TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Gerd Kukemilk

DESIGN AND IMPLEMENTATION OF A CONFIGURATION MANAGEMENT DATABASE COMPATIBLE WITH ITIL AND E-ITS

Master's thesis

Supervisor: MSc Toomas Lepik
Co-Supervisor MBA Thomas Lepik

Tallinn 2024

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Gerd Kukemilk

ITIL-I JA E-ITS-IGA ÜHILDUVA KONFIGURATSIONIHALDUSE ANDMEBAASI DISAIN JA RAKENDAMINE

Magistritöö

Juhendaja:	MSc Toomas Lepik
Kaasjuhendaja	MBA Thomas Lepik

Tallinn 2024

Author’s declaration of originality

I hereby certify that I am the sole author of this thesis. I have referred to all the used materials, references to the literature, and the work of others. This thesis has not been presented for examination anywhere else.

Author: Gerd Kukemilk

12.05.2024

Abstract

Managing and securing IT assets in growing organisations is challenging, especially given compliance requirements with standards like the Estonian Information Systems Security Standard (E-ITS) and the Information Technology Infrastructure Library (ITIL). This thesis aims to narrow the gap between these frameworks by designing a Configuration Management Database (CMDB) model that integrates with both.

A literature review revealed no CMDB data models capable of mapping services to E-ITS and ITIL requirements. The proposed model fills this gap by automating asset management, facilitating risk and impact analysis, and improving compliance with E-ITS and ITIL. Implemented in a prominent Estonian government organisation, the model demonstrated its potential to streamline business and technical processes while simplifying auditing.

Future work will refine the data model for detailed reporting and impact analysis. Provided solution enhances accuracy, efficiency, and compliance, providing better visibility into the organisation's IT infrastructure.

Keywords: E-ITS, ITIL, CMDB, Asset Management, Risk Analysis

This thesis is written in English and is 8 pages long, including 9 chapters, 11 figures and 6 tables.

Annotatsioon

IT-varade haldamine ja turvamine kasvavates organisatsioonides on keeruline, eriti arvestades vastavusnõudeid sellistele standarditele nagu Eesti Infoturbe Standard (E-ITS) ja rahvusvaheliselt tunnustatud infotehnoloogia juhtimise parimad praktikad (ITIL). Käesoleva lõputöö eesmärk on ületada nende raamistike vahelised lõhed, kavandades Konfiguratsioonihalduse andmebaasi (CMDB) mudelit, mis integreerib neid.

Kirjanduse analüüsi tulemusel võis jõuda järeldusele, et ükski varem käsitletud CMDB mudel ei suuda teenuseid E-ITS ja ITIL nõuetega siduda. Pakutud andme mudel täidab selle lünga, automatiseerides varade halduse, hõlbustades riskide ja mõju analüüsi ning parandades vastavust E-ITS standardi ja ITIL raamistiku vahel. Arendatud andme mudel rakendati ühes tuntud Eesti riigiasutuses ning demonstreeris oma potentsiaali äriliste ja tehniliste protsesside sujuvamaks muutmisel, lihtsustades samas auditite läbiviimist.

E-ITS ja ITIL nõuetele vastavuse hõlbustamiseks sisaldab andme mudel meetodit teenuste kaardistamiseks varadega ning riskide ja mõju analüüsi läbiviimiseks. CMDB andme mudel võimaldab paremat nähtavust kogu IT-taristule, pakkudes seeläbi suuremat täpsust, tõhusust ja vastavust.

Märksõnad: E-ITS, ITIL, CMDB, Varahaldus, Riski analüüs

See magistritöö on kirjutatud inglise keeles ja on 88 lehekülge pikk, sisaldades 9 peatükki, 11 joonist ja 6 tabelit.

List of abbreviations and terms

DPI	Dots per inch
ITIL / ITILv4	Information Technology Infrastructure Library Version 4 – a framework of best practices for IT service management that helps organisations assess risks and implement procedures to log and respond to incidents.
ISO/IEC 27001	An international standard providing guidelines for establishing, implementing, maintaining, and improving an Information Security Management System (ISMS).
ISO/IEC 27002	A part of the ISO/IEC 27000 family, focusing on information system security management and offering comprehensive security controls and best practices.
E-ITS	Estonian Information Security Standard – a standard based on the German BSI IT-Grundschutz methodology, designed to be compatible with ISO/IEC 27001 and mandatory for Estonian government institutions and organisations.
CMDB	Configuration Management Database – a database that stores information about hardware and software assets (Configuration Items or CIs) and their relationships for managing IT assets and services.
PDCA	Plan, Do, Check, Act – a four-stage iterative principle for continuous improvement, often used in quality and information security management.
CI	Configuration Item – a component of an IT infrastructure, such as hardware, software, or documentation, that is managed and tracked within a Configuration Management Database (CMDB).
COBIT	Control Objectives for Information and Related Technologies – a framework for IT governance and management, offering guidance on aligning IT strategies with business objectives.
BPMN	Business Process Model and Notation – a modelling language and notation for visualising and documenting business processes.
TOGAF	The Open Group Architecture Framework – a framework that provides an approach for designing, planning, implementing, and governing an enterprise information technology architecture.
ITSM	IT Service Management (ITSM), focuses on aligning IT services with business requirements and improving overall service delivery.

ISMS	Information Security Management System (ISMS)
DSRM	The Design Science Research Methodology (DSRM).
DOD3046	Department of Defence’s military standard 3046.
OOP	Object Oriented Programming
SSH	Secure Shell is a protocol by which two computers are connected via a secure channel.
WMI	Windows Management Instrumentation (WMI) is a framework within Microsoft Windows that allows system management and monitoring across networks by providing information and control over system components.
OS	Operating System

Table of Contents

List of Figures

List of Tables

Introduction

In today's rapidly evolving technology landscape, managing and tracking changes to IT assets within an organisation has become increasingly complex. In addition, as organisations grow, so do their infrastructure and service catalogue, making it more challenging to maintain accurate asset management and to ensure the security of those assets.

The Estonian government has introduced the Estonian Information Security Standard (E-ITS), a mandatory standard for all Estonian government institutions and organisations, to help tackle this issue. The E-ITS is based on the German BSI IT-Grundschutz methodology and is designed to be compatible with ISO/IEC 27001 [1], [2].

A significant aspect of the E-ITS involves tracking, monitoring, and managing an organisation's IT assets to facilitate risk analysis, provide a general overview of the organisation's security posture and assist with keeping track of responsibilities [1]. A typical asset management approach involves using a Configuration Management Database (CMDB), which contains a complete catalogue of assets, including processes and services that the organisation owns or provides [3], [4].

E-ITS does not specify how or what tool to use for asset management or in what detail to manage the assets, as this is left for the implementer to decide. For this reason, the author of this thesis provides a possible solution that should be general enough to be reused by any organisation. A CMDB is a perfect tool in this situation, mainly because it has been a known concept for decades, although it has gained and lost popularity over the years [4], [5], [6].

1.1 Structure of the Thesis

This thesis is organised as follows:

- **Chapter 2: Research Methodology**

  This chapter introduces the research methodology used, specifically the Design Science Research Methodology (DSRM). It outlines the six stages of DSRM and provides an overview of how each stage is applied in the context of this thesis.

- **Chapter 3: Problem Statement**

  This chapter represents the first stage of the DSRM, "Problem Identification and Motivation." It analyses the challenges in integrating ITIL and E-ITS and introduces the research problem addressed in this thesis.

- **Chapter 4: Literature Review**

  This chapter corresponds to the second stage of the DSRM, "Objectives of a Solution." It explores existing research and frameworks related to ITIL, E-ITS, and CMDBs to investigate the current state of integrating Information Security Management System (ISMS) processes into service management. It identifies clear objectives for the unified CMDB data model.

- **Chapter 5: CMDB Data Model Design and Development**

  This chapter aligns with the third stage of the DSRM, "Design and Development." It presents the proposed CMDB data model's design principles, constraints, and architecture.

- **Chapter 6: Implementation Plan**

  This chapter corresponds to the fourth stage of the DSRM, "Demonstration." It outlines a step-by-step guide for implementing the proposed CMDB data model and demonstrates how it integrates ITIL and E-ITS management processes.

- **Chapter 7: Implementation and Evaluation**

  This chapter aligns with the fifth stage of the DSRM, "Demonstration." It describes using the artefact to solve the identified problems, detailing the practical application and initial testing.

- **Chapter 8: Observed Results at Organisation A**

  This chapter covers the sixth stage of the DSRM, "Evaluation." It assesses how effective and efficient the implemented CMDB data model is within the organisational context, reviewing the results and feedback.

- **Chapter 9: Summary**

  This chapter concludes the thesis, serving as the "Communication" stage of the DSRM. It summarises the research findings, discusses the implications of the proposed CMDB data model, and outlines future research opportunities.

2 Research Methodology

The thesis follows the DSRM (Design Science Research Methodology) as the research methodology for this thesis because it offers a structured and iterative approach to designing, developing, and evaluating innovative solutions in the information systems research [7].

The DSRM was followed to develop an integrated CMDB data model that aimed to facilitate combining both ITIL and E-ITS principles. The outcome is an actionable data model that could be used to build a CMDB. Development of this data model followed an iterative approach that first outlined a hypothesis, which was followed by designing an initial version of the data model, validating it in practice, and finally redesigning it as necessary, and repeating the process until a successful data model was achieved.

2.1 Overview of The Design Science Research Methodology

The DSRM is a structured framework that underpins the entire lifecycle of design-oriented research [7]. It is composed of six steps:

- problem identification and motivation,
- objectives of a solution,
- design and development,
- demonstration,
- evaluation, and
- communication

Each step guides the research from a concept to a working solution.

The initial section of this thesis elaborates on problem identification and motivation, articulating integration challenges between ITIL and E-ITS within CMDB data models. As part of this, the author conducted a literature review that underpins the theoretical foundation for the thesis.

Next, the objectives of a solution are determined, laying out clear terms of functionality and impact of the possible solution.

The third section is about design and development, consisting of details about crafting the CMDB data model, incorporating features from ITIL and E-ITS. Following this, the demonstration step describes how the model operates in real-world scenarios, ensuring it meets the needs identified at the outset.

The evaluation section measures the performance of the provided solution against the objectives. Finally, a communication section will encapsulate the research findings, detailing the problem's significance, the developed data model's utility, and the rigour behind its design and effectiveness.

This six-step process ensures a thorough and academically sound approach to designing a solution to a real-world problem, providing a robust foundation for this thesis.

2.2 The DSRM Process: A Step-by-Step Approach

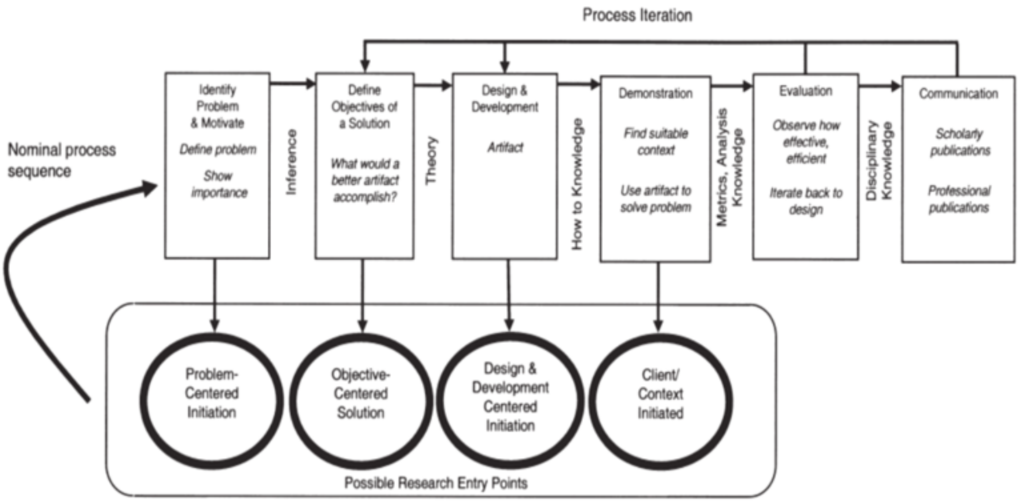
Process Iteration
Nominal process sequence
Identify Problem & Motivate
Define problem
Show importance
Inference
Define Objectives of a Solution
What would a better artifact accomplish?
Theory
Design & Development
Artifact
How to Knowledge
Demonstration
Find suitable context
Use artifact to solve problem
Metrics, Analysis Knowledge
Evaluation
Observe how effective, efficient
Iterate back to design
Disciplinary Knowledge
Communication
Scholarly publications
Professional publications
Problem-Centered Initiation
Objective-Centered Solution
Design & Development Centered Initiation
Client/ Context Initiated
Possible Research Entry Points

Figure 1: DSRM Process Iteration [7]

The Design Science Research Methodology (DSRM) process in this thesis is undertaken as follows:

- **Problem Identification and Motivation:** The research begins by pinpointing the specific challenges in integrating ITIL with E-ITS. This step involves understanding the context and why an integrated solution is necessary. This aligns with the "Problem-Centered Initiation" entry point shown (Figure 1).

- **Objectives of a Solution:** With the problem identified, the research sets definitive goals for the CMDB data model, outlining the desired functionality and its impact on IT service management and information security. This step correlates with the "Define Objectives of a Solution" stage (Figure 1).

- **Design and Development:** In this phase, the initial version of the CMDB data model is crafted, integrating the key features of ITIL and E-ITS to meet the identified needs. The design and development of the CMDB data model align with the "Design and Development-Centered Initiation" phase depicted (Figure 1).

- **Demonstration:** The model is tested in real-world scenarios to show that it works as intended and effectively solves the identified problems. This corresponds to the "Demonstration" step in (Figure 1).

- **Evaluation:** The CMDB data model's performance is measured against the objectives using qualitative feedback and quantitative data, ensuring that the solution provides tangible benefits. This step is shown as "Evaluation" (Figure 1).

- **Communication:** The final step involves sharing the research outcomes, highlighting the importance of the problem, the benefits of the artefact, and the thorough process behind its creation and validation. This is represented by the "Communication" step (Figure 1).

2.3 Iterative Development in DSRM

Iterative development is a core component of the DSRM process applied in this thesis. Instead of designing the perfect solution in one go, the research followed a cyclical process of continuous improvement (Figure 2):

1. **Hypothesis and Initial Design:** The process began by developing a hypothesis about how the CMDB data model should function and creating an initial design based on this.
2. **Validation and Feedback:** The initial design was implemented practically, and feedback was gathered on its performance and usability.
3. **Refinement:** The feedback was used to identify areas for improvement and refine the design. This involved changing the functionality or structure of the CMDB data model.
4. **Re-implementation:** The refined design was rolled out, and its performance was re-validated, ensuring that the modifications had improved the model.
5. **Repeat the Cycle:** This cycle of design, feedback, refinement, and re-implementation was continued until the CMDB data model met all the set objectives and performed effectively in its intended environment.

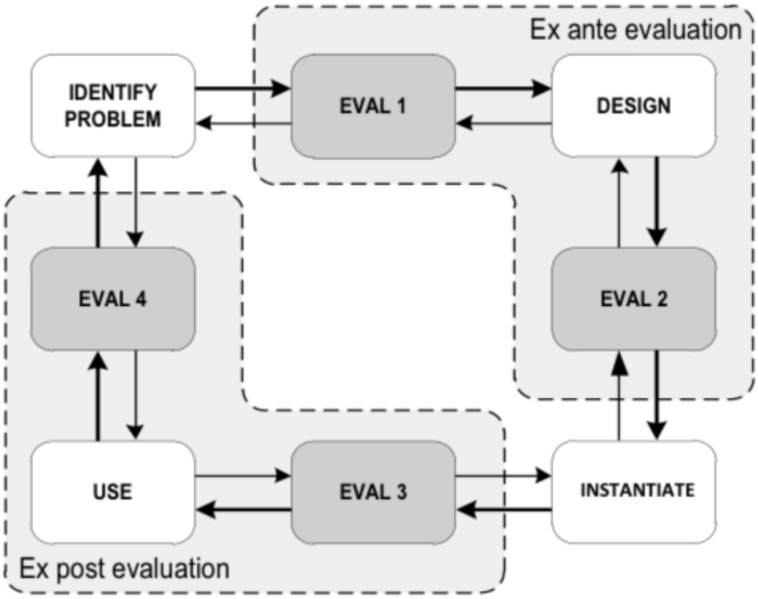

Figure 2: Evaluation Activities within the DSR Process
(Adapted from Sonnenberg and vom Brocke (2012)) [7]

This iterative approach allowed flexibility and responsiveness to new insights and feedback, ensuring that the final CMDB data model is practical and effective.

3 Problem Statement

Existing research on Configuration Management Databases (CMDB) reveals a lack in an unified approach toward integration with different frameworks, specifically with E-ITS while ensuring comprehensive business process-to-asset mapping and accurate protection controls assignment. This thesis aims to fill this gap by proposing a novel CMDB data model design that combines ITIL and E-ITS frameworks for better compliance, security, and asset management.

As the first stage of the Design Science Research Methodology (DSRM), which is problem identification and motivation, this thesis investigates E-ITS and ITIL to identify common ground and propose a unified CMDB data model that ensures compatibility between these practices. This is achieved through a literature review process.

3.1 Research Questions:

- What organisational opportunities and potential efficiency gains are possible from managing a central CMDB database for ISMS and ITSM?
- How does using a centralised CMDB database affect the data quality of the CMDB itself and the efficiency of ITSM and ISMS management processes?

3.2 Scope and Objectives

This master's thesis aims to design and implement a unified Configuration Management Database (CMDB) data model to automate asset management and improve compliance with the E-ITS standard and the ITIL framework.

The research primarily concentrates on developing a theoretical model that helps bridge the gap between ITIL and E-ITS. It allows two separate processes, IT Service Management (ITSM) and Information Security Management Systems (ISMS), to utilise a single dataset. The author is aware of specific tools that are already under development, such as Cybsis [8] and Kordon [9], that are designed to help organisations apply the E-ITS standard and include an aspect of asset management. Still, these tools are not suitable as configuration management solutions because of (still) lack of the possibility for integration with different data sources. Moreover, this work does not focus on comparing commercial nor open-source CMDB software applications and their features, as this is outside the scope. Most commercial software allows for modifications of data models or requires clients to build their own data models. Therefore, this thesis aims to provide a guideline on how to build a unified CMDB data model rather than recommend specific software solutions.

The specific objectives of this thesis are:

1. **Design** a unified CMDB data model that encompasses the requirements of E-ITS and ITIL.
2. **Evaluate** the impact of the unified CMDB data model on asset management practices.
3. **Investigate** the potential of the proposed data model for ensuring accurate service-to-asset mapping.
4. **Pilot** the model within a prominent Estonian government organisation to evaluate its practicality and benefits.

4 Literature review

Corresponding to the second stage of the DSRM, "Objectives of a Solution," this chapter explores existing research and frameworks related to ITIL, E-ITS, and CMDBs.

The author conducted a literature review to investigate the present practices of combining the ITIL with Information Security Management standards such as ISO 27001 and E-ITS. While researching, it was impossible to find any literature on merging ITIL (any version of ITIL) with E-ITS because the latter is much too recent and requires more research. Thus, the author relies on older frameworks that, according to E-ITS authors, comply with the E-ITS [10]. Even then, there was very little previous work to rely on, specifically work that would incorporate an example of a configuration management database and show how it could be applied in merging ITIL with an ISMS standard.

4.1 Literature Review Process

4.2.1 Search Strategy:

The search strategy included these databases:

- IEEE Xplore
- ResearchGate
- Academia.edu
- IOPScience
- Ester
- TalTech library

Keywords and Search Combinations:

The following keywords and Boolean operators were used:

- **Keywords:**
  - "ITIL"
  - "ISO 27001"
  - "Configuration Management Database (CMDB)"
  - "Information Security Management"
  - "Asset Management"

- Search Combinations:

  1. "ITIL" AND "ISO 27001"
  2. "ISO 27001" AND "CMDB" AND "ITIL"
  3. "Information Security Management" AND "ITIL" AND "ISO 27001"
  4. "Asset Management" AND "ISO 27001" AND "ITIL"
  5. "E-ITS" OR “Estonian information security standard”
  6. "Configuration Management Database OR CMDB"

The following results were found after using the search combinations (Table 1). The table represents the database to the search combination (query) result:

Table 1: All search results from various databases

Database to query results	1	2	3	4	5	6
IEEE Xplore	5	0	1	0	566	1813
ResearchGate	-	-	-	-	-	-
Academia.edu	4624	4233	14736	335917	533	9265
IOPScience	4	0	2	0	13	42
Ester	0	0	0	0	0	85
TalTech library	27	0	7	0	393	82756

ReserachGate results were excluded because ResearchGate did not give an exact number of results found; the results are always 100 articles.

4.2 Inclusion and Exclusion Criteria

The author applied the following inclusion and exclusion criteria.

Inclusion criteria:

1. Articles published in peer-reviewed journals, on the E-ITS portal, in conference proceedings, or that are master's theses.
2. Articles focusing on integrating ITIL and ISO/IEC 27000 family of standards or E-ITS for effective asset and information security management.
3. Articles discussing CMDB, information security management, or risk management in the context of ITIL and ISO/IEC 27000 family of standards.
4. Articles about the integration of ITIL and Security Management.

Exclusion criteria:

1. Articles not published in English or Estonian
2. Articles without a clear focus on ITIL and ISO/IEC 27001 integration, mapping, or asset management
3. Articles primarily focus on topics other than CMDB, such as information security management, risk management, and service management.
4. Articles that concentrate on CMDB in a specific area, such as networking
5. Articles that include unrelated technologies or topics that aren’t relevant to the author’s research goals
6. Articles that concentrate on explaining what a CMDB database is this topic sufficiently covered already.

After applying the inclusion and exclusion criteria and doing a short review of the abstract and conclusions of the articles that were found, a small number of articles was chosen. The author concedes that there are a lot more relevant articles and data sources out there. Still, the author decided that the chosen articles are sufficient to determine the research objectives and gain an insight into the state of the art. The fact is that there isn’t much research done around CMDB and E-ITS, and the author's research topic is needed and relevant to the Estonian IT community.

4.3 Synthesis

In information security and IT service management, structured frameworks are undeniably crucial. Among the most predominant of these frameworks for service management is the Information Technology Infrastructure Library version 4 (ITILv4) and for Information Security Management ISO 27001 [11], [12], [13].

ITIL is a framework of service management best practices, empowering organisations to streamline service delivery and enhance customer experience [12], [14].

Complementing ITIL is ISO 27001, which is an internationally recognised standard detailing the establishment, improvement, and maintenance of an Information Security Management System (ISMS) [13], [15]. This standard does not merely operate in silos. Instead, it integrates seamlessly with other standards, bridging the potential gaps in the security management [14], [16].

Both ITIL and ISO 27001 employ the Plan-Do-Check-Act principle [11]. This cyclic method encourages organisations to consistently refine and adapt their processes in response to the evolving dynamics of IT and security [14], [17].

Integrating ITIL and ISO 27001 leads to potential security and service delivery improvements. However, this integration presents challenges, including the documentation preparation for the ISO 27001 [15] and the intricacies of melding the ITIL processes [12]. With these challenges, many organisations need guidance on the best approach [16].

This review aims to investigate this integration, exploring the challenges and benefits of merging ITIL and ISO 27001. The review provides a comprehensive understanding by analysing various sources, assisting organisations in merging IT service management and information security practices. This, in turn, will lead us to understand how to do the same with E-ITS and ITIL, as E-ITS is designed to be compliant with ISO 27001.

This research cannot directly delve into E-ITS as there needs to be more literature on this topic alone.

4.3.1 Enhancing IT Operations with CMDB for ITIL, E-ITS and ISO27001 Compliance

The Configuration Management Database (CMDB) is crucial for managing IT services and assets and ensuring security within IT frameworks such as ITIL and the Estonian Information Security Standard (E-ITS) [2], [18]. A CMDB goes beyond a mere data repository by enabling organisations to manage their IT infrastructure with greater visibility and control, supporting service delivery and security protocols.

A CMDB's integration with the ITIL framework in IT service management promotes an organised approach to managing IT services. This structured repository offers a complete inventory and detailed mapping of IT assets, which is crucial for effective service management. The CMDB's role in this framework ensures that IT services are delivered efficiently and aligned with business goals, thereby improving user satisfaction and supporting strategic IT decision-making [3].

The CMDB provides a unified view of IT assets for asset management, detailing their attributes and interdependencies. This comprehensive visibility allows for effective asset lifecycle management, from procurement to disposal, aligning IT resources with business needs and optimising operational efficiency [19].

Security management benefits significantly from CMDB integration, particularly in adherence to E-ITS. The CMDB supports identifying and managing information security risks and facilitating the application of E-ITS security measures by providing insights into IT assets and their configurations. This alignment with E-ITS strengthens an organisation's security measures and ensures compliance with this standard, which is critical for protecting sensitive information and maintaining trust [2].

In conclusion, CMDB's integration within ITIL and for E-ITS compliance is instrumental in advancing IT service management, asset management, and security. The CMDB enables centralised management of IT assets, offering insights necessary for effective service delivery, efficient asset utilisation, and robust security management.

4.3.2 The Role of a Configuration Management Database in ITIL and ISO 27001

The Configuration Management Database (CMDB) is paramount in IT service management and information security, functioning beyond a mere data repository [4], [20]. CMDB's integration into ITIL and ISO 27001 frameworks aids in honing IT operations and enhancing security management.

Susanto and his team emphasise the integral role of the CMDB in rendering a comprehensive view of the IT infrastructure, which proves vital for efficient monitoring, incident logging, and timely response. As organisations tread the complex paths of ITIL workflows and ISO 27001 security measures, the CMDB emerges as an essential tool. It assists in consistent asset tracking, understanding interdependencies, and proactively identifying and rectifying vulnerabilities [5].

Moreover, the CMDB's significance becomes apparent when ITIL's service management principles are integrated with ISO 27001's security standards [17].
It creates a cooperative environment, combining the best aspects of both standards, resulting in a combined and forward-thinking approach to managing services and security [15].

4.3.3 Tangible Benefits of Integration

Combining ITIL and ISO 27001 in IT service management and information security brings several benefits, as shown by different studies [12], [15], [17]. While each standard improves organisational efficiency, they offer more significant advantages combined.

Merging ITIL and ISO 27001 helps streamline processes, avoiding duplication and unnecessary work. This combination has increased operational efficiency and cost savings for companies [17].

ITIL focuses on structured and quality-driven IT service management. When combined with ISO 27001, which emphasises information security, the result is a higher quality of processes. This means that IT services are effective and secure, enhancing the overall operations in the organisation [12].

ISO 27001 values accurate data and structured processes. Studies have shown that ITIL's detailed service management can help meet ISO 27001's standards, improving the management of security processes [12]. This merger ensures security measures are both appropriate and up-to-date with organisational needs.

Integrating these standards gives a more precise and fuller picture of the IT environment, making it easier for IT managers and security professionals to tackle challenges proactively. This makes the organisation more flexible and quick to respond [15].

The combination of ITIL and ISO 27001 builds on the strengths of ITIL's focus on service management and ISO 27001's dedication to information security. This creates a robust framework that benefits the entire organisation [12].

Another critical advantage is shared responsibility. Research has highlighted that both standards promote collective accountability—ITIL for delivering high-quality services and ISO 27001 for strengthening security. This shared focus ensures everyone in the organisation is committed to maintaining high standards [15].

In summary, combining ITIL and ISO 27001 transforms organisations, offering clear benefits like cost reductions and better security. This integrated approach helps organisations confidently face the challenges of the digital world [15], [17].

4.3.4 Obstacles to Implementation of the Standards

Integrating standards and frameworks like ITIL and ISO 27001 into organisational practices, along with Configuration Management Databases (CMDBs) deployment, introduces a series of challenges. These challenges span from technical to cultural aspects and underscore the complexity of aligning IT service management with information security standards. A synthesis of the insights from different studies [2], [16], [19] reveal several obstacles for an organisation.

1. **Expertise and Resource Constraints**: Implementing ISO 27001 necessitates a profound understanding of information security standards, often surpassing the available resources within organisations. The challenge is compounded by the need for specialised expertise, employment, and the translation of complex

security standards concepts into actionable strategies [16].

2. **Selecting the Appropriate Standard**: Choosing the most suitable information security standard from several options is challenging, especially since each has unique characteristics. This selection process is critical as it dictates the organisation's approach to managing its information security risks and controls [16].

3. **Cultural and Organisational Shifts**: Adopting standards like ISO 27001 and integrating ITIL and CMDB systems into existing practices demand significant cultural and organisational changes. This includes moving towards a centralised approach to configuration management and embracing new security controls, which may affect the existing company culture and operational habits [2], [19].

4. **Data Accuracy and System Integration**: Implementing a CMDB effectively requires ensuring the accuracy and completeness of data, which poses a substantial challenge. Moreover, integrating CMDBs with existing IT systems can present complexities that must be carefully managed to avoid disrupting existing workflows [19].

5. **Lack of Automated Verification Tools**: A specific challenge identified for the Estonian Information Security Standard (E-ITS) is the need for more unified tools to verify the implementation of security measures automatically. Relying on manual processes for this verification is inefficient and prone to errors, making it difficult for organisations to assess and demonstrate their compliance confidently [2].

These obstacles highlight the intricate balance organisations must achieve between technical adjustments, resource allocation, and cultural transformation. Overcoming these challenges requires meticulous planning, commitment to continuous improvement, and a strategic approach to integrating IT service management and information security practices within the organisation. The journey towards successful standard implementation and integration is complex but crucial for enhancing IT governance, service delivery, and security protocols.

4.3.1 The Conclusions from the Literature Review

Integrating ITIL and ISO 27001 into organisations comes with challenges, including needing specific skills, adapting the company culture, and ensuring processes and systems work well together. For instance, accurate data management, choosing the proper standards, and merging new systems with the old are critical areas that require careful handling [19]. The need for automated tools to verify security measures complicates things further, making manual checks a slow point for efficiency [2].

However, these challenges also present opportunities for significant improvement in how services are delivered and how secure information is managed. Integrating ITIL and ISO 27001 and a solid CMDB system can lead to comprehensive improvements across service delivery, information security, and organisational resilience [2], [19]. These standards support one another and, with the insights from CMDBs, equip organisations to address changes in the digital landscape confidently.

To overcome these obstacles, organisations need a strategy that includes improving organisational culture, optimising processes, and encouraging ongoing improvement. Success depends on the collective effort of all stakeholders, dedicated planning, and a solid commitment to these initiatives [2], [19].

In conclusion, integrating ITIL and ISO 27001 poses challenges but offers a path to greater efficiency, security, and operational excellence. Embracing the challenges, being open to change, and striving for the best practices can prepare organisations to succeed in the interconnected and digital age [16].

5 Configuration management database model design and development

This section delves into the design considerations, constraints, and the detailed model development process, aligning with the design and development stage of the DSRM methodology [7]. Building on the objectives outlined in earlier discussions, this work aims to formulate a model that meets the requirements of both standards management processes. The author firmly believes in the proposed approach's cost-efficiency and potential to significantly enhance the visibility and understanding of the organisation's asset landscape. These objectives serve as the cornerstone for the model's design philosophy.

5.1 Basic Concepts and Components of a CMDB

Understanding the core principles of a Configuration Management Database (CMDB) is essential to lay the groundwork for the design and development phases. This understanding will facilitate the identification of crucial Configuration Items (CIs) necessary for our design.

A CMDB is essentially a repository that catalogues all organisational assets [2], [3], [4], [21]. It's vital to note that the term "assets" encompasses many entities, including hardware, software, personnel, and processes [3]. However, not every asset is automatically a part of the CMDB. The selection of assets to be included as CIs is determined based on the specific requirements and goals of the CMDB design [2], [4], [21], [22].

With the concept introduced, defining a CMDB and its constituent elements is essential. This effort utilises insights from the ITILv4 (ITIL) [23] and the US Department of Defence’s military standard 3046 (DOD3046) [24], both of which provide a framework for configuration management.

Definition for Configuration Item

ITIL: *"Any component that needs to be managed in order to deliver an IT service."* [23]

DOD3046: *"A product or an aggregation of products that accomplish an end-use function and requires separate identification. An item is designated as a CI for purposes of additional configuration management focus due to its complexity, logistic support requirements, or acquisition strategy or because it is intended to undergo configuration status accounting or verification and audit separately from other items. Configuration items are end items or major components of end items, which typically have performance requirements allocated to them and documented in their own specification."* [24]

Comment [3]: In simpler terms, it can be understood as any asset in the organisation that is used to deliver value. It could be a chair, a car, a piece of software, a database server, the data in the database or people. The logic here is that all assets or parts of assets can be configuration items, but not all assets are configuration items, only the chosen ones.

In the context of this research, a configuration item is defined as an asset or a component of an asset that has been chosen to be inserted into the configuration management database to keep track of.

Definition of the Configuration Record:

ITIL: *"Configuration record is a record containing the details of a configuration item (CI). Each configuration record documents the lifecycle of a single CI. Configuration records are stored in a configuration management database."* [23]

DOD3046: No equivalent definition

Comment: In ITIL, an entry to a CI is called a configuration record, while in DOD 3046, there is no such distinction between a CI and a record itself. It is helpful to have this distinction, as it allows us to talk about a type of an object and an object of the type.

Definition of Configuration:

ITIL: *"An arrangement of configuration items (CIs) or other resources that work together to deliver a product or service. It can also be used to describe the parameter settings for one or more CIs."* [23]

DOD3046: *"The functional and physical characteristics of existing or planned hardware, firmware, software or a combination thereof, as detailed in requirements and technical documentation and ultimately achieved in a product."* [24]

Comment: Configuration is a collection of configuration items that, when combined, make up a product or a basis of a service. In essence, when we interconnect a group of CIs, it will make up a configuration.

Definition of Configuration Management:

ITIL: *"Configuration management provides information on the CIs that contribute to each service and their relationships: how they interact, relate, and depend on each other to create value for customers and users. This includes information about dependencies between services. This high-level view is often called a service map or service model and forms part of the service architecture."* [23]

DOD3046: *"An engineering and management process which ensures the configuration of an item is known and documented and changes to an item are controlled and tracked for purposes of establishing and maintaining consistency of a product's performance, functional, and physical attributes with its requirements, design, and operational information. "* [24]

Comment: This process maintains all the configuration items in the database; now, configuration records are inserted through this process.

Definition of Configuration Management (CMDB):

ITIL: *"A database used to store configuration records throughout their lifecycle. The CMDB also maintains the relationships between configuration records."* [23]

DOD3046: *“The CSA system shall retain a complete historical record of the information defining the CI and associated components and shall be capable of generating historical configuration baseline information at major points in the lifecycle of the product.”* [24]

Additionally, it is helpful to highlight this excerpt from the ITIL v4 standard: *“Configuration information can be stored and published in a single configuration management database (CMDB) for the whole organisation, but it is more common for it to be distributed across several sources. In either case, it is important to maintain links between configuration records so that people can see the full set of information they need and how the various CIs work together.”* [23]

The description provides insight into the foundational elements and the structure of a Configuration Management Database (CMDB). It clarifies that a CMDB may comprise various data sources consolidated into one comprehensive database or exist as a singular database entity. Although consolidating diverse data sources into a single database might present challenges, centralising data access through one platform is considerably advantageous for operational coherence and efficiency [25].

Equipped with an understanding of the CMDB's fundamental components, the discussion advances towards the conceptual framework of the model being developed. The first step of this process involves identifying the essential configuration items (CIs) necessary for a foundational yet practical CMDB. These CIs will be the humble beginnings of fulfilling the primary requirements of an organisation's change and service management.

In the next chapter, the author identifies the configuration items necessary for managing the information security management system, specifically in the Estonian Information Security Standard context.

5.2 Requirements for the Configuration Management Database.

The requirements for a configuration management database depend somewhat on the organisation or company where it is being implemented because the asset groups can be quite different [3], [22]. For example, a hospital would have many other types of assets than an IT company or a military base.

This thesis is scoped around Estonian Government IT organisations, and, in this context, there are a lot of commonalities, which leads the author to believe that a baseline standard can be achieved.

The author is setting up the premise as such.

1. The organisation provides IT-related services, such as hosting, IT support, online services, etc. It needs to maintain servers, network devices, and software.
2. The organisation already follows either ISO20000 standard or ITIL practices or is willing and ready to do so to improve its majority and maintenance processes.
3. The organisation must implement E-ITS standards and pass the auditing process mandatorily or at least want to do that voluntarily.

These assumptions allow the author to scope the design without considering organisations whose primary services are not IT-related. Therefore, their configuration items might differ from those the author is interested in. However, this doesn't mean the ideas in this thesis wouldn't be helpful elsewhere; it simply indicates that the author's experience is limited to the IT domain.

Identified requirements for a configuration management database by investigating the organisation's structure and goals.

1. The CMDB must support the automation of infrastructure and other systems by providing comprehensive asset data and facilitating detailed grouping of assets based on intent and responsible staff members.
2. The schema of the CMDB must be extensible without redesigning the entire model.
3. It must be possible to track the interconnections of applications to determine their relationships and dependencies for impact analysis.
4. The CMDB database must provide server groups suitable for managing E-ITS controls.

5. Business processes or contracts that contain Service Level Agreements (SLA) and information security requirements should have logical, queryable connections to assets.

Requirement 1: CMDB must support the automation of infrastructure.

It must provide a grouping principle for servers or devices such that the groups do not allow overlapping assets.
For example, the server group in the monitoring system requires that no server is put into two groups because otherwise, alerting systems that rely on group-based alerting, such as Zabbix or Prometheus, would start sending alerts on the same issue twice and thus overwhelm the receiver with irrelevant information.

Requirement 2: The schema of the CMDB must be extensible.

The author provides a basic schema that would satisfy the set requirements in this thesis, but there are more things possible outside of the scope. The schema must allow for easy extensibility with additional configuration items without changing the core.

Requirement 3: It must be possible to track interconnections of applications to determine their relationships and dependencies.

Some of the activities that one must do either in a service management system (SMS) or an information security management system (ISMS) are:

- Incident management
- Availability management
- Risk assessment
- Access management
- Responsibility management
- IT Service Governance

For each of these activities, it is necessary to have an overview of assets and information on how these assets are interconnected. For example, an output of this database of interconnected CIs could be impact analysis and availability analysis, where it is necessary to understand what would happen if a service or system component stopped working.

Requirement 4: The CMDB database must provide server groups suitable for managing E-ITS controls.

E-ITS is an ISMS system that is a so-called baseline security standard that provides a set of modules with controls that should be implemented in an organisation [1]. These modules need to be applied to the objects of interest ("target objects" as E-ITS defines them [20]) that need to be protected. For example, the "Linux/Unix server" module must be applied to all servers running Linux or Unix operating systems.
A corresponding CI in the CMDB needs to be created, allowing the organisation to manage these objects of interest in a grouped manner so the staff wouldn't need to correlate the modules to each piece of asset separately for the E-ITS [1].

Requirement 5: Business processes or contracts that contain Service Level Agreements (SLA) and information security requirements should have logical, queryable connections to assets.

Both E-ITS and ITIL must be able to track availability from an ITIL perspective to ensure that the service level agreement goals that have been agreed upon in contracts with clients are met [23]. For the E-ITS side of things, it is necessary to track availability to meet the "Confidentiality, Availability, Integrity" (CIA) levels usually set in the contracts or must be met due to the risk assessment result. E-ITS requires that the security level be set for the business process, and from there, the required level must propagate to the assets needed for the process. The CMDB database needs to facilitate this requirement from the E-ITS [20].

Assessment of Requirements

A good CMDB would support the organisation in all stages of the ITIL service lifecycle [7], providing accurate and comprehensive information about configuration items (CIs) and their relationships. However, this thesis focuses specifically on asset management for information security management and service management, which represent critical components of the ITIL service lifecycle. The CMDB data model proposed in this thesis aims to facilitate better integration between the Information Technology Infrastructure Library (ITIL) and the Estonian Information Systems Security Standard (E-ITS) by enabling accurate tracking, grouping, and mapping of assets. This focus aligns closely with the Service Design and Service Transition stages, ensuring that new or changed services are correctly documented and compliant with security standards before being moved into production.

5.3 Design

The best approach to building a CMDB database is to start small and work iteratively to add necessary CIs. This approach aligns well with the DSRM process and ITIL and ISO27001's Plan, Do, Check, Act iterative development process.

The author plans to design the data model iteratively, working through the abstraction layers from the bottom up and top down until they meet in the middle.

The author defines six abstract layers, similar to previous works in this field [12], [31]. Since the CMDB database is envisioned as a graph, the layers themselves are unimportant for the design. However, they are helpful tools for making a mental map and imagining how the pieces should fit together.

All layers are listed from highest to lowest as such:

1. **Service layer:** Based on ITIL and E-ITS, documentation must list all the organisation's services. ITILv4 defines service as *"delivering value to customers by facilitating outcomes they desire without the need for them to own specific costs and risks."* [23]. In the CMDB design, there must be a type of CI (or

many) that documents the service as understood in ITIL and E-ITS.

2. **Process or Contract layer:**
   ITIL defines a process as *"a set of interrelated or interacting activities that transform inputs into outputs to achieve a specific objective."* [23]. If we consider that the objective is to provide a service, there may be processes that support other processes, which we could term as sub-processes. However, the primary goal of these processes is still to deliver a service [23].

   The CMDB does not need to document every process in detail or capture the complexities of the process flow. This level of detail is better suited for other tools, such as BPMN documentation software. However, the CMDB should reference such documentation. It must include a Configuration Item (CI) that links to this documentation and has fields for information on the security level, as required by E-ITS. If defining a process is not feasible, a contract object can be used as a substitute.

3. **Data layer:** The data layer consists of data processed or accessed by applications in the application layer and utilised by processes in the process layer. This data is conceptualised as an abstract entity, integral for all process activities, whether accessing or processing. It's linked to relevant documentation and connects with the Process and Application layers.

   The idea for a data layer is not original to the author and is described in The Open Group Architecture Framework (TOGAF). When updating or replacing applications, planning for data migration—including master, transactional, and reference data—becomes crucial [26]. This process should outline migration requirements and detail the necessary data transformation, cleansing, and refinement to meet the new application's specifications. Therefore, documenting data in the CMDB design is crucial for maintaining data integrity and facilitating seamless interaction between processes and applications.

4. **Application layer:** The Application layer encompasses all aspects of an application and its components, focusing on data processing or access. While the Process layer outlines how data is used within a process, the Application layer

specifies the locations where data is processed or accessed. Additionally, the Data layer serves as a bridge, connecting processes with their respective applications.

5. **Assets:** The Assets layer details every asset within the CMDB. Without manual intervention, these assets lack context explaining why they are part of the database. This model assigns context through a hierarchy, from the service to the application layer. Assets below the application level are included in the database exactly as they are captured by software tools using automated processes.

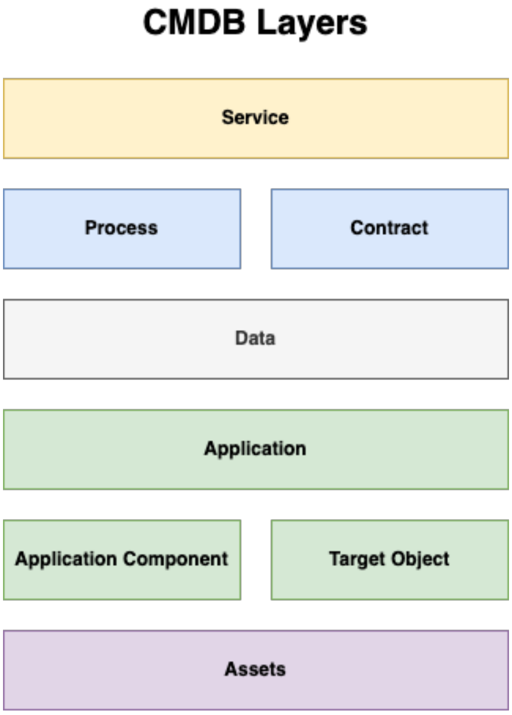

Figure 3: CMDB Layers

5.3.1 Design Considerations

The envisioned model adopts a graph-based structure, where Configuration Items (CIs) are nodes, and the relationships between these CIs are represented as edges. Crucially, this model facilitates the inclusion of annotations directly on these edges, providing context and details about the relationships between assets. This approach is implementable in any CMDB software that supports the creation of directed references (or directed edges) and can attach comments to these references.

During the model's development, it became apparent that the Configuration Items (CIs) resemble the concept of aggregation in object-oriented programming. Thus, looking at the problem from this perspective can help gain valuable insights.

Aggregation is a type of association that represents a "has-a" relationship between objects, where one object is a container, and the others are contents. This relationship implies that while the contained objects belong to the container, they can also exist independently. Here is how the two concepts align:

1. **Containment and Ownership:** In both cases, there's a notion of containment without strict ownership. A CI can be part of multiple configurations or services in a graph-based CMDB. Much like in OOP, an object can be aggregated by various other objects in the database [4]. This represents a flexible relationship where components maintain their independence while contributing to the functionality of the whole.

2. **Hierarchy and Dependency:** Both models express hierarchical relationships and dependencies. In aggregation, objects form structures that show how higher-level objects are composed of lower-level ones [27], [28]. Similarly, a graph-based CMDB visually depicts how higher-level services or systems depend on underlying components, allowing for an understanding of critical dependencies [4].

3. **Navigability:** In aggregation, the relationship allows navigation from the container to the contained objects [29]. This is paralleled in CMDBs, where the relationships between CIs allow tracing dependencies and impacts across the network [4]. This navigability is crucial for impact analysis, troubleshooting, and planning changes [3], [4].

4. **Modelling Complex Relationships:** Both techniques allow for modelling complex relationships in a manageable and understandable way. In OOP, aggregation helps break down complex systems into simpler, manageable objects with clear relationships [30]. Similarly, a graph-based CMDB breaks

down the IT environment into interrelated components, making it easier to manage and understand [3].

This realisation helps us understand the limitations and pitfalls that we might encounter down the road. Knowing that the concept aligns well with aggregation is encouraging, as this will mean that we also have loose coupling, primarily throughout our CIs.

Loose coupling is preferred over tight coupling because it allows for easier code maintenance, extendibility, and scalability, among other things [30], [31].

Aggregation, however, has a notable downside: complexity, which is the enemy of a successful CMDB project [6]. In other words, tracking complexity and continuously monitoring the database's health is very important.

5.3.2 Design Proposal

In a previous chapter, the author outlined requirements for the configuration management database. The author considers each requirement in the designing phase and proposes a solution.

1. **CMDB must support automation in the organisation.**

   Problem analysis: There needs to be a CI that can be used to group hosts and, in turn, can be used to manage host groups in other systems, such as monitoring systems.

   The main problem the monitoring system tries to solve is detecting application issues. Once issues are detected, the organisation's staff that oversees maintaining this application must be alerted. This also corresponds to E-ITS requirements of monitoring requirement, specifically in the " *OPS.1.1.1: IT-haldus üldiselt*" module.

   Alerts should only be sent to staff responsible for the asset experiencing issues. This implies that when grouping assets like servers, the grouping must be specific to the administrators managing them. In other words, a host group should not include servers overseen by different administrators.

Referencing the layers the author established in the previous chapter (See Figure 3), the layer where the grouping can be done is the application layer. However, the issue remains that there can be separate responsibilities within the context of one application. For example, the database can be managed by Admin A, the application servers can be maintained by Admin B, and the application itself is maintained by Admin C.

This establishes the initial criteria for categorising assets and the future Configuration Items (CIs) on which this categorisation will be based. The primary criterion is that the CI must be defined by responsibility.

To satisfy this criterion, the author suggests introducing a CI named “application component” (Figure 4). This strategy involves not linking all hosts directly to the Application CI. Instead, it utilises an intermediary CI, which facilitates modelling the division of responsibility. This method ensures that responsibilities are appropriately allocated, reflecting the operational and management structure within the system.

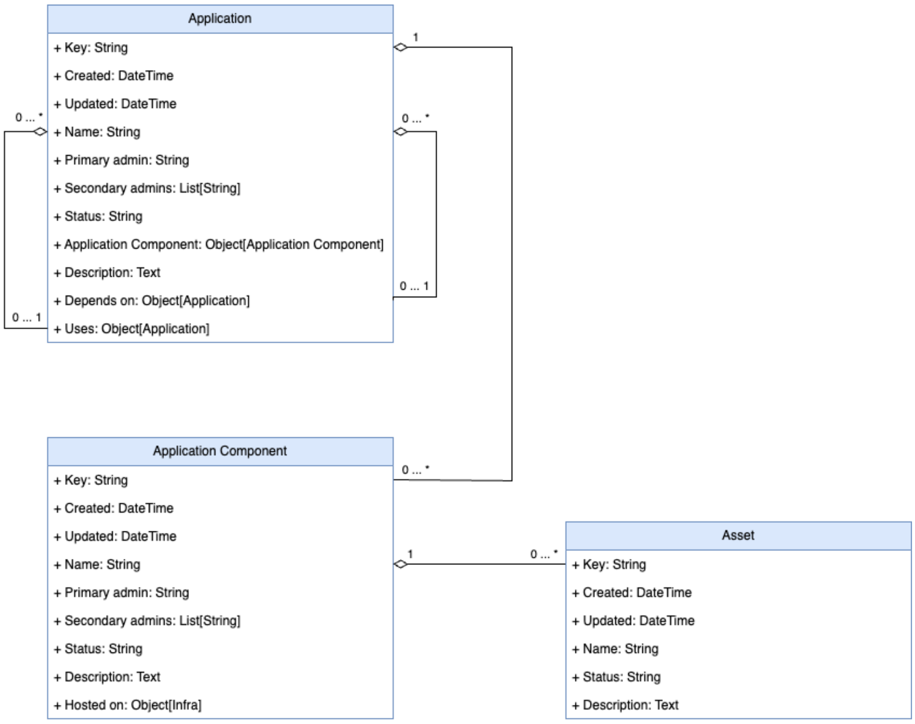

Figure 4: Application Component and Relations to Application and Assets

It is tempting to define relationships between application components. Still, from the author's experience, this is not a good idea as it makes the model too complicated to manage and keep up to date with minimal or no benefit. It seems to be sufficient to have relationships between application CIs.

To effectively group assets under the Application Component, we also need a schema for assets and asset type CI-s that are needed. The initial schema and selection of asset types are chosen by the author as such (Figure 5).

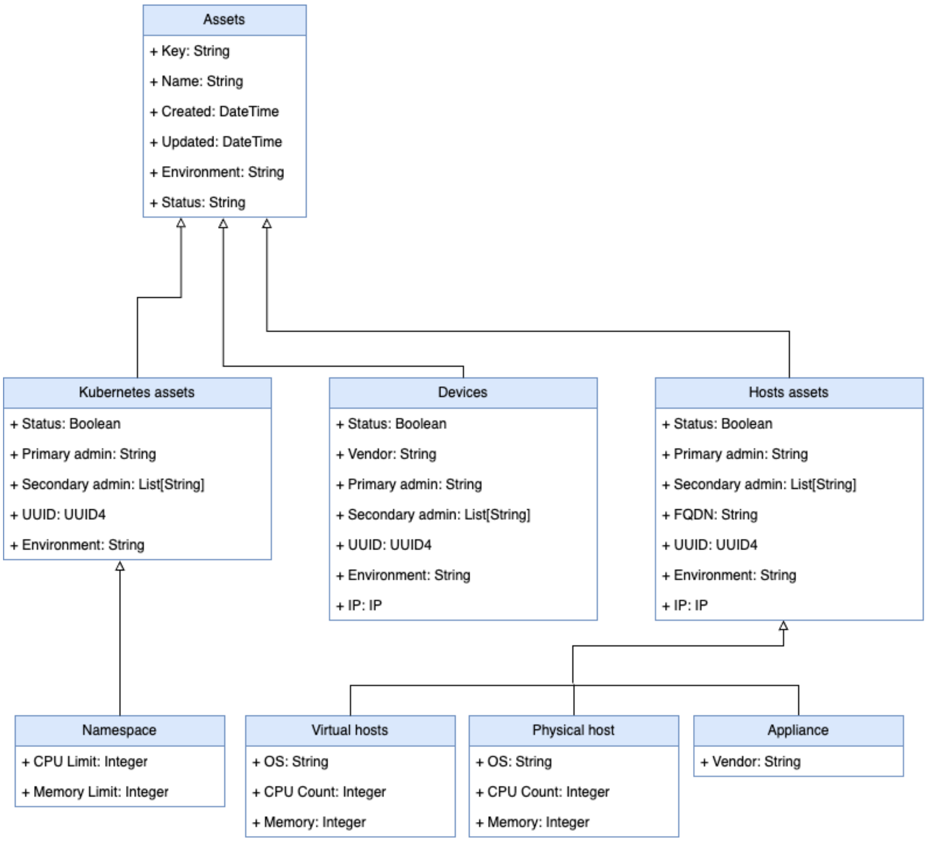

Figure 5: Assets Schema as UML Class Diagram.

Proposed solution: Introduce a CI called “Application Component” that is used to model the responsibility of the assets (Figure 4). It should have at least the following attributes:

- Name: (The name of the application component)
- Primary administrator: (The name/username of the administrator who is responsible for the maintenance of this asset)

- Secondary administrator: (The administrator or a list of administrators who would be second in line and act as a backup to the primary administrator)
- “Hosted on” is a list of assets (Kubernetes namespace, virtual hosts, physical hosts, network devices) installed on the application component.
- Status: Tracking the lifecycle of this CI like any other is necessary, so we need a status attribute that holds its state.
- Environment: Most organisations have production and test environments at the very least, and thus, we should have separate applications for test and production environments. We will create an attribute that documents it.

This is not an extensive list of attributes that this CI can or should have. The guiding principle is to start small, or in other words, less is more. This is important to follow because many CMDB projects fail due to complexity [4], [6]. Attributes should be added only if necessary and, even then, only if the attribute is needed for each CMDB record with the CI type. In essence, there should not be any CI record in the database that has unset attributes. From the author’s experience, if this happens, it’s a good indicator that something has gone wrong in the design phase.

2. **The schema of the CMDB must be extensible.**

Problem analysis: The author is aware that this design does not currently cover all possible ITIL requirements, and this thesis aims to design a model that can be used as a foundation upon which to build. The current goal is to show that the model meets E-ITS implementation and ITIL configuration management requirements.

Thus, the model needs to be extendable. As stated, the basic principle is that the model is a graph. As a graph, it’s possible to introduce any new CIs and create relations between them without changing existing relations, which means extending the model is easy.

Proposed solution: The design of the CMDB data model heavily relies on references between objects to document the context of the object itself. As mentioned, the model is envisioned as a graph, where each CI is a node, and the references between CIs are edges. This approach allows users to track and document context situations and relationships between applications, servers, services, and even users if necessary. Building everything out as a graph allows us to rearrange references whenever needed without changing the information inside nodes. Much like aggregation in object-oriented programming [27], [32].

3. **It must be possible to track the interconnections of applications to determine their relationships and dependencies for impact analysis.**

   Problem analysis: The ITILv4 framework addresses the need to systematically evaluate the potential impact of changes within the IT environment on services and stakeholders [3], [4], [33]. This is crucial for maintaining service integrity and value creation as adjustments are made.

   The core issue it tackles is the balancing act between implementing necessary changes and ensuring they do not adversely affect the service delivery or stakeholder interests. Impact analysis within ITILv4 is intended to identify these effects beforehand, ensuring that alerts and responses are tailored and directed appropriately [34]. This necessitates a structured approach to categorising services and assets in a manner that aligns with their management and oversight responsibilities. The process must ensure that any change-related notifications or actions are directed explicitly to the relevant parties, much like how a monitoring system must send alerts only to those directly responsible for an asset. Consequently, services and assets must be grouped and managed to reflect the specific administrative domains within the organisation, preventing cross-notification issues and ensuring clarity in responsibility and response.

   Proposed solution: The design facilitates cross-referencing of application CIs to determine their dependencies. This method will enable the analysis of the impact of a failure in one application on another application or the entire system. Ideally, two types of references would be beneficial: a “uses” reference, indicating that one application interacts with another without relying on it for functionality, and a “depends-on” reference, highlighting that one application's functionality is contingent on the other (See Figure 4).

4. **The CMDB database must provide server groups suitable for managing E-ITS controls.**

   Problem analysis: E-ITS requires the organisation to identify “target objects”, which are assets the organisation needs to protect. These assets can be network devices, data, physical assets, etc.

The target object will be assigned modules from E-ITS that will determine the controls that need to be implemented on these assets [20]. The types of controls (basic, standard, and high) are chosen based on the level of protection requirement. The level of protection requirement must be determined at the business process level in [20], [22]. Thus, the protection requirement criterion is implied on all assets that fall under a business process. Therefore, a reference chain from the process level to the assets is necessary.

Proposed solution: The Configuration Item (CI) "application component" is ideally suited for managing E-ITS controls (Figure 7). Since the “application component” CI already has a restriction around responsibility, it’s possible to add another restriction, which should be the intent or context of the assets in the group. Essentially, one application component should group assets that are interchangeable with each other or similar in their responsibility in the network.

Given its design for group hosts according to their specific responsibilities and context, the "application component" aligns excellently with the E-ITS requirement. It enables hosts to be organised into clearly defined groups that precisely match the intentions of E-ITS modules. For example, an application component that organises the database cluster would include only those hosts essential for database functionality and managed by the database administrator, directly supporting the focused objectives of E-ITS's database-specific modules.

5. **Business processes or contracts that contain Service Level Agreements (SLA) and information security requirements should have logical, queryable connections to assets.**

   Proposed solution: There will be connections from the Service CI to the Process or Contract CI, which in turn has edge connections to the Application CI, and through the application component CI that is connected to the Application, it has a path to the hosts (See Figure 6 and Figure 4).

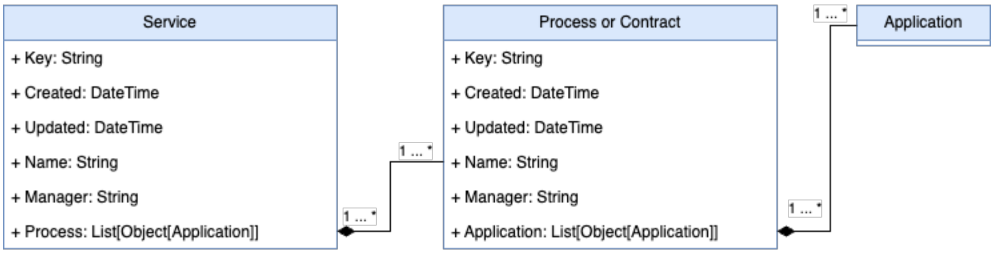

Figure 6: Services Relation to Asset Through the Application CI

This setup will allow the user to query all hosts necessary for a given service and, therefore, also for a given process. This allows the system to propagate the security requirements from the process to the hosts. Additionally, it will enable the ITIL quality assurance and capacity planning processes to use the input from the CMDB system.

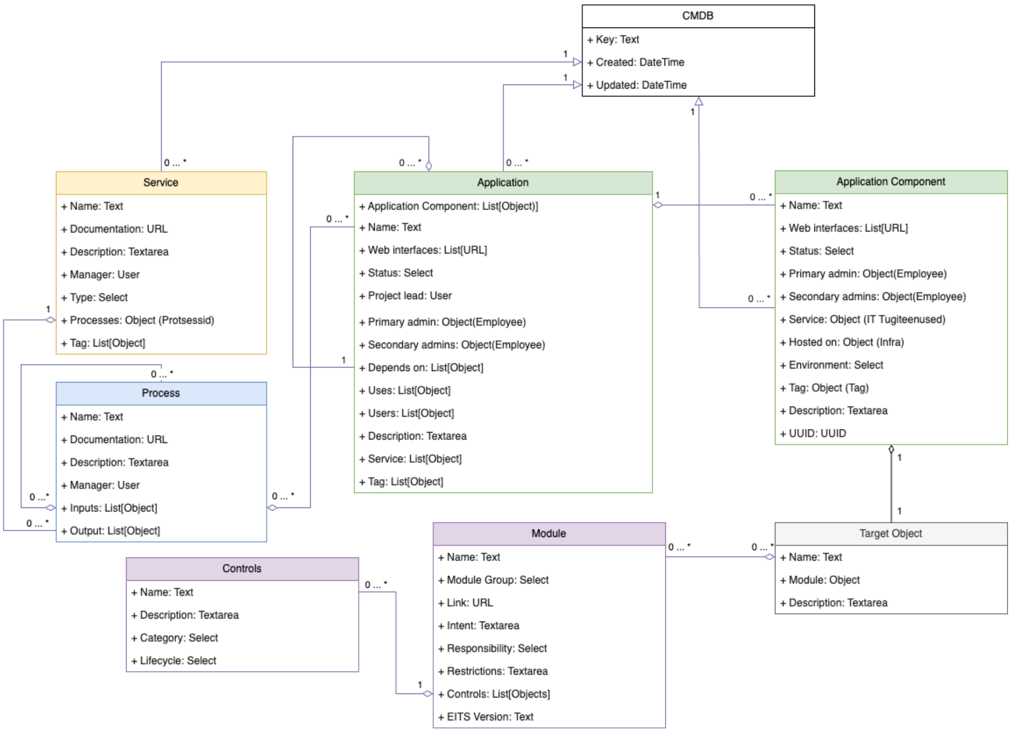

Figure 7: Class Diagram of the CMDB Data Model with E-ITS CI-s.

While the model descriptions align with the principles of aggregation from object-oriented programming (Figure 7), it is more accurate to conceptualise this as a directed graph (Figure 8). Viewing it as a graph helps one grasp the infrastructure and services more intuitively, which in turn helps prevent errors when expanding the model.

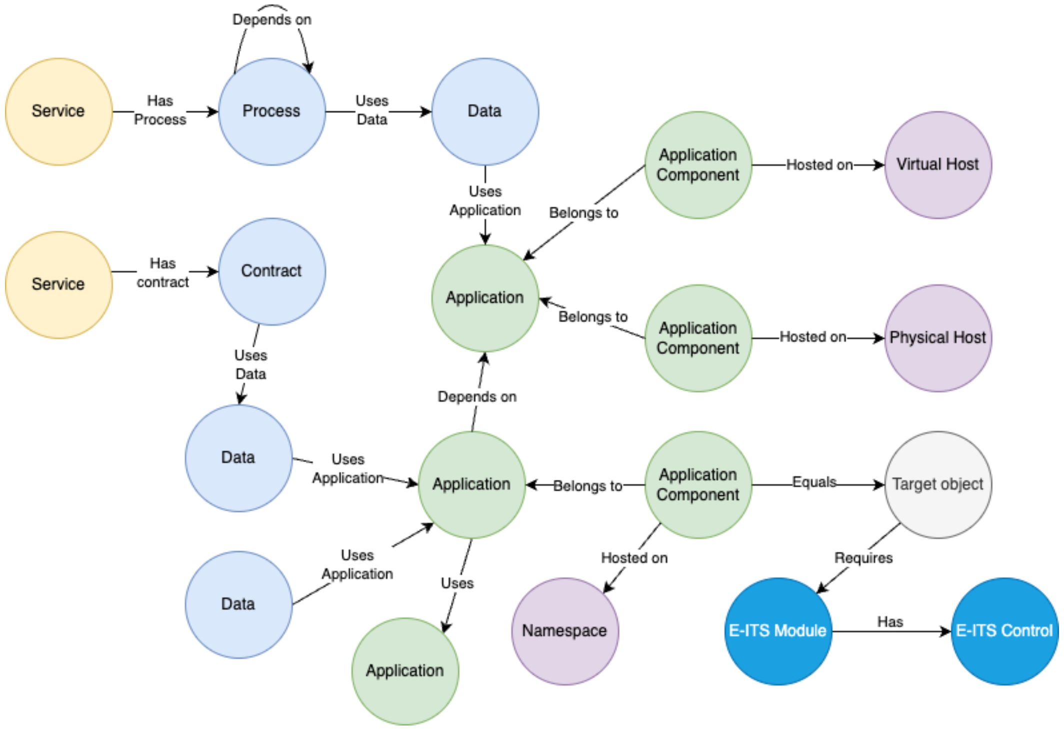

Figure 8: Graph Representation of the CMDB Data Model as Designed.

The final design of the CMDB data model contained these CI-s that are used to create context around assets (Table 2).

Table 2: List of CMDB CI-s for Creating Context.

configuration Item (CI)	Description	Role in CMDB
Service CI	Represents services offered by the organization.	Documents service attributes like service levels and compliance requirements.
Application CI	Represents individual applications.	Documents dependencies and relationships, connecting to both process and service CIs.
Process CI	Represents business processes.	Connects to application CIs and includes attributes like protection requirements for documenting workflows and security.
Contract CI	Represents agreements with clients or partners.	Similar to process CIs but specific to legal and

		contractual obligations, especially in hosted service contexts.
Data CI	Represents data handled by applications.	Connects processes with applications, emphasizing data handling and security.
Hosted Service CI	Similar to service CIs but specific for hosted scenarios.	Includes CIs between the hosted service and the application, detailing attributes for protection requirements.
Assets CI	A basic unit in CMDB represents physical and virtual assets like servers or devices. Specific assets in a schema are up to the implementer to decide. This thesis is around the context.	Used in the initial asset collection and categorization phases, fundamental for tracking and management of assets. Assets collected and categorised in this thesis are shown in (Figure 5)
Application Component CI	An intermediary CI used to model divisions of responsibility within applications.	Defines responsibility areas and connects various assets under a single application, managing asset responsibilities.
Target Object CI	Represents specific assets or groups of assets that need to be protected according to E-ITS standards.	Identifies and categorizes assets requiring specific security measures under E-ITS, facilitating targeted protection controls.
E-ITS Module CI	Represents a set of specific E-ITS guidelines applicable to a particular type of asset or IT infrastructure.	Provides a structured approach to applying E-ITS security standards to CIs, ensuring compliance across the organization.
E-ITS Control CI	Specific controls within an E-ITS Module that need to be implemented on Target Objects.	Ensures that each Target Object meets the required E-ITS security controls, documenting and managing compliance efforts.

6 Implementation Plan

This chapter revolves around the "Design and Development" stage of the Design Science Research Methodology (DSRM), where the author focuses on planning the implementation of the Configuration Management Database (CMDB) model. Essentially, the author designs the model's implementation procedure into the organisation’s infrastructure and management processes.

The design and implementation of the configuration management database were done in stages and were scoped around building the data model without delving too deeply into the organisation's details.

Although ITIL suggests finding stakeholders and analysing the organisation's requirements before anything, the author knowingly skipped this Field [28]. The reason is the hypothesis that a generally usable data model can be achieved by looking only at the standards and CIs that typically need to be in the model.

This implementation plan builds the model from the bottom up (See Figure 3) by collecting information about the assets first, which is a supported approach in the “Implementing ITIL Configuration management” [4]. This is followed by grouping the assets into applications through the application component configuration item. The documentation of Services and Processes then follows this stage. This, in turn, is followed by documenting E-ITS target objects and referencing E-ITS modules. The final phase of the implementation plan is to test and validate the model, which will end in refinement. Below is the visualisation of the entire process (See Figure 9).

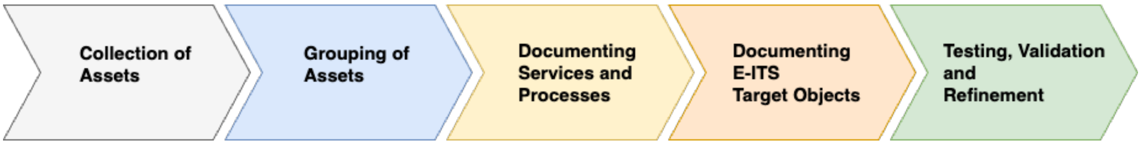
Collection of Assets
Grouping of Assets
Documenting Services and Processes
Documenting E-ITS Target Objects
Testing, Validation and Refinement

Figure 9: Stages of the Implementation Plan.

6.1 Collection of Assets

At this stage, the author chose which assets were most relevant and collected all the information about those assets into an asset management database (Figure 5) without any context on why those assets exist.

The author was interested in the assets essential for hosting applications and those required by hosting platforms. In the initial iteration, the author excluded office appliances like printers, workstations, Wi-Fi access points, etc.

To ensure all relevant assets were accurately mapped, the author utilised various sources, including the configuration of the virtualisation cluster and software specifically designed for discovering assets such as servers and network appliances. Given its potential role as a control set later, it was crucial to avoid using the monitoring system as a source for the asset list.

The objective at this stage was to identify as many assets as possible, intending to categorise these assets and build context around them in subsequent phases.

The information and CI-s we collected about the assets were based on availability and necessity. In essence, the author had a short list of essential information about the assets, such as IP-s, hostnames and fully qualified domain names of hosts. Anything else was out of scope for the first iterations, such as the number of cores, installed applications, or servers' other resources.

The author collected basic information in the first iteration of this process for later use for capacity planning and correlating the collected data with monitoring.

In the first iteration, the author collected information on the following list of assets (Figure 5):

- Virtual hosts: Hosts created on a hypervisor such as VMWare or Xen.
- Physical hosts: Hosts that have been installed on bare metal servers.
- Device: A device that is either a network device or an appliance.

- Kubernetes namespaces: Namespaces are logical groups in Kubernetes that contain a set of containers. A namespace can be considered a host, such as a virtual machine.

6.2 Grouping of Assets

Grouping assets under a CI helps the author understand their context. By adding context, the author intends to document why the assets have been introduced to the infrastructure.

The author has established that several assumptions and requirements are made about the infrastructure being covered by the CMDB database, and the grouping of assets is the main activity that will set the foundation on which everything else can be built.

In this implementation stage, it is assumed that most of the assets to be grouped have been collected and stored in an asset management database. Their attributes have also been documented as well as possible. To group them, a minimum list of attributes must be included in the assets (Figure 5).

The minimum list of assets and attributes that were collected in the pilot project:

Table 3: Asset and Attributes.

Asset Type	Asset Attributes	Asset Type	Asset Attributes
1. Virtual Host	IPs and network interfaces	3. Device	IPs and network interfaces
	Operating system		Operating system
	Hostname		Hostname
	Status		Status
	FQDN		FQDN
	Environment		Environment
2. Physical Host	IPs and network interfaces	4. Kubernetes Namespace	IPs and network interfaces
	Operating system		Status
	Hostname		FQDN
	Status		Environment
	FQDN		
	Environment		

This asset and attribute list was sufficient for the first iteration and implementation of the CMDB database model. More attributes necessary to collect are expected to become apparent when the model has been in use, such as location and hypervisor. This approach aligns with the DSRM research process and CMDB building best practices [3], [4]. It's always best not to optimise before it's necessary and not to collect information without it being needed [4].

In the design phase, the author envisioned a CI called the "Application Component" (Figure 4), which is a node in the graph between the asset, such as a virtual host and the application CI (Figure 4). This is sufficient to document the context of why the asset is required in the infrastructure and show the user of the CMDB database how the assets relate to other assets in the organisation.
At this stage, the implementation team in Organisation A only needed information about two configuration items: the Application and Application Component (Figure 4 and Table 3). So, it is best to scope all activities around these items to avoid getting overwhelmed and confused.

To complete this stage, it is necessary to include the systems administrators and/or service managers in the organisations—people who know the assets in question. Most often, the only staff in the organisation who know precisely how and why an asset is introduced to the infrastructure are those in charge of maintaining it.

The best approach forward:

1. Document the requirements necessary for grouping assets under Application Component CI
2. Let the systems administrator do the actual grouping.
3. Let their managers validate that the grouping has been done according to the documentation.

After this has been completed, the plan expects that there will still be mistakes and inconsistencies. Still, it's possible to validate it further when the data is being used in automation, and all errors can be fixed later. In this step, it's mainly necessary to get the initial data and implement the model.

To further validate that grouping is being done, it’s possible to implement one more control. This is to establish a burndown chart of unmapped assets. Because the goal is to map all assets to application components, it’s assumed that no asset that is not connected to an application component should exist. Thus, we can expect these outcomes.

1. Start establishing context around organisation infrastructure by knowing why each asset has been introduced and its purpose.
2. Discover assets without reason to exist in the infrastructure and thus eliminate them.
3. Discover basic groups that serve as the basis for monitoring groups.
4. Establish a group of assets that we will use as the “target object” in E-ITS as the target for security controls.
5. Increase the security of the infrastructure by eliminating the possibility of anyone running hosts/devices in the network that are unknown and potentially malicious.
6. Establish dependency connections between “Application components” that will be used for service management to manage outages of services and plan for maintenance.
7. Documenting responsibility and attaching each asset to a staff member who maintains it.

This stage of the implementation plan is crucial and will determine the overall project's success. It’s essential to communicate all these outcomes to the management and highlight the immediate benefit of completing this stage successfully.

Communicating with management often and reporting on the findings is expected to be very helpful. This approach is expected to keep morale high and goals clear.

This stage must end with establishing a process that requires administrators or similar staff responsible for maintaining assets to keep the CMDB database Cis “Application” and “Application Component” up to date. Otherwise, it’s necessary to start all over by the end of the iteration.

6.3 Documenting Services and Processes

Having a clear understanding of what services the organisation provides to its clients and to itself is necessary from both the ITSM view and ISMS view for different reasons.

ITSM (Information Technology Service Management) manages the service lifecycle by documenting issues with them, such as outages. During this lifecycle, it’s also necessary to inform clients of planned outages and perform capacity planning Field [28]. For these reasons, it is essential to understand which assets are used to maintain those services.

From the ISMS (Information Security Management System) point of view, especially from E-ITS requirements, it’s necessary to understand which processes the organisation has and what levels of protection requirements these processes need [1].

E-ITS sets three levels of protection requirements: normal, high, and very high. Different controls need to be implemented on the assets to fulfil these requirements.

1. **Establish the list of services that the organisation provides to its clients.**

   The definition of a service is given above, but it’s necessary to clarify it with the attributes of a business service.
   The properties of a service [23] as the author understands them:

   a. Service creates value for the customer.
   b. The owner of the service holds ownership of all risks.
   c. The organisation holds ownership of the costs of maintaining the service.
   d. The organisation’s leadership can discontinue the service.
   e. The owner of the service is in complete control of the processes that are maintained for the service.

A business service must have all these properties. Suppose any of those properties is missing, for example. In that case, the organisation does not have the right to discontinue the service or does not control the processes, and another organisation owns the control. The service in question should be regarded as a hosted service.

By grouping services into two categories and defining specific properties for services, the author achieves the following goals:

1. There is a clear distinction between services that can be documented with processes.
2. A list of services that should have contracts established with partners that contain SLA goals and security requirements can be established. The necessity becomes apparent when precise protection requirements need to be documented.
3. A list of required processes becomes apparent for risk assessment.

We initially define four types of services and differentiate them by the properties we established above and their relationship to the organisation.

- **Hosted service:** A service hosted with the organisation by another entity legally different from the organisation and bought from the organisation.
- **Business service:** A service that the organisation maintains and offers to its clients matches all the above properties.
- **External service:** A service that is bought by the organisation and is hosted somewhere else. In this relationship, the organisation is the client.
- **Supporting service:** A service maintained for the organisation to support processes needed to provide a business service or a hosted service. This service is not sold directly to customers but is a component of the value stream.

Finally, we need to establish the minimum list of attributes that we need to find out about the services.

Minimum attributes for all services that are needed to map out:

1. The name of the service
2. The organisation that is the owner of the service
3. A link to the SLA document contract or requirements or the main business process that drives the service.
4. Contact of the organisation that is responsible for the service.
5. Description of the service

6. Connection to the application that is used for the service (Ideally through a contract or business process object and a data object).

2. Establish and document value streams and processes required to maintain a service lifecycle.

To help in this stage, it is necessary to understand what a process is.

Quote from ITIL v4: *"Definition: Process A set of interrelated or interacting activities that transform inputs into outputs. A process takes one or more defined inputs and turns them into defined outputs. Processes define the sequence of actions and their dependencies."* [23]

In essence, the staff and automation systems of the organisation do specific tasks such that the output of the series of actions taken is the service itself. Processes can be interconnected to other processes, creating value streams.

It's better to abstract processes in the first stage and go into detail in later iterations to avoid getting bogged down in this stage and drowning in detail. In the first iteration, a relationship needs to be established between assets and services through an abstract process CI.

Additionally, CMDB is not the correct place to map out business processes in detail. For this, the organisation should use dedicated process modelling software such as 2c8 or similar that allows users to model business processes in, for example, the BPMN modelling language. CMDB should, however, have process objects that contain the link to the correct process model documentation and other important information about the process.

Minimum list of attributes that were needed for the processes.

1. Name
2. Service (a reference to the service)
3. Primary process (This allows to connect one process to another in a linked list such that it's possible to document the entire value stream)
4. Owner (Owner of the process)

5. Link to model (It should contain a URL to the BPMN model)
6. Link to documentation (A link to external documentation for the process. It’s best if the link is to an index that may contain links to other documentation relating to the process)
7. Applications (reference to the applications that are used in the process)

6.4 Documenting E-ITS Target Objects

If previous stages have gone according to plan and design, it is possible to introduce a new configuration item called “target object”. This is required to adhere to the E-ITS management plan, which requires the organisation to map E-ITS modules to host groups or hosts [20].

It was decided to add a new configuration item instead of reusing a previously created one for one specific reason. We do not add attributes to configuration items that were designed for other purposes in mind, thus adhering to the composition principle. There should be a new CI for each purpose unless adding an attribute to the existing CI makes more sense and describes the existing CI better.

Additionally, it allows more granular access control and simplifies the object's management by keeping attributes per CI to a minimum.

6.5 Plan for Testing, Validation, and Refinement

This chapter outlines the plan for testing, validating, and refining the model once it has been implemented.

1. Testing and validation.

The implementer must test the model in a real scenario to ensure it meets expectations. To this end, it is necessary to select a few of the business services and model them into the CMDB from start to finish with all the Ci records in place.

After this, stakeholders should validate the model, gathering feedback for usability, accuracy, and completeness.

2. **Iterative Refinement.**

Use stakeholder feedback and test results to refine the CMDB data model, making necessary adjustments to improve functionality, usability, compliance support and documentation. Introduce new attributes to CIs where needed but analyse attributes thoroughly so that no attribute is introduced to any CI-s that would not be necessary for all records under the CI.

3. **Training and Documentation.**

Develop user guides and training materials for the CMDB system. Conduct training sessions for all relevant personnel to ensure effective adoption and use of the system. Introduce processes and other tools, such as web forms, in Jira to streamline the maintenance of the CMDB database.

6.6 Measuring Success

It is assumed that the organisation where the CMDB database is being implemented, as described in this thesis, has a working monitoring system that lists all the assets in the database.

Additionally, for the success measurement to work best, the monitoring system must have an auto-discovery or auto-registration feature set up so that the assets that are being monitored are set up automatically.

If these requirements are met, as they were in the organisation where this model is being tested and developed, it is possible to check if these conditions are met:

Conditions for the CMDB data model's asset and service management success.

1. Each asset in the monitoring system is also present in the CMDB database.
2. Each asset in the CMDB database is covered with one application component.
3. Each application component is a part of an application.
4. Each application relates to a service or a process object.

5. The monitoring system is configured to monitor each asset in the CMDB database. It alerts the admins and management staff listed under the application component and application CI records.

Conditions for the success of the E-ITS integration.

1. A “target object” CI record (“sihtobjekt”) has been generated based on the application component.
2. The target object has appropriate relationships with each module from E-ITS, which can help determine which controls need to be implemented.
3. The administrators will receive automatic Jira tickets for each control they must implement. As the number of unresolved tickets reaches zero, it will become possible to track their status. The author will consider it a success when the tickets have been created and assume that the control implementation is outside this thesis's scope.

Concrete measurements that can be done:

1. A burndown list of assets in the monitoring system that is not found in the configuration management databases assets list needs to reach 0 assets.
2. A list of assets that have not been classified under any application component must reach 0 assets.
3. All application components need to belong to an application.
4. Each application is associated with at least one business process with a protection requirement set to adhere to E-Its requirements.

The testing, validation, and refinement phase ensures the unified CMDB data model achieves its intended objectives. By following an iterative approach, the model is continually improved to deliver accurate asset management, compliance, and risk analysis for ITIL and E-ITS frameworks.

7 Implementation and evaluation

This chapter represents the “**Demonstration**” stage of the Design Science Research Methodology (DSRM) process. It discusses the practical application and impact of the implemented model at Organisation A, evaluating how well it met the project's objectives and effectiveness in a real-world setting. The discussion also covers the iterative improvements made to the model, aligning with the DSRM process, and how these enhancements have influenced the overall efficacy and integration of the system within the organisation's existing infrastructure.

The model was implemented using the design and implementation plan described in previous chapters. In Organisation A, the implementation plan was preceded by communication and agreements from the management. The author demonstrated the model's design and implementation plan to the management in detail and got their agreement and support.

The author was allocated some resources, namely a three-person team to help realise the author's design and communicate with the employees to integrate the developed model. This work was done iteratively by designing a prototype data model and testing it against the available data about the organisation. This, too, was done continuously, and the model was iteratively improved, aligning with the DSRM process (Figure 2).

Additionally, all administrators were communicated to prioritise assistance from the CMDB team in collecting the necessary data.

7.1 Collection of Asset Information.

Currently, in the asset information collection process, we are in the "Demonstration" stage of DSRM, where the author demonstrates the implementation plan and the data model in practice, in this section specifically, the implementation plan. This involves assessing the efficacy of the implemented solutions against the defined objectives and identifying areas for further refinement.

In the context of asset information collection, the Design Science Research Methodology (DSRM) was utilised to guide the development and refining of technological solutions within the organisation's IT infrastructure.

DSRM emphasises creating and evaluating IT artefacts intended to solve identified problems, which, in this case, involved the efficient and secure collection and management of asset data.

Asset data collection was streamlined through two primary methods:

- **Integration with Virtualization Systems**: This method provided detailed insights into both virtual and physical hosts, improving the accuracy of asset tracking.
- **Specialised Network Scanning Software**: Utilized to discover connected devices and collect operating system data, enhancing the comprehensiveness of asset information.

Since the organisation has many assets and networks, it was necessary to build a network of proxies that collected information from many different networks and aggregated it to a central server.

Proxied infrastructure design has many positive effects. The main positive attribute is that there is one proxy per network. It is a contained system and is not directly connected to the server. Thus, compromising one proxy doesn't affect the work of the central server or other proxies. The proxy server scans its network and compiles an information package that is then uploaded to a central git repository. The server will process all packages in the central git repository periodically and update the database accordingly. The repository can also be used as a backup system for disaster recovery when all data needs to be rebuilt and other backup methods have failed.

Proxied architecture splits the work so the server is not overloaded with polling tasks. It also allows network administrators to potentially have fewer rules between networks because the scans don't have to go beyond one network subnet. Finally, it allows more granular access management, as each proxy has its management user, thus making it easier to contain risks of account takeover to one network.

A few issues were discovered that needed to be fixed in the next iteration of design and development. It is tough to track the lifecycle of assets in the asset management database. Choosing a uniqueness criterion such as an IP address or the hostname is common, but both can change over the asset's lifetime.

To solve the problem of a unique identifier being needed to track hosts correctly and not insert duplicates in the database, we decided to add a UUID identifier to each host and namespace in Kubernetes.

The UUID will be saved in a metadata file in the server and as a custom parameter in Kubernetes and virtualisation software for virtual hosts. The main rule is that the UUID may only be generated once the server has been created and not modified, even when it is restored from backup.

During this step of the implementation and demonstration, these possible improvements were discovered (Table 4):

Table 4: Results and Discoveries from the First Implementation Iteration

Feature	Description
Decentralized Data Collection	Built a network of proxies for collecting information from various networks, aggregated to a central server.
Enhanced Security and Stability	Each network operates its own proxy, creating a contained system. Compromising one proxy does not impact the central server or other proxies.
Efficient Data Processing	Proxy servers scan networks, compile information packages, and upload to a central git repository. The central server periodically processes these packages.
Disaster Recovery Support	The central git repository also serves as a backup system for disaster recovery, useful when primary data needs reconstruction.
Reduced Server Load	The architecture distributes workload to prevent overloading the central server with polling tasks.

Simplified Network Management	Fewer network rules are needed because scans are contained within single network subnets, simplifying management.
Improved Access Management	Each proxy has its own management user, enhancing granularity in access control and reducing account takeover risks.
Unique Identifier Implementation	Introduced a UUID identifier for each host and namespace in Kubernetes to track assets uniquely and avoid database duplicates.
Persistent UUID Use	UUIDs are generated once upon server creation and are not modified, even when restored from backup, ensuring consistent tracking.

These results demonstrate the proactive steps taken to enhance data management, security, and operational efficiency within the organisation's IT infrastructure.

7.2 Grouping of Assets

In the "Grouping of Assets" stage of the implementation, under the Design Science Research Methodology (DSRM), we are still primarily involved in the "Demonstration" phase, where the methods and models developed are applied and tested in a practical scenario. This stage validates the grouping strategy and its implementation in the organisational context to ensure it effectively meets the predefined objectives. This is crucial for demonstrating the functionality and effectiveness of the asset grouping methods in a live environment, leading up to further evaluation and refinement.

To prepare for this stage, we checked that all assets inserted into the CMDB database were represented in the monitoring system. An automated script was set up to compare two sets of asset information and compile a report showing missing hosts from the CMDB database. The goal was to independently collect the same asset data that the monitoring system contains. Then, at a later point, the reverse would be done to see if the CMDB data collection resulted in found hosts or other assets that have not been put into monitoring.

Once the asset difference has been resolved and the sets are such that the set of assets in the CMDB database contains all assets or more than the set of assets that are in the monitoring system,

Grouping was done based on the application and its components. Each application was identified based on the hosts found and grouped so that each component was grouped as a separate application component object in the database.

This essentially means that, for example, if an application had a database component and a web application component, each was grouped under a different application component object. After this, a directional relationship was set up between the objects describing which application component depended on the other. In this example, the web application would depend on the database. Additionally, dependencies were set up with other application components, such as virtualisation or Kubernetes clusters.

Each application component was connected to an application that would describe the whole. Necessary fields for each application component, such as administrators, managers, descriptions, etc., were filled out. This process aimed to group all assets from the previous step under one such application component group. This group was later used in the monitoring system to group all assets and set up alerts based on the administrator and manager information from the CMDB.

The success of this process was measured by the hosts that were not grouped under any application component. The main idea is that a single asset that is not part of an application should not exist.

The outcome was that the organisation discovered well over a hundred virtual machines that were left ungrouped and thus without a reason to exist. These hosts were long-forgotten lab environments, legacy systems that had not been powered down for one reason or another, or hosts created for migration and then left on and forgotten about. These situations are common in any organisation, but all are rediscovered in the process and powered down.

The organisation already had its first win through this grouping process, where many resources were freed through clean-up. As a second win, the organisation was able to automate the monitoring system alerting setup.

At the end of this process, the author evaluated the outcomes and identified a list of beneficial results that are listed in the table below (Table 5):

Table 5: Beneficial Results of the Grouping of Assets

Outcome	**Description**
Outcome of the Grouping Process	Discovered over a hundred ungrouped and thus unnecessary virtual machines, which were subsequently powered down.
Improved Asset Utilization	Optimized resource utilization by identifying and decommissioning unneeded virtual machines, reducing unnecessary energy and maintenance costs.
Enhanced Security Posture	Deleting unused servers increased security posture by reducing attack surface and by removing potentially unpatched servers.
Increased Operational Efficiency	Automation of monitoring system set-up reduced manual monitoring administration tasks, leading to increased efficiency in operations.
Better Compliance Management	The structured approach to asset management and clear documentation facilitated easier compliance with IT governance and regulatory requirements.
Improved incident management	Greatly increased asset overview and responsibility documentation increased the ability to manage incidents.
Optimized IT Spend	Decommissioning unnecessary assets and automating parts of the monitoring process potentially lowered IT operational costs.

These are only a few beneficial outcomes from the mapping that the author identified. Potentially, many more will be discovered in later iterations, especially when the CMDB data model has been adopted by a wider set of users in the organisation.

7.3 Service Management Improvements

This chapter outlines the practical implementation of service management using the developed CMDB data model as part of the "Demonstration" phase of the Design Science Research Methodology (DSRM).

The CMDB data model was enriched with detailed service management components, utilising the existing structures within Organisation A. This stage was essential for demonstrating how the model operates within an existing framework and validating the integration of service and application components in a live environment.

When the author started building the CMDB, Organisation A already had working service management in place, which meant the author could reuse much of the information already available from another system.

The organisation also documented and modelled all its business processes using the BPMN modelling language in specialised software. This allowed us to add references to the specialised software from the CMDB as URL references.

Four different types of services were identified in the implementation plan. A configuration item was implemented in the CMDB database for each type, and each service item was directly connected to the application CI in the first iteration.

As per the DSRM methodology, a second iteration was conducted after the model was evaluated, and a layer of process items and configuration items representing contracts was added.

The process CI-s and the contract CI-s were necessary to document the level of protection needed per E-ITS requirements. The result of the CMDB data model looked like this.

First, the upper level is the service object containing the attributes documented in the design.

The business service object has a layer of processes attached to it. The process CIs are connected to the application CI-s and contain the "protection requirement" attribute.

In the future, creating a separate configuration item containing more information on risk assessment and protection requirements might be necessary, which would then be referenced in the service records.

The Hosted Service object needed a layer of configuration items between it and the application that would, similarly to the process CI for the business service object, contain the attributes for the protection requirement level. The best candidate for such a CI was the necessary contract between Organisation A and its client organisations. This CI would be, in essence, very similar to the process CI in the CMDB.

The idea behind having separate CIs for contracts and processes is that Organisation A is responsible for its business processes and thus needs to assess the needs for security itself. At the same time, when Organisation A acts as a hosting partner, it essentially hosts an application that is part of someone else's business processes. Thus, Organisation A is not responsible for assessing the security needs. The contracts must detail the security requirements; therefore, the hosting cost is directly affected depending on the necessary security level. More security measures are always also more expensive.

During this implementation process, it was discovered that another layer to describe data would be beneficial. A configuration item that would describe the nature of data that is being processed by an application or stored in a database would be the ideal CI to represent the actual attributes for security, as well as help to describe multi-tenancy situations where an application can process data for more than one client at a time, and the service level agreements for clients, are different. This CI was not envisioned in the original design document and was the direct result of the iterative approach of the implementation process and the research methodology.

The data layer was not implemented by the time of the writing of this thesis because documenting all the data records is an ongoing process and will take several months to complete. It is quite possible that during the implementation of the data CI, more changes to the model will need to be made, such as possibly adding a data CI between the references of applications themselves to document the paths the processing of data takes in the infrastructure.

The current model is set up so that a business service record is connected through a reference edge to the process record, which is then connected to the application CI record.

The hosted service records are similarly set up, with the difference that a Contract CI record is used instead of a process CI record.

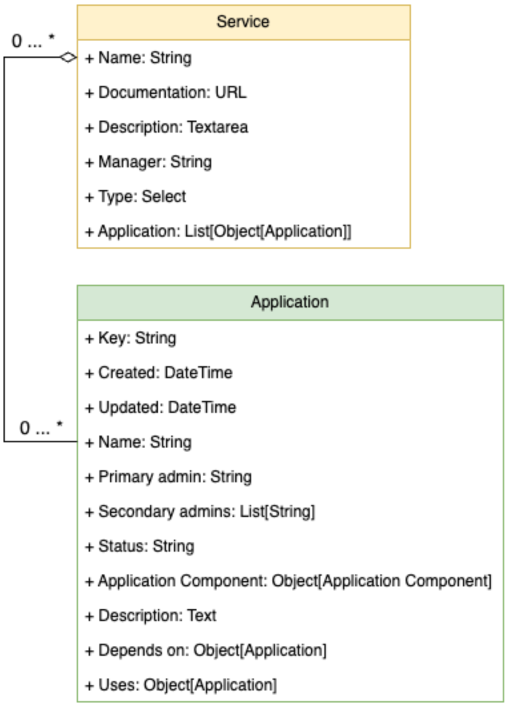

Figure 10: Simplified Service to Application Connection

"Supporting service CI" records are currently directly connected to the "**Application CI"** (Figure 10 **)** records because, in the first iteration, a layer for processes was deemed unnecessary. With the need to manage E-ITS, it has become necessary to add a process layer, but this would require an extensive documentation effort for the CMDB team, which is currently underway. In the meantime, a general medium level for security measures is adopted for all supporting services. As the documentation progresses and security assessment occurs, the level will be adjusted accordingly.

7.4 E-ITS ISMS Management using the CMDB Data Model

E-ITS ISMS management has many processes that must be completed to become compliant. Not all of them require a CMDB or can be automated, so this chapter will only concentrate on asset security controls management.

Once the organisation had mapped all the assets to the assets database and created the application component CIs with all the necessary attributes filled with accurate information, it was possible to start creating the target object CIs.

For each application component, one target object type record was created and mapped one-to-one with the application component. This process was done automatically and heavily relied on the accuracy of the information available.

As mentioned before, E-ITS provides detailed measures to be implemented to protect assets from cyber criminals. These measures are grouped under modules that each have a theme, such as a Linux Unix server for general measures applied to Unix-type operating systems or a module such as Webserver for relevant measures that need to be implemented on applications that can be classified as web servers, such as apache2.

The author created configuration items for each module in CMDB and asked administrators to map relevant modules to target objects. To help the administrators do this effectively and accurately, the author conducted training sessions and a support channel where each admin could ask for assistance.

Following this step in the implementation, the author had to create a means to set security requirement levels on all the target objects. E-ITS requires this done at the process level [20]. The problem discovered is that not all applications in the CMDB are related to processes that would describe the assets accurately. For example, if the process in question is the management of operating systems, it wouldn't describe the contents of the servers and their security requirements accurately if a security assessment were conducted.

This is because many applications set up are introduced to assist a hosted service and are covered by a contract. Suppose the application is a website or a web application hosted for a third organisation. All its components, such as the database, load-balancing, and DDOS protection, are set up only to serve this website. The organisation itself has no business process that maintains the website's contents. As such, no business service can be described around the website itself, only around hosting it.

Therefore, a contract CI was introduced to replace the process CI (Figure 7). The organisation would then be able to set up all the necessary references from the hosted service to the assets.

Similarly, we documented processes and the organisation's business processes and were able to assign protection requirements now accurately to process-level records such as the business processes and contract objects.

With that, the author had all the necessary pieces set up to start the next phase of the E-ITS management process. We could write an automated script to create a ticket for each security method in a module assigned to a target object.

Each ticket created was also assigned to the administrator in charge of the application component on which the target object record was based. The modules have three sets of security methods: the basic level, the average-level methods, and the high-level methods.

By default, all target objects require a normal level of protection. Still, if the business process related to the target object determines high- or very-high-security requirements, tickets will also be created from the high-level methods list.

Implementing all security methods for every target object is unnecessary; this would be much too expensive. Thus, keeping the tickets to a high level to a minimum is preferable. This does not necessarily mean that the assets would be significantly more at risk due to that. It just means expensive services to mitigate DDOS attacks, such as developing high availability for everything, are unnecessary and impractical.

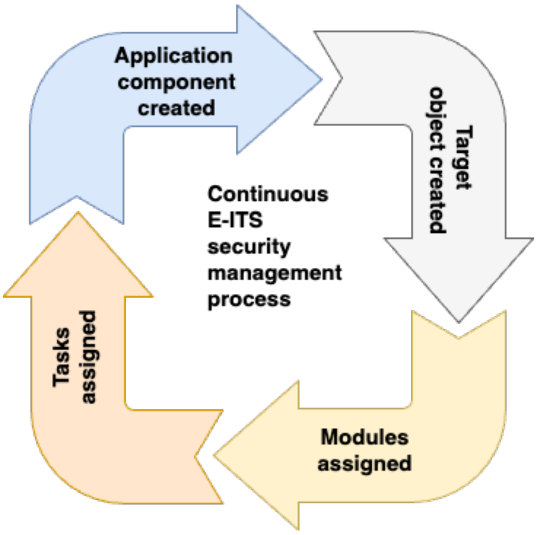
Application component created
Target object created
Continuous E-ITS security management process
Modules assigned
Tasks assigned

Figure 11: Continuous E-ITS Security Controls Assignment Process

At the time of writing this thesis, the process of resolving all the tickets that have been created is still ongoing. It is yet unknown if the outcome will be a success or how long the process will take, but the expectation is that it will yield a good result. Everyone involved is supportive, and the plan is to set up an ongoing process of managing these tickets to achieve a constant state of strong security posture.

This also means that auditors can check the state at any time without warning, and the organisation can show that it is compliant.

8 Observed Results at Organisation A

This chapter targets the 'Evaluation' stage of the Design Science Research Methodology (DSRM); the goal is to assess the implementation of the implemented model within Organisation A.

Designed to integrate IT Service Management (ITSM) and Information Security Management System (ISMS) practices, the model aimed to connect services and business processes to E-ITS target objects. This chapter evaluates the outcomes from the model's deployment to identify necessary changes for the future, list lessons learned, and explicitly answer the initial research questions.

This assessment will refine the model further and optimise its effectiveness in real-world settings. By the end of this chapter, we aim to have conclusively responded to these questions, providing clear insights and actionable recommendations based on the observed results.

8.1 Improved Monitoring Systems

An integration with monitoring systems was planned and created with the implemented CMDB. The goal was to automate asset grouping in the monitoring systems and automate the alerting system by using the information about administrators and their responsibility mapping from the CMDB data.

The goal was successfully achieved, and two separate monitoring systems were integrated with the configuration management database.

Automating monitoring management corresponds or aligns with the ITIL practice and E-ITS controls:

ITIL v4 Practice: Monitoring and Event Management

This integration aligns with ITIL v4's emphasis on the automation of monitoring systems, which includes the need to systematically record and respond to changes within the IT infrastructure. Key activities involve:

- Event Detection: Identifying potential issues through changes in the system.
- Event Logging: Documenting every event systematically.
- Event Review: Analysing events to determine their implications.
- Event Response: Responding to events to mitigate any negative impacts.

E-ITS Requirements Related to Monitoring

The automation also supports E-ITS's focus on comprehensive monitoring for security management, incident response, and operational continuity:

- Security Monitoring: Ensuring continuous log data analysis to detect security incidents (DER.1.M6).
- Performance Monitoring: Continuously monitoring server systems to quickly address any exceedances of established thresholds (SYS.1.1.M23).

Results and Outcomes

Enhanced Accuracy and Efficiency: Automated host, user and alert group creation significantly improved the monitoring systems' effectiveness and manageability.

Expanded Coverage and Improved Awareness: The system now covers thousands of servers across hundreds of groups, enhancing visibility and situational awareness. Alerts are set to notify relevant personnel of performance degradations or service interruptions.

Future improvements and lessons learned: The solution assumed that only administrators and their managers are interested in the servers they are responsible for. While this is true in most cases, there is sometimes a need to add more people to the group. Thus, the model needs to be re-evaluated to implement a way to manage user groups granularly.

8.2 Improved Incident Response and Impact Analysis

The integration of the CMDB was methodically approached to enhance incident response and impact analysis. This integration aimed to leverage detailed mappings of assets to their responsible administrators to enhance the organisation's incident management capabilities and impact analysis processes.

The CMDB was successfully integrated with the incident management systems, streamlining incident response procedures and improving the effectiveness of impact analyses, which is crucial during IT incidents.

This enhancement is crucial because it aligns with specific ITIL practices and E-ITS requirements, emphasising efficient incident management and thorough impact analysis.

ITIL v4 Practice: Incident Management

The ITILv4 book “ITIL® Foundation ITIL 4 Edition” underlines the importance of automated and structured incident management. It details the necessity for:

- **Systematic observation:** Monitoring incidents as they occur.
- **Event logging:** Recording details of all incidents.
- **Event review:** Analysing incidents to assess their impact and causes.
- **Event response:** Responding appropriately to mitigate and resolve incidents.

E-ITS Requirements Related to Incident Response

The E-ITS standards specify the need for effective incident response to maintain security and operational continuity. Relevant sections and practices within E-ITS include:

- **Incident Monitoring and Management:** Ensuring continuous observation and management of incidents to maintain security integrity. This practice aligns with the E-ITS module DER.4: Emergency and Incident Management, which focuses on the capability to manage incidents effectively to ensure business continuity and rapid recovery.

Enhanced Incident Response: The CMDB's integration made incident response more efficient through automated alerts and mappings, reducing the time to respond to and resolve incidents.

Improved Impact Analysis: The ability to analyse incident impacts was significantly enhanced, providing detailed insights into potential and actual effects on IT operations and services.

Operational Continuity: The enhancements contributed to better operational continuity practices, aligning with E-ITS's focus on maintaining operations during and after incidents.

8.3 Simplified E-ITS Management and Security Control Assignment

The deployment of the CMDB facilitated the systematic management of E-ITS by automating the mapping of business processes to specific target objects and applying E-ITS security controls to these objects. This allowed for a structured assignment of security controls, effectively managed by designated administrators documented within the CMDB.

ITIL v4 Information Security Management Practice

The CMDB supports ITIL v4 Information Security Management practices by ensuring that security controls are embedded within service management processes. It balances preventing, detecting, and correcting security incidents, aligning security controls with the organisation’s risk appetite.

E-ITS Requirements Related to Security Management

According to the E-ITS Implementation Manual [20], the E-ITS framework mandates the application of security controls that are specific to the security requirements of target objects identified within the organisation’s infrastructure. The CMDB helps map these target objects with appropriate E-ITS modules, ensuring systematic control management and compliance. This approach supports the E-ITS goal of a structured and sustainable information security management system that adapts to changes in the environment and learns from security incidents.

Results and Positive Outcomes:

- **Enhanced Security Management:** The CMDB's automation and clarity allowed for precise and systematic management of E-ITS controls, improving security management across the organisation.

- **Streamlined Compliance and Security Posture:** The CMDB streamlined compliance processes by aligning target objects with the required E-ITS modules, leading to a more robust and enforceable security framework. This method ensures compliance with E-ITS standards and enhances the efficiency and effectiveness of the organisation’s security management practices, aligning operational efforts with strategic security goals.

8.4 Answers to the Research Questions

Question 1: *What organisational opportunities and potential efficiency gains are possible from managing a central CMDB database for ISMS and ITSM?*

Managing a central CMDB database significantly enhances organisational efficiency by centralising asset information and simplifying management tasks and compliance activities. This centralised approach facilitates better decision-making and optimises IT service management by ensuring all asset information is consistent, up-to-date, and easily accessible. The CMDB's role in improving operational and compliance-related processes has been clearly demonstrated through:

- **Simplified Audit Processes:** Automating asset and process mapping significantly streamlines audit procedures, reducing the time and costs associated with compliance checks.

- **Enhanced Security Management:** By systematically managing security controls, the CMDB improves adherence to the E-ITS security standard, ensuring all assets meet necessary requirements.

- **Improved IT Service Delivery:** Accurate and readily available asset information enhances IT service management, improving service delivery and user satisfaction.

- **Improved Risk Management:** Detailed asset information with mapped relationships between services and assets enables more effective risk assessments, allowing IT teams to address vulnerabilities proactively.

- **Efficient Change Management:** With a comprehensive understanding of asset configurations and dependencies, IT teams can plan and execute changes more effectively, minimising potential disruptions.

- **Improved incident management:** With detailed asset-to-service mapping, the scope and impact of the service interruption can be easily determined.

Question 2: *How does using a centralised CMDB database affect the data quality of the CMDB itself and the efficiency of ITSM and ISMS management processes?*

Utilising a centralised CMDB improves the quality of data stored about IT assets, enhancing the efficiency of IT service management and information security management systems. High-quality, accurate data is crucial for effective risk management, change management, and compliance, as it allows for proactive and informed decision-making.

During the implementation and evaluation of the model in Organisation A, it was apparent that the data quality in the CMDB database improved significantly after it was used in ISMS processes and ITSM processes. The reason is that more departments were now interested in the accuracy and availability of the data.

Data accuracy was significantly more prioritised by the leadership of Organisation A, and CMDB-s importance was significantly more emphasised. This solved the problem of the organisation's cultural barriers, [16], [19] talked by Yuvaraja Chinthapatla and Heru Susanto et.al.

Additionally, it encourages cooperation in the organisation through shared goals and problems. As the CMDB is adopted by more departments in wider areas, such as the financial department for service management quotations for clients, it’s expected that data quality and scrutiny will only increase, leading to even better data quality processes and controls.

8.5 Lessons Learned and Future Work

Lessons learned

While implementing the model at Organisation A, the author primarily focused on technology and solutions, neglecting the importance of communication despite understanding its significance from previous readings [16], [35].

This decision was not due to an oversight but was a strategic choice based on the assumption that a bottom-up approach would require fewer participants and expedite the process. This approach did speed up certain aspects, yet it also led to numerous unforeseen obstacles. These were mainly due to confusion within the organisation and a general lack of interest and cooperation, issues that could have been mitigated by simply communicating more broadly about the project's progress.

Furthermore, the author noted a common misunderstanding among personnel regarding ITIL concepts such as services, business processes, and applications. These terms were often used interchangeably and incorrectly, indicating the need for well-defined terms, complete with examples, before training on the use of the CMDB database and how to document services within it. Without this clarity, the quality and utility of the CMDB data varied significantly.

Future work

The author believes that with the advent of generative AI, such as ChatGPT, it would be feasible to automate the contextualisation process of assets into application components and services. This automation would significantly enhance the data quality and increase the complexity and detail of the CMDB database infrastructure. Additionally, implementing automated risk assessments on the CMDB dataset could further extend the research presented in this thesis.

The domain of configuration management databases is vast, offering extensive opportunities for research, particularly with the current advancements in data processing and automation. Most importantly, a comprehensive case study focused on developing a standardised CMDB data model is needed, which could serve as a robust foundation for further research into related areas.

The research acknowledges certain limitations due to its primary focus on Estonian governmental institutions, which may restrict the generalizability of findings to different organisational contexts. Future studies could explore the model's application across various organisations and service contexts. Furthermore, the evolving IT and security standards landscape will necessitate continuous refinement of the CMDB data model to incorporate new practices and technologies.

9 Summary

The culmination of this thesis aligns with the Design Science Research Methodology's communication phase, documented comprehensively in this final chapter. This research pioneered the integration of the Information Technology Infrastructure Library (ITIL) with the Estonian Information Security Standard (E-ITS) through the strategic use of a Configuration Management Database (CMDB) model. Focused on enhancing IT service and security management within expanding organisations, particularly within the Estonian government's digital framework, the study was driven by the imperative to effectively manage and contextualise IT assets.

From an academic viewpoint, this thesis enriches the discourse on integrating IT service management and security standards, offering a new perspective on managing IT services and information security with a CMDB data model. Practically, it provides actionable insights for organisations aiming to enhance their IT management practices and navigate the complexities of IT infrastructure and security compliance.

The research identified significant operational efficiency and security management enhancements by employing a unified CMDB data model. This model streamlined asset management processes, thereby aiding organisations in maintaining a precise inventory of IT assets. This facilitated compliance with E-ITS and ISO/IEC 27001 standards, bolstering security posture. Implementing the CMDB data model within a prominent Estonian government entity demonstrated tangible benefits, simplifying audit processes and enhancing various business operations. This practical application underlined the model’s efficacy in real-world settings, underscoring its potential for facilitating continuous improvement and systematic risk management, which are vital for robust information security management.

This research marks a progressive step towards refined IT service management and security practices, highlighting the intricacies of managing IT assets in a rapidly evolving digital environment. It sets the stage for future explorations into integrating IT service management with information security, a critical endeavour for securing and optimising digital infrastructures in the contemporary landscape.

10 References

[1] "E-ITS." Accessed: Apr. 01, 2023. [Online]. Available: https://eits.ria.ee/et/avalehe-menueue/eits-v2022-en/

[2] T. Lepik, "Principles of Automatic Verification of the Implementation of Security Measures in the Estonian Information Security Standard," Magistritöö, Tallinn University, Institute of Digital Technologies, Tallinn, 2023.

[3] L. Miller, *Configuration Management and CMDB For Dummies, ServiceNow Special Edition*. Hoboken, NJ: John Wiley & Sons, Inc., 2019. [Online]. Available: https://www.wiley.com

[4] L. Klosterboer, *Implementing ITIL configuration management*. Pearson Education, 2007.

[5] T. Solovjova, *Kuluefektiivse konfiguratsioonihalduse andmebaasiga automaattuvastussüsteemi loomine ettevõtte ITIL protsesside toetamiseks. Creating a Cost Efficient Autodiscovery System with Automatic CMDB Integration to Support ITIL Workflows*. 2016.

[6] M. Brenner and M. Gillmeister, "Designing CMDB data models with good utility and limited complexity," in *2014 IEEE Network Operations and Management Symposium (NOMS)*, 2014, pp. 1–15. doi: 10.1109/NOMS.2014.6838375.

[7] J. vom Brocke, A. Hevner, and A. Maedche, "Introduction to Design Science Research," 2020, pp. 1–13. doi: 10.1007/978-3-030-46781-4_1.

[8] "Cybsis - Eesti infoturbestandardi (E-ITS) rakendamise tööriist." Accessed: May 12, 2024. [Online]. Available: https://eits.raulwalter.com/

[9] "Kordon. The straightforward GRC platform." Accessed: May 12, 2024. [Online]. Available: https://kordon.app/

[10] "E-ITS." Accessed: May 08, 2024. [Online]. Available: https://eits.ria.ee/et/avalehe-menueue/tutvustus/eits-saamislugu

[11] K. Beckers, S. Hofbauer, G. Quirchmayr, and C. C. Wills, "A Method for Re-using Existing ITIL Processes for Creating an ISO 27001 ISMS Process Applied to a High Availability Video Conferencing Cloud Scenario," in *Availability, Reliability, and Security in Information Systems and HCI*, A. Cuzzocrea, C. Kittl, D. E. Simos, E. Weippl, and L. Xu, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 224–239.

[12] M. Sykes and N. Landman, "ITIL and ISO/IEC 27001-How ITIL can be used to support the delivery of compliant practices for Informaton Security Management Systems," *Fox IT Ltd and QT&C Group Ltd*, 2010.

[13] H. Susanto, M. N. Almunawar, and Y. Tuan, "Information Security Management System Standards: A Comparative Study of the Big Five," *Int. J. Electr. Comput. Sci. IJECS-IJENS*, vol. 11, Jan. 2011.

[14] M. Motii and E. Semma, "Towards a new approach to pooling COBIT 5 and ITIL V3 with ISO/IEC 27002 for better use of ITG in the Moroccan parliament," *International Journal of Computer Science Issues*, vol. 14, pp. 49–58, Jun. 2017, doi: 10.20943/01201703.4958.

[15] R. Sheikhpour and N. Modiri, "Mapping Approach of ITIL Service Management Processes to ISO/IEC 27001 Controls," *JOURNAL OF COMPUTING*, vol. 3, pp. 117–124, Jan. 2011.

[16] H. Susanto, M. N. Almunawar, and Y. Tuan, "Information Security Challenge and Breaches: Novelty Approach on Measuring ISO 27001 Readiness Level".

[17] R. Sheikhpour and N. Modiri, “A best practice approach for integration of ITIL and ISO/IEC 27001 services for information security management,” *Indian Journal of Science and Technology*, vol. 5, pp. 2170–2176, Feb. 2012, doi: 10.17485/ijst/2012/v5i3.1.

[18] A. Bakhoff, J. Talik, and K. Loopman, “Eesti infoturbestandardi rakendamine Atlassiani toodetega.” Accessed: Apr. 06, 2024. [Online]. Available: https://blog.twn.ee/et/eits-rakendamine

[19] Y. Chinthapatla, “Mastering Digital Complexity: The Role of Configuration Management Database (CMDB) in Modern Infrastructure Management,” *International Journal of Management IT and Engineering*, vol. 14, pp. 1–8, Mar. 2024.

[20] Riigi Infosüsteemide Amet, “E-ITS v2022 IMPLEMENTATION MANUAL.” 2022. [Online]. Available: https://eits.ria.ee/

[21] M. Johnson, M. Ryder, and J. Sorensen, *IT Service Modeling for the CA CMDB*. Emereo Publishing, 2012.

[22] “Eesti infoturbestandardi rakendamine Atlassiani toodetega | TWN blog.” Accessed: Apr. 06, 2024. [Online]. Available: https://blog.twn.ee/et/eits-rakendamine

[23] ITIL® Foundation, *ITIL® Foundation ITIL 4 Edition*, 4th ed. AXELOS, 2019.

[24] US Department of Defence, “DEPARTMENT OF DEFENSE INTERIM STANDARD PRACTICE CONFIGURATION MANAGEMENT.” US Department of Defence, Mar. 06, 2013.

[25] Y. Chinthapatla, “Empowering IT Infrastructure Management With CMDB,” Jan. 2024.

[26] “Phase C: Information Systems Architectures - Data Architecture.” Accessed: Apr. 09, 2024. [Online]. Available: https://pubs.opengroup.org/architecture/togaf91-doc/arch/chap10.html

[27] “Keep it Flexible: How Loose Coupling Boosts Software Reliability.” Accessed: Apr. 01, 2024. [Online]. Available: https://www.codereliant.io/keep-it-flexible-how-loose-coupling-boosts-software-reliability/

[28] G. Booch, R. A. Maksimchuk, M. W. Engle, B. J. Young, J. Conallen, and K. A. Houston, *Object-Oriented Analysis and Design with Applications*, 3rd ed. Pearson Education, 2007.

[29] “Object Oriented Aggregation.” Accessed: Apr. 09, 2024. [Online]. Available: https://atomicobject.com/oo-programming/object-oriented-aggregation

[30] K. Sandoval, “The Difference Between Tight and Loose Coupling | Nordic APIs |,” Nordic APIs. Accessed: Mar. 31, 2024. [Online]. Available: https://nordicapis.com/the-difference-between-tight-coupling-and-loose-coupling/

[31] “What’s The Difference Between Tight And Loose Coupling? | Clean Commit.” Accessed: Mar. 31, 2024. [Online]. Available: https://cleancommit.io/blog/whats-the-difference-between-tight-and-loose-coupling/

[32] “OOP: Inheritance vs. Aggregation | Baeldung on Computer Science.” Accessed: Apr. 09, 2024. [Online]. Available: https://www.baeldung.com/cs/inheritance-aggregation

[33] “Business Impact and Risk Analysis in ITIL Service Design.” Accessed: Apr. 10, 2024. [Online]. Available: https://blog.masterofproject.com/business-impact-risk-analysis/

[34] “ITIL Change Management Risk Assessment.” Accessed: Apr. 10, 2024. [Online]. Available: https://changemanagementinsight.com/itil-change-management-risk-assessment/

[35] S. Maes, “CMDB BEST PRACTICES: HOW TO SUCCESSFULLY IMPLEMENT CMDB IN YOUR ORGANIZATION,” vol. IFS Blog, Jun. 2023.

Appendix 1 – Non-exclusive licence for reproduction and publication of a graduation thesis[1]

I Gerd Kukemilk

1. Grant Tallinn University of Technology free licence (non-exclusive licence) for my thesis Design and Implementation of a "Configuration management database Compatible with ITIL and E-ITS", supervised by Toomas Lepik
   1.1. to be reproduced for the purposes of preservation and electronic publication of the graduation thesis, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright;
   1.2. to be published via the web of Tallinn University of Technology, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright.
2. I am aware that the author also retains the rights specified in clause 1 of the non-exclusive licence.
3. I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

12.05.2024

[1] The non-exclusive licence is not valid during the validity of access restriction indicated in the student's application for restriction on access to the graduation thesis that has been signed by the school's dean, except in case of the university's right to reproduce the thesis for preservation purposes only. If a graduation thesis is based on the joint creative activity of two or more persons and the co-author(s) has/have not granted, by the set deadline, the student defending his/her graduation thesis consent to reproduce and publish the graduation thesis in compliance with clauses 1.1 and 1.2 of the non-exclusive licence, the non-exclusive license shall not be valid for the period.

Appendix 2 – CMDB CI attributes list

Table 6: CMDB Basic CIs and their Attributes.

Entity	Attribute	Intent
CMDB	Key	A unique identifier for the CMDB record.
	Created	The date and time when the CMDB record was created.
	Updated	The date and time when the CMDB record was last updated.
Service	Name	The name of the service being provided.
	Documentation URL	A link to documentation or more information about the service.
	Description	A detailed description of the service.
	Manager	The user responsible for managing the service.
	Type	The classification or type of service (e.g., IT, customer service).
	Processes	Related processes that are part of or utilize the service.
	Tags	Labels or keywords associated with the service for easier search.
Application	Application Component	The components or parts that make up the application.
	Name	The name of the application.
	Web interfaces	URLs for the application's web-based interfaces or endpoints.
	Status	The current operational status of the application.
	Project lead	The user leading the project for the application.
	Primary admin	The main administrator or employee responsible for the application.
	Secondary admins	Additional administrators or support staff for the application.
	Depends on	Other CMDB items that the application depends on.
	Uses	Resources or services that the application uses.
	Users	People or systems that use the application.
	Description	A detailed description of the application.

	Service	Services associated with the application.
	Tags	Labels or keywords associated with the application.
Process	Name	The name of the process.
	Documentation URL	A link to documentation for the process.
	Description	A detailed description of the process.
	Manager	The user responsible for the process.
	Inputs	Required inputs for the process to function.
	Output	Expected outputs resulting from the process.
Module	Name	The name of the module.
	Module Group	The group or category to which the module belongs.
	Link URL	A hyperlink to more information or the location of the module.
	Intent	The purpose or intended use of the module.
	Responsibility	The party or role responsible for the module.
	Process Version	The version of the process that the module uses or refers to.
	Controls	Security or governance controls related to the module.
Target Object	Name	The name of the target object.
	Module	The module associated with the target object.
	Description	A detailed description of the target object.
Controls	Name	The name of the control.
	Description	A detailed description of the control.
	Category	The classification or category of the control.
	Lifecycle	The stage of the lifecycle that the control is associated with.