065192.

[43] R. Ikeda and J. Widom, "Data lineage: A survey," Stanford InfoLab, Tech. Rep., 2009.

[44] F. Pezoa, J. L. Reutter, F. Suarez, M. Ugarte, and D. Vrgoč, "Foundations of JSON Schema," in *Proceedings of the 25th International Conference on World Wide Web*, Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2016, pp. 263–273, ISBN: 9781450341431. DOI: 10.1145/2872427.2883029. [Online]. Available: https://dl.acm.org/doi/10.1145/2872427.2883029.

[45] W3S, *JSON - Introduction*, 2022. [Online]. Available: https://www.w3schools.com/js/js_json_intro.asp (visited on 02/22/2022).

[46] W3S, *JSON vs XML*, 2022. [Online]. Available: https://www.w3schools.com/js/js_json_xml.asp (visited on 02/22/2022).

[47] T. DeMarco, "Structure analysis and system specification," in *Pioneers and Their Contributions to Software Engineering*, Springer, 1979, pp. 255–288.

[48] C. Gane, "Structured systems analysis: Tools and techniques," *Englewood Cliffs*, 1978.

[49] E. Yourdon, *Modern structured analysis*. Yourdon press, 1989.

[50] P. D. Bruza and T. Van der Weide, *The semantics of data flow diagrams*. Citeseer, 1989.

[51] Y. Tao and C. Kung, "Formal definition and verification of data flow diagrams," *Journal of Systems and Software*, vol. 16, no. 1, pp. 29–36, Sep. 1991, ISSN: 01641212. DOI: 10.1016/0164-1212(91)90029-6. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/0164121291900296.

[52] W. S. Davis and D. C. Yen, *The information system consultant's handbook: systems analysis and design*. Boca Raton, Fla.: CRC Press, 1999, OCLC: 45787705, ISBN: 978-1-4200-4910-7 978-1-315-21974-5 978-1-351-82694-5 978-1-351-83563-3. [Online]. Available: http://www.crcnetbase.com/isbn/9781420049107 (visited on 12/16/2021).

[53] P. G. Larsen, N. Plat, and H. Toetenel, "A formal semantics of data flow diagrams," *Formal Aspects of Computing*, vol. 6, no. 6, pp. 586–606, Dec. 1994. DOI: 10.1007/bf03259387. [Online]. Available: https://doi.org/10.1007/bf03259387.

[54] The Open Group, *Using The TOGAF 9.1 Framework With The ArchiMate 3.0 Modeling Language*, 2017. [Online]. Available: https://www.vanharen.net/blog/using-the-togaf-9-1-framework-with-the-archimate-3-0-modeling-language/.

[55] The Open Group, *Physical Elements*, 2019. [Online]. Available: https://pubs.opengroup.org/architecture/archimate3-doc/chap11.html#_Toc10045429.

[56] N. Laskowski, *Data lake governance: A big data do or die*, 2016. [Online]. Available: https://www.techtarget.com/searchcio/feature/Data-lake-governance-A-big-data-do-or-die (visited on 03/11/2022).

[57] N. Miloslavskaya and A. Tolstoy, "Big Data, Fast Data and Data Lake Concepts," *Procedia Computer Science*, vol. 88, pp. 300–305, 2016, ISSN: 18770509. DOI: 10.1016/j.procs.2016.07.439. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1877050916316957.

[58] S. García, J. Luengo, and F. Herrera, *Data Preprocessing in Data Mining*. Jan. 2015, vol. 72, ISBN: 978-3-319-10246-7. DOI: 10.1007/978-3-319-10247-4.

[59] "Cron," in *Expert Shell Scripting*, Berkeley, CA: Apress, 2009, pp. 81–85, ISBN: 978-1-4302-1842-5. DOI: 10.1007/978-1-4302-1842-5_12. [Online]. Available: https://doi.org/10.1007/978-1-4302-1842-5_12.

[60] P. Vassiliadis, A. Simitsis, and S. Skiadopoulos, "Conceptual modeling for ETL processes," in *Proceedings of the 5th ACM international workshop on Data Warehousing and OLAP - DOLAP '02*, New York, New York, USA: ACM Press, 2002, pp. 14–21, ISBN: 1581135904. DOI: 10.1145/583890.583893. [Online]. Available: http://portal.acm.org/citation.cfm?doid=583890.583893.

[61] The Open Group, *Business Layer*, 2019. [Online]. Available: https://pubs.opengroup.org/architecture/archimate3-doc/chap08.html#_Toc10045365.

[62] CDP, *Who We Are*, 2022. [Online]. Available: https://www.cdp.net/en/info/about-us.

[63] The Open Group, *Application Layer*, 2019. [Online]. Available: https://pubs.opengroup.org/architecture/archimate3-doc/chap09.html#_Toc10045389.

[64] The Open Group, *Technology Layer*, 2019. [Online]. Available: https://pubs.opengroup.org/architecture/archimate3-doc/chap10.html#_Toc10045407.

[65] ISO/IEC, *ISO 25000 Software and Data Quality*, 2022. [Online]. Available: https://iso25000.com/index.php/en/iso-25000-standards/iso-25010?start=6.

[66] V. Khatri and C. V. Brown, "Designing data governance," *Communications of the ACM*, vol. 53, no. 1, pp. 148–152, 2010, ISSN: 0001-0782. DOI: 10.1145/1629175.1629210. [Online]. Available: https://dl.acm.org/doi/10.1145/1629175.1629210.

[67] TU Delft, *Kick off form MSc thesis project Cosem / EPA / MoT*, 2022. [Online]. Available: https://d2k0ddhflgrk1i.cloudfront.net/Studentenportal/Faculteitspecifiek/TBM/Onderwijs/Master/GraduationPortal/1.MScThesisProject-Kick-offFormv.11.2021.pdf.

[68] The Open Group, *Relationships*, 2019. [Online]. Available: https://pubs.opengroup.org/architecture/archimate3-doc/chap05.html#_Toc10045310.

Acronyms

ADM Architecture Development Method. vi, 28, 29

AI Artificial Intelligence. 3

API Application Programming Interface. iii, 22, 23, 24, 25, 32, 33, 34, 35, 36, 41, 42, 43, 44, 45, 46, 47, 49, 55, 56

CAS Carbon Accounting Solution. ii, iii, iv, v, vi, vii, 2, 3, 4, 6, 8, 11, 14, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 29, 34, 35, 36, 37, 38, 39, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 73, 74

CGI Conseillers en Gestion et Informatique. ii, iii, 2, 3, 4, 6, 34, 36, 37, 38, 39, 40, 43, 45, 48, 49, 52, 54, 55, 56, 57, 73, 74

CO_2 Carbon Dioxide. ii, iii, 1, 2, 34, 35, 36, 55, 57

CO_2e Carbon dioxide equivalent. iv, 11, 12, 13, 14, 15, 16, 20, 35, 36, 38, 39, 40, 45, 46, 49, 56, 57

CSR corporate social responsibility. ii, iii, 17, 19, 25, 32, 33, 34, 35, 36, 40, 41, 43, 44, 45, 55

DBA Database Administrator. 21, 35

DBMS Database Management System. iii, 21, 29, 43, 45, 55

DFD data flow diagram. vi, 25, 26, 27, 50, 51, 74

DM data mining. 31

DSA Data Staging Area. 35

DSR Design Science Research. 4

DSRM Design Science Research Methodology. ii, vi, vii, 5, 8, 25, 36, 37, 47, 48, 57

EA Enterprise Architecture. ii, iv, v, vi, vii, 4, 6, 8, 11, 18, 21, 37, 38, 44, 45, 46, 48, 52, 54, 55, 56, 57, 67, 68, 73

EAM Enterprise Architecture Model. ii, iii, iv, vii, 1, 2, 3, 4, 6, 8, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 34, 36, 37, 44, 47, 48, 49, 50, 52, 54, 55, 56

ESG environmental, social and governance. 1, 20, 21, 39, 40

ETL extraction-transformation-loading. vi, 34, 35, 42, 45, 55

EU European Union. ii, 1

FRQ functional requirements. 17

FS Feature Selection. 31

GHG Greenhouse Gas. ii, 1, 2, 12, 19, 20, 21, 27, 39, 55

GHGP Greenhouse Gas Protocol. 12, 14, 19

GWP Global Warming Potential. 12, 14, 15, 20, 27, 32, 34, 38, 42, 49, 55

HVAC Heating, Ventilation and Airconditioning. 42, 56

IaaS Infrastructure as a Service. 21

ICT Information and Communications Technology. ii, vii, 1, 2, 4, 6, 8, 25, 34, 37, 44, 47, 48, 50, 52, 55, 56, 57, 73, 74

IEA International Energy Agency. 2

IoT Internet of Things. 6, 20, 24, 29, 30, 34, 35, 36, 38, 42, 44, 45, 53, 55, 56

IPCC Intergovernmental Panel on Climate Change. 20, 38

IS Information Systems. 4, 16, 21, 23, 25

IT Information Technology. 4, 28, 30

JSON JavaScript Object Notation. iii, vi, 22, 23, 24, 25, 33, 34, 35, 44, 45, 49

kWh kilowatt-hour. 6, 46, 57

ML Machine Learning. 3

MoT Management of Technology. 48, 57, 58

NFRQ non-functional requirements. 17

OSDU Open Source Data Universe. ii, 1, 2, 3, 4, 10, 11, 21, 35, 45, 47, 48, 56, 74

PaaS Platform as a Service. 3, 21

RFD Research Flow Diagram. 1

SaaS Software as a Service. ii, 1, 2, 3, 4, 21, 54, 56, 57

SBT Science-Based Target. iv, vi, 21, 32, 35, 36, 38, 39, 40, 42, 45, 55

SBTi Science Based Target Initiative. 21, 35, 39, 45

StaaS Storage as a Service. 21

TOGAF the Open Group's Architecture Framework. ii, 28, 29

UI User Interface. ii, iii, 41, 42, 43, 44, 45, 46, 55

UML Unified Modelling Language. 28, 56

UNFCCC United Nations Framework Convention on Climate Change. 1

VPN Virtual Private Network. 43

XML eXtensible Markup Language. 23

Glossary

carbon accounting The integration of aspects of climate change mitigation into accounting. 1, 6

Appendices

A Alternative terms for carbon accounting

Table A.1: Alternative terms used for carbon accounting. [4].

Scale	Alternatively used terms
National scale	Account for CO_2 emissions, carbon emissions accounting, (GHG) emission accounting, CO_2 (emission) accounting, carbon storage accounting, accounting for biospheric carbon exchange, biospheric carbon accounting, carbon stock accounting, Canada's national forest carbon . . . accounting, C accounting, GHG inventory carbon accounting, (sub-) national emission accounting, (national) carbon footprint accounting, accounting for climate change, full carbon (and GHG) accounting, accounting for greenhouse gas (GHG) emissions, consumption-based emission accounting, accounting for 'emissions from consumption', greenhouse gas (GHG) accounting, accounting for virtual carbon, carbon debt accounting, climate change accounting, accounting for carbon and other GHG emissions, accounting for sulfur dioxide emissions, national GHG accounting, physical carbon accounting, political carbon accounting, market-enabling carbon accounting, financial carbon accounting, social/environmental carbon accounting, national and project level carbon accounting, global-scale carbon accounting, national scale carbon accounting, accounting for carbon stock changes, carbon and land use accounting, environmental accounting for GHG, footprint accounting
Project scale	Project-based greenhouse gas accounting, accounting for biospheric carbon stock changes, carbon accounting system, accounting for carbon sequestration, (average) carbon stock accounting, trade-based carbon sequestration accounting, carbon trade accounting, project level (carbon) accounting, forest carbon accounting, greenhouse gas (GHG) mitigation accounting, green accounting, physical carbon accounting, emissions accounting, GHG inventory accounting, accounting for carbon sequestered, carbon flow accounting, accounting of non-CO_2 gases, (full) accounting of forest carbon stock changes, accounting of forest carbon sinks and sources, carbon tax accounting, carbon credit accounting, GHG project accounting, Kyoto forest carbon accounting, accounting emissions reductions, account for GHG reductions, carbon (C) accounting, forest management C offset accounting, climate accounting, (carbon) offset accounting, greenhouse gas offset accounting, carbon sequestration accounting, national forest carbon accounting, accounting of carbon stocks and fluxes, greenhouse gas (GHG) accounting, accounting for carbon sinks, accounting for emissions, full net–net carbon accounting
Organizational scale	(Carbon) emissions accounting, accounting for emission rights, GHG project accounting, (corporate) greenhouse gas (GHG) emissions accounting, national and corporate emissions accounting, carbon sinks accounting, accounting for greenhouse effect, CO_2 (emissions) accounting, carbon emission and sequestration accounting (CES accounting), carbon financial (statement) accounting, accounting for emission allowances, carbon cost accounting, accounting for carbon, carbon data accounting and reporting, accounting for carbon emission permits, accounting for the European Union's new emissions trading scheme, pollution offset accounting, 'carbon footprint' accounting, carbon credit accounting, (corporate) greenhouse gas (GHG) accounting, supply-chain GHG accounting, accounting for emission reductions, carbon business accounting, life cycle carbon cost accounting, carbon strategic management accounting, emissions rights accounting, whole life carbon accounting, corporate level (carbon) accounting, climate change accounting, CO_2 accounting, carbon storage accounting methods, accounting for greenhouse gas emissions, 'engineering accounting', greenhouse gas (GHG) foot-printing, carbon capture and storage, accounting for carbon regulatory obligations, accounting for held emission credits and offsets, accounting and assuring corporate GHGs, accounting for carbon commodities products, accounting of that permit, accounting of GHGs, accounting for climate change, carbon-related accounting, non-financial accounting and reporting framework with regards to carbon, accounting for CO_2 flows, carbon management accounting, monetary carbon accounting, physical carbon accounting, carbon capital expenditure accounting, carbon flow accounting, carbon capital impact accounting, accounting for externalities such as carbon pollution, accounting and reporting of carbon emissions
Product scale	(GHG) emissions accounting, carbon flow accounting, CO_2 accounting, forest biomass carbon pools accounting, accounting for greenhouse gas emissions, climate accounting, accounting for carbon footprints, greenhouse gas accounting, carbon accounting life cycle assessment, greenhouse gas and carbon accounting

B List of Interviewees for Requirements Elicitation

Table B.1: Interviewees for requirements elicitation.

Code	Country	Experience Level	Job Title	Role / Knowledge
IV1	NL	Senior	Consultant Data Engineering	Data Science
IV2	NL	Senior	Consultant	Data engineering
IV3	NL	Senior	VP Consulting Expert	Sustainability Advisor
IV4	NL	Senior	CSR Manager	Sustainability Reporter
IV5	SW	Senior	Consultant	Sustainability
IV6	NL	Senior	Director Consulting Services	Delivery / Data Universe
IV7	FI	Senior	Lead Consultant Regulatory Compliance and Risk	ESG regulation
IV8	NL	Senior	VP Consulting Expert	Advanced Analytics
IV9	DE	Senior	VP Consulting Internal	Computer Science
IV10	NL	Senior	Consultant	Software Architect; electrical grids, IoT
IV11	NL	Medior	Consultant	Game Design, Data visualisation
IV12	NL	Senior	Consultant	IoT
IV13	DE	Senior	VP Consulting Expert	CSR

C Interview Protocol For Requirements Elicitation - Architecture

Date:

Opening Statement

You are being invited to participate in a research study for a Master Thesis titled [Organizational decarbonization: a data architecture model for carbon accounting, monitoring and quantification]. The Master Thesis is conducted by Prashant Badaltjawdharie from Delft University of Technology.

The purpose of the research study is to develop a data architecture model, which captures how organizations can use open source data platform implementations to foster carbon accounting, monitoring and quantification in emission scope two. Scope two represents one of the largest sources of GHG emissions globally: the generation of electricity and heat now accounts for at least a third of global GHG emissions (source: https://ghgprotocol.org/sites/default/files/standards/Scope%202%20Guidance_Final_Sept26.pdf). On both an academic as practical level there are many unknowns with regards to carbon accounting, monitoring, and quantification facilitated by open source EA-model platform implementations. I aim to identify requirements on several but not limited to the following aspects: (a) data sources and types (b) data configurations (c) carbon accounting and reporting standards (e.g. Greenhouse Gas Protocol and EN ISO 14064-2:2012) (d) automatic data conversion to carbon equivalents (CO_2e) (e) time intervals for carbon footprint display (e.g. monthly or (near) real-time).

Since CGI aims to become a net-zero carbon emissions organization by 2030, and aims to offer ICT-services to clients with regards to decarbonization, I would like to interview several CGI members. Your input is important for eliciting requirements for the EA-model. Your input may furthermore introduce new aspects for data requirements. Eventually, the gained insights through this research study can contribute to CGI's business performance and innovativeness.

I would appreciate if you take the time to participate in the interview. The interview consists of 12 questions and is divided in two parts. Part one aims to identify what CGI's current practices are for carbon accounting. Part two focuses on eliciting requirements for the EA-model. The interview should take approximately 30 - 45 minutes of your time. Your responses are voluntary and will be confidential. All responses will be compiled together and analyzed as a group. I believe there are no risks associated with this research study; however, as with any online related activity the risk of a breach is always possible.

You will receive a copy of the master thesis after completion.

Interview Questions Requirements Elicitation	
Part one: Analysis of Existing Systems	
A1.	What is your role in the development of a tool for carbon accounting?
Part two: Requirements Elicitation EA-model	
A2.	For which stakeholders (both internal and external) is carbon accounting data relevant?
A3.	To which standards for carbon quantification do you want the information system to comply?
A3.1	To which standards for carbon monitoring do you want the information system to comply?
A3.2	To which standards for carbon reporting do you want the information system to comply?
A4.	What functionalities should the data platform provide at the bare minimum?
Part two: Requirements Elicitation EA-model - Data Requirements	
A5.	In emission scope two, which data sources and types are most important to collect for the data platform?
A6.	Who should be able to access the data?
A7.	How should data be stored?
A8.	What is a reasonable time interval for the data platform to display carbon data?
A9.	Do you foresee problems with sharing data between the electricity supplier and CGI?
A9.1.	How can daily data sharing between electricity supplier and CGI be ensured? (i.e. hold the electricity supplier accountable?)
A9.2.	How can trustworthiness of data be ensured with regards to data exchange between electricity supplier and CGI?
A9.3.	In which format should data be shared with CGI?
A9.4.	How can traceability of GHG emissions data be ensured in the data platform?
Interview Wrap-up	
10.	Do you have any final comments?
11.	Do you recommend other CGI members whom I should interview?

Ending Statement

I sincerely thank you for participating in the interview. The results of the interview will be analyzed to formulate requirements for the EA-model. The results will be shared with the interviewees. It is possible that post-interview it is necessary to ask follow-up questions. Please indicate if you consent with answering possible follow-up questions post-interview.

D ArchiMate Elements and Description

D.1 Business Layer Elements

Element	Description	Notation
Business actor	Represents a business entity that is capable of performing behavior.	Business actor
Business role	Represents the responsibility for performing specific behavior, to which an actor can be assigned, or the part an actor plays in a particular action or event.	Business role
Business collaboration	Represents an aggregate of two or more business internal active structure elements that work together to perform collective behavior.	Business collaboration
Business interface	Represents a point of access where a business service is made available to the environment.	Business interface
Business process	Represents a sequence of business behaviors that achieves a specific result such as a defined set of products or business services.	Business process
Business function	Represents a collection of business behavior based on a chosen set of criteria (typically required business resources and/or competencies), closely aligned to an organization, but not necessarily explicitly governed by the organization.	Business function
Business interaction	Represents a unit of collective business behavior performed by (a collaboration of) two or more business actors, business roles, or business collaborations.	Business interaction
Business event	Represents an organizational state change.	Business event
Business service	Represents explicitly defined behavior that a business role, business actor, or business collaboration exposes to its environment.	Business service
Business object	Represents a concept used within a particular business domain.	Business object
Contract	Represents a formal or informal specification of an agreement between a provider and a consumer that specifies the rights and obligations associated with a product and establishes functional and non-functional parameters for interaction.	Contract
Representation	Represents a perceptible form of the information carried by a business object.	Representation
Product	Represents a coherent collection of services and/or passive structure elements, accompanied by a contract/set of agreements, which is offered as a whole to (internal or external) customers.	Product

Fig. D.1: Business Layer Elements and Descriptions [61].

D.2 Application Layer Elements

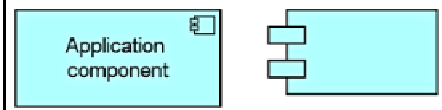
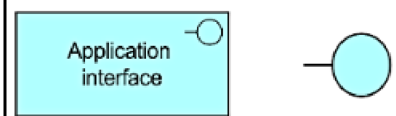
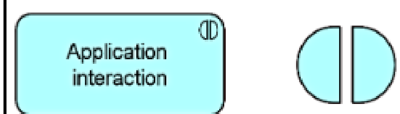
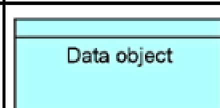
Element	Definition	Notation
Application component	Represents an encapsulation of application functionality aligned to implementation structure, which is modular and replaceable.	
Application collaboration	Represents an aggregate of two or more application internal active structure elements that work together to perform collective application behavior.	
Application interface	Represents a point of access where application services are made available to a user, another application component, or a node.	
Application function	Represents automated behavior that can be performed by an application component.	
Application interaction	Represents a unit of collective application behavior performed by (a collaboration of) two or more application components.	
Application process	Represents a sequence of application behaviors that achieves a specific result.	
Application event	Represents an application state change.	
Application service	Represents an explicitly defined exposed application behavior.	
Data object	Represents data structured for automated processing.	

Fig. D.2: Application Layer Elements and Descriptions [63].

D.3 Technology Layer Elements

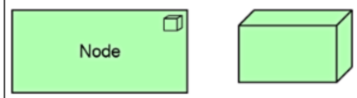
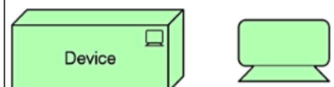
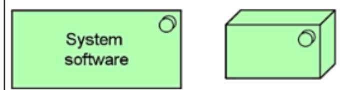
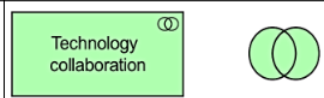
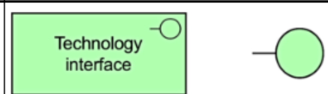
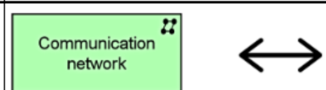
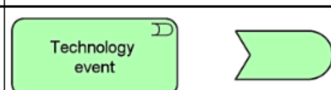
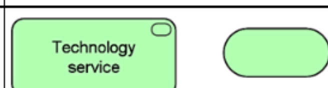
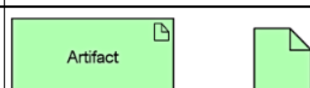
Element	Definition	Notation
Node	Represents a computational or physical resource that hosts, manipulates, or interacts with other computational or physical resources.	
Device	Represents a physical IT resource upon which system software and artifacts may be stored or deployed for execution.	
System software	Represents software that provides or contributes to an environment for storing, executing, and using software or data deployed within it.	
Technology collaboration	Represents an aggregate of two or more technology internal active structure elements that work together to perform collective technology behavior.	
Technology interface	Represents a point of access where technology services offered by a node can be accessed.	
Path	Represents a link between two or more nodes, through which these nodes can exchange data, energy, or material.	
Communication network	Represents a set of structures that connects nodes for transmission, routing, and reception of data.	
Technology function	Represents a collection of technology behavior that can be performed by a node.	
Technology process	Represents a sequence of technology behaviors that achieves a specific result.	
Technology interaction	Represents a unit of collective technology behavior performed by (a collaboration of) two or more nodes.	
Technology event	Represents a technology state change.	
Technology service	Represents an explicitly defined exposed technology behavior.	
Artifact	Represents a piece of data that is used or produced in a software development process, or by deployment and operation of an IT system.	

Fig. D.3: Technology Layer Elements and Descriptions [64].

D.4 Relationships

Structural Relationships		Notation	Role Names
Composition	Represents that an element consists of one or more other concepts.		← composed of → composed in
Aggregation	Represents that an element combines one or more other concepts.		← aggregates → aggregated in
Assignment	Represents the allocation of responsibility, performance of behavior, storage, or execution.		← assigned to → has assigned
Realization	Represents that an entity plays a critical role in the creation, achievement, sustenance, or operation of a more abstract entity.		← realizes → realized by
Dependency Relationships		**Notation**	**Role Names**
Serving	Represents that an element provides its functionality to another element.		← serves → served by
Access	Represents the ability of behavior and active structure elements to observe or act upon passive structure elements.		← accesses → accessed by
Influence	Represents that an element affects the implementation or achievement of some motivation element.	+/-	← influences → influenced by
Association	Represents an unspecified relationship, or one that is not represented by another ArchiMate relationship.		associated with ← associated to → associated from
Dynamic Relationships		**Notation**	**Role Names**
Triggering	Represents a temporal or causal relationship between elements.		← triggers → triggered by
Flow	Represents transfer from one element to another.		← flows to → flows from
Other Relationships		**Notation**	**Role Names**
Specialization	Represents that an element is a particular kind of another element.		← specializes → specialized by
Relationship Connectors		**Notation**	**Role Names**
Junction	Used to connect relationships of the same type.	(And) Junction Or Junction	

Fig. D.4: Relationships and Descriptions [68].

E Evaluation Framework for Enterprise Architecture Modelling of CAS

E.1 List of ICT-Architects for Model Evaluation, Demonstration and Improvement

List of CGI's ICT-Architects involved in model evaluation, demonstration and improvement. The code EA_x stands for Enterprise Architect$_{1,\ 2,\ \ldots,\ n}$.

Table E.1: ICT-Architects involved in model evaluation, demonstration and improvement.

Code	**Country**	**Experience Level**	**Job Title**	**Role / Knowledge**
EA1	NL	Senior	Principal ICT-Architect	ICT-Architecture
EA2	NL	Senior	Director Consulting Expert	Agile, DevOps, Emerging Technologies
EA3	NL	Senior	Director Consulting Expert	ICT-Architecture
EA4	NL	Senior	Senior Consultant	ICT-Architecture
EA5	CA	Senior	Senior Consultant	ICT-Architecture

E.2 Evaluation Aspects of Enterprise Architecture Model Components

Table E.2: Evaluation aspects of Enterprise Architecture components

Enterprise Architecture Component	**Evaluation Aspects**
Requirements	Completeness, correctness, quality
	Coverage by models
	Demonstration by models
Modelling	Conform ArchiMate modelling language; relationships, correct elements
	Separation of ICT-architecture in correct modelling layers
	Business Process Modelling; correctness, coherence, readability
	Application Modelling; relationships with data, data aggregation, dashboarding
	Technology Modelling; does the physical infrastructure support the purpose?

E.3 Overview of Feedback on Enterprise Architecture Model

In Table E.3 the feedback received from the ICT-Architects is summarized. The feedback was given in 7 sessions. Per session, the feedback is coded as F_{na} with n being the session number ranging from $[1-7]$ and a being the feedback ranging from $[a, b, .., n]$.

Table E.3: Overview of Feedback on Enterprise Architecture Model

Session	**Code**	**Feedback**
1	F_{1a}	Split the components in the correct modelling layer, respecting ArchiMate's order of modelling (i.e. start with business modelling, then application modelling, then technology modelling)
	F_{1b}	ICT-solutions start with business processes, so model the business process first
	F_{1c}	Consider modelling data flow diagram (DFD)s in ArchiMate
	F_{1d}	Database update process can be deleted, it is 'obvious' that this needs to happen
	F_{1e}	Does your text support the models? Keep asking yourself this.
2	F_{2a}	Show how your models change (modifiability) when CGI decides to add electric cars/gas equipment to CAS
	F_{2b}	Show how your models change (adaptability) when CGI decides to offer CAS to a client
3	F_{3a}	Remove data stores from DFD, they don't add to logical data flows
	F_{3b}	emphasize that services, applications and functions are exactly that by typing it out
4	F_{4a}	The dashboard in the app. layer is sufficient? Do we just look at a dashboard?
5	F_{5a}	DFDs have no starting point, call it data source/data origin.
6	F_{6a}	The models are far more readable and demonstrate the requirements for a big part
7	F_{7a}	Modularity (which is a non-functional requirement) is also demonstrated by your models due to 'gelaagdheid' i.e. modelling in layers
	F_{7b}	Security and authorization subsystems (FRQ_7) can not be properly modelled in ArchiMate
	F_{7c}	Data lineage is demonstrated if you explain this is provided by OSDU's reference architecture.
	F_{7d}	modifiability and adaptability are clearly demonstrated; by using cloud servers adaptability is furthermore demonstrated, since cloud servers can run on different systems (Windows, Linux)
	F_{7e}	[EA1] states: "The models are sufficient for master level research work and they cover about 90% of requirements"