IMPLEMENTATION OF ROBOTIC PROCESS AUTOMATION IN THE COST ACCOUNTING PROCESS AT THE KEHA CENTRE

Lappeenranta–Lahti University of Technology LUT

Master's Programme in Industrial Engineering and Management, Innovation and Logistics

2026

Suvi Pieksemä

Examiners: Professor Marko Torkkeli, D.Sc. (Tech.)

Postdoctoral researcher Päivi Haapalainen, D.Sc. (Econ.)

ABSTRACT

Lappeenranta–Lahti University of Technology LUT

LUT School of Engineering Sciences

Industrial Engineering and Management

Suvi Pieksemä

Implementation of robotic process automation in the cost accounting process at the KEHA Centre

Master's thesis

2026

60 pages and 9 figures

Examiners: Professor Marko Torkkeli, D.Sc. (Tech.) & Postdoctoral researcher Päivi Haapalainen, D.Sc. (Econ.)

Keywords: Robotic Process Automation (RPA), cost accounting, public sector, Design Science Research (DSR), Excel-based workflow

This master's thesis examines how Robotic Process Automation (RPA) can be applied to reduce the manual effort required in the KEHA Centre's cost accounting process. Cost accounting is carried out under the Finnish State Treasury's shared cost accounting model, and the selected execution phase relies heavily on Excel-based calculation files and repeated transfers of cost data into the Kieku system. The aim of the study was to design, implement, and evaluate an automation solution that reduces repetitive execution work, improves consistency in the process, and supports the production of more up-to-date cost information.

The study follows a Design Science Research (DSR) approach, and the RPA solution was developed iteratively together with Palkeet. The automation targets a selected execution phase in which financial experts first prepare the required input data in a definition document. After that, the robot runs the macro-based calculations in standardized Excel templates and transfers the resulting cost datasets into Kieku in a predefined sequence. In a typical run, the execution phase includes around 20 separate Kieku transfers, one for each calculation step or template output.

The results show that the implementation reduced repetitive manual execution work and improved the predictability and consistency of the cost accounting run. The expert role remains essential, especially in input preparation, exception handling, and final reconciliation, but automating the execution phase creates a realistic basis for running the cost accounting cycle more frequently than the current twice-yearly schedule when resources allow. The findings also indicate that RPA is particularly well suited to rule-based and standardized public-sector finance processes that follow a shared national model.

TIIVISTELMÄ

Lappeenrannan–Lahden teknillinen yliopisto LUT

LUTin insinööritieteiden tiedekunta

Tuotantotalous

Suvi Pieksemä

Ohjelmistorobotiikan käyttöönotto KEHA-keskuksen kustannuslaskentaprosessissa

Tuotantotalouden diplomityö

2026

60 sivua ja 9 kuvaa

Tarkastajat: Professori Marko Torkkeli, TkT ja Tutkijatohtori Päivi Haapalainen KTT

Avainsanat: Ohjelmistorobotiikka (RPA), kustannuslaskenta, julkinen sektori, suunnittelututkimus (DSR), Excel-pohjainen työnkulku

Tässä diplomityössä tarkastelen, miten ohjelmistorobotiikkaa (RPA) voidaan hyödyntää KEHA-keskuksen kustannuslaskentaprosessin työlään suoritusvaiheen keventämisessä. Kustannuslaskenta tehdään valtion yhteisen kustannuslaskentamallin mukaisesti, ja arjessa työ nojaa edelleen vahvasti Exceliin sekä toistuviin tietojen siirtoihin Kieku-järjestelmään. Tavoitteeni oli suunnitella, toteuttaa ja arvioida automaatioratkaisu, joka vähentää rutiininomaista suorittamista, tekee tekemisestä yhdenmukaisempaa ja tukee ajantasaisempaa kustannustietoa.

Työ toteutettiin suunnittelututkimuksena (Design Science Research, DSR), ja ratkaisua kehitettiin iteratiivisesti yhdessä Palkeiden kanssa. Automatisoitava vaihe alkaa siitä, kun talousasiantuntija on koonnut ja syöttänyt tarvittavat tiedot määrittelydokumenttiin. Tämän jälkeen robotti käynnistää laskentapohjien makroajot, vie laskennan vaiheet läpi ja siirtää laskennan tuottamat kustannustiedot Kiekuun erillisinä raportointidatasetteinä (tyypillisesti noin 20 siirtoa yhtä ajokertaa kohden) ennalta määritellyssä järjestyksessä. Lopullinen täsmäytys ja raportointi tehdään edelleen Kiekun raporttien perusteella.

Tulosten perusteella automaatio vähensi selvästi toistuvaa manuaalista suoritustyötä ja teki suoritusvaiheesta ennustettavamman ja yhdenmukaisemman. Asiantuntijatyö ei kuitenkaan poistu prosessista: esivalmistelu, poikkeusten käsittely sekä täsmäytys ovat edelleen keskeisiä vaiheita. Kun suoritusvaiheen kuormitus pienenee, kustannuslaskentaa on realistisempaa ajaa nykyistä useammin kuin kaksi kertaa vuodessa resurssien niin salliessa. Tiheämpi ajorytmi tukee ajantasaisempaa kustannustietoa ja voi auttaa havaitsemaan täsmäytyksissä ilmeneviä poikkeamia aiemmin. Työ osoittaa myös, että valtion yhteisen mallin mukaisessa, sääntöpohjaisessa ja standardoidussa prosessissa RPA soveltuu hyvin rutiinisuorituksen automatisointiin.

ACKNOWLEDGEMENTS

I would like to thank my supervisors and examiners for their guidance and valuable comments during the thesis process. I am also grateful to the KEHA Centre and Palkeet for the opportunity to carry out this work in a real development setting and for the practical support provided during the implementation. I would also like to thank my family and those close to me for their encouragement and support throughout the process.

In Kouvola 27.05.2026

Suvi Pieksemä

ABBREVIATIONS

AI	Artificial Intelligence
DSR	Design Science Research
HR	Human Resources
IA	Intelligent Automation
IT	Information Technology
KEHA	Employment, Development and Administration Centre
ML	Machine Learning
NLP	Natural Language Processing
Palkeet	Finnish Government Shared Services Centre for Finance and HR
RPA	Robotic Process Automation

Table of contents

Abstract

Acknowledgements

Abbreviations

1 Introduction 12
2 Theoretical framework 14
2.1 The state's common cost accounting model 14
2.2 Process development in public financial administration 17
2.3 Robotic process automation (RPA) 18
2.4 Design science research (DSR) approach 21
2.5 Previous research and practical experiences in automation 23
3 Research methodology 26
3.1 Research approach 26
3.2 Research design and process 26
3.3 Data collection methods 28
3.4 Evaluation and validation 29
3.5 Reliability and validity 30
4 Current state analysis (case: KEHA Centre) 32
4.1 Overview of the KEHA Centre’s cost accounting process 32
4.2 Process flow and responsibilities 34
4.3 Identified manual work phases and challenges 36
4.4 Development needs and automation potential 38
5 Solution design and implementation 40
5.1 Design of the RPA solution 40
5.2 Technical implementation 42
5.3 Process description of the automated step 43
5.4 Testing and iterative development 46
5.5 Evaluation of the solution 47
6 Deployment and results 48

6.1 Deployment of the RPA solution 48
6.2 Results of the implementation 49
7 Discussion 51
7.1 Interpretation of the results 51
7.2 Contribution to theory and practice 53
7.3 Limitations of the study 54
7.4 Future development and further research 55
8 Conclusions 56
References 58

Figures

Figure 1. Overview of the Finnish state's shared cost accounting model (Valtiokonttori 2017).

Figure 2. Allocation phases of the shared cost accounting model (Valtiokonttori 2017).

Figure 3. Comparison of RPA, intelligent automation, and artificial intelligence based on key characteristics (Deloitte 2018; George James Consulting 2024; Rapid Innovation 2024).

Figure 4. Timeline of the research and development process.

Figure 5. Role of the KEHA Centre within the Finnish state cost accounting framework.

Figure 6. Current (as-is) cost accounting process at the KEHA Centre.

Figure 7. Logical structure of the RPA solution.

Figure 8. Target (to-be) process after RPA implementation.

Figure 9. Qualitative comparison of key process characteristics before and after automation.

DECLARATIONS

Turnitin

The originality of this thesis has been reviewed with the Turnitin similarity checking service.

AI usage

The author of the thesis, Suvi Pieksemä, used the following AI-tools during the preparation of the thesis:

1. OpenAI, ChatGPT GPT-5.3
    a. Purpose of use: Structuring of the thesis
    b. Explanation of the use of the tool: Used as a writing aid to improve structure and readability.
2. Microsoft Copilot
    a. Purpose of use: Language polishing and formatting support
    b. Explanation of the use of the tool: Used to suggest minor language edits (grammar, wording) and to support formatting for consistency.

Responsibility

The author, Suvi Pieksemä, takes full responsibility for the content of this thesis and has reviewed and edited the content generated by the possible use of AI tools.

1 Introduction

Public sector organizations, including Finnish government agencies, are under increasing pressure to provide transparent and timely financial information while operating with limited resources and under strict regulation. (Valtiovarainministeriö 2014; Valtiovarainministeriö 2020). Still, a large share of everyday administrative work is done manually. Digitalisation is often discussed as a long-term development objective, but it also affects the practical day-to-day work of public organizations. Many administrative routines still involve repetitive manual work, so even modest process improvements can make the work more predictable and reliable (Palkeet 2024).

Cost accounting is one area where high expectations meet very real resource constraints. In the public sector, cost accounting supports financial management, decision-making, and resource allocation by turning operational activities into comparable cost information. In Finland, agencies follow the State Treasury's shared cost accounting model, which provides a clear framework and standardized coding (Valtiokonttori 2017). Even with that framework in place, the everyday execution can still be surprisingly manual, repetitive, and slow.

At the KEHA Centre, the cost accounting cycle includes several steps where data is prepared and checked in Excel and then transferred to the Kieku system for reporting. In practice, this means repeated rounds of making sure the right fields are filled, formats match what the templates expect, and entries follow the required coding structure before anything can be submitted. This kind of work is detail sensitive. It takes time, it interrupts other tasks, and it also creates room for small errors from copying and pasting or manual input that can later show up in the final figures. As decision-making in the public sector has become more information-driven, the importance of timely and reliable cost information has increased (Valtiovarainministeriö 2020). The process and system descriptions presented in this thesis are based on State Treasury guidance, relevant legislation, Palkeet materials, KEHA Centre process documentation, and observations made during the design, testing, and implementation of the RPA solution.

In this situation, Robotic Process Automation (RPA) is a practical option because it does not require major changes to the underlying systems (Deloitte 2018). Instead, an RPA robot can automate repetitive, rule-based tasks by following the same steps a user would take in

existing applications. This fits the KEHA Centre's operating environment rather well, since the work mainly takes place between Excel templates and Kieku, and changes to shared platforms are typically slow due to governance and regulatory requirements.

I design and implement an RPA solution for one selected step in the KEHA Centre's cost accounting process. It simply automates a repetitive execution phase where cost data is processed in Excel and then transferred into Kieku for reporting. The robot carries out the repeatable steps in the same way each time, which reduces manual effort and avoidable errors and leaves experts more time for preparation, validation, and interpretation.

To build the solution in a way that stays systematic and academically grounded, I use the Design Science Research (DSR) approach. DSR fits this work because it supports both building something concrete and testing whether it works in practice. The main research question of the study is: How can Robotic Process Automation be implemented in the KEHA Centre's cost accounting process to improve efficiency and accuracy? This main question is addressed through the following supporting sub-questions:

- What are the main manual work phases in the current cost accounting process?
- How can RPA be used to improve the efficiency and accuracy of this process?
- What kind of benefits and challenges emerge from the implementation?

The thesis follows the work from context to implementation and results. The theoretical background comes first (cost accounting in the Finnish state context, process development, RPA, DSR approach and the previous research). After that, I describe how the study was carried out and then move into the KEHA Centre case, where I map the current workflow and the main problem areas. The final chapters describe the solution design, deployment, and results, and then discuss what the case means in practice and in relation to earlier work. The thesis closes with conclusions that bring the main findings together and point out what they mean for the KEHA Centre and for similar public-sector finance processes.

2 Theoretical framework

Before moving into the case analysis and the implementation work, I introduce the concepts that the solution builds on. The KEHA Centre's cost accounting process is shaped by the State Treasury's shared cost accounting model, by public-sector requirements for transparency and accountability, and by the practical constraints of shared systems such as Kieku. For that reason, this chapter first explains the shared model, then discusses process development in public financial administration, and introduces Robotic Process Automation. I also describe the Design Science Research (DSR) approach used in this thesis and end the chapter with a short review of earlier research and practical experiences related to automation in the public sector.

2.1 The state's common cost accounting model

For the KEHA Centre, the State's shared cost accounting model works as a practical framework that shapes how cost information is produced and reported. It determines what is calculated, how allocations are carried out, and what kind of datasets end up in Kieku. This level of standardisation also makes automation realistic. When the rules, coding structures, and outputs are stable, a robot can execute repetitive steps reliably as long as the work remains aligned with official guidance and the causality principle (Valtiokonttori 2017; Deloitte 2018).

The Finnish state's shared cost accounting model is a standardized framework developed by the State Treasury. Its role is to make cost information more comparable and usable across agencies by defining a common structure for how costs are tracked, allocated, and reported. In practice, it turns everyday operational activities into cost information that can support decision-making, operational management, and resource allocation within and between state organizations (Valtiokonttori 2017).

The model is grounded in legislation. Finnish legislation requires state agencies and ministries to monitor the effectiveness, efficiency, and productivity of their operations. This is one of the reasons why the common cost accounting model is used across the state administration (Valtion talousarviolaki 423/1988, section 12; Valtion talousarvioasetus

1243/1992, section 55). Alongside these legal requirements, the shared model supports the management of accounting units by providing up-to-date and comparable information on the cost structure of operations (Valtiokonttori 2017).

At the core of the shared model is a common tracking object structure used across the Finnish state administration, where the functional tracking object is the central element. The model then defines how costs from support functions and paid absences are allocated to core and steering functions, enabling agencies to calculate the total costs of core operations. Agencies can extend allocations further (for example to outputs, projects, or services) depending on their own information needs and objectives. When the coding structures are applied consistently, the resulting cost information can be processed and compared in a standardized way across agencies. Figure 1 gives a high-level view of the shared cost accounting model (Valtiokonttori 2017).

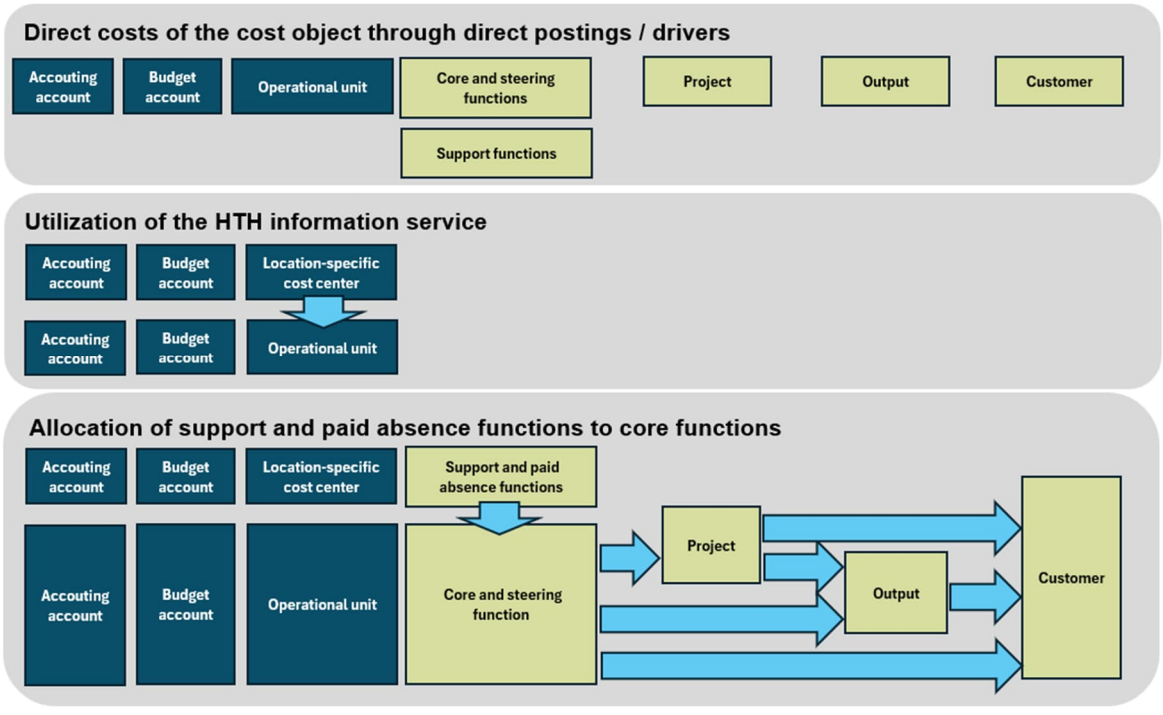

Figure 1. Overview of the Finnish state's shared cost accounting model (Valtiokonttori 2017).

The State Treasury's guidance also defines a minimum model, where specific activity categories are allocated to core and steering functions through five phases. This gives agencies a shared baseline while still leaving room for unit-specific detail, depending on internal processes, information needs, and accounting objectives. Agencies may use the model with varying levels of detail, but the allocations still need to follow the causality principle. The amount of work involved can also differ depending on the allocation methods and rules used. Figure 2 shows the five allocation phases (Valtiokonttori 2017).

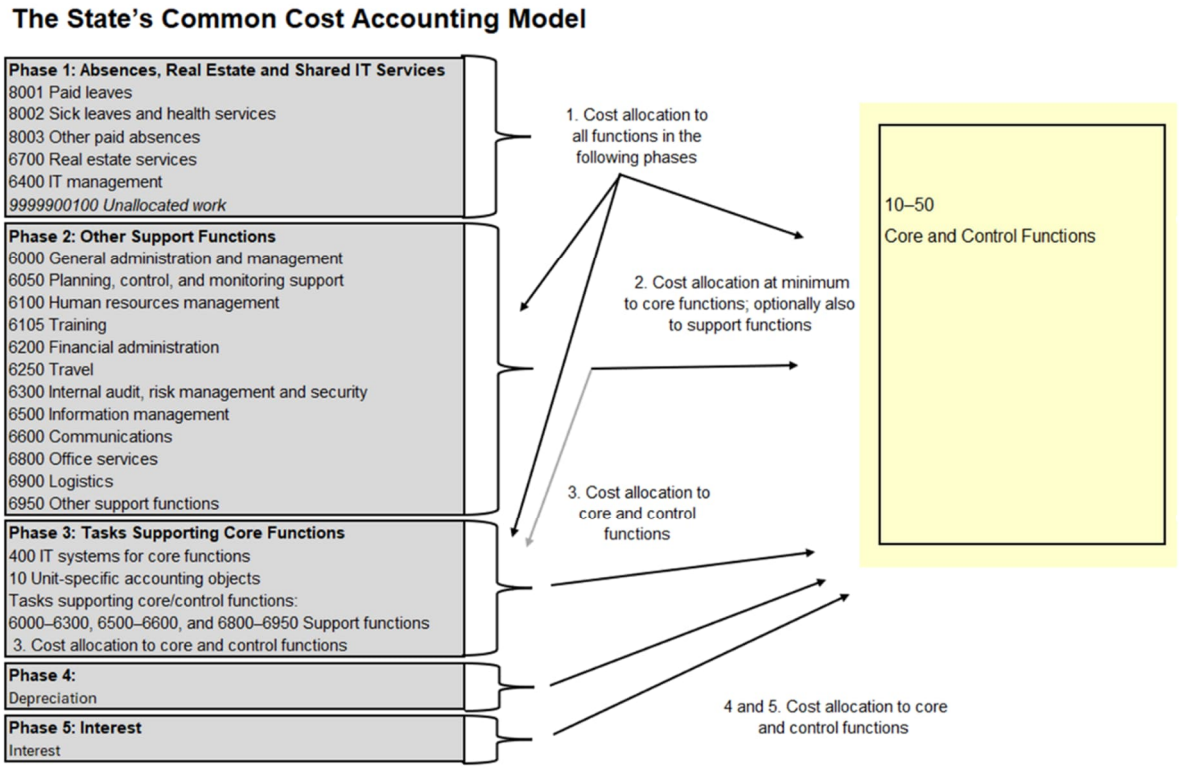

Figure 2. Allocation phases of the shared cost accounting model (Valtiokonttori 2017).

The shared model creates a solid foundation for transparent and comparable cost information across the state administration. Still, the value of that information depends on how efficiently the work is organised and how reliably the routines produce high-quality datasets. Process development addresses this practical side of the model. Since the model is applied as part of everyday work, agencies need processes that are consistent enough for routine use but still flexible when requirements change.

2.2 Process development in public financial administration

In the Finnish state context, process development goes hand in hand with reforms in financial administration and digitalisation. The shared cost accounting model does not run itself; its value depends on the routines that produce the inputs, run the cycle, and turn results into reporting. This is why many improvements focus on redesigning the workflow and clarifying responsibilities rather than changing the model. Clear, well-documented processes also support reliable financial information for decision-making and reporting, and they make it easier to apply automation where the work is repetitive and rule-based (Valtiovarainministeriö 2020).

Process development in the public sector also has a different starting point than in private organisations. Legislation, the state budget, and public service obligations set boundaries for what can be changed and how quickly. In financial management, work is further shaped by guidance from the State Treasury and the Ministry of Finance, and by shared national systems such as Kieku (Valtiovarainministeriö 2020).

The aims of public-sector processes extend beyond productivity and economic efficiency alone. They also need to support openness, equal treatment, and the practical fulfilment of legal and administrative responsibilities. The Public Governance Strategy (2020) notes that "public governance builds the sustainable daily life of the future and a working and secure society under all conditions," which in many ways reflects the broad role of public administration in society. Since work is often coordinated across several stakeholders and organizations, clear and well-defined processes help support everyday operations (Valtiovarainministeriö & Kuntaliitto 2020).

In Finland the financial administration development aims to improve efficiency, consistency, and the overall quality of information across public administration. In practical terms, this means better use of resources, less manual work, and reporting that is more comparable between organisations. The goal is to support decision-making with timely, transparent, and comparable financial information for both agency-level management and government-level steering (Valtiovarainministeriö 2014).

Many priorities are set through shared state-level policies, including financial management policy and public governance policy. These emphasise active financial management,

evidence-based management, customer orientation, and societal impact. When these priorities are taken seriously, process development becomes more than administrative housekeeping. It is also strategic work that affects the public sector's capacity to renew and the reliability of everyday operations (Valtiovarainministeriö & Kuntaliitto 2020).

Responsibility structures and implementation realities also matter, because they determine who can change what and how improvements move from plans to routine work. The Ministry of Finance and the State Treasury guide Finnish state financial administration. The Ministry is responsible for broader strategic direction, whereas the State Treasury supports practical development work and implementation in everyday operations (Valtiovarainministeriö 2014; Valtiokonttori 2017). In everyday operations, Palkeet is involved in shared finance and HR services as well as in the support and maintenance of related systems (Palkeet 2024). At the same time, development work is often carried out through different programmes and initiatives linked to financial strategy, digitalisation, and automation. These programmes usually define practical goals, schedules, and responsibilities for the implementation work (Valtiovarainministeriö 2014). Agencies typically contribute through pilots, working groups, and feedback channels. Agencies usually participate through pilot projects, different working groups, and practical feedback from everyday work. Shared systems such as Kieku and the common cost accounting model are meant to make reporting practices more consistent across the state administration (Valtiokonttori 2017). At the same time, agencies still need some room to apply these practices according to their own operational needs (Palkeet 2024).

2.3 Robotic process automation (RPA)

In the KEHA Centre case, the strongest automation candidate is a rule-based workflow that moves from Excel to Kieku, which makes RPA a natural starting point. Public sector agencies are under sustained pressure to improve operational efficiency, reduce manual workload, and respond to tighter reporting and governance requirements. Often the rules are clear but applying them consistently still requires a large amount of repetitive execution work. Robotic Process Automation (RPA) automates rule-based tasks by working through existing applications, so it can often be implemented without changing the underlying information systems (Deloitte 2018). Intelligent automation builds on traditional RPA by

combining it with technologies such as machine learning and natural language processing. As a result, it may also support more demanding tasks and assist experts in practical decision-making situations (Rapid Innovation 2024).

In the public sector, these technologies have been applied especially in finance and HR tasks where large volumes of data are processed with established rules or transferred between systems. Typical examples include cost allocation routines and travel expense processing. A concrete Finnish example is Palkeet's automation solution that supports the processing and transfer of Excel-based information into the Kieku system. Solutions of this kind can improve data quality, reduce errors, and free up expert capacity for work that requires interpretation and judgment (Palkeet 2024).

Figure 3 compares RPA, intelligent automation, and artificial intelligence through characteristics that matter in public-sector operations, such as rule-based execution, data requirements, and the level of autonomy.

Feature	Robotic Process Automation (RPA)	Intelligent Automation (IA)	Artificial Intelligence (AI)
Purpose	Automate rule-based, repetitive tasks	Combine RPA and AI capabilities to automate more complex and decision-based processes	Simulate human cognitive functions and learn from data
Learning Ability	No learning (follows fixed logic)	Includes learning components such as natural language processing (NLP) and machine learning (ML)	Capable of learning, adapting and improving from data
Use Case Scope	Routine tasks (data entry, copy-paste, system integrations)	Complex tasks (document analysis, approvals, smart customer service)	Advanced tasks (prediction, natural language understanding, image recognition)
Adaptability	Static and rule-driven	Semi-adaptive (adjusts to workflows, user inputs, and exceptions)	Highly adaptive (continuously improves and evolves)
Technology Components	Desktop bots, macros, workflow automation	RPA + AI technologies (e.g. NLP, ML)	Neural networks, ML models, generative AI, advanced analytics
Typical Use Cases	Filling forms, transferring data between systems	End-to-end case processing, chatbots, automated decision support	Forecasting, pattern detection, autonomous systems
Value in Public Sector	Speeds up manual work, reduces human error	Improves service delivery, reduces processing time, enhances process transparency	Enables strategic decision-making and predictive governance

Figure 3. Comparison of RPA, intelligent automation, and artificial intelligence based on key characteristics (Deloitte 2018; George James Consulting 2024; Rapid Innovation 2024).

In practical financial administration, the main benefits of RPA are shorter processing cycles and less manual input, especially in repetitive validation and system-to-system transfer tasks. In the KEHA Centre's context, this refers to running standardized Excel calculations and moving the resulting datasets into Kieku with the same checks repeated across each dataset. Because the robot executes an identical sequence each time, RPA supports processing consistency and offers a cost-effective way to automate rule-based work without changing existing systems. When the task is stable and rule-based, RPA can often be implemented quickly, and it is commonly used for data transfers, verification tasks, and file handling (Deloitte 2018; Rapid Innovation 2024). In the KEHA Centre case, the fit comes from standardized templates and predefined allocation rules, which keep the inputs and outputs stable for automation.

RPA has clear limits. It cannot make independent decisions, and it needs explicit handling for exceptions. RPA works best for clearly defined, repetitive processes, which limits its usefulness in more complex situations. For this reason, RPA is often seen as one step in a broader automation journey that may later include more advanced technologies such as machine learning and artificial intelligence. These limitations are especially relevant in public sector environments, where exceptions and regulatory interpretations often require human judgment (Rapid Innovation 2024; George James Consulting 2024).

The work concentrates on Robotic Process Automation (RPA) in the KEHA Centre's cost accounting process and targets manual effort in processing Excel-based cost data and transferring the resulting datasets into Kieku. Although the study stays focused on RPA, Palkeet and other state-level actors are also exploring broader forms of intelligent automation for more complex processes. In this context, targeted RPA works as a practical first step. It delivers value in stable, rule-based tasks and can later be complemented with more advanced capabilities where they are justified (Palkeet 2024; Rapid Innovation 2024).

Because the work involves building, testing, and evaluating a solution in a real setting, the study needs an approach that supports structured development. Design Science Research

(DSR) fits this purpose and offers a well-established framework for designing and assessing practical artifacts such as automation solutions.

2.4 Design science research (DSR) approach

Design Science Research (DSR) is an approach where real-world problems are addressed by designing, implementing, and evaluating concrete artifacts such as processes, systems, or technical solutions. At the same time, it combines scientific inquiry with practical development work and aims to produce both a useful solution and knowledge that can later be communicated and built on further (Hevner et al. 2004). In this thesis, DSR provides the methodological basis for developing and testing the automation solution used in the KEHA Centre's cost accounting process.

DSR is particularly well suited to public-sector development, where improvements must be made within the constraints of legislation, official guidance, and technical standards. It also provides a relatively structured way to design and evaluate new approaches such as automation. At the same time, the development work remains connected to practical operational needs and accountability requirements (Hevner et al. 2004; Gregor & Hevner 2013; Hardy & Williams 2011).

The thesis follows Peffers et al.'s (2007) Design Science Research process model, which consists of six consecutive phases:

1. **Problem identification and motivation:** Define the research issue as well as the underlying practical need.
2. **Define the objectives for a solution:** Establish the goals for the solution, which will serve as the basis for evaluating the artifact.
3. **Design and development:** Develop the actual artifact, such as a process or an automation solution.
4. **Demonstration:** Test the artifact function in a real or simulated environment.
5. **Evaluation:** Compare the performance of the artifact with the objectives set.

6. **Communication:** Document the research results for both academic and practitioner audiences.

Using this type of structured process makes the development work easier to manage in practice, and it also makes the design choices and evaluation steps easier to justify academically.

Another widely used foundation for DSR is the framework by Hevner et al. (2004), which summarises seven guiding principles for design science research:

1. **Design as an artifact.** The research must produce a concrete solution, such as a model or system.
2. **Problem relevance.** The problem being addressed must be relevant to practical operations.
3. **Design evaluation.** The developed artifact must be systematically evaluated, for example through testing or expert feedback.
4. **Research contributions.** The research should generate new insights, principles, or practices.
5. **Research rigor.** The theoretical and methodological foundation of the study must be reliable and well justified.
6. **Design as a search process.** Design is viewed as an iterative process that explores and tests alternative solutions.
7. **Communication of research.** The research must be communicated to both academic and practitioner audiences.

These frameworks help keep the development work both practical and academically rigorous. In this study, Peffers' process model structures the development and evaluation phases, and Hevner's principles are used as a guide for research quality, relevance, and transparent reporting.

A central part of DSR is that the artifact is built and tested in an actual setting. Evaluation can be quantitative (for example, reductions in processing time or error rates) or qualitative (for example, user feedback or expert assessments). In the public sector, evaluation is often supported through methods such as document analysis, pilot testing, and comparative

assessment. This helps ensure that the designed solution meets operational objectives and that the findings are credible beyond a single design iteration (Gregor & Hevner 2013; Hardy & Williams 2011).

DSR brings scientific reasoning and practical design work together so that the outcome has academic merit and concrete usefulness for the organisation being developed (Gregor & Hevner 2013).

2.5 Previous research and practical experiences in automation

Earlier research suggests that automation can ease manual effort in administrative and financial work, but the gains are not automatic. They depend on whether the process itself is suitable for automation and on how carefully the solution is introduced (Syed et al. 2020; Aguirre & Rodriguez 2017). Public-sector examples describe much the same pattern. In practice, RPA has mainly been applied to recurring tasks where the same steps are repeated in shared systems and routine reporting work, often with fairly standardised data structures (Asatiani & Penttinen 2016; Deloitte 2018; George James Consulting 2024; Rapid Innovation 2024). This is close to the execution phase examined in this thesis, where calculations are run in Excel and the results are then transferred into Kieku.

A fairly consistent message in the RPA literature is that software robots are most useful in work that follows clear rules, uses structured data, and repeats the same steps over time. Syed et al. (2020) describe RPA in simple terms as software that carries out routine tasks through existing systems in a way that resembles what a user would normally do. What stands out in their discussion is not only the basic idea itself but also the kinds of effects that are reported most often: less manual work, smoother execution, and more consistent handling of routine tasks. Aguirre and Rodriguez (2017) reach a similar conclusion and link RPA especially to routine back-office work with predictable outcomes. Their discussion highlights the same kinds of benefits that appear throughout the literature: faster execution, higher productivity, and fewer errors when the process is stable enough to support automation.

The same point comes through when earlier studies discuss process standardisation and clearly defined workflows. Asatiani and Penttinen (2016), for example, describe RPA above

all in repetitive routine processes where tasks follow explicit rules and system interactions stay predictable. That description fits the KEHA Centre case rather well. The selected step is not a loosely defined knowledge task but a recurring execution phase built around Excel calculations and standardised Kieku reporting routines.

At the same time, the literature leaves little doubt that automation still needs to be governed and monitored rather than simply switched on and left to run. Willcocks, Lacity, and Craig (2015) discuss this through questions of governance, organisational control, and clear responsibility for the automated work. Bagheri and van de Wetering (2024) make a closely related argument: technical implementation alone is not enough if the organisation is not ready, the governance practices are weak, or the process has been selected poorly. This is also visible in the KEHA Centre case. Financial experts still prepare the source data, review the outputs, and make sure the final reporting remains aligned with existing accounting practices.

A Finnish practical example can be found in Palkeet's automation work in routine public-sector administration. According to Palkeet (2024), RPA has in some cases been combined with selected intelligent capabilities to make routine processes faster and more reliable, and the reported effects are not limited to speed alone. The material also points to changes in how employees experience the quality and predictability of their work. The solution developed in this thesis follows the same broad logic, although in a narrower setting: it automates the handling of Excel-based cost accounting data and transfers the results into Kieku at the KEHA Centre.

International studies also show that public-sector automation comes with both clear opportunities and familiar constraints. Hardy and Williams (2011) note that automation in this context has to work within complex regulation, legacy systems, and strong transparency requirements. Even so, those constraints do not rule automation out. Rather, they shape the conditions under which it can be useful. A similar point appears in George James Consulting's (2024) discussion from a more practice-oriented perspective: there are many situations in government work where carefully scoped RPA is already enough, without bringing in more advanced artificial intelligence.

Seen together, these studies and practical examples point in much the same direction. RPA seems to be most useful in public-sector finance work when the process is already stable

enough to be repeated in a consistent way, the data structures are clear, and responsibilities remain visible rather than disappearing behind the technology. The literature repeatedly links automation with shorter processing times, less manual work, and more consistent handling of data and reporting tasks (Syed et al. 2020; Deloitte 2018; Palkeet 2024). The same discussion also keeps circling back to rather practical questions: how traceability is maintained, who remains responsible for the work, and where the limits of automation become visible. Those questions do not go away in implementation. If anything, they become more important in situations where the process still calls for interpretation, user involvement, change management, or human judgment (Willcocks et al. 2015; Isaac 2025; Aguirre & Rodriguez 2017; George James Consulting 2024; Rapid Innovation 2024). In other words, public-sector RPA is presented less as a stand-alone technology and more as a controlled way to make repetitive work more manageable within existing organisational and system constraints.

What earlier studies have discussed in detail are the general benefits of RPA and the challenges related to adoption and implementation. Much less attention has been given to public-sector cost accounting processes where automation has to work inside shared reporting structures, formal governance requirements, and highly standardised financial routines. This thesis speaks to that narrower question through a Finnish case in which RPA is introduced in one execution phase of cost accounting under the State Treasury's shared model. The KEHA Centre case therefore adds a concrete example of how automation can be applied in a regulated financial management environment and provides the starting point for the process analysis and implementation described in the following chapters.

3 Research methodology

This chapter describes how the study was carried out and how the solution was evaluated. It explains how Design Science Research (DSR) guided the work, what material and observations were used during development, and how reliability and validity were considered in a practical case setting.

3.1 Research approach

I apply the Design Science Research (DSR) approach because the main goal is practical. It is to develop a working RPA solution for the KEHA Centre and to assess whether it actually improves the execution phase of cost accounting. At the same time, the work needs to be reported and evaluated in a research-driven way. DSR fits this combination well, because it links theory and design choices to hands-on development and requires that the artifact is evaluated in its intended context (Hevner et al. 2004; Peffers et al. 2007; Gregor & Hevner 2013).

In practice, I rely on two established DSR frameworks. Peffers et al.'s (2007) six-phase process model provides the structure for how the work moved from problem identification to design, demonstration, evaluation, and reporting. Hevner et al.'s (2004) seven principles, in turn, help ensure that the study remains both relevant to the organisation and rigorous in how the artifact and findings are justified and communicated.

Peffers' process model and Hevner's principles provided a practical structure for the work and supported iterative development. The study moved from defining the problem and objectives to designing, testing, and evaluating the automation solution in the KEHA Centre context.

3.2 Research design and process

The study progressed as an iterative design and test cycle. I moved back and forth between understanding the process, implementing the robot, and checking whether the solution

behaved correctly in the target workflow. This was important because the automation depends on small but critical details, such as input formats, template behaviour, and how Kieku accepts datasets. To keep the work systematic, the overall process is described using Peffers et al.'s (2007) six-phase DSR model.

I started by mapping the current cost accounting workflow and identifying where manual effort accumulates and where errors are most likely to occur. Based on that analysis, I defined the objectives for the automation with a clear focus on the execution phase: reducing repetitive work, improving consistency of system entry, and lowering the risk of errors caused by copying and pasting or manual input. The RPA solution was then designed and developed together with Palkeet so that the implementation would match the real templates, access rights, and Kieku environment. Once a working version was available, the solution was demonstrated through test runs in a simulated operating setup, using last year's figures as test material in a test environment that mirrors the KEHA Centre workflow.

After the demonstrations, I evaluated the solution against the objectives using both technical observations and expert input. The evaluation considered execution time (a typical automated run of the execution and transfer phase took about 30 minutes compared to the earlier multi-day manual cycle), the absence of run-time errors in successful runs, and correctness checks based on totals across the chain (template totals and report-level sums compared against the transfer files). In addition, I used qualitative feedback from KEHA Centre financial experts collected through informal discussions in internal meetings and joint sessions to understand how the solution fits the workflow in practice. I then documented the design choices, the implementation, and the observed outcomes so that the results can be understood both in an academic context and by practitioners who work with similar public-sector finance processes.

Because the work was iterative, insights from each test round directly shaped the next development steps. For example, I tightened validation rules, improved error handling, and adjusted the execution sequence. This made it possible to arrive at a solution that is both technically stable and practically usable in the KEHA Centre's cost accounting setting.

Figure 4 shows the Design Science Research process used in the study as a timeline of research and development work.

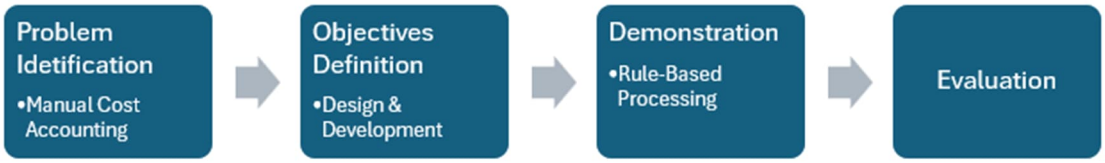

Figure 4. Timeline of the research and development process.

The work started with problem identification grounded in the current state of cost accounting at the KEHA Centre. This phase focused on challenges related to manual, Excel-based work, such as high workload, the risk of calculation and data-transfer errors, and limited process transparency. Based on this problem context, the objectives of the study were defined, with an emphasis on designing an automation solution that supports cost accounting activities within existing system constraints.

In the objectives definition phase, the focus shifted from analysis to concrete design choices. I specified the functional requirements for the RPA solution and designed rule-based processing logic aligned with KEHA Centre practices and the Kieku system. In the demonstration phase, the artifact was implemented and applied in the KEHA Centre context by automating the selected cost accounting tasks and the related data transfer steps using predefined rules.

The evaluation phase assessed the solution in its organisational context. Alongside execution time and technical correctness, the evaluation considered qualitative changes in work, process control, transparency, and error risk. These phases show how the study combined a DSR structure with practical development at the KEHA Centre, moving from problem identification to solution evaluation.

3.3 Data collection methods

The empirical material for this thesis comes from two main sources: (1) documents that define how cost accounting and the workflow from Excel to Kieku should be carried out, and (2) observations from iterative development and test runs of the robot. I used these inputs to understand the current process, to translate rules and requirements into automation logic,

and to capture what changed during implementation. The description of the cost accounting system and workflow in this thesis is based on State Treasury guidance, relevant legislation, Palkeet materials, and case-specific process documentation from the KEHA Centre.

The document analysis covered guidance and supporting material from the State Treasury, Palkeet, and the KEHA Centre. These documents describe the shared cost accounting model, the required coding and allocation principles, and the practical constraints that matter for automation (for example, what the Excel templates produce and how the outputs need to be entered into Kieku).

Alongside document analysis, I gathered observations during iterative testing in the design and development phases. Each test round showed whether the robot performed the intended steps (macro runs, validations, and Kieku transfers), whether the produced datasets matched expectations, and where the process broke due to input variations or edge cases. These findings were used immediately to refine the automation logic and to make exception handling more reliable.

To complement the technical tests, I collected feedback from financial experts who work with the process. Their input helped me assess whether the robot's behaviour is understandable and trustworthy in day-to-day use, and whether the solution actually reduces the most repetitive execution work without weakening expert control over validation and reconciliation. Together with test observations, this qualitative feedback supports the practical relevance of the solution and strengthens the interpretation of the evaluation results.

3.4 Evaluation and validation

The evaluation of the RPA solution focused on verifying its functionality and accuracy and on assessing how it contributed to process improvement. The evaluation was guided by Design Science Research principles, which emphasize both practical applicability and research rigor in the KEHA Centre context (Hevner et al. 2004).

Because the work sits at the intersection of Excel templates, system entry in Kieku, and expert judgment in reconciliation, I treated evaluation as both a correctness check and a usability check, so that the solution can be trusted in routine use.

I assessed the solution through test runs in a real and simulated operating environment by checking whether the robot reliably completed the intended steps and produced correct transfers into Kieku. I also compared the automated execution with the earlier manual approach in terms of time consumption, error risk, and consistency across the dataset sequence.

Alongside these observations, I used feedback from key users and financial specialists to evaluate usability and users' confidence in the solution. This helped me judge whether the solution fits the workflow in practice, not only whether it works technically.

Validation was supported through repeated cycles of testing, adjustment, and re-testing. The purpose of these cycles was to confirm that the same inputs lead to the same outcomes, that the robot handles expected variations in the data, and that any exceptions are identified clearly for expert review.

3.5 Reliability and validity

This thesis combines development work with an organisational case, which makes reliability and validity central to how the results should be read. Reliability refers to whether the development and evaluation steps are described clearly enough that the work could be repeated in the same setting. Validity concerns whether the evaluation reflects the objectives of the study and the realities of the KEHA Centre's cost accounting workflow.

Reliability was supported by following a documented DSR process and by recording the key design choices, test rounds, and fixes made during development. The automation logic itself is based on standardized inputs and explicit rules (the definition document, agreed Excel templates, and a fixed Kieku transfer sequence), which reduces random variation between runs. Repeating the same test runs also helped us to confirm that the robot behaves consistently under the same conditions.

Validity was strengthened by using more than one source of evidence instead of relying on a single viewpoint. Document analysis clarified what the process is supposed to look like, testing showed how the robot performs in practice, and expert feedback helped interpret whether the changes are meaningful in day-to-day work. Involving both Palkeet's IT specialists and software developers and the KEHA Centre's financial experts also supported

construct and content validity, because it kept the solution aligned with both the technical constraints of the automation and the practical realities of the cost accounting workflow.

The combination of transparent documentation, iterative testing, and expert involvement provides a credible basis for the results presented in Chapter 6 and for considering how similar automation could be approached in comparable public-sector finance contexts.

4 Current state analysis (case: KEHA Centre)

This chapter examines the current cost accounting process at the KEHA Centre within the Finnish State's shared cost accounting model. The aim is to identify the major manual work phases, challenges, and development needs that form the basis for automation. In the present process, a significant share of expert working time is spent on repetitive verification and data transfer tasks rather than analytical or development work.

The chapter starts by outlining the KEHA Centre's cost accounting process and its position in the state-level framework. It then describes the current process flow, the Excel templates used in the work, and the roles of key actors such as Palkeet and the State Treasury. The chapter closes by highlighting where manual effort accumulates, where inefficiencies are most visible, and where human error is most likely to occur.

The observations in this chapter provide the practical starting point for the design choices described in Chapter 5. By documenting the workflow in detail, it becomes possible to focus automation on tasks that matter most in terms of workload, consistency, and data quality, and to evaluate later whether the changes actually improve the process.

4.1 Overview of the KEHA Centre's cost accounting process

The KEHA Centre carries out cost accounting under the Finnish State's shared cost accounting model, coordinated by the State Treasury. The model is meant to produce consistent and comparable cost information across agencies for financial steering, decision-making, and performance management (Valtiokonttori 2017). For the KEHA Centre, the outputs are used to monitor the use of resources and to make public spending more transparent. Figure 5 shows where the KEHA Centre sits in this framework and how the work links to the State Treasury, Palkeet, and the Kieku system.

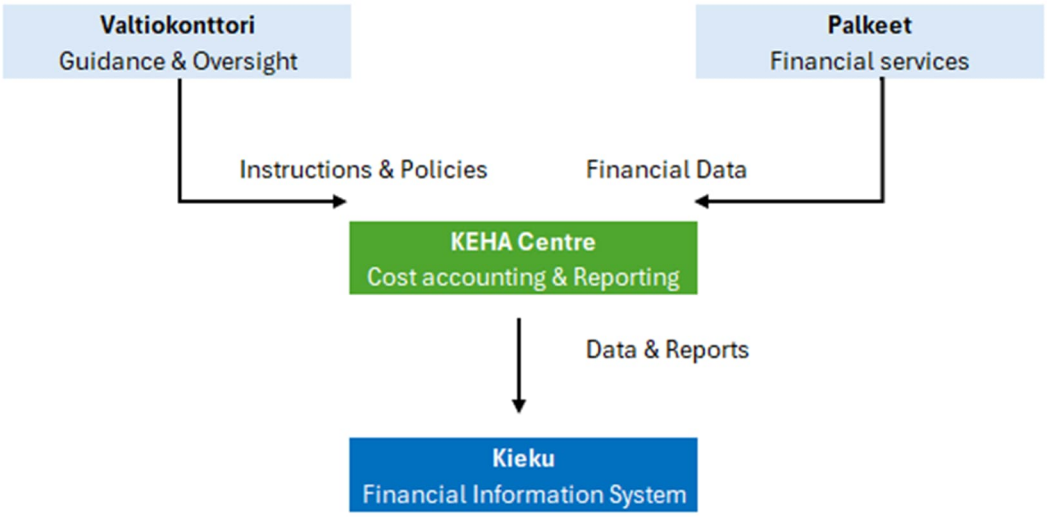

Figure 5. Role of the KEHA Centre within the Finnish state cost accounting framework.

The cost accounting cycle combines automated elements with manual work. Cost data is collected from several systems and operational records, and financial experts compile, check, and organise the inputs in Excel. The main preparation happens in a definition document, where inputs, parameters, and allocation drivers are maintained. Once the definition document is ready, macros in the Excel calculation templates are used to run the calculations. A typical cycle consists of roughly 20 allocation phases implemented as Excel-based calculation steps or templates, although the exact number can vary depending on the datasets and reporting needs. Each phase produces its own output dataset, and the outputs are then transferred into Kieku as separate reporting datasets in a fixed sequence.

Because the templates follow a standardized structure and include basic validations, many missing values and inconsistencies become visible already during preparation, instead of only appearing at the final reporting stage. In practice, this includes totals-based checks across the chain: template totals and report-level sums are compared against the transfer files, both within the Excel workflow and before data is moved into Kieku. These checks help surface issues earlier, but the final figures are still confirmed through reconciliation and Kieku reporting.

The cycle is currently run twice per year in order to keep cost information reasonably up to date. Completing one run typically requires working through roughly 20 calculation steps

and then checking the outputs across the full dataset sequence before and after transfer to Kieku. In practice, a full run takes about three working days, depending on data readiness and the time needed for expert checks. Since the work relies heavily on spreadsheets and repeated verification, it demands careful quality control and it also creates room for small entry or checking errors.

This Excel-based workflow is relatively new at the KEHA Centre. The earlier approach to the cost accounting cycle was organised differently, and the current templates and division of tasks reflect the organisation's recent efforts to standardise how the work is done.

Standardisation has made the cycle easier to manage and more transparent, but the execution work is still repetitive and takes time. A large share of the effort goes into running the same calculation steps, transferring datasets into Kieku, and verifying that each step has been completed correctly. This is the main reason automation is being pursued. If the routine execution can be handled more consistently, experts can spend more time on validation and interpretation, and a more frequent run cycle becomes realistic when resources allow.

The KEHA Centre's cost accounting process is well structured and supports transparency and efficient use of resources. In Section 4.2, I describe the workflow in phases and clarify the main responsibilities. This also makes it easier to see where manual work accumulates and where automation can help.

4.2 Process flow and responsibilities

At the KEHA Centre, cost accounting follows the state-level shared model and a standardized structure. For clarity, I describe the workflow in four main phases: (1) collecting and preparing the inputs in the definition document, (2) running the processing and calculations in the Excel calculation templates, (3) transferring the final datasets into Kieku, and (4) reconciling and reporting based on Kieku reports.

The cycle starts with gathering cost information from several internal and external sources, such as accounting records, payroll data, and operational inputs. The material is then exported into Excel templates that follow the State Treasury's cost accounting guidelines. Getting the inputs ready often requires coordination between KEHA Centre financial experts and Palkeet, especially when source data needs clarification or formats need to be aligned.

In the definition document, financial specialists bring the inputs together for the calculation run. They enter and update parameters and allocation drivers, check that the source data is complete and internally consistent, and make the small corrections that prevent later calculation or reporting issues. The definition document follows the shared cost accounting model and acts as the central handover and approval point: once it is ready, the calculation templates can be run and the outputs can be generated in a controlled way.

Once the definition document is ready, the work moves into the calculation phase. The Excel calculation templates apply the predefined allocation keys and rules to distribute costs through a series of allocation phases, where each phase produces its own output dataset. Macros run the calculation logic step by step and generate the transfer files needed for system entry. If an error occurs, the financial expert investigates whether the root cause is in the source report format (for example, a report that was created or exported in an incompatible structure) or in the inputs entered in the definition document. The expert then corrects the issue before the run continues.

After the calculations, the results are transferred from Excel into Kieku, the official financial reporting platform for all state agencies. In practice, this means repeating the same entry-and-upload routine for each allocation-phase dataset: preliminary information required for the upload is entered, and the corresponding transfer file is then uploaded to the designated location in Kieku. The sequence matters because later phases and reconciliation depend on earlier outputs being available in the correct place.

In the current workflow, the transfers are performed manually and involve repeated entry actions and checks, which makes this phase both slow and sensitive to input mistakes. Reconciliation begins only after all allocation phases have been successfully run and their datasets transferred into Kieku. The reconciliation itself follows a separate template aligned with the shared cost accounting model, and final reporting is completed based on the reconciled figures.

Responsibilities are shared across several actors. KEHA Centre financial experts compile and validate the inputs, approve the run readiness, and remain responsible for reconciliation and interpretation of results after the datasets have been transferred into Kieku. Palkeet supports the workflow by maintaining the automation platform and starting the robot on

request when a run is scheduled or needed. The State Treasury maintains the shared cost accounting model and issues guidance to ensure that agencies follow national standards.

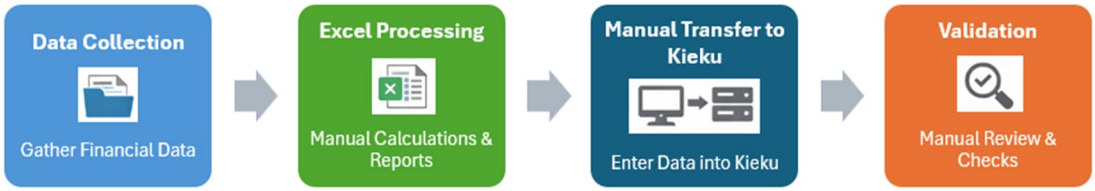

Figure 6. Current (as-is) cost accounting process at the KEHA Centre.

Figure 6 traces the flow from initial data collection to the manual transfer into Kieku. It makes visible where the process relies on repeated Excel-based validation and human decisions, and where information is re-entered or reconciled across systems.

It also underlines how much manual processing and validation takes place before system entry.

4.3 Identified manual work phases and challenges

The KEHA Centre's cost accounting process still includes several manual steps that require considerable expert effort and slow down the overall cycle. Even though the process follows a standardized structure defined by the State Treasury, much of the practical work is carried out through Excel-based handling, repeated checks, and stepwise transfers between tools. The combination of many small steps and strict ordering requirements makes the execution phase sensitive to interruptions and increases the risk that issues are only noticed later in reconciliation.

Data preparation and validation are among the most time-consuming parts of the cycle. A practical challenge is that source data does not always arrive in a uniform structure, which means that reports often need to be reshaped into the expected format before the templates (and later the robot) can use them reliably. In addition to format alignment, experts cross-check entries for completeness and consistency and ensure that the coding structure is respected. Because the calculation is executed through multiple templates, similar checks are repeated across steps to make sure each dataset is ready for transfer to Kieku.

A second major workload driver is the manual transfer of results from Excel into Kieku. Because each calculation step produces a separate reporting dataset, the transfer phase is repeated many times and the step order matters. In practice, the same entry-and-upload routine is performed repeatedly, and it is easy to place a dataset in the wrong location or skip a step when the work is interrupted. These mistakes are often not visible immediately; instead, they may surface later as a totals mismatch during reconciliation, which then requires experts to trace back where in the chain the error occurred.

Limited automation also affects timeliness. Since one cost accounting cycle takes about three working days and requires significant expert effort, the work is typically completed only twice per year. In practice, the three-day duration is driven by the repeated Kieku transfers across the dataset sequence, the checks and corrections around each step, and the fact that the work often proceeds with interruptions as other tasks take priority in between. The result is that the execution phase becomes a workload peak that is difficult to schedule more often. Even with automation, financial experts are still needed to prepare the inputs and to review and interpret results; the main automation potential lies in taking over the repetitive execution and transfer work so that the cycle is less laborious to complete.

The process also relies heavily on individual expertise. In practice, this can lead to variation in what is checked first, how strictly intermediate totals are compared, and how exceptions are documented and investigated when a mismatch appears. When documentation is incomplete, experts may need to reconstruct the chain and repeat checks to regain confidence in the results, which adds to both workload and cycle time.

These challenges point to three clear development priorities: reducing repetitive manual work, improving data accuracy, and ensuring consistent execution of the process. RPA can support these priorities because the targeted tasks are rule-based and repeatable, the Excel workflow is standardized, and the outputs follow a defined structure for transfer to Kieku. At the same time, the workflow remains led by experts. Financial specialists continue to prepare inputs, handle exceptions, and validate outcomes, while the robot takes over the most repetitive execution steps.

4.4 Development needs and automation potential

The current state analysis highlights clear opportunities to improve both efficiency and reliability in the KEHA Centre's cost accounting process. In practice, the main development need is to reduce the risk that the dataset transfer "chain" breaks: when many datasets are handled one by one, a missed step or an incorrect order can interrupt the cycle or create inconsistencies that are hard to trace afterwards. Another recurring issue is that totals mismatches may only become visible late in the cycle, when the work has already progressed and fixing the root cause requires backtracking. These are the kinds of repetitive, rule-based activities where automation can provide the most value without changing the underlying systems.

Automating the execution and transfer work can reduce manual workload, lower the risk of input mistakes, and improve consistency between reporting periods. It can also strengthen auditability and traceability in concrete ways: by keeping a run log of key steps, by saving an error listing to an agreed team workspace and notifying the responsible expert by email, and by making it easier to confirm what was run and when (for example, the run date and the input material version used). In reconciliation, these traces help clarify where a deviation emerged instead of leaving experts to reconstruct the chain from memory and scattered notes.

Another development need relates to timeliness. With the cycle currently completed only twice per year and taking roughly three working days to run, the availability of up-to-date cost information is limited. While automation can reduce the hands-on execution work and make the run easier to schedule, the main practical constraint for increasing run frequency is upstream input preparation and the expert resources required for it. If the organisation chooses to organise these resources differently, more frequent runs (for example quarterly) become more realistic. A higher run frequency would also provide practical benefits: reconciliation issues and deviations would surface earlier, and management would have access to more up-to-date cost information for steering and follow-up.

From an automation point of view, the process fits RPA well because the execution work is stable and repeatable across cycles. The calculation is based on standardized Excel templates and macros, the datasets follow a predefined sequence, and the shared cost accounting model keeps the main rules and coding structures consistent. In addition, the transfer step in Kieku

is largely the same each time: the same entry and upload routine is repeated for each dataset. Together, these characteristics create conditions where a robot can execute the sequence reliably without altering the underlying financial systems.

Because the KEHA Centre applies the State Treasury's shared cost accounting model, similar automation could potentially be adapted for other government agencies operating under the same framework. At the same time, some parts do not transfer directly: agency-specific data sources, template versions and reporting views, as well as local workspaces and access-right arrangements shape how the execution step can be automated in practice. There are also clear limits to what should be automated at this stage. Final reconciliation remains expert work, because it requires interpretation and accountability beyond rule-based execution. Even with these boundaries, the approach aligns with national goals related to digitalisation and knowledge-based management and illustrates how standardized processes can enable reusable automation solutions when local adaptations are handled explicitly.

With the current workflow and automation opportunities clarified, the next part of the thesis moves into solution design and implementation. The focus is on one process step where automation can reduce repetitive execution work while keeping preparation, validation, and reconciliation led by experts in place.

5 Solution design and implementation

Chapter 4 showed where manual effort and error risk concentrate in the KEHA Centre's cost accounting workflow. This chapter turns those observations into a concrete RPA solution by focusing on one well-defined execution step where automation can reduce repetitive work without changing the underlying systems.

The chapter explains how the selected step was moved from a manual workflow to a semi-automated one. It describes the design goals and logic of the solution, the technical implementation, and the process changes introduced by the robot.

The solution was developed through iterative testing and refinement, and its performance was assessed against the objectives set for efficiency, accuracy, and consistency.

5.1 Design of the RPA solution

The RPA solution was designed to automate a specific execution phase in the KEHA Centre's cost accounting process. Based on the current state analysis, the selected target was the most repetitive part of the workflow: running the macro-driven Excel steps and transferring the outputs into Kieku dataset by dataset. This phase was also where small manual mistakes are easiest to make, because the same entry and upload routine is repeated many times. From the experts' perspective, it is a necessary but time-consuming "routine task" that interrupts other work.

In this thesis, the selected step refers to the execution phase from start to finish where (1) the macro-driven calculation runs are executed in the Excel calculation templates and (2) the resulting outputs are transferred into Kieku as separate reporting datasets. In a typical cycle, this means entering or uploading approximately 20 datasets (one per calculation step/template) into their respective locations in Kieku before reconciliation and reporting can take place. A run was considered successful when the robot completed the sequence without run-time errors, the totals matched the agreed reference checks, and all calculation templates were transferred into Kieku so that expert-led reconciliation could begin.

The main design goals were to reduce manual workload in the most repetitive execution steps and to lower the risk of transfer mistakes. In practice, this meant that the robot must follow the same dataset sequence each time, stop on deviations instead of continuing silently, and make it clear to experts where and why a run failed. At the same time, the solution needed to fit the existing workflow and system landscape, as changes to the underlying financial systems were out of scope. For this reason, the robot was designed to work on top of the current tools by replicating the actions previously carried out by financial experts.

The design work started by mapping the selected step in detail, including its inputs, decision rules, validation criteria, and expected outputs in Kieku. The definition document was treated as the key handover point before execution: it follows the state's shared cost accounting model and brings the required parameters and drivers into a consistent format for the templates. This made it possible to keep expert responsibility clear. Once the definition document is ready, the robot can execute the repeatable run-and-transfer sequence. Special attention was paid to data dependencies and exception cases, because these define where automation can proceed fully and where human review is required.

The design also accounts for the continued role of human expertise. Financial experts prepare and verify the inputs before execution (especially the definition document), and they validate the results afterwards through reconciliation and reporting once all datasets have been transferred into Kieku. During the run, expert involvement is mainly needed only if the robot flags an exception, which is not expected to occur in every cycle but must be handled when it does. In this way, the solution supports professional judgment and accountability rather than replacing it.

The resulting RPA solution is a rule-based robot that executes predefined tasks in a consistent and repeatable way. It integrates into the existing cost accounting workflow and takes over the most repetitive execution and transfer work while keeping expert-led control points in place. Final reconciliation is deliberately left outside the robot's scope in this version, and it remains expert work that may be supported further in future development.

This step was selected because it combines high repetition with high error sensitivity, and it consumes expert time on routine execution rather than on analysis. The key requirements were a fixed dataset sequence, totals-based validation, clear stop-and-notify handling for

exceptions, and a clean handover to expert reconciliation once all templates have been transferred into Kieku.

5.2 Technical implementation

The solution runs on Palkeet's automation platform, but the robot is configured for the KEHA Centre's use and can be run as part of KEHA's own cost accounting workflow. After the inputs have been prepared in the definition document, the robot runs the macro-driven calculations in the Excel templates and transfers the resulting reporting datasets into Kieku in a controlled sequence.

The robot can be run on a schedule on a pre-agreed date or started manually when needed. In the current operating model, the cost accounting expert agrees the timing and sends a separate notification when the run is ready to start. For manual runs, the expert requests Palkeet to start the robot. The run frequency has so far been twice per year, and the target is to move toward more frequent runs (for example quarterly) when resources allow. For each run, the robot retrieves the required Excel files from the KEHA Centre's shared team workspace to which the robot has access.

Once the files are available, the robot opens the definition document and the calculation templates and transfers the required inputs into each template before the macros are run. A central validation point is that sums match across the chain: the totals in the templates are compared against the totals in the transfer files, and the report-level sums are cross-checked against the same transfer files before data is moved into Kieku. If a required file or input is missing or a check fails, the robot stops and produces an error listing for the cost accounting expert.

After each calculation step, the robot enters the resulting dataset into Kieku before continuing to the next step. Because the run produces multiple reporting datasets, the same entry and upload routine is repeated across the full dataset sequence. For each dataset, the robot navigates to the correct location in Kieku, completes the entry, and confirms that the action succeeded before continuing. If an error occurs, the robot records a simple message indicating where the run failed and delivers an error listing in the shared workspace and a notification by email to the cost accounting expert.

The implementation was refined through repeated test runs and joint review sessions. Early versions exposed practical issues such as access-right constraints and input-format differences that can interrupt macro runs or Kieku entry. These issues were addressed by tightening the sum checks and by making failures visible to the expert through the error listing and notification. The robot supports the execution and transfer work, but the final reconciliation is still carried out by an expert and begins once all calculation templates have been successfully run and transferred into Kieku.

As implemented, the robot carries out the selected execution step in a controlled and repeatable way within the KEHA Centre workflow. It reduces the hands-on effort required for running the templates and transferring datasets into Kieku, while keeping the key control points like sum checks, exception handling, and final reconciliation, with the financial experts.

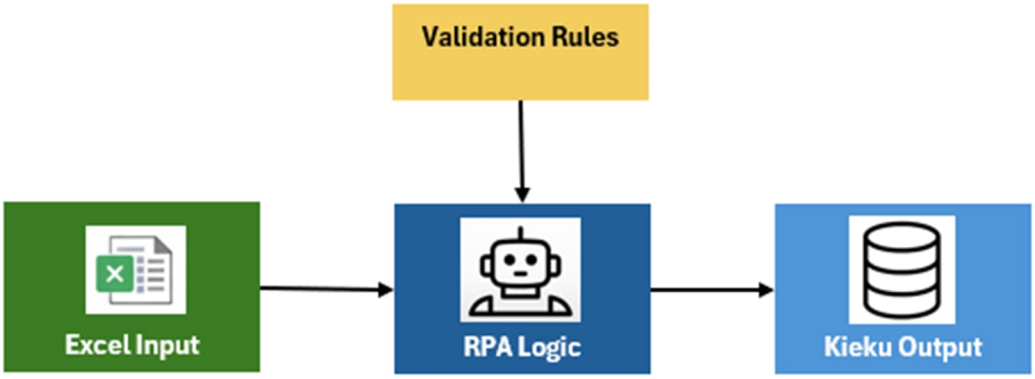

Figure 7. Logical structure of the RPA solution.

Figure 7 shows the main building blocks of the solution. The implementation keeps input reading, validation and transformations, and the Kieku entry routines as distinct parts, which makes the robot easier to test and troubleshoot and supports incremental changes.

5.3 Process description of the automated step

Here the automated step covers the full execution phase from start to finish: running the Excel-based calculations and moving the outputs into Kieku for reporting. Once financial experts have completed the definition document (inputs and allocation drivers), the robot

moves the inputs into the calculation templates, runs the macros, and then transfers the outputs into Kieku as separate reporting datasets. In practice, the Kieku transfer includes both entering preliminary information and uploading a file for each dataset. In a typical cycle, this results in roughly 20 Kieku entries/uploads, one for each calculation step or template.

Before automation, experts completed this execution phase manually. They ran the macros in each calculation template and then moved the outputs into Kieku dataset by dataset, repeating the same entry and upload routine many times. The core checks were practical: totals were compared against what the templates produced, and it was verified in Kieku that each dataset ended up in the correct location before continuing.

Because the transfer work is repetitive and detail-sensitive, it requires careful attention. In practice, one immediate sign that the allocations have been run is that accounts starting with 999 appear in the Kieku reports. However, the final correctness is only confirmed after reconciliation, when the reconciliation calculation has been completed and the results match as expected. For this reason, execution can be supported by automation, but expert validation and interpretation remain necessary after the run.

With RPA in place, this execution phase is now partly automated. After the definition document is ready, the robot runs through the calculation templates in sequence. For each step, it moves the required inputs into the template, runs the macro, checks that the output totals are in line with expectations, and then completes the Kieku routine (entering the preliminary details and uploading the dataset file) before moving on. The step order matters because datasets need to be processed in a defined sequence and placed in the correct location in Kieku. If the robot detects a deviation, it produces an error message so the run does not continue silently with incorrect data.

The robot does not remove the need for human involvement. Financial experts still prepare and verify the inputs before execution and review exceptions when the robot raises an error message. After the datasets have been transferred into Kieku, experts use Kieku reports (including the appearance of 999-series accounts) as an initial indicator that the run has gone through, but the final validation comes from reconciliation and the reconciliation calculation. This keeps accountability and control with the experts while the robot handles the repetitive execution work.

Before a new cost accounting run, the robot performs a reset run that clears the previous allocations from Kieku. This reset is necessary because otherwise the same allocations would be entered multiple times on top of earlier results. In other words, the execution needs to start from a clean state in Kieku before the new datasets are transferred.

In practice, the work now looks different: once experts have prepared the definition document (for example, the report links the robot retrieves, the driver data for the templates, and the relevant organisational units and activities for the phase), the robot executes the repeated run-and-transfer sequence. Experts then focus on input readiness, handling any exceptions the robot flags, and confirming the results through reconciliation and reporting in Kieku.

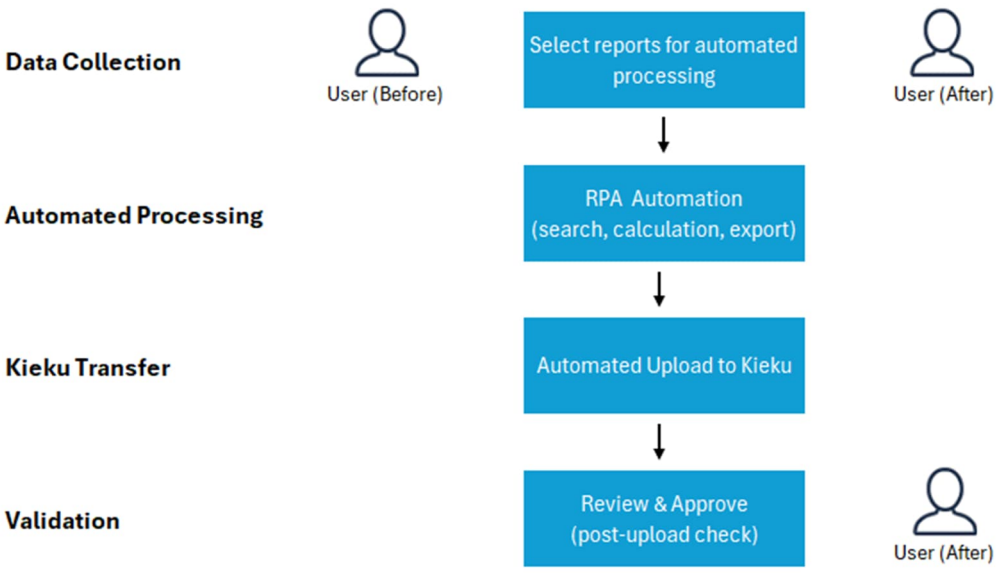

Figure 8. Target (to-be) process after RPA implementation.

Figure 8 describes the target process after automation. The robot carries out the repetitive data handling, validation, and transfer routines, while financial experts remain responsible for input preparation and for reviewing the results and reconciliation in Kieku.

5.4 Testing and iterative development

Testing started early and continued throughout development, with Palkeet carrying the main responsibility for running the tests in the automation environment. I supported the work by clarifying process details and providing additional information when questions came up, especially around inputs, expected totals, and what should be visible in Kieku after each transfer.

The first version covered the core steps of the execution phase and was tested using last year's figures as reference material, with the aim of reaching the same totals as in the previous year's reference outputs. Early tests revealed practical issues that affected repeatability, most commonly access-right problems in the target environment and cases where the input data in Excel was in the wrong format for the templates and macros.

When a failure occurred, the robot stopped the run and produced an error listing to an agreed location for review. The issue was then corrected (for example by fixing the input format or updating the access-related setup), and the change was verified in the next test run. This test–fix–retest loop was repeated until the same input produced the expected totals and the full transfer sequence completed without run-time errors.

Development and testing were done in collaboration with Palkeet and the KEHA Centre financial experts. Palkeet ran the technical tests and contacted me when additional process detail was needed to interpret a failure or confirm an expected outcome. In internal meetings and joint sessions, we discussed how the robot currently behaves and how it should behave in the target workflow. We also gathered the experts' views on the robot handling the execution tasks and identified practical improvements for checks, error messages, and future development to better support monitoring and reconciliation.

By the end of the main test rounds, the robot handled the input variations encountered in the development material and produced results that matched the reference totals from the previous year when the same assumptions were used. If exceptions occurred, they were surfaced through the error listing for expert review rather than being hidden inside the run. Overall, this provided a workable level of stability for the initial deployment, while leaving room to refine checks further as new edge cases emerge in continued use.

5.5 Evaluation of the solution

The evaluation asks whether the automation meets the objectives set for the study and whether it improves the cost accounting execution phase in a way that matters in practice. I focus on reduced manual work, accuracy and consistency of the transfers, and what was learned during implementation and early use.

The core comparison is between the earlier manual execution phase and the automated run. In both cases, the same work needs to happen: Excel calculations must be completed and the outputs must be entered into Kieku across the full dataset sequence. The evaluation therefore looks at how much hands-on effort the execution requires, how consistently the sequence can be completed, and how reliably the outputs end up in the right place in Kieku.

To verify correctness, I compared the totals produced by the Excel templates with the values transferred to Kieku and confirmed the results through the Kieku reports used in reconciliation. I also monitored that the robot completed the predefined checks and the full transfer sequence, and that the successful runs finished without run-time error messages.

The workflow was exercised repeatedly during the development phase to confirm that execution remained consistent in practice. Under typical conditions, the automated phase took around 30 minutes from start to finish. In addition to these technical observations, usability was assessed through feedback gathered from KEHA Centre financial experts in internal meetings and joint working sessions.

The feedback focused on how well the solution fits everyday work and whether it improves the clarity of the execution phase, especially from the perspective of monitoring the run and reconciling results afterwards. Together, these observations form the basis for the results reported in Chapter 6, where I describe what changed after deployment and how the automated execution compares with the earlier manual approach.

6 Deployment and results

This chapter describes how the RPA solution was brought into the KEHA Centre's cost accounting workflow and what changed after it went into use. The focus is on the deployment approach and on the practical results observed in the execution phase.

The results are described using both technical observations from test runs and feedback from financial experts who work with the process. Together, these sources show how automation affected execution time, data quality, and everyday use.

The chapter also notes what worked well in the first deployments and where expert attention is still needed, so the reader gets a realistic picture of what the automation does and does not cover.

6.1 Deployment of the RPA solution

The RPA solution was introduced into the KEHA Centre's cost accounting workflow together with Palkeet. The main priority during deployment was to integrate the robot into the existing Excel and Kieku workflow without disrupting ongoing financial work. Because the robot operates through the same templates and user interfaces that are already in use, deployment did not require changes to the underlying IT infrastructure.

Deployment was carried out in stages so that the robot could be tested in a controlled way before broader use. The early runs focused on two essentials: the robot had to run the macro-driven calculation steps in the Excel templates after the definition document was ready, and it had to transfer each resulting dataset into Kieku reliably and in the correct order. The team paid particular attention to input format consistency, exception handling, and the stability of the Kieku entry routines.

Financial experts were closely involved throughout deployment and they were still closely involved in the process even after automation was introduced. They prepared and reviewed the necessary source data before the robot was run and checked the results afterwards to make sure everything looked correct. The final reporting and reconciliation work continued

to rely on Kieku reports and the organization's normal accounting practices, so the overall control process remained largely unchanged.

The deployment also required close collaboration between the KEHA Centre and Palkeet to confirm that the robot behaves consistently in the Kieku environment. The work focused on practical details such as system navigation, access rights, and run-time behaviour during the repeated dataset transfers. By the end of deployment, the robot fit into the workflow in a way that supports everyday work while preserving expert oversight.

6.2 Results of the implementation

The clearest change after implementation was the reduction in repetitive manual execution work. Earlier, experts had to run the calculation steps and transfer the outputs into Kieku manually across the full dataset sequence. With the robot in place, the same sequence is executed consistently and in the correct order, and experts mainly monitor the run and focus on exceptions and checks. Under typical conditions, the execution phase from start to finish takes approximately 30 minutes. Before automation, completing the same execution and transfer work typically took about three working days in the normal flow of work, because the steps had to be handled one by one across multiple datasets alongside repeated checks and interruptions. This comparison focuses on the execution and transfer phase. Preparation by experts, exception handling, and reconciliation remain necessary parts of the overall cycle. The three-day estimate refers to the typical calendar duration of completing the execution and transfers in practice, not three days of uninterrupted hands-on work.

Accuracy and consistency improved mainly because execution became more standardized. The robot follows the same sequence every time, runs the same macro-driven calculations, and transfers datasets into Kieku in a repeatable way. This reduces the likelihood of errors caused by copying and pasting or manual entry during execution. In the observed runs, no run-time errors were linked to the robot, and it completed the full dataset sequence in the correct order within a single run. Any discrepancies were found during reconciliation in Kieku and were related to the input data or allocation drivers rather than the automated execution itself.

Efficiency improved because the robot took over the most time-consuming execution activities. The repeated macro runs and the sequential transfer of multiple datasets into Kieku can now be completed with less manual effort and fewer interruptions. The approximately 30-minute execution time covers the macro-based calculation run and the Kieku transfers. Input preparation and the final reconciliation still require expert work and are carried out outside the automated run. Over time, this makes it more realistic to run the cost accounting cycle more frequently than before, provided that upstream input preparation and reconciliation capacity are available.

Feedback from financial experts supported the technical observations. They described the main benefit as the removal of repeated Kieku entry or upload actions across the dataset sequence, which makes the execution phase easier to manage. They also noted that the run is more predictable, while they still retain control over validation, exceptions, and the final reconciliation. In their feedback, three themes came up consistently: reduced routine workload, improved predictability of execution, and preserved expert control.

The results show that RPA can improve efficiency and consistency in a cost accounting execution phase that is repetitive, rule-based, and built around standardized templates. In the KEHA Centre case, the main benefit came from automating the Excel calculation run and the stepwise Kieku transfers while keeping expert-led preparation and reconciliation in place.

7 Discussion

This chapter reflects on what the implementation changed in practice and why those changes occurred. It links the observations back to the research questions and to the concepts discussed in the theoretical framework, and it considers what the case suggests for similar public-sector finance processes.

The discussion focuses on three themes: what explains the observed efficiency and quality improvements, what the case adds to earlier work and to public-sector practice, and what the main limitations and development opportunities are.

7.1 Interpretation of the results

The KEHA Centre case shows that RPA is a good match for structured, rule-based execution work in cost accounting. The largest time savings came from the phase where macros are run in Excel and the resulting datasets are entered into Kieku one by one across the full dataset sequence. When that repetitive execution was moved to the robot, expert time shifted toward inputs, checks, and the final reconciliation.

Improved accuracy and consistency follow directly from standardized execution. Manual work can vary between users and across busy periods. In contrast, the robot runs the same macros, follows the same step order, and transfers datasets into Kieku in the same way every time. This reduces errors caused by copying and pasting and makes missed checks less likely. The case underlines a practical point that standardization is not just a nice-to-have for automation. It is what makes reliable automation possible.

The biggest speed improvement came from removing the repeated manual steps in the execution phase. A typical run now finishes in about 30 minutes, whereas earlier the same execution and transfer work typically took about three working days in the normal flow of work. Preparation and reconciliation still require expert work, but the automated execution makes runs easier to schedule and lowers the day-to-day friction of completing the cycle. Over time, this creates a realistic basis for increasing the run frequency.

The case also makes the human role very clear. Automation handles routine execution, but experts remain responsible for preparing inputs, deciding how to handle exceptions, and interpreting the results during reconciliation. The solution was designed to bring exception cases to experts' attention instead of trying to resolve them automatically. This keeps control over reporting with the people who are accountable for the numbers.

These findings are also in line with the earlier literature discussed in Chapter 2.5. Previous studies have linked RPA especially to stable, rule-based processes with structured data and clearly defined task sequences, and the KEHA Centre case supports that view in a public-sector cost accounting setting (Syed et al. 2020; Asatiani & Penttinen 2016). The case also shows quite clearly that the expert role does not disappear once the repetitive execution is automated. The work changes instead. More time is spent on making sure the inputs are ready, dealing with exceptions when they appear, and checking the results afterwards in reconciliation. This fits well with earlier observations that governance, traceability, and human oversight remain important even when the execution itself is automated (Willcocks et al. 2015; Bagheri & van de Wetering 2024).

	Before Automation	After Automation
Manual Cost Accounting Work	Excel calculations, manual upload	User selection & post-upload review
Risk of Calculation and Data Entry Errors	Formula handling, copy-paste	Rule-based processing
End-to End Cost Accounting Cycle	Sequential manual steps	Automated end-to-end run
Process Control & Transparency	Implicit, user-dependent	Explicit, rule-based & auditable

Figure 9. Qualitative comparison of key process characteristics before and after automation.

Figure 9 brings the before-and-after differences together at a high level. It summarizes how repetitive execution work decreased, how the run became more consistent and predictable, and how expert effort moved away from routine system entry toward preparation, reconciliation, and interpretation.

7.2 Contribution to theory and practice

One practical contribution of this case is that it makes the implementation work more concrete in a public-sector finance setting. The process was highly regulated and already fairly standardised, which could easily be seen only as a constraint. In this case, those same features also made automation easier to apply. The selected step followed clear rules, repeated the same actions from one run to the next, and relied on standardised Excel files and recurring Kieku transfer routines. That combination made it possible to automate the work without changing the underlying systems.

From a theoretical perspective, the case adds a more specific example to earlier RPA research by showing how automation can be applied in a public-sector cost accounting process that operates under shared reporting structures and formal governance requirements. Earlier literature has often discussed RPA at a more general level or in back-office settings without focusing closely on cost accounting under a common state model. The KEHA Centre case adds detail to that discussion by showing that standardised routines and shared structures do not only constrain automation but can also create the conditions that make it workable in practice.

The case also connects process development and automation in a practical way. Standardization made the work more transparent and comparable, and automation reduced the effort required to run the process. This matters in public organisations, where shared systems and governance requirements often limit the scope for large IT changes (Hardy & Williams 2011; Valtiovarainministeriö 2020).

The case also shows how Design Science Research can be used to develop and evaluate an automation solution in a public-sector setting. Iterative development, real-world testing, and a clear link to theory helped keep the artifact practical while maintaining a research-driven structure.

For the KEHA Centre, the work provides a concrete example of how automation can be integrated into the cost accounting workflow. The implementation shows that one well-chosen execution phase can improve efficiency and consistency. Because the process follows the State Treasury's shared model, the same idea may also fit other agencies that work under similar rules, provided that templates, data sources, and working practices are adapted locally. The work shows how theory-guided design choices can translate into measurable, practical improvements in a real public-sector cost accounting workflow.

7.3 Limitations of the study

The case and the evaluation setup also set clear boundaries for what can be concluded from the results. The results are based on one selected process step in the KEHA Centre's cost accounting workflow. The step was well suited for automation, but the findings do not extend automatically to the full cost accounting process or to other finance processes where decisions are less structured or where data is less standardized.

The work was carried out in one organisational context, the KEHA Centre. Even though the process follows the State Treasury's shared cost accounting model, templates, system configurations, and working practices vary between agencies. This means the solution cannot be copied directly to another organisation without local adjustments. At the same time, the case shows what a comparable development effort can look like for agencies operating under the same shared model.

The evaluation relies on short-term observations during implementation and early testing. Because the solution is still new, longer-term effects, such as sustained efficiency gains, maintenance effort, and changes in user practices, remain outside the scope of this study.

This study also uses qualitative feedback from financial experts involved in the process. The feedback helped to understand how the solution works in practice and how it supports daily tasks. At the same time, the observations are based on individual experiences, meaning that not all impacts of the solution can be measured or fully quantified. The limitations discussed above make longer-term evaluation important. Additional cases would also help assess how well the approach scales across public-sector financial processes.

7.4 Future development and further research

This section outlines a few realistic next steps that build on what was learned in the KEHA Centre case. One clear next step is to look for other repetitive execution steps in the cost accounting cycle that follow stable rules and formats. This thesis focused on one well-defined phase, but similar patterns can also appear elsewhere in the cycle, especially where data is moved between templates or entered into systems in the same way each run. Extending automation step by step would make it easier to keep control over quality while gradually reducing routine work.

Another development area relates to how the outputs are used after each run. If the Kieku outputs and reconciliation notes were captured in a more consistent way, reporting and follow-up work could become easier and less dependent on manual compilation. In practice, this would mean paying attention to how the results are documented and how they flow into the management views that are used after the cycle.

The case also raises the question of whether more advanced tools are worth adding around the execution phase. One practical use would be to help experts notice unusual patterns in inputs or outputs earlier, so that reconciliation effort can be targeted where it matters most. Another would be to support recurring exception handling, for example by grouping similar exception types or drafting short notes for documentation. These kinds of additions would still need careful consideration in terms of transparency, data protection, and governance requirements in the public sector.

Because the work follows the State Treasury's shared model, a similar implementation may also fit other agencies that use the same framework. Even so, a practical rollout would need to account for local templates, data sources, and working practices. Testing the approach in another agency would show what carries over directly and what needs to be adjusted.

A longer follow-up period would also help answer practical questions that do not show up in early testing. It would clarify what kind of maintenance the robot needs, how stable the templates and system routines remain over time, and how users adapt their working practices when routine execution work is reduced. In short, the case suggests that public-sector finance automation works best when it is developed in small, well-controlled steps, with clear ownership of inputs, checks, and reconciliation.

8 Conclusions

This thesis shows what Robotic Process Automation (RPA) can realistically improve in a selected execution phase of the KEHA Centre's cost accounting process. The work was carried out as Design Science Research (DSR) and focused on designing, implementing, and evaluating an automation solution that fits the constraints of the Finnish State's shared cost accounting model.

The main research question of this thesis was how Robotic Process Automation can be implemented in the KEHA Centre's cost accounting process to improve efficiency and accuracy. Based on the results, RPA can be implemented effectively by targeting the repetitive execution phase in which macro-driven Excel calculations are run and the resulting datasets are transferred into Kieku in a predefined sequence. The first supporting sub-question asked what the main manual work phases in the current cost accounting process are. The results show that these phases are concentrated in input checking, repeated execution of the Excel-based calculation steps, and the stepwise transfer of datasets into Kieku. The second supporting sub-question asked how the selected execution phase can be automated in practice within the existing Excel and Kieku environment. The results indicate that this phase can be automated by using a rule-based robot that works with the existing Excel templates, follows the agreed transfer sequence, performs predefined checks, and stops when deviations occur. The third supporting sub-question asked what kinds of benefits and challenges emerge from the implementation. The implementation produced clear benefits in reduced routine work, faster execution, and more consistent processing, while the main challenges remained related to input quality, exception handling, and the continuing need for expert validation and reconciliation.

The selected phase covers the execution work from start to finish after the inputs have been prepared in the definition document. In the implemented solution, the robot transfers the inputs into the Excel templates, runs the macro-driven calculation steps, and transfers the outputs into Kieku as separate reporting datasets along a predefined sequence. In typical runs, this automated execution was completed in approximately 30 minutes. The clearest improvements were a reduction in repetitive manual work and a more consistent and predictable execution of the cost accounting run.

The case also makes it clear that cost accounting remains expert work. Financial specialists still prepare and validate the inputs, handle exception cases, and complete reconciliation and reporting based on Kieku reports after the datasets have been transferred. The value of RPA in this setting comes from taking over routine execution while leaving accountability and judgment with the experts.

From a research perspective, the thesis demonstrates how a design-oriented approach can be used to build and evaluate a working automation artifact in a regulated public-sector finance context. The case reinforces a practical requirement for RPA: process standardisation and stable data structures. When inputs, calculation logic, and system entry routines are well defined, RPA can be implemented effectively without modifying core systems.

The findings are also consistent with earlier literature that presents RPA as most useful in stable, rule-based processes with clear data structures and well-defined responsibilities. In that sense, the KEHA Centre case does not only confirm what earlier studies have suggested about public-sector automation, but adds a concrete example from cost accounting under the State Treasury's shared model. It shows that shared reporting structures and governance requirements do not only limit automation but can also provide the stability that makes it workable.

Even partial automation can deliver meaningful benefits when it targets repetitive, rule-based work. For the KEHA Centre, the implemented solution provides a practical basis for running the cost accounting cycle more efficiently and, over time, potentially more frequently than the current twice-yearly practice as resources allow. A more frequent run cycle would make outputs available sooner and help identify reconciliation issues earlier. Increasing the run frequency would still require that upstream input data and expert reconciliation capacity are available at the desired pace. Because the work follows the State Treasury's shared model, similar solutions may be feasible in other agencies, although local templates, data sources, and working practices would need adaptation. More advanced automation, such as detecting unusual patterns, could support experts in preparation and reconciliation as long as transparency and governance requirements are met.

References

Aguirre, S. & Rodriguez, A. (2017). Automation of a Business Process Using Robotic Process Automation (RPA): A Case Study. Teoksessa Figueroa-García, J., López-Santana, E., Villa-Ramírez, J. & Ferro-Escobar, R. (toim.), Applied Computer Sciences in Engineering. WEA 2017. Communications in Computer and Information Science, vol. 742. Cham: Springer, 65–71. Available: DOI: 10.1007/978-3-319-66963-2_7.

Asatiani, A. & Penttinen, E. (2016). Turning robotic process automation into commercial success – Case OpusCapita. Journal of Information Technology Teaching Cases, 6(2), 67–74.

Bagheri, S. & van de Wetering, R. (2024). RPA Ready? Unlocking Organizational RPA Readiness. Teoksessa Helms, R. et al. (toim.), Disruptive Innovation in a Digitally Connected Healthy World. Switzerland: Springer, 14–28.

Deloitte. (2018). The Robots Are Here: Transforming the Public Sector with Robotic Process Automation. Accessed: 1.7.2025. Available: https://www.deloitte.com/content/dam/assets-zone3/ca/en/docs/services/consulting/2024/ca-en-the-robots-are-here-aoda.pdf

Finlex. (1988). Laki valtion talousarviosta (423/1988). Accessed: 1.7.2025. Available: https://www.finlex.fi/fi/laki/ajantasa/1988/19880423

Finlex. (1992). Asetus valtion talousarviosta (1243/1992). Accessed: 1.7.2025. Available: https://www.finlex.fi/fi/laki/ajantasa/1992/19921243

George James Consulting. (2024). Automation or AI for Government: When Automation Will Do for Now. Accessed: 2.7.2025. Available: https://www.georgejamesconsulting.com/post/automation-or-ai-for-government-when-automation-will-do-for-now

Gregor, S. & Hevner, A. R. (2013). Positioning and Presenting Design Science Research for Maximum Impact. MIS Quarterly, 37(2), 337–355.

Hardy, C. A. & Williams, S. P. (2011). Assembling E-Government Research Designs: A Transdisciplinary View and Interactive Approach. Public Administration Review, 71(3), 405–413.

Hevner, A. R. et al. (2004). Design Science in Information Systems Research. MIS Quarterly, 28(1), 75–105.

Isaac, J. (2025). Automation in Public Sector: Balancing Efficiency and Accountability. Accessed: 23.7.2025. Available: https://www.researchgate.net/publication/393662100_Automation_in_Public_Sector_Balancing_Efficiency_and_Accountability

Syed, R. et al. (2020). Robotic Process Automation: Contemporary themes and challenges. Computers in Industry, 115.

Palkeet. (2024). Visio, arvot ja strategia. Accessed: 1.7.2025. Available: https://palkeet.fi/palkeet/visio-ja-strategia/

Palkeet. (2024). Älykkään automaation palveluilla tehokkuutta rutiinitoihin. Accessed: 4.7.2025. Available: https://palkeet.fi/uutiset/alykkaan-automaation-palveluilla-tehokkuutta-rutiinitoihin/

Peffers, K. et al. (2007). A Design Science Research Methodology for Information Systems Research. Journal of Management Information Systems 24(3), 45–77.

Rapid Innovation. (2024). Intelligent Automation in the Public Sector. Accessed: 2.7.2025. Available: https://www.rapidinnovation.io/post/intelligent-automation-in-the-public-sector

Valtiokonttori. (2017). Valtion yhteinen kustannuslaskentamalli – ohje. Accessed: 1.7.2025. Available: https://www.valtiokonttori.fi/maaraykset-ja-ohjeet/valtion-yhteinen-kustannuslaskentamalli/

Valtiovarainministeriö. (2006). Ministeriöiden kustannuslaskennan kehittäminen. Työryhmämuistio 1/2006. Accessed: 1.7.2025. Available: https://vm.fi/documents/10623/1169938/Ministeri%C3%B6iden+kustannuslaskennan+kehitt%C3%A4minen+1_2006/3fe26603-0c97-4ed3-bb1b-1346f323acb8/Ministeri%C3%B6iden+kustannuslaskennan+kehitt%C3%A4minen+1_2006.pdf

Valtiovarainministeriö. (2014). Valtion taloushallinnon strategia – tavoitetila vuoteen 2020. Accessed: 1.7.2025. Available:

https://vm.fi/documents/10623/1107144/Taloushallintostrategia_julkaisu_7_2014.pdf/44da5c98-6a64-4a51-ac48-ad98314b683f/Taloushallintostrategia_julkaisu_7_2014.pdf

Valtiovarainministeriö. (2020). Valtion taloushallinnon strategia – kohti ennakoivaa ja vaikuttavaa taloushallintoa. Accessed: 1.7.2025. Available: https://vm.fi/hallintopolitiikka/valtion-taloushallinto

Valtiovarainministeriö & Kuntaliitto. (2020). Julkisen hallinnon strategia – Kestävää arkea yhdessä rakentaen. Accessed: 1.7.2025. Available: https://julkaisut.valtioneuvosto.fi/bitstream/handle/10024/162571/Julkisen_hallinnon_strategia_2020.pdf?sequence=1&isAllowed=y

Willcocks, L., Lacity, M. & Craig, A. (2015). The IT Function and Robotic Process Automation. The Outsourcing Unit Working Research Paper Series, Paper 15/05. London: London School of Economics and Political Science. Accessed: 20.5.2026. Available: https://researchonline.lse.ac.uk/id/eprint/64519/1/OUWRPS_15_05_published.pdf