UNIVERSITY OF TARTU
Faculty of Science and Technology
Institute of Computer Science
Software Engineering Curriculum

Daryna Pyshchyk

An Approach for Engineering Software Ecosystems Based on an Industrial Case Study

Master’s Thesis (30 ECTS)

Supervisor(s): Mohamad Gharib, Ph.D.

Tartu 2025

An Approach for Engineering Software Ecosystems Based on an Industrial Case Study

Abstract:
Software Ecosystems (SECO) can be described as a complex collaborative network of software platforms, components, and services that interact and evolve together in a shared operational environment. Compared to traditional software engineering, SECO engineering represents a significant challenge, particularly in requirements gathering, architectural design, and stakeholder coordination, as traditional software engineering approaches cannot fully resolve particular needs of SECO. Although previous research has analysed specific stages of the SECO life cycle, such as requirements, architectural design, or quality assurance, a comprehensive domain-independent approach that guides the overall process of designing and developing SECO is still lacking. Additionally, most current SECO consider humans as an integral part of the SECO, which adds to the complexity of the aforementioned problems. Human-centred design standards like ISO 9241-210 offer valuable guidance for designing human-centred systems, yet they are not tailored to the specific dynamics of SECOs.

This thesis addresses the challenge of engineering human-centred SECO by developing a practical approach that has been constructed by integrating SECO best practices with the experiences used for the development of the PHArA-ON (Pilots for Healthy and Active Ageing in Europe) project, which establishes a digital SECO that supports well-being, activity, and social participation among Europe's ageing population. This work proposes a five-stage human-centred SECO engineering approach. The stages are: (1) context-of-use specification, where SECO goals, stakeholder roles, and tasks are identified; (2) requirements specification, aligning stakeholder needs with SECO objectives; (3) design of a SECO reference architecture; (4) implementation planning and incremental deployment of SECO components; and (5) ongoing evaluation through key performance indicators and stakeholder feedback. Each stage iterates on the others, ensuring adaptability to evolving requirements and technologies. In addition, it offers detailed recommendations and tailored results for each phase of the process.

To validate the approach, semi-structured interviews were conducted with SECO experts, confirming its applicability and highlighting areas for refinement. The primary contribution is a domain-independent and practical method that guides SECO developers through systematic, human-centred processes, enhancing interoperability, stakeholder alignment, and sustainability of SECO.

Keywords:
Software Ecosystems (SECO); human-centred design; ISO 9241-210; industrial case study

CERCS:

P170, Computer science, numerical analysis, systems, control

Lähenemisviis tarkvara ökosüsteemide inseneerimiseks tööstusliku juhtumiuuringu põhjal

Lühikokkuvõte:

Tarkvarapõhiseid ökosüsteeme (SECO) võib kirjeldada kui keerukat koostöövõrgustikku tarkvaraplatvormidest, komponentidest ja teenustest, mis tegutsevad ja arenevad koos ühises töökeskkonnas. Võrreldes traditsioonilise tarkvarainseneeriaga kujutab SECO inseneeria endast märkimisväärset väljakutset, eriti nõuete kogumisel, arhitektuuri kavandamisel ja sidusrühmade koordineerimisel, kuna traditsioonilised tarkvarainseneeria lähenemisviisid ei suuda SECO erivajadusi täielikult lahendada. Ehkki varasemad uuringud on analüüsinud SECO elutsükli konkreetseid etappe, näiteks nõudeid, arhitektuurilist kavandamist või kvaliteedi tagamist, puudub endiselt terviklik, valdkonnast sõltumatu lähenemisviis, mis juhendaks SECO kavandamise ja arendamise kogu protsessi. Lisaks käsitlevad enamik tänaseid SECO-sid inimest kui ökosüsteemi lahutamatut osa, mis suurendab nimetatud probleemide keerukust. Inimkeskse disaini standardid, nagu ISO 9241-210, pakuvad väärtuslikke juhiseid inimkesksete süsteemide loomiseks, kuid need ei ole kohandatud SECO spetsiifilisele dünaamikale.

Käesolev väitekiri käsitleb inimkeskse SECO inseneeriaga seotud väljakutset, töötades välja praktilise lähenemisviisi, mis on loodud SECO parimate tavade integreerimise teel ja PHArA-ONi (Pilots for Healthy and Active Ageing in Europe) projekti arenduskogemuste põhjal, millega luuakse digitaalne SECO, mis toetab heaolu, aktiivsust ja sotsiaalset osalemist Euroopa vananeva elanikkonna seas. Töö pakub välja viieetapilise inimkeskse SECO inseneeria lähenemise. Etapid on: (1) kasutuskonteksti määratlemine, kus tuvastatakse SECO eesmärgid, sidusrühmade rollid ja ülesanded; (2) nõuete kirjeldamine, mis seob sidusrühmade vajadused SECO eesmärkidega; (3) SECO referentsarhitektuuri kavandamine; (4) teostusplaneerimine ja SECO komponentide järkjärguline juurutamine; ning (5) pidev hindamine võtmetulemusnäitajate ja sidusrühmade tagasiside kaudu. Iga etapp itereerib teistega, tagades kohanemisvõime muutuvate nõuete ja tehnoloogiatega. Lisaks pakub lähenemine iga etapi jaoks üksikasjalikke soovitusi ja kohandatud tulemusi.

Lähenemise valideerimiseks viidi läbi poolstruktureeritud intervjuud SECO ekspertidega, mis kinnitasid selle rakenduskõlblikkust ja tõid esile täiendamisvõimalusi. Peamine panus on valdkonnast sõltumatu ja praktiline meetod, mis juhendab SECO arendajaid süstemaatilise inimkeskse protsessi kaudu, parandades SECO interoperabiilsust, sidusrühmade ühtlustatust ja jätkusuutlikkust.

Võtmesõnad:

tarkvara ökosüsteemid (SECO); inimesekeskne disain; ISO 9241-210; tööstuslik juhtumiuuring

CERCS:

P170, Computer science, numerical analysis, systems, control

Contents

1 Introduction **7**

2 Background **9**
2.1 Ecosystems: Key Concepts 9
2.2 ISO 9241-210 11
2.3 The PHArA-ON project 12

3 Methodology **14**

4 The process for constructing the approach **16**
4.1 SECO Context-of-use description 16
4.1.1 In PHArA-ON 17
4.2 SECO Requirements Specification 19
4.2.1 In PHArA-ON 23
4.3 Producing SECO design solutions 28
4.3.1 In PHArA-ON 30
4.4 Evaluating the SECO 35
4.4.1 In PHArA-ON 35

5 An approach for SECO engineering **38**
5.1 SECO context of use specification 38
5.1.1 Specify SECO goals and initial KPIs 38
5.1.2 Identify and classify stakeholders' roles, needs and tasks 41
5.2 SECO requirements specification 41
5.2.1 Systematic elicitation of SECO requirements aligned with defined goals 41
5.2.2 Incorporate stakeholders requirements with SECO goals 43
5.2.3 SECO requirements analysis 43
5.2.4 SECO requirements consolidation 46
5.3 Develop SECO design solution 46
5.3.1 Design SECO reference architecture 46
5.3.2 Validate SECO reference architecture 48
5.4 SECO implementation 48
5.4.1 Map reference architecture to implementation components and technical design 48
5.4.2 Plan and align development tasks among SECO contributors . . 50
5.4.3 Develop/adapt core platform and components with continuous technical documentation 50
5.4.4 Deploy incremental releases and communicate changes 51

5.5 SECO evaluation . . . 51
5.5.1 Specify SECO evaluation method and refine KPIs . . . 53
5.5.2 Set up centralised KPI collection and traceability mechanisms . 53
5.5.3 Perform evaluation and analyse cross-organisation results . . . 54

6 Validation **55**
6.1 Methodology . . . 55
6.2 Results and Discussion . . . 55

7 Limitations and future work **58**

8 Conclusions **59**

References **65**

Appendix **66**

A Semi-Structured Interview Plan **66**
II. Licence . . . 69

1 Introduction

With the growing complexity of some software applications, managing their design and development has become a key challenge. Consequently, the concept of a Software Ecosystem (SECO) has gained attention as an effective approach to managing and reducing this complexity by enabling seamless interaction and integration among various software components composing such complex systems [Bos09]. An SECO refers to an interconnected network of software components that operate within the same environment [Bos09, HD12]. Initially derived from ecological principles, the SECO concept has since been extended to encompass social and business domains, eventually finding significant relevance in software development, where it reflects the relationships between organisations, individuals, and software entities [HD12]. Consequently, unlike traditional development pipelines, SECOs operate as complex, decentralised networks that allow the creation, modification, and exchange of value in dynamic ways [HF18a]. This underscores the inadequacy of most traditional development approaches in addressing the challenges posed by SECOs.

Unsurprisingly, SECOs have gained traction in various industries [AS16]. Despite their adoption, there are gaps in SECO research, as identified in literature reviews [AS16, WWS+22, MH13, FFFLG14] and field studies [SM15]. For example, existing research often focusses on specific SECO components, domain-specific applications, or isolated development phases such as requirements engineering or architecture design [AS16, WWS+22, FFFLG14].

It is well acknowledged that accurate, complete, and consistent requirements help outline developer responsibilities, streamline time-to-market decisions, and enhance platform usability. Additionally, a robust SECO architecture ensures interoperability and supports external development on SECO platforms [AS16]. Consequently, the requirements and architecture of SECO have received considerable attention [AS16]. However, existing approaches appear limited to properly dealing with the requirements and architecture of SECO due to the complexity of these systems that involve various stakeholders that often have conflicting needs [KDKB14], which requires updating the architecture to meet the changes in requirements. This is confirmed by interviews with SECO developers, who reveal that leveraging past insights is crucial to designing new SECOs, but significant challenges remain in specifying accurate, complete, and consistent requirements and architecture to ensure SECO quality [SM15]. To this end, a comprehensive, domain-independent approach for the design and implementation of SECO remains a critical research gap [AS16, MH13, SM15].

This thesis aims to address these gaps by proposing a practical, human-centred design approach for the engineering of SECO [Int10]. This approach encompasses key phases of the SECO development life cycle, incorporating best practices from real-world industrial projects, namely the PHArA-ON project (Pilots for Healthy and Active Ageing in Europe)[1]. The PHArA-ON project aims to develop a SECO that improves the quality of life of Europe's ageing population through the integration of various digital technologies and services. Specifically, the PHArA-ON project utilises real industrial experience and provides valuable insights into understanding and engineering human-centred SECO. For these reasons, it has been used as a baseline for developing our approach.

The primary contribution of this work will be a practical human-centric design approach to engineering SECO, enabling SECO developers to design SECOs that meet the requirements of various stakeholders. To validate the approach developed, we will conduct interviews with SECO experts to verify clarity and completeness, alignment with practice, usability, and possible adoption barriers. Feedback received from experts will be used to iteratively refine the approach to better meet the needs of SECO developers.

The rest of this thesis is structured as follows. Section 2 introduces and elaborates on the fundamental principles of SECOs. Following this, Section 3 details the methodological framework used in the study. Section 4 delves into the procedural aspects of constructing an engineering approach for SECOs. Section 5 discusses the SECO engineering approach that has been developed. Section 6 validates the SECO engineering process through semi-structured interviews with experienced SECO practitioners. Section 7 summarises the main limitations observed during the study and outlines the directions for future research and industrial adoption. Section 8 concludes the thesis by highlighting the main contributions and their implications, after which references and appendices are provided for completeness. In addition, AI-based tools were used exclusively as supporting instruments for processing existing text to improve formatting and overall text clarity.

[1] https://www.pharaon.eu/

2 Background

2.1 Ecosystems: Key Concepts

The term "Software Ecosystem" (SECO) first appeared in 2005 and was defined as "a collection of software products that have some degree of symbiotic relationships" [MH13]. In 2009, Bosch expanded on this, describing SECO as "a set of software solutions to enable, support, and automate the activities and transactions of actors in the associated social or business ecosystem and the organisations that provide these solutions" [Bos09]. Hanssen et al. [HD12] presented a more general definition, incorporating previous definitions, describing SECO as a set of software platforms used by internal and external developers, together with a community of domain experts who collaboratively develop component solutions to meet user needs.

The concept of an ecosystem, which originated from ecology, has been adapted to social and commercial contexts. In a business ecosystem, actors include companies, individual customers and their associated processes that share knowledge and information [Bos09]. In an SECO, the primary distinction is the use of a platform that provides a common technological foundation [HF18a]. A platform is typically a product line architecture with shared components; Software becomes part of an SECO when its owners choose to make the platform available to external parties [Bos09]. The main difference in SECO relays in the value consumption process is that they are not consumed at the end of the pipeline, but are created, changed, exchanged, and consumed in different ways and locations [HF18a]. In contrast to managing SECO, opening up an SECO platform to allow external developers and companies to form a community provides significant benefits, such as enhanced offerings for customers and reduced development complexity [AS16]. In addition to improving customer services, Bosch [Bos09] identified other reasons for adopting SECO, including accelerating innovation by collaborating with partners and transforming successful features developed by partners into platform components. Many existing successful SECO platforms started as single products and once gained popularity and success, they opened platforms for additional component developers. Such examples can be fond among EBay, Google, Facebook, Apple, iOS, Salesforce and Dropbox [HF18a].

Typically SECOs consist of actors, artefacts, services, relationships, transactions, and networks [FFFLG14]. The functioning of a SECO includes the interconnection of many actors with different roles, each covering important aspects. Several studies highlight major roles in SECO with slight variations, such as keystone players, niche players, and users (see Figure 1) [KDKB14, Val13, AS16]. Keystone players create communities of actors by developing a platform that allows most of the value creation to be done by other actors. Niche players create specialised functions for the SECO and generally constitute the majority of actors [KDKB14]. Some studies also include users as the third role in defining the group of stakeholders that consume the services

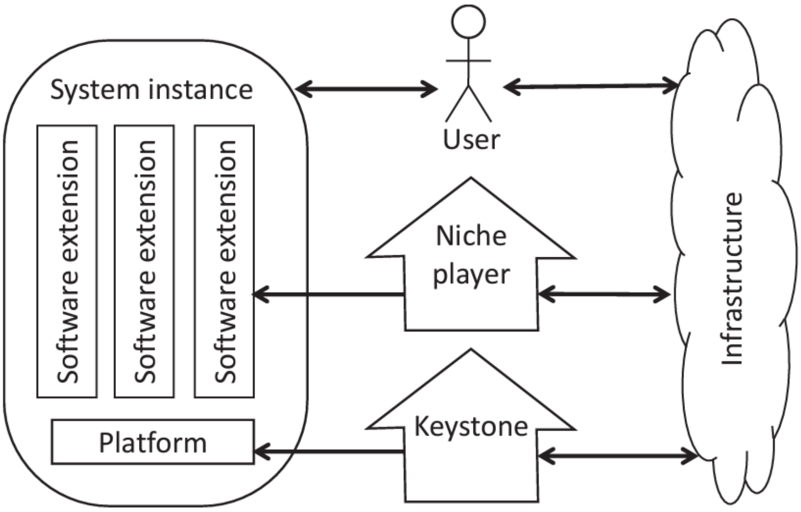

Figure 1. Technical and organizational concepts in a SECO [AS16]

proposed by SECO without offering their own [KDKB14]. Such differences in actors' roles compared to other systems result in split in operational knowledge distributed among actors. As a result, no one has a full picture of the system which adds another level of difficulty [AS16]. In addition, some studies view an SECO as composed of three different layers: technical, business, and social [HD12]. The technical layer includes the engineering architecture and the development processes of the SECO. The business layer encompasses management, value creation, and decision-making. The social layer relies on communication and interaction between actors [AS16].

The SECO research area has gained significant popularity in recent years, as evidenced by various reviews in the literature [WWS+22, FFFLG14, AS16, HD12, HF18a, AMAGC22]. These studies highlight the growing importance and challenges of SECO. Hanssen et al. [HD12] conducted a systematic literature review, noting gaps in SECO definitions and analytical models, with limited research in real-world contexts. Fotrousi et al. [FFFLG14] explored the use of Key Performance Indicators (KPIs) to assess SECO health and sustainability, identifying gaps in practical application. Knauss et al. [AMAGC22] analysed requirements definition and prioritisation in SE, particularly focussing on balancing openness with confidentiality. Challenges include managing cross-cutting concerns, stakeholder selection, and scaling efficiently. Hedges [HF18a] examined the role of the SECO architecture, focussing on strategic and governance challenges. Axelsson and Skoglund [AS16] addressed quality assurance issues in SECOs and proposed a research agenda to improve multi-organisational quality practices. Wulfert et al. [WWS+22] developed design principles for SECO to enhance interoperability, simplify developer participation, and promote growth, but only within the e-commerce

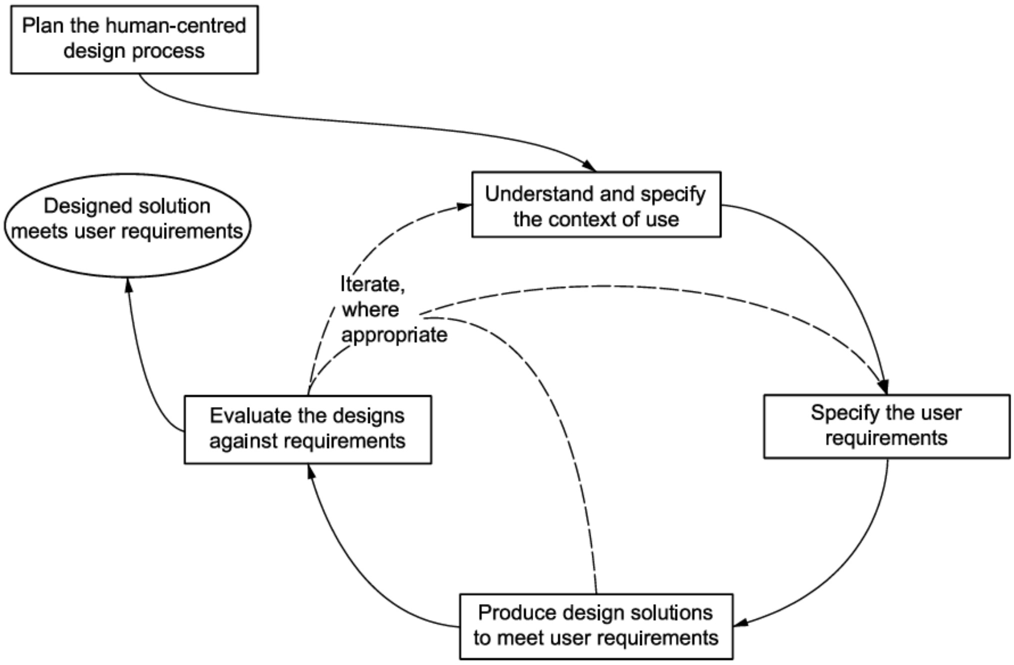

Figure 2. Interdependence of human-centred design activities [Int10]

domain. Collectively, these studies show the main trends in current SECO research with a focus on requirements, architecture, or quality and indicate the lack of comprehensive approaches to engineering SECO, covering all aspects, applicable across domains, and connected to real-world contexts.

2.2 ISO 9241-210

ISO 9241-210:2010 is an international standard that provides guidance on human-centred design for interactive systems [Int10]. It defines principles and activities that help ensure that the systems are usable and meet the user needs effectively. The standard promotes understanding of users, including them throughout development, evaluating designs with user feedback, and adopting an iterative process. It emphasises designing the entire user experience and assembling multidisciplinary teams [Int10].

ISO 9241-210 is intended for those managing design processes and can be applied across diverse development models. It supports improving usability, accessibility, satisfaction, and overall human well-being in the use of technology [Int10]. The standard suggests that the following human-centred design activities be implemented throughout the life cycle (see Figure 2):

1. **Understand and specify the context of use**: Identify users, their tasks, environ-

ments, and any constraints to ensure that design decisions are based on real-world situations.

2. **Specify the user requirements**: Translate the understanding of context into clear user needs and usability goals that the system must meet.

3. **Produce design solutions**: Develop and refine design alternatives that address the specified user requirements. These can range from sketches to interactive prototypes.

4. **Evaluate the design against the requirements**: Use user feedback and usability testing to assess how well the solutions meet user needs. This step helps identify improvements and supports iterative design.

These activities are repeated iteratively, with users involved throughout the process, to ensure continuous alignment with user expectations and system effectiveness [Int10].

2.3 The PHArA-ON project

The PHArA-ON SECO engineering process serves as the basis for our approach. PHArA-ON (Pilots for Healthy and Active Ageing in Europe) is a project developed under the EU Horizon 2020 programme to address the growing need for smart solutions that support the well-being of older adults. With a growing preference among older adults to stay in their own homes, combined with a shortage of caregivers and the high cost of home care, there is an increasing demand for technologies that can assist in active ageing. The main goal of the PHArA-ON project is to improve the quality of life of older adults by addressing their health, well-being, and social engagement needs at multiple pilot sites in Europe. The SECO primarily serves older adults, healthcare professionals, and formal and informal caregivers. To meet the needs of these users, a user-centric approach has been adopted, which allows them to access various services focused on well-being, social interaction, and training.

The purpose of the PHArA-ON project is to design and implement the PHArA-ON ecosystem, a network of people, software, and hardware that work dynamically and evolve to bring value to all stakeholders [oTUFS+21]. The project will follow a human-centric approach to maximise the usability and acceptance of the PHArA-ON SECO among all stakeholders. The project begins by gathering the requirements of stakeholders, particularly older adults, at each pilot site. These requirements are revisited and refined during the development, deployment, and pre-validation phases of the pilots. Continuous feedback from stakeholders guides the evolution and improvement of the PHArA-ON SECO, ensuring that it remains relevant, effective, and capable of meeting the dynamic needs of its users.

PHArA-ON addresses several key challenges, including making technology more elderly-friendly, defining and monitoring health status, enabling non-intrusive monitoring, and fostering social cohesion. Additional challenges involve creating personalised care plans, reducing loneliness through digital tools, improving accessibility through IT platforms, improving cultural awareness, ensuring indoor environmental quality, and supporting caregivers with more personalised services. To overcome these challenges, PHArA-ON leverages advanced services, devices, and tools to build an integrated, customisable, and interoperable platform. This platform, developed using technologies provided by consortium members or third parties through open calls, utilises IoT, AI, and other innovations to create a scalable solution that seamlessly integrates services such as health monitoring, cognitive stimulation, and community building [GFN+24].

Requirements for the PHArA-ON SECO have been gathered from six pilot locations: Spain (Murcia and Andalusia), Portugal, the Netherlands, Slovenia and Italy - each focussing on different challenges. The SECO will undergo two phases of validation: pre-validation and large-scale pilot (LSP) validation at these pilot sites, ensuring its effectiveness and readiness for broader implementation [GFN+24]. By treating human operators as core elements of the system and integrating their feedback throughout the project lifecycle, PHArA-ON aims to create a supportive environment that enhances well-being, encourages social interaction, and provides accessible and effective care for older adults.

3 Methodology

The proposed human-centred design approach for engineering Software Ecosystems (SECOs) follows a Design Science Research (DSR) approach as described by Bell et al. [BdCI+07]. DSR involves identifying the problem, developing a solution, and evaluating how well the solution works [AAS+04]. DSR is a research approach that focusses on creating and testing new artefacts that solve practical problems. These artefacts can be models, methods, frameworks, or systems that help address specific needs. DSR involves an iterative process, in which the artefact is refined through several iterations of testing and improvement [BdCI+07]. This method combines design and research to ensure that the results are scientifically sound and useful in practice. By applying DSR, we develop an approach for SECO engineering that is suitable for real-world situations, ensuring the outcomes are relevant, practical, and contribute to the field's knowledge base. The methodology in terms of its four steps is shown in Figure 3, and is described as follows:

1. **Problem Identification**: As described earlier, the complex, collaborative, and decentralised nature of SECOs requires a customised approach for each stage of development. This includes having well-defined requirements and architecture to ensure effective integration, interoperability, and overall quality. An analysis of studies covering the main stages of SECO development - requirements, design, solution development, and evaluation - highlighted a lack of guidance for designing and implementing SECOs through all of them. This thesis addresses this gap by proposing a structured and practical approach to SECO engineering for the defined development lifecycle, incorporating best practices from the industry.

2. **Approach Design**: Drawing on insights from the PHArA-ON project, this Approach Design integrates established best practices and proven standards to form a comprehensive framework for SECO engineering. Such best practices prioritise reliability, scalability, and efficiency [Som15]. The framework structures the development of SECO into four iterative phases: Context of Use (CoU) specification, Stakeholders' Requirements (SR) definition, Design Solution (DS) production, and Evaluation of the Design Solution (EDS) guided by ISO 9241-210 [Int10]. Each phase includes a step-by-step methodology that ensures internal consistency and alignment with real-world strategies.

   Importantly, the approach also adopts a human-centred design in every phase: defining the context of use, eliciting user needs, or iteratively refining the solution, the user perspective is integral. This human-centred emphasis helps create SECO solutions that are not only robust but also intuitive, accessible, and closely tailored to stakeholder expectations. In addition, the framework outlines key concepts, recommended activities, and the intended results for each phase, ensuring a repeatable and actionable development process.

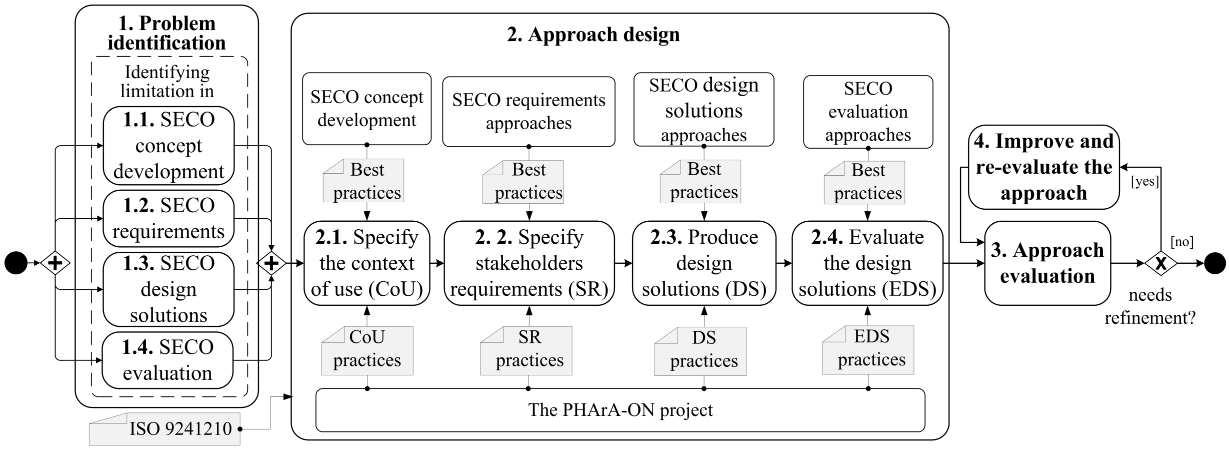

Figure 3. DSR Method Application

By combining industry best practices with user-centric practices, this structured approach offers SECO designers a clear path to manage the entire SECO lifecycle, from initial scope and requirements to final evaluation of the solution. The finer details of each phase are discussed in Section 4.

3. **Approach Evaluation**: The evaluation involves interviews with SECO developers to gather feedback on the proposed approach. Experts will be selected from key areas, including SECO requirements, architecture, design and evaluation, with the participation of professionals involved in the PHArA-ON project. These experts will assess the practicality and comprehensiveness of the approach within their respective domains, identify areas for improvement, and validate its real-world applicability based on their experience with SECO engineering. Their insights will provide valuable input on the framework's relevance from an industry perspective, ensuring that it addresses the actual needs of SECO stakeholders. An in-depth explanation of this step will be provided in Section 6.

4. **Improve and Re-evaluate the Approach**: This phase focusses on making improvements based on feedback from expert reviews. We identify the limitations of the approach and explain how these areas can be improved in the future.

Using this methodology, the research aims to fill existing gaps in the engineering of SECO, providing practical solutions that can be used in different domains and meet the needs of stakeholders.

4 The process for constructing the approach

This section discusses the process for constructing the SECO engineering approach (depicted in Step 2 in Figure 3). The approach has been constructed by integrating SECO best practices with the experiences used for the PHArA-ON ecosystem, considering the four key activities in the human-centred design process offered in ISO 9241-210 [Int10]. Highlights how the human-centred design activities outlined in the standard are implemented in real-world SECO engineering processes. The PHArA-ON project serves as a practical example of applying these principles in a large-scale, health-orientated SECO. In PHArA-ON, human-centred design is evidenced through active user participation, co-design activities, and iterative evaluation across diverse pilot sites. This underscores PHArA-ON as a strong example of how the principles of ISO 9241-210 [Int10] can guide the development of SECO. In the rest of this section, we present and discuss how each of the four key steps in the approach (e.g., 4.1. SECO Context-of-use description, 4.2. SECO Requirements Specification, 4.3. Producing SECO design solutions, and 4.4, Evaluating the SECO) was developed, using the best practices of SECO engineering, followed by how it was followed and used in PHArA-ON the project. The findings of this analysis lay the foundation for the SECO engineering approach, rooted in human-centred design activities, which is presented and discussed in Section 5.

4.1 SECO Context-of-use description

SECO is usually developed to achieve a specific goal or a set of goals. Certain studies offer their interpretations of SECO definitions alongside the articulation of SECO objectives. For example, the work by Knauss et al. [KDKB14] highlights a method for following the SECO definition known as *The five-themes approach*. This involves the establishment of a Product Management Committee (PMC) that includes members of the technical, development, and business sectors to outline the strategic themes of SECO, thereby allowing it to attract potential new participants [KDKB14].

The paper by Christensen et al. [CHKM14] illustrated the development of SECO that was driven by the objective of addressing integration issues within telemedicine systems that resulted in data duplication in various formats, missed opportunities for reuse due to overlapping architectures and proprietary designs, and reduced feasibility of cohesive telemedicine systems, thus complicating system construction. These challenges were identified through an analysis of Danish Telemedicine SECO, revealing issues such as fragmented IT frameworks, increased development expenses, and inadequate interoperability [CHKM14]. The SECO described in the article by Kruize et al. [KWS+16] where the goal of SECO was to create a shared environment in which many different agricultural software solutions can be seamlessly combined, helping farmers manage processes more effectively.

The paper by Cioroaica et al. [CPBS21] further contributes to this topic by proposing a structured model for the SECO goals. The authors classify the goals into three levels: strategic goals (e.g., reducing emissions), tactical goals (e.g., forming collaborations between SECO entities) and operational goals (e.g., reducing vehicle acceleration to improve fuel efficiency). This hierarchy of goals helps SECO participants align actions with high-level objectives and supports trust evaluation. Moreover, they introduce the concept of anti-goals, which define boundaries for trust reasoning by identifying potentially harmful or undesirable outcomes. This model enables both top-down planning and bottom-up detection of deviations, supporting dynamic goal management in SECOs. In addition, the paper proposes a formalism for goal representation that supports expression in natural language, traceability across abstraction levels, and run-time analysis. This formalisation allows SECO participants to evaluate progress, detect deviation, and refine strategies as needed, thus improving transparency and system adaptability.

In addition to the identification of goals, stakeholders and their definition of roles are considered as part of the definition of the SECO context of use. According to several research [KDKB14, HJB13, AS16], the stakeholder roles in SECOs are commonly divided into three main categories based on the relationship with SECO: *keystone players*, *niche players*, and *users*. *Keystone players* typically drive development and provide the core platform, focussing on requirements derived from their direct experience with the product. For example, senior developers prioritise managing technical debt, while technical leads concentrate on architectural decisions. Development leads allocate and oversee requirements execution, and support staff facilitate customer-driven requirements. Product managers specify strategic requirements aligned with the overall vision of the product [KYB+18]. *Niche players* contribute specialised components and are influenced by the technical decisions of keystone players. Their requirements elicitation process mirrors that of keystone players, but is restricted by dependencies on core platform decisions [KYB+18]. *Users*, through service consumption and feedback, refine and shape SECO [AS16]. The elicitation of their requirements is focused on strategic goals and operational needs [KYB+18].

4.1.1 In PHArA-ON

In PHArA-ON from the beginning, the initial goal was defined as part of the European H2020 project to enable smart and active living for the ageing population of Europe [oTUFS+21]. Then, on the basis of it, ten top-level goals (called challenges in the context of the PHArA-ON project)) were defined (see Table 1). These challenges focused on improving the well-being and autonomy of elderly people through technology, addressing aspects such as health monitoring, social cohesion, accessibility, caregiver support, and environmental quality. Each challenge (PCH1–PCH10) highlighted a specific set of needs, such as non-intrusive monitoring, personalised care plans, and the promotion of digital tools for connectivity and inclusion [oTUFS+21]. The respective Consortium Members (CM) were assigned tasks related to the SECO goals, deriving initial requirements, architecture, and design solutions, and were also involved in the evaluation of these solutions relevant to these PCHs.

ID	Description
PCH1	The behaviour and the approach of elderly to friendly technological devices
PCH2	Health status definition and its progress over time
PCH3	Non-intrusive monitoring and alarm triggering
PCH4	Promote social cohesion
PCH5	Define specific personalized care plan on the basis of user's needs
PCH6	Reduce loneliness and enhance autonomy through connectivity and digital tools
PCH7	Promote accessibility and provision of proximity services through IT platforms
PCH8	Promote capacity building and awareness on citizenship and cultural traditions
PCH9	Indoor environmental quality
PCH10	Support to caregivers toward more efficient and personalized care services

Table 1. PHArA-ON Challenges (PCH) [GFN+24]

In addition, in the PHArA-ON project, stakeholders were organised into multiple levels, reflecting their diverse roles and contributions to the SECO. The keystone players in PHArA-ON included the Platform and Software Components Provider and the Solution Provider, responsible for the overall architecture, integration, and governance of the SECO [oTUFS+21]. They ensured the interoperability of services and technologies between various pilot sites.

The niche players were represented by various service providers such as Emergency Service Providers, Transportation Service Providers, Digital Skills Course Providers, and Delivery Providers, who introduced innovative solutions tailored to specific pilot needs [oTUFS+21]. Their role was crucial in expanding the functionality of the SECO and filling in places keystone players may not address directly.

Users in PHArA-ON, such as Older Adult and Informal and Formal Caregivers, Healthcare Professionals, Family Members, Friends / Peers, took an integral role in the co-design and validation processes [oTUFS+21]. Their feedback loops were essential for ensuring that the SECO remains user-centric and responsive to evolving demands. Furthermore, the PHArA-ON project also incorporated other roles, such as experts, researchers, and other participants, who guided the compliance and scalability of the SECO [oTUFS+21].

4.2 SECO Requirements Specification

Requirements specification plays a crucial role in SECO engineering by defining functional and non-functional requirements, managing trade-off, and ensuring SECO sustainability [HD12]. It involves elicitation, analysis, classification, prioritisation, and validation of system requirements to ensure alignment with the needs and goals of stakeholders [Som15]. In the remainder of this section, we discuss the most common approaches for each of these activities.

Requirements elicitation encompasses the process of gathering system requirements by observing existing systems, discussing with users and stakeholders, and analysing tasks. This may involve creating models and prototypes to clarify the system's needs [Som15].

The researchers propose different methods that can be used to specify the requirements in SECO. A study by Chirumamilla et al. [CNDGS18] indicates that the goal-orientated requirements engineering (*GORE*) such as the modelling of *i* * modelling or *KAOS* as well as reference models are one of the ways to describe SECO requirements. *GORE* is a structured approach to requirements engineering that focusses on goals as primary entities to capture what a system should achieve [vL01]. The *GORE goals* represent the desired properties or states, ranging from high-level strategic goals to technical details. The goals here can be categorised into various types. As an example, the goals could be functional (specifying user-specific system services) and non-functional (specifying quality attributes such as security, safety, performance, usability, flexibility, and interoperability) [vL01].

The elicitation of stakeholder requirements has been explored in certain studies. The research by Knauss et al. [KBKD12] discussed approaches to capture some of the stakeholders' requirements through end-user observation within SECO, aiming to increase user participation and understanding their needs. We briefly describe these methods as follows:

Methods such as *DO-BE-FEEL* [LLBS18], *HOW-NOW-WOW* [ET13], and empathy maps [FSOC15] provide valuable tools for gathering and organising stakeholders' goals.

- *DO-BE-FEEL* emphasises capturing what users need to do, who they want to become, and how they want to feel when interacting with the system [LLBS18].
- *HOW-NOW-WOW* categorises ideas based on feasibility and innovation: 'Now' investigates the current problematic situation, 'Wow' envisions a preferred future state, and 'How' develops actionable solutions [ET13].
- *Empathy maps* assist in designing business models from the perspective of the customer. They go beyond demographic characteristics to develop a deeper understanding of the customer's environment, behaviour, aspirations, and concerns, fostering a greater degree of empathy for specific individuals [FSOC15].

Scenario modelling is a commonly used method used to visualise specific use cases and interactions, making abstract goals more tangible [SMMM98]. *Use cases* are iteratively developed alongside the analysis. Each *scenario* is described as 'a sequence of events representing one possible pathway through a use case' [SMMM98]. Consequently, multiple scenarios can be defined for a single use case, with each scenario illustrating a potential sequence of events. Another way of defining requirements could be *user stories*. Taking into account the recommendations provided in [Mai12], an effective definition of requirements in user stories often involves understanding the context in which the system will operate and clearly defining the role of the user, the desired capabilities and the goals. So, user stories should be precise, avoiding ambiguity, and consider how new features will impact user behaviour.

The study in [MPLL$^+$15] highlights the benefit of an agent-orientated approach for complex systems. Various methodologies such as *Prometheus*, *Gaia*, *INGENIAS*, and *ROADMAP* can be used, each providing different approaches, methods, and notations. Despite their differences, they share common modelling concepts, including agents or roles to represent interactions, goals instead of functional requirements to capture system behaviour, and quality goals for non-functional requirements. These concepts offer a natural way to describe software systems and their broader sociotechnical context.

Requirements Analysis involves interpreting and structuring the requirements collected to ensure that they are clear, complete, and consistent [Som15]. Damian in el. defines a bunch of challenges that must be added to SECO during that stage. Keystone players face significant difficulties in meeting the diverse needs of stakeholders, maintaining a balance between their own interests and the health of the SECO, and fostering collaboration in competitive settings [DLJ$^+$21]. Also, for keystone players prioritising self-interest excessively may push away niche creators, while ignoring innovative requirements can lead to a loss of competitive edge. To address these issues, solutions involve the implementation of collaborative requirements processes that encourage co-innovation, the adoption of strategies that maintain the health of the SECO, and the use of structured guidelines for decision making [DLJ$^+$21]. According to [DLJ$^+$21], the experiences of Xero and Shopify serve as illustrative examples, demonstrating the success of such strategies in navigating the intricate dynamics of SECOs.

Requirements classification is the process of grouping unstructured requirements into categories [Som15]. Often requirements are classified into two broad categories *functional* and *non-functional* requirements that can include several subcategories [Som15]. *Functional requirements* typically emerge from user needs and define the behaviour of the system, while *non-functional requirements (or quality requirements)* represent essential quality attributes and have a broad and global impact on software systems and play a crucial role in linking technical requirements with business objectives

[AMAGC22, DLJ+21, CNDGS18]. In addition, domain-specific requirements that originate in the system application domain rather than the specific needs of system users must also be considered during classification [Som15]. This might involve introducing new functional requirements independently, placing limitations on current functional requirements, or specifying the procedures to perform certain computations [Som15].

Focussing on *non-functional requirements*, the study by [AMAGC22] underscores the link between such requirements and SECO health. It highlights how quality attributes such as performance and usability can foster trust and potentially increase the robustness and productivity of an SECO. The authors also point out trade-off that can affect the vibrancy of the SECO [AMAGC22]. Balancing these trade-off is therefore crucial to preserving both software quality and overall SECO health [AMAGC22].

At the same time, *cross-cutting concerns* are requirements or issues that influence multiple parts of a SECO and cannot be handled solely by one component or actor [KDKB14]. These are not necessarily non-functional requirements; they can also involve functional aspects that span various parts of the SECO. Typical cross-cutting concerns include responsibilities like security, privacy, or interoperability, which often require coordination between the platform and its peripheral actors. When different customers request features that overlap in scope, practitioners must decide whether these concerns should be managed at a central or a more peripheral level in the SECO [KDKB14].

Interoperability is cross cutting concern that defined as the ability of different systems or components to exchange and use shared information in a meaningful way, typically by adhering to common standards and interfaces [RCL14, WBSE15, Gas15]. It ensures that the equipment or software of various providers can communicate seamlessly, enabling efficient collaboration in a network environment [Gas15].

Taking into account such cross-cutting concerns as *security* and *data privacy* the following definitions can be found in the literature. *Security* is defined as a concept standardised by the International Organisation for Standardisation that involves maintaining the confidentiality, integrity and availability of information, with additional attributes such as authenticity, accountability and reliability [PRRT14]. In addition, *cybersecurity* protects data from attackers and is essential for individuals and organisations, playing a key role in preventing cyber threats [RB22].

Confidentiality ensures that information is not disclosed to unauthorised individuals or entities, while data availability ensures that data remain accessible to authorised users when necessary [PRRT14]. *Privacy*, recognised as a fundamental human right, is related to controlling what others or information systems can know about an individual. In regulatory contexts such as the European Union and the United States, the notice-and-choice model (consent-based privacy) is commonly employed to ensure individuals have control over their personal data [PRRT14].

Some studies explore *emotional requirements* as a domain-specific concern, focussing on how users want to feel when interacting with the system and the community around it [MPLL+15, GFN+24, TSP+19]. These requirements are considered essential as functional or quality goals because they influence acceptance, trust, and continued engagement with the SECO [TSP+19, GFN+24]. By addressing emotional needs, the SECO can go beyond performance or usability goals to support the emotional well-being of its participants, fostering stronger relationships between stakeholders and encouraging user loyalty [MPLL+15].

Requirements Prioritisation plays a crucial role in managing stakeholder needs, helping to determine which requirements should take precedence based on factors such as stakeholder influence, feasibility and impact[Som15]. For instance, a project's technical team might prioritise system performance, while users may focus more on usability, leading to potential clashes in requirements. In such cases, if conflicts persist, an additional sub-action, such as *requirements negotiation*, can be employed to resolve differences [Som15]. Various requirement prioritisation and negotiation models are applicable as described in [Val13]:

- *EasyWinWin*: A collaborative requirement negotiation framework supported by software tools that guide stakeholders to mutually satisfactory agreements [Val13]. The software facilitates this process by offering structured workshops, interactive feedback sessions, and prioritisation tools that allow stakeholders to identify key requirements, address conflicts, and reach consensus efficiently. This approach emphasises cooperation during decision making.

- *StakeSource2.0*: A Web-Based Tool for Prioritisation of Distributed Requirements [Val13]. It constructs social networks of stakeholders, collects their recommendations, and highlights conflicts to facilitate resolution. The tool gathers recommendations through surveys, collaborative ranking methods, and feedback collection [Val13].

- *Requirements Value Chains*: Proposed by Fricker [Fri10], this model integrates negotiation and network theory to manage the flow of requirements in SECOs. Optimise communication bottlenecks and stakeholder management by strategically assigning power in negotiations [Val13].

- *MAP Method*: Focusses on aligning shared goals and activities among networked organisations to facilitate requirement analysis and decision-making [Val13].

Requirements Validation is a critical step to confirm that documented requirements accurately capture the needs of users and stakeholders while remaining feasible and testable [Som15]. Typical requirement validation involves reviewing them with stakeholders to verify whether they capture real needs, remain consistent with other system requirements, and are feasible and verifiable. This usually includes asking whether they actually reflect user goals (*validity*), whether they conflict with anything (*consistency*), whether anything is omitted (*completeness*), whether they're practically achievable (*feasibility*), whether they can be tested (*verifiability*), and whether they're stated in a way that everyone can understand [Som15]. The study Chirumamilla et al. [CNDGS18] mentioned the importance of requirement validation in SECO requirements engineering by performing specification reviews. This case highlighted how well-organised reviews foster successful collaboration and help resolve ambiguities between SECO members.

4.2.1 In PHArA-ON

The PHArA-ON project demonstrates how a mix of various methods provides flexibility in specifying requirements, combining goal-orientated requirements engineering, agent-orientated goal modelling, and scenario modelling.

The stakeholder engagement in the PHArA-ON project was followed by a co-design methodology, ensuring that the perspectives of all actors were reflected from the initial phases. Stakeholder goals were elicited through three methods without digital technologies (face-to-face co-creation seminars, phone interviews and review of literature/previous projects) and four methods that involve digital technologies (online interviews, online questionnaires, virtual co-creation seminars, and semi-virtual co-creation seminars) [oTUFS$^+$21].

For example, Italy used face-to-face co-creation seminars (pre-COVID), phone interviews, and online interviews involving both older adults and stakeholders. The Netherlands focused on reviewing previous projects and employing online questionnaires and virtual seminars. Slovenia could only hold a semi-virtual co-creation seminar for older adults, while Portugal and Andalusia primarily conducted virtual co-creation seminars. Murcia supplemented its co-design activities with an online questionnaire. In particular, face-to-face efforts were carried out before the COVID-19 outbreak, after which most activities moved to remote formats. This distribution highlights how the pilots adapted their methods to local contexts and pandemic constraints, ensuring that they still involved older adults and various stakeholders in gathering user goals.

The *DO-BE-FEEL* framework was utilised to structure user input in designing a system. It involved identifying functional goals (*'Do'*) for user tasks, quality goals (*'Be'*) for system attributes like ease of use and security, emotional goals (*'Feel'*) for user experiences such as confidence and empowerment, and stakeholder roles (*'Who'*) involving older adults and caregivers. A uniform representation of the workshop results was chosen to align different pilot projects and enhance integration.

The requirements were first organised into goals models based on the result of the elicitation of the requirements. *DO-BE-FEEL* method allowed easy mapping to the *goal models* (see Figure 4).

Goal models were used to represent the needs of stakeholders in a container that included four components: *functional goals*, *quality goals*, *emotional goals*, and *stake-*

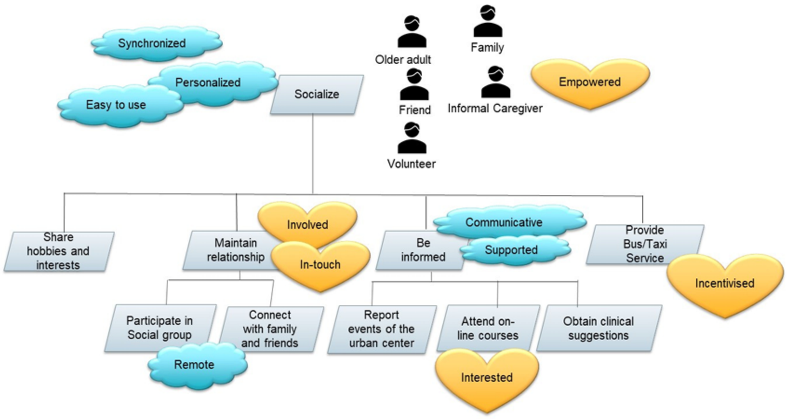

Figure 4. Example of the goal model of socialization from PHArA-ON project [oTUFS+21]

holder roles [MPLL+15]. The *goal models* were then transformed into *scenarios*, *user stories*, and *use cases*. Initially, *scenarios* were developed to represent the requirements based on *goal models*. *Scenarios* could also contain *sub-scenarios* linked to *sub-goals*. The scenarios were structured in a tabular format, specifying elements such as goals, initiators, triggers, and activity steps [oTUFS+21]. Each step represented an activity that could involve multiple roles, and these roles encapsulated the actual components or devices involved, ensuring platform independence (see Figure 5). Additional columns in the scenario table outlined quality goals, emotional goals, and planned technologies [oTUFS+21]. These scenarios guided developers and integrators. If new software components were required, scenarios could be supplemented with user stories [oTUFS+21]. For technical details such as system interactions or adjustments, use cases were used to explain specific needs [oTUFS+21].

The PHArA-ON *security requirements* were elicited from several sources such as software engineers, project stakeholders, security norms, and domain specialists in Active and Healthy Ageing [GJM+24]. The specification of the PHArA-ON security requirements incorporated insights from numerous international standards and frameworks, ensuring a comprehensive, secure, and regulatory compliant strategy for data protection and system security [GJM+24].

Taking into account the challenges of the PHArA-ON project, initially identified at the beginning of the project, the challenges were addressed during the requirement elicitation process. These challenges were assigned to target scenarios and addressed through goal models and scenarios developed during elicitation [oTUFS+21]. The requirements were systematically aligned with these challenges to ensure that the technological solutions directly responded to the user's needs, integrating user input to refine both the goal models and the scenarios. As an example of an Italian pilot, the mapping process ensured

SCENARIO PUCS_F01							
Goal		Be in touch with the caregiver					
Initiator		Older Person					
Trigger		Action by the Older Person					
Innovation		Assuring the formal or informal caregiver that everything is well and occurs naturally through following daily routine by the older person					
DESCRIPTION/ OUTLINE		Activities by the Older Person that enable him/her being in touch with the Carer/Relative and reassure the Carer/Relative that everything is well					
Condition		Step	Activity	Roles involved	Quality goals	Emotional goals	Technologies
Loop	Sequence	1	The Older Person communicates that he/she is well (Scenario PUCS_F02)	Older Person, In-touch Monitor	Accessible, Flexible, Invisible to others, Part of routine, Mobile, Attractive	Empowered, Independent, Engaged, In touch	Non-intrusive Alarm System by Happy Older Adult Ltd.
		2	The In-touch Monitor confirms that the communication has been registered	In-touch Monitor, Older Person		In touch	
	Option	3	The Carer/Relative acknowledges the communication by the Older Person	Carer/Relative, In-touch Monitor		Cared about	
Interleaved		4	Inform Carer/Relative	Older Person, In-touch Monitor, Carer/Relative		In control	
		5	Check on Older Adult	Carer/Relative, Older Person		Reassured	

Figure 5. Example of scenario of being in touch with the caregiver [oTUFS+21]

that the challenges were systematically linked to the goals and scenarios, making the requirements that were elicited actionable (see Table 2). PCH2 (health status definition and progress monitoring) was linked to the PGM I01 goal model and the scenarios PUCS I01 and PUCS I01.1, ensuring a structured approach to tracking and assessing older adults' health. Similarly, PCH4 (Promoting social cohesion) was associated with PGM I02 and PUCS I02, focussing on solutions that improve social engagement. The PCH5 challenge (Defining personalised care plans) was connected to PGM I01 and PUCS I01, ensuring that the care interventions were tailored to individual needs [oTUFS+21]. Furthermore, PCH10 (Supporting caregivers for more efficient and personalised services) shared the same mapping as PCH2, reinforcing the need for solutions that facilitate health monitoring and tailored support for caregivers and older adults.

This structured mapping of challenges to goal models and scenarios allowed the Italian pilot to translate user needs into concrete technological solutions. It ensured that the requirements elicited guided the selection and implementation of the appropriate technologies.

Requirements Classification: In PHArA-ON, a *goal-orientated requirement engineering approach* was used, classifying each requirement into one of three categories: functional, quality (non-functional) or emotional [oTUFS+21]. In practice, they treated functional requirements as functional goals, non-functional requirements as quality goals, and user feelings or motivational aspects as emotional goals [oTUFS+21]. This method helped address both the practical features of the solutions (e.g., 'Monitor health', 'Provide

Challenge	Goal model code	Scenario code
PCH2: Health status definition and its progress over time	PGM_I01	PUCS_I01, PUCS_I01.1
PCH4: Promote social cohesion	PGM_I02	PUCS_I02
PCH5: Define specific personalized care plan on the basis of user's needs	PGM_I01	PUCS_I01
PCH10: Support to caregivers towards more efficient and personalized care services	PGM_I01	PUCS_I01, PUCS_I01.1

Table 2. Example of challenges and their corresponding codes on Italian pilot in PHArA-ON project [oTUFS+21]

emergency alert') and the quality and emotional dimensions important to older adults (e.g., ease of use, sense of security, empowerment). In addition, *emotional requirements* described how users want to experience the system on an emotional level, for example feeling safe, confident, or socially connected [oTUFS+21, GFN+24]. During the development of the PHArA-ON project, it was ensured that emotions such as reassurance, engagement, or reduced loneliness would not be overlooked. Instead of subsuming them under quality goals, they were elicited and validated as first-class requirements, then incorporated into design decisions and final evaluations of the system [oTUFS+21].

In addition, *cross-cutting requirements* were addressed in the PHArA-ON, including *security*, *privacy*, and *interoperability*.

The *security* requirements specification was informed by multiple international standards and frameworks to ensure a comprehensive, secure and compliant approach to data protection and system security. The key standards used in the PHArA-ON cybersecurity framework covered various aspects of information security, health informatics, and cybersecurity governance [GJM+24]. So, for IT and information security, internationally recognised standards provided guidelines for managing security risks, defining controls, and ensuring data protection [GJM+24]. For *data privacy*, PHArA-ON adopted a Privacy By Design approach to ensure compliance with the General Data Protection Regulation (GDPR) throughout the project life cycle [TPRP24]. The approach integrates both traditional information security goals, confidentiality, integrity, and availability, and privacy-specific goals such as transparency, and intervenability. To ensure compliance with privacy regulations, PHArA-ON maintains records of processing activities, conducts Privacy Impact Assessments, and adheres to ISO 27701 [ISO22] for the management of privacy information [TPRP24].

The *interoperability* in The PHArA-ON project employs multiple platforms tailored to different pilot scenarios rather than enforcing a single unified platform. The PHArA-

Scenario ID	Goal of the Scenario	Activity Number (from the Scenario)	Activity (from the Scenario)	Priority
PUCS_F01	Be in touch with the caregiver	1	The Older Person communicates that he/she is well (Scenario 2)	3
		2	The In-touch Monitor confirms that the communication has been registered	2
		3	The Carer/Relative acknowledges the communication by the Older Person	1
		4	Inform Carer/Relative	3
		5	Check on Older Person	3
PUCS_F02	Communicate	1	The Older Person looks at pictures	3
		2	The Older Person likes pictures	3
		3	The Older Person sends a message to the Carer/Relative	3

Figure 6. Example of prioritisation of activities for the scenarios from PHArA-ON project [oTUFS+21]

ON interoperability strategy focusses on evolving these platforms with new features, increasing standardisation, and applying open-source solutions when possible [RB+23]. These platforms are integrated into PHArA-ON's to ensure communication through the Secure Inter-Platform Interoperability Solution. Functional requirements include the ability of each platform to support multiple communication protocols, enable semantic data exchange, and ensure compliance with PHArA-ON's Secure Inter-Platform Interoperability Solution. In addition, third-party integration was allowed, ensuring that external contributors and service providers can connect their solutions through standardised APIs and open interoperability tools. [RB+23].

Requirements Prioritisation One of the ways is using the *streetlight approach* proposed in [FdMF+19] that was used in the PHArA-ON project (see Figure 6). The *streetlight approach* was applied to prioritisation activities between pilot sites. The approach involves summing the priorities assigned by individual sites to derive a final prioritisation. For borderline activities, such as those with one low priority and one maximum priority, internal discussions were conducted to confirm the activity. This method ensured a clear focus on actionable tasks and encouraged stakeholder consensus, promoting effective alignment and prioritisation of requirements.

Requirements Validation PHArA-ON project illustrates effective validation practices. In this project, user and stakeholder requirements were validated through a structured three-step process. This included completeness checks to ensure all functionalities and requirements were captured, consistency checks to resolve any conflicts among requirements, and realism checks to confirm technical feasibility. These steps facilitated the early identification and resolution of issues, minimising redesigns, and improving the reliability of the SECO.

In addition, the PHArA-ON project to ensure a structured approach to ***consolidate the SECO requirements***, the process involves elaborating these into technical requirements. Comprehensive requirements were formulated for the selected implementation and testing, with a focus on those of utmost priority and some of intermediate priority. [oTUFS+21]. This was achieved through a table that outlines the identifier of the scenario, the goals, the details of the activity, the technologies involved, and the corresponding technical requirements. The technical requirements referenced at least one of the listed technologies. Following human-centred design [Int10] and agile software engineering principles [Som15], requirements that involve stakeholders (e.g., older people or caregivers) are formulated using 'the stakeholder should be able to... with the help of technology.' Other technical requirements were framed as 'the technology should be able to...' [oTUFS+21].

4.3 Producing SECO design solutions

SECO design solutions involve defining modular architectures, ensuring interoperability, and incorporating governance frameworks to support scalability and sustainability [MH13]. *SECO architecture* defined as the structural and strategic design of digital systems, ensuring that interconnected organisations, platforms, and technologies work efficiently [HF18b]. In the research by Hedges et al.[HF18b] highlights key aspects of this concept, focussing on business ecosystems, platform models, governance, and architectural principles. [HF18b] mentioned four key principles for designing a successful SECO platform architecture: *simplicity*, ensuring ease of understanding at a high level; *resilience*, preventing failures in isolated components from disrupting the entire SECO; *maintenance*, allowing cost-effective modifications through modular design and standardised interfaces; and *extensibility*, allowing the platform to support unforeseen future capabilities [HF18b].

As noted in [Bos10], the implementation of the design of the SECO architecture also faces several significant challenges. It must provide a stable and predictable interface for external developers while supporting various levels of integration with external solutions, such as user interfaces, workflows, and data [Bos10]. In addition, the architecture must ensure seamless integration between external products, adding another layer of complexity. To maintain system security, the SECO architecture must minimise the risks associated with defective or malicious external code [Bos10]. This often requires implementing robust security measures and certification processes. This requires careful, forward-thinking design.

A *reference architecture* in the context of SECO serves as a foundational blueprint that defines common structures, design principles, and best practices to address recurring challenges within a specific domain by providing shared guidance to streamline development, promote interoperability, and allow collaboration between various actors within SECO [KM16]. According to a study by Knodel et al. [KM16] the use of the SECO reference architecture mitigates the risks of architectural mismatches, promotes innovation through collaboration, and accelerates the time to market for SECO participants. In addition, it helps maintain the sustainability of SECO by providing a framework to manage variability and align with changing market and stakeholder needs [KM16].

Common modelling frameworks for SECO architecture according to Anwar et al [AG19] include *ArchiMate* (for cross-layer visualization), *TOGAF* (structured enterprise methodology), *FAML* (agent-based modelling), *ISO/IEC/IEEE 42010* (architectural context and stakeholder-focused), *SABSA* (security alignment), and *ITIL* (service management best practices). These frameworks help ensure clarity, consistency, and adaptability between the business, application, and technology layers.

As an example presented in the article by Kruize et al. [KWS$^+$16] the architecture design model in the reference architecture for Farm SECO was applied using a structured and modular approach. They divide the system into key components, including actors, a platform, an open software enterprise, business services, and ICT components [KWS$^+$16]. These elements worked together to ensure seamless integration and interoperability. A SECO model was adopted, which allowed multiple vendors to develop modular software components that could be integrated into a unified system. This approach followed a best-of-breed methodology, allowing users to customise their farm management systems by linking different ICT components [KWS$^+$16]. The architecture emphasised modularity and configurability by defining the components of the atomic and composite application. This allowed for the flexible assembly of software solutions that could dynamically support various business processes. In the article by Kruize et al. [KWS$^+$16], the validation of the reference architecture was performed by comparing its design with the initial requirements. First, it was checked whether the proposed architecture met the functional and non-functional needs they had identified [KWS$^+$16]. The reference architecture was then applied to two existing SECOs, examining how well it captured their core elements and supported the integration of various ICT components [KWS$^+$16]. By verifying requirement coverage and testing the architecture in practice through these mappings, they demonstrated that the model could describe, compare, and reveal gaps or improvements.

SECO implementation involves the practical implementation and integration of the principles of the SECO design in the functioning of SECO [ZCM22]. The study by Zajdel et al. [ZCM22] addresses the development of open source SECO and explores the usage of *DevSecOps* for the management of deployments. *DevOps* automates the continuous delivery of software updates, focussing on correctness and reliability. *DevSecOps* extends this by integrating security practices with development and operations [ZCM22]. *DevSecOps* adds automated security stages, such as static application security testing (SAST) and source composition analysis (SCA), to the continuous integration pipeline (CI). Typically, code is committed, built, tested, and then tested for vulnerabilities or licence issues, which can be wasteful if severe problems arise late in the process. To address this, Zajdel et al. [ZCM22] propose moving open source usage checks (SCA) to the beginning of the pipeline. This shift ensures that critical vulnerabilities are caught early, reducing wasted effort and limiting the temptation to deploy risky code. Furthermore, study [WWS+22] proposes some design principles regarding the development of SECO that highlight the need for coding guidelines to improve code quality and communication, the establishment of a central repository for collaborative development and versioning, and the provision of tool templates to streamline setup time. They recommend creating a dedicated sandbox environment for testing and quality checks, offering demo scenarios and workflow testing mechanisms for greater clarity and control of SECO development processes, encouraging user interface prototyping with pre-tested design elements to unify user experiences, and ensuring real-time analytics for error detection and personalized offerings [WWS+22]. By embedding these practices, an SECO can significantly strengthen collaboration, quality assurance, and stakeholder engagement.

4.3.1 In PHArA-ON

The PHArA-ON project examined various potential reference architecture models, evaluating each based on factors such as scalability, interoperability, and focus on user needs. The chosen architecture model balanced modularity and interoperability, enabling seamless integration of services across various pilot sites while remaining user-focused [GCN+21]. Key reference architectures, including IoT models such as ISO/IEC CD 30141, IEEE P2413, oneM2M, RAMI 4.0, and IIRA, as well as EU project architectures, were analysed [GCN+21]. As a result, the modified *4 + 1 view model* was chosen, with additional views to improve its applicability [GCN+21] and was applied as follows:

1. *The PHArA-ON conceptual model* provides a description of the core concepts, their attributes, and their interrelationships. The goal was to ensure a consistent understanding between stakeholders, which helps in future collaborations and architectural evaluations [GCN+21]. The PHArA-ON Concept Model was structured as a System of Systems (SoS), where independent systems, represented by PHArA-ON pilots, collaborate towards a shared goal while maintaining a degree of autonomy (see Figure 7). Each pilot operated independently, ensuring operational independence while still contributing to the overall PHArA-ON SECO. The management of these pilots is also decentralized, with each led by different teams fulfilling similar roles, establishing managerial independence. The

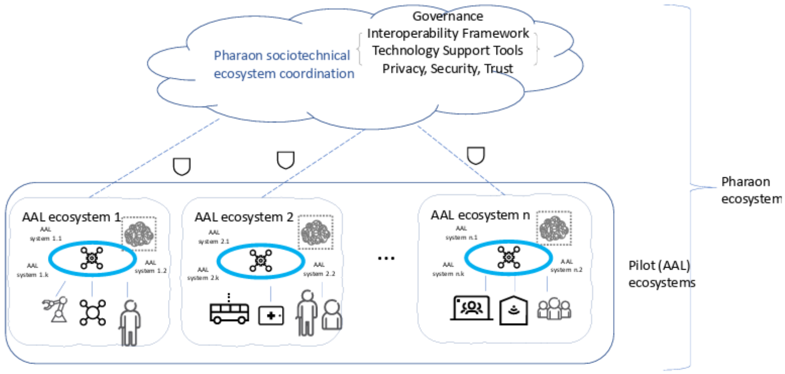

Figure 7. PHArA-ON SECO concept [GCN+21]

PHArA-ON SECO architecture delivers functionalities to users through a structured multi-layered system. The Device and Network Layer handles data collection, sensing, actuating, and connectivity. The Platform Layer ensures integration and interoperability of components, while the Service Layer supports application development and data analysis. The Application Layer focuses on user interaction with interfaces and tools for data visualization and control. The Collaboration and Processes Layer integrates with enterprise systems and business processes for seamless cross-domain interactions. Cross-cutting functions across all layers address interoperability, security, and privacy, enhancing the system's robustness and adaptability.

2. *The Process View* in the PHArA-ON system described its dynamic aspects, focusing on system processes, their communication, and runtime behaviour. It outlined key interactions between architectural blocks and components using sequence diagrams. Scenario realization flows demonstrated how functional blocks interacted to support user scenarios defined during requirement specification. Furthermore, different data flow examples illustrated how sensor data was pushed to the platform after pre-processing, while applications and services retrieved stored data based on predefined criteria through Context Management. These flows ensured efficient interaction between system components, balancing centralized and distributed processing while supporting various deployment models, including edge and cloud environments.

3. *The Information Layer* in the PHArA-ON SECO was defined to establish a shared domain-agnostic language that facilitated compatibility and interoperability between system components. This layer defined the relationships between infor-

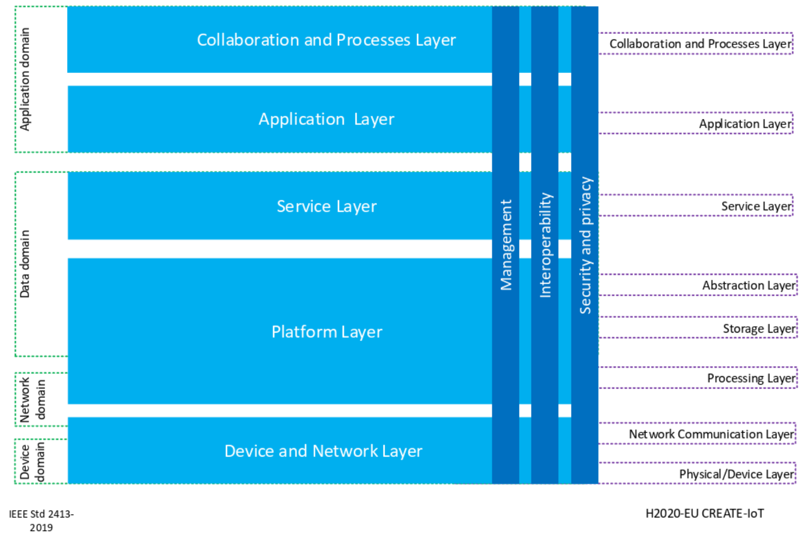

Figure 8. Example of PHArA-ON layered model and main cross cutting functions [GCN+21]

mation objects and structured data and metadata organization. The PHArA-ON SECO followed a Data-Centric Architecture, where data remained the core asset, independent of applications, ensuring long-term consistency and accessibility [GCN+21]. Unlike traditional systems where applications were tightly coupled with their own data models, PHArA-ON employed a unified semantic data model that integrated all functionalities. The architecture relied on ontologies, which defined data relationships and structures, similar to relational database tables, but offering richer domain models [GCN+21].

4. *The Communication View* in the PHArA-ON architecture facilitated seamless interaction between various technological components while minimizing integration efforts. It involved multiple logical flows to ensure effective data transmission and processing (see Figure 9).

5. *The Physical/System View* of the PHArA-ON architecture described the system through its physical components rather than its functional ones. It considered devices, subsystems, and networks without focusing on specific implementations. The system was divided into three main groups: Devices and Sensors, Network, and Data Center [GCN+21]. The PHArA-ON architecture ensured that data storage aligned with the platform layer, where computational nodes hosted various platform, service, and application layer functions. Some nodes were dedicated to managing environments, such as Kubernetes Master Nodes, to coordinate the

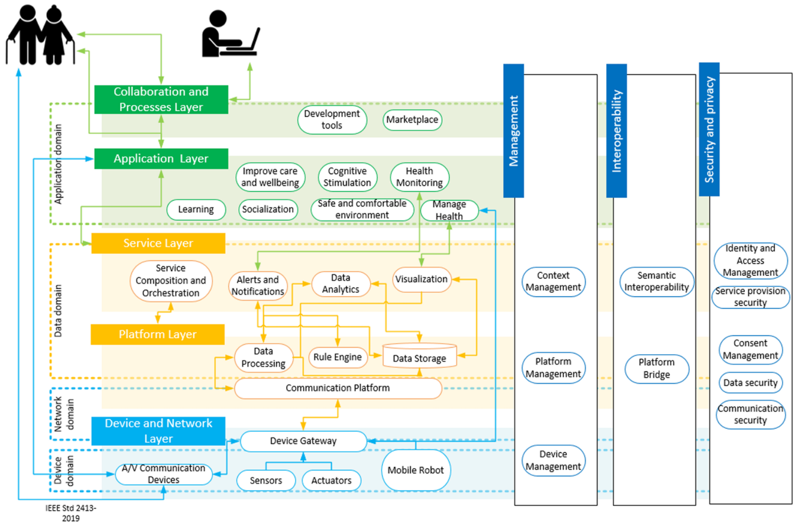

Figure 9. Example of Communication flow from PHArA-ON [GCN+21]

system's operation.

During *validation of the SECO reference architecture* in the PHArA-ON project, the objective was to refine and expand the technical requirements while identifying and validating the architectural components that aligned with those requirements. The interconnections between these components were assessed and classified according to their suitability level [GCN+21, oTUFS+21]. If necessary, new architectural components were introduced or existing ones were adjusted according to practical use cases. The architectural components were ensured to align with the revised technical requirements, incorporating feedback from the pilot testing phases. Subsequently, this agile approach supported the evolution of the PHArA-ON SECO in a user-centric manner, ensuring that it met the needs of users and met the final objectives of the project [GCN+21].

Then, the PHArA-ON project structured its development task planning using a work package-based approach, ensuring that each phase of the project was systematically managed and aligned with the overall goals. The development tasks were organised into a structured timeline with work packages assigned to different partners, each responsible for specific aspects such as interoperability, cybersecurity, pilot integration, and evaluation [RB+23]. Task execution was managed through iterative development cycles, allowing adjustments based on feedback from real-world pilot deployment [RB+23]. Furthermore, a centralised project repository (hosted on GitLab) provided access to the latest software developments and ensured alignment between different pilot sites [DGV+22].

The project also adopted Agile methodologies, incorporating incremental delivery

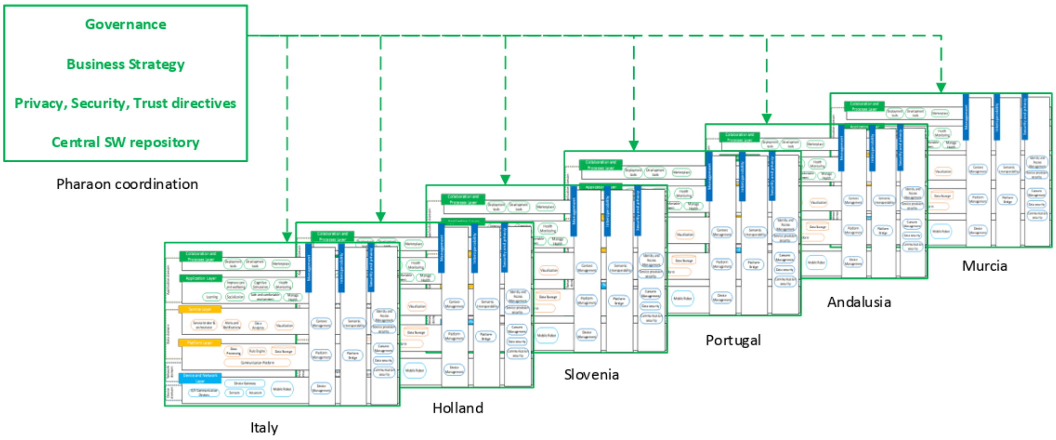

Figure 10. PHArA-ON SECO implementation [GCN+21]

models and continuous feedback loops. This approach allowed teams to iteratively refine solutions, prioritise high impact tasks, and integrate user feedback into the design, development, and validation phases [DGV+22]. The PHArA-ON SECO was implemented by combining a shared reference architecture with local adaptations at each pilot site. Each pilot reused existing solutions, such as the Onesait Healthcare Data Platform in Murcia, and supplemented them with PHArA-ON-provided components for data storage, analytics, governance, and device management [GCN+21]. These templates defined uniform APIs, which enabled standardised data extraction, execution of analytics, and device interaction (see Figure 10).

Although each pilot customised the architecture to its particular needs, the PHArA-ON **coordination layer** oversaw the governance, privacy, security and trust directives. A central software repository contained open source modules, PHArA-ON-specific components (such as IoT gateways) and technical documentation [GCN+21]. This modular design allowed pilots to include or omit certain layers based on existing technologies and requirements at each site.

In order to handle the deployment process, PHArA-ON adopted a *DevSecOps* based *CI/CD* process to integrate multiple technologies from different organisations under a flexible quality assurance approach. The project set up a shared cloud environment to support these pipelines [DGV+22]. Since some technologies had their own specific requirements and build tools, PHArA-ON allowed teams to use the project-wide *CI/CD* infrastructure, their private systems, or a mixture of both. This approach included Per-technology (pipelines to build, package, test, and deploy individual components) and Per pilot (pipelines that allow partial integration and testing between multiple partners) [DGV+22]. As a result, teams could adapt to constraints, potentially migrate pipelines to the shared environment, and still maintain consistent *DevSecOps* practices throughout the project.

4.4 Evaluating the SECO

SECO evaluation is the systematic assessment of structure, performance, and sustainability [AS16]. The evaluation of SECOs involves analysing various aspects, including architecture, governance, quality assurance, and interoperability, to ensure their efficiency and long-term viability [AS16].

The *health of a SECO* refers to its ability to survive over time while remaining adaptive and productive [SdSANM+17]. According to Amorim et al. [SdSANM+17], it was characterised by *robustness* (resilience of the SECO against disruptions), *productivity* (the capacity to generate continuous value) and *creation of niches* (ability to foster innovation and new opportunities). In addition, the authors highlight that metric-based assessments are commonly used for the evaluation of SECO health. This approach focusses on predefined indicators to help quantify the health of the SECO by evaluating factors such as software contributions, developer retention, and SECO adaptability [SdSANM+17].

The study by Fotrousi et al. [FFFLG14] presents a systematic mapping study on the use of Key Performance Indicators (KPIs) in SECO. According to the authors, both health and sustainability are key performance objectives commonly found in complex systems [FFFLG14]. Effective SECO management requires coordinating actors, software, and business models across business, technical, and social dimensions. They recommend platform owners to use KPIs to benchmark and monitor SECO behaviour [FFFLG14]. These indicators serve as early warnings for potential problems and help measure the health and sustainability of SECO. They define qualities like *size*, *financial aspects*, *satisfaction*, *performance*, *ability to mitigate risks*, and *compatibility* [FFFLG14]. *Size* reflects growth and scale, while diversity captures heterogeneity and openness. *Financial aspect*s include investment, costs, and pricing. *Satisfaction* covers usability, trust, reputation, and accessibility. *Performance* is assessed through efficiency, resource utilisation, and precision. The *ability to mitigate risks* involves security, reliability, and availability. *Compatibility* measures interoperability and portability, and maintainability reflects flexibility and adaptability. These attributes provide a way to systematically evaluate and improve SECOs [FFFLG14].

4.4.1 In PHArA-ON

In PHArA-ON the specification of the SECO evaluation methods and KPIs was carried out through a structured framework. The evaluation approach was designed to generate evidence on the impact of created services, focussing on health maintenance, user acceptance, and cost effectiveness. A harmonised methodology was implemented to ensure that each pilot site follows a shared procedure for data collection and impact

KPIs	Methods for data collection	Data collection timing
Sustainability category (Primary end-point)		
QoL	1) EQ-5D-3L (CarerQ-7D for formal/informal carers)	Baseline (M00), intermediate (M06) and final analysis (M12)
Social Isolation	1) UCLA Loneliness scale version 3	Baseline (M00), intermediate (M06) and final analysis (M12)
Cost Analysis	1) Cost Utility Analysis (and ICER) 2) Willingness to Pay	Final analysis (M12)
QoL category (secondary end-point)		
Users' acceptance	1) SUS questionnaire 2) ICT technologies usage	PS per PS (ref. par. 3.10)
Innovation and Growth category (secondary end-point)		
Local replication plans	No. of local replication plans for AHA-IoT services	Final analysis (M12)

Figure 11. Example of PHArA-ON's Global KPIs for end-points assessment

assessment. This process is guided by an "Evidence Generation Protocol," which defines the indicators, targets, and tools to measure the results. The evaluation framework follows a co-design approach to identify a mandatory set of common KPIs applicable across all pilot sites, ensuring consistency and comparability. The KPIs were developed iteratively, taking into account local SECO needs and PHArA-ON's overall objectives. Through workshops and collaborative meetings, the methodology was refined to support the replication and scale-up of services, addressing key factors such as financial sustainability, procurement models, and service organisation. A shared set of global KPIs (see Figure 11) was established to be measured in different phases of the protocol (baseline, intermediate, and final) using standardised and validated data sources. These KPIs were classified as sustainability (e.g. quality of life, social isolation and cost analysis), user acceptance, and innovation and growth (e.g. local replication plans). The study design included two cohorts: an experimental group that tested the full set of PHArA-ON technologies and a control group without technology but under the same conditions.

In addition, in PHArA-ON, the Smart Mobile Independent Living Evidence (SMILE) platform was developed as a centralised web-based system to collect and process quantitative common KPIs. The platform ensures data quality and analysis consistency between different pilot sites while adhering to GDPR principles. To achieve this, SMILE incorporates anonymisation techniques, access control mechanisms, and secure data storage to protect sensitive information. The platform is designed to store anonymised user profiles, define structured datasets, and manage questionnaire-based data collection. It enables automated scoring of responses and facilitates the generation of reports for further analysis.

PHArA-ON utilised the SMILE platform to perform a structured evaluation of KPIs at the pilot sites, ensuring a robust approach to analysing the results of the intervention. The data processing pipeline involved multiple steps to maintain accuracy and reliability. Missing values were handled through multiple imputation techniques, reducing biases and ensuring data completeness. The analysis process included a statistical description of independent variables at both the local and global levels, providing information on the determinants of SECO such as demographic data from users and baseline conditions. The evaluation framework also compared the outcome measures at different time points, using statistical significance tests to assess the impact of interventions. Furthermore, PHArA-ON leveraged advanced data mining techniques to identify key influencing factors and predict trends in KPI performance.

5 An approach for SECO engineering

The SECO engineering approach was developed using ISO 9241-210 [Int10] since social components are considered an integral part of any SECO [HD12]. SECOs support dynamic collaboration, adaptability, and stakeholder interaction, distinguishing them from traditional isolated software systems [HD12, Bos09]. To ensure that the SECO remains healthy and sustainable, its engineering process must consider its unique aspects, incorporating key quality attributes such as performance, usability, security, interoperability, and niche creation [FFFLG14, AS16]. To effectively address this challenge, the following approach has been developed for SECO engineering. The approach consists of five main stages: *context of use specification*, *requirement specification*, *SECO design solution*, *design implementation*, and *evaluation* (see Figure 12). Each stage corresponds to a human-centred engineering activity, reflecting the principles of ISO 9241-210 [Int10] for interactive systems, but extended to address the unique demands of SECOs. We deliberately separate the design-related activity into two stages — *SECO design solution* and *design implementation* — because merging conceptual design and implementation often blurs responsibilities, increases complexity and can lower the quality of both outcomes, especially when teams tackle them in parallel. This split therefore provides clearer focus and better coordination within a SECO context while still aligning with the ISO principles.

The stages form a cyclical process: Following evaluation, insights and improvements are integrated into the previous stages, facilitating ongoing SECO refinement. This cyclical approach acknowledges that SECOs evolve as new actors join, requirements change, and technologies mature. By integrating the roles of requirements gathering, architectural design, solution implementation, and iterative validation, the figure underscores a holistic, human-centred path for engineering SECOs.

5.1 SECO context of use specification

In this section, we examine the initial specification of the context of use within SECOs, describing the steps necessary to fully analyse stakeholders, their roles, tasks, and initial goals (see Figure 13).

5.1.1 Specify SECO goals and initial KPIs

This action involves identifying and defining the strategic objectives of SECO and assigning at least one measurable Key Performance Indicator (KPI) to each goal in order to evaluate progress towards these objectives. The goals should align with the overall vision of the SECO, ensuring that all stakeholders have a clear understanding of the expected outcomes and the long-term direction. To ensure the feasibility of these goals, it is required to analyse the current advances in SECO development, review existing or

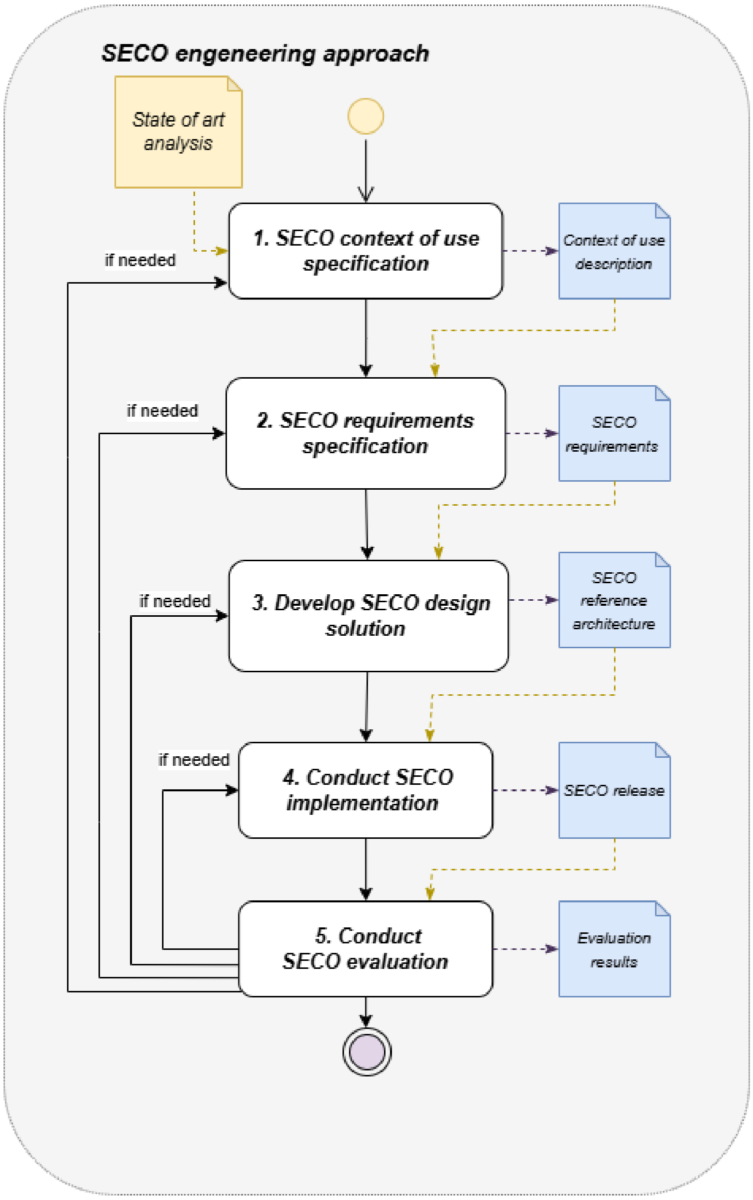

Figure 12. SECO engineering approach

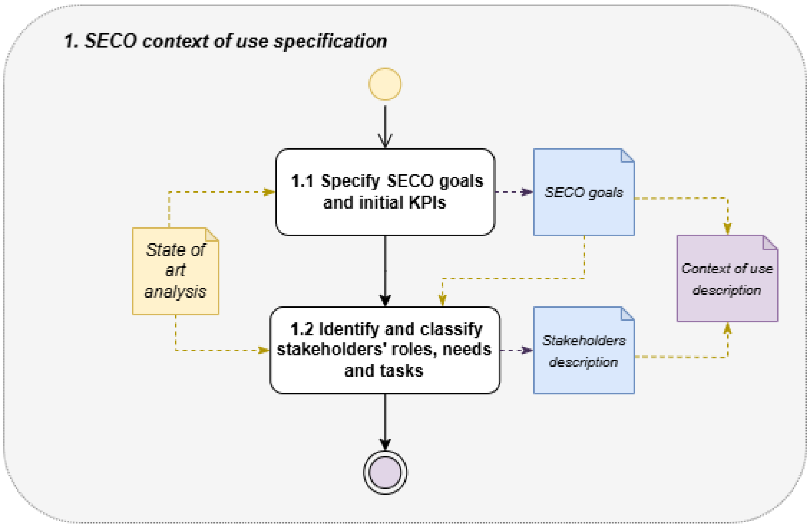

Figure 13. SECO context of use specification process

comparable SECOs, and perform a preliminary evaluation of the target system along with prospective user and stakeholder groups. Where applicable, domain-specific regulations and strategic documents should also be analysed to ensure alignment within the domain. This initial objective establishes a deeper context and directs the main focus of the system.

In addition, we recommend refining the initial SECO goal into *sub-goals* that improves clarity and focus, helping to maintain and improve SECO. This division facilitates the distribution of specific tasks among stakeholders, allowing each participant to work towards objectives aligned with their expertise and interests. To better address the various *sub-goals* SECO is targeting, which may be important to different stakeholders, we recommend assigning dedicated SECO managers. These managers will be responsible for gathering requirements from stakeholder groups and aligning these with SECO's specific *sub-goals*. Afterward, each manager will determine the appropriate methods tailored to their assigned objective.

For enhanced clarity in measurement and evaluation, each SECO sub-goal must incorporate specified initial KPIs. These KPIs will act as benchmarks to assess the progress and effectiveness of SECO initiatives.

Furthermore, it is advisable to classify the SECO goals into hierarchical levels—*strategic*, *tactical*, and *operational*—as this helps ensure traceability between high-level intentions and concrete implementations. A formalised representation of goals can support clarity, consistency, and analysis, enabling the SECO to adapt as new participants and concerns about trust emerge. Formalisation also facilitates running-time reasoning about trust and alignment of stakeholder expectations, as demonstrated in recent research on trust-based digital SECOs [CPBS21].

5.1.2 Identify and classify stakeholders' roles, needs and tasks

Based on the analysis of similar SECOs and defined SECO goals, in this phase, specific *roles* and *responsibilities* of stakeholders are identified. This ensures that all contributions and responsibilities are well understood. The specification of stakeholder roles is a crucial step as it influences how requirements will be generated [KYB^{+}18]. In addition, this ensures that all contributions and responsibilities are well understood as different stakeholders could follow different goals [HF18a].

Although the categorisation of stakeholders into *keystone players*, *niche players*, and *users* provides a foundational understanding of the dynamics of SECO, these categories are not entirely comprehensive or mutually exclusive. They serve as top-level classifications that can be refined further into more specialised subcategories based on the SECO goals. Many other roles are possible, such as *dominators*, *standardization organisations*, *resellers*, *operators*, and others [HJB13, HD12]. This layered and flexible approach ensures that SECOs can adapt to changing requirements and technological advances, fostering innovation and resilience. To streamline these efforts and enhance the adaptability of SECO, we suggest integrating the mentioned activities by classifying the SECO stakeholders into coherent groups, including technical/components' providers/developers, platform developers, and different types of end-users such as *direct* and *indirect*. This classification enables more targeted communication, fosters collaborative decision-making, and ensures that all stakeholder needs are systematically addressed.

5.2 SECO requirements specification

The SECO requirements specification phase begins by considering the context of use, which encompasses the goals of the SECO, the stakeholders' roles. The primary focus of this phase is to examine the input from stakeholders to determine the SECO components and compile the initial list of SECO requirements. This process includes executing the following steps (refer to Figure 14):

After completion of the design and implementation phases, these steps can be reiterated to ensure that requirements continue to meet the needs of stakeholders and the project goals. This iterative process helps to refine and adapt requirements as necessary.

5.2.1 Systematic elicitation of SECO requirements aligned with defined goals

Elicitation of SECO requirements involves the thorough collection, identification and refinement of the needs, expectations, and limitations of stakeholders within an SECO based on previously defined SECO goals and sub-goals. Given the multifaceted, dynamic, and interconnected nature of SECOs, it is crucial to acquire efficient requirements. This

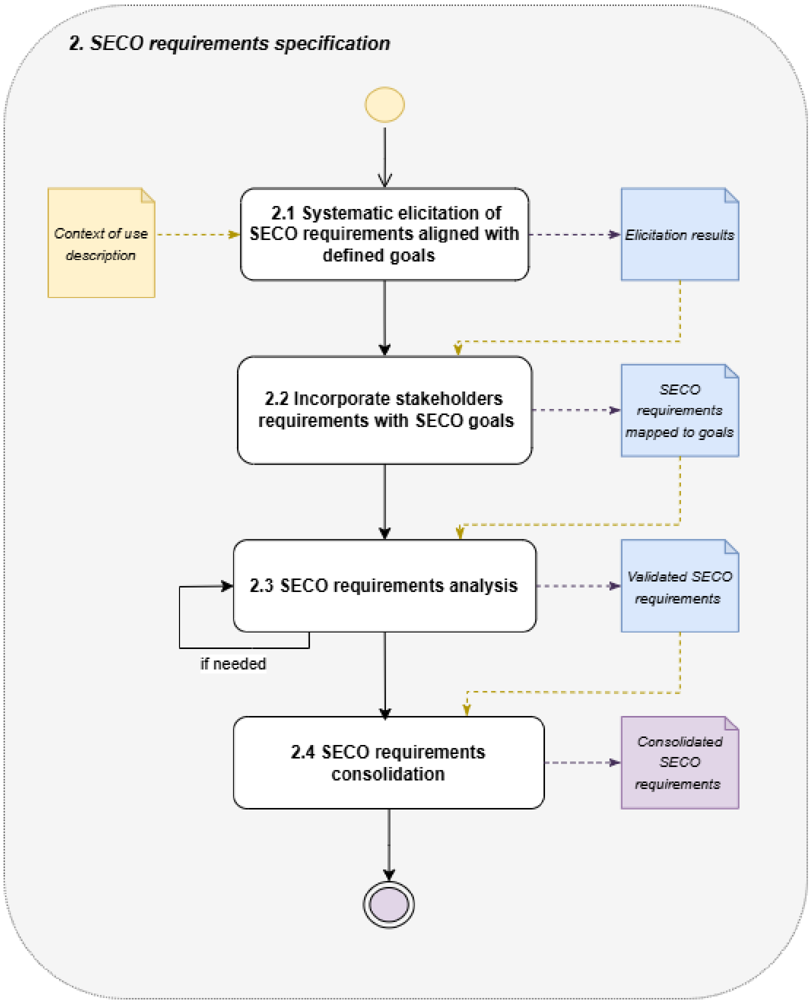

Figure 14. SECO requirements specification

step ensures that the requirements are not only articulated with clarity and aligned with the objectives of SECOs but are also capable of adapting to the evolving characteristics of SECOs.

Certain requirements can be derived directly from the defined SECO goals and sub-goals, providing a structured basis for the elicitation process. Based on that, every manager tasked with assigning SECO sub-goals is advised to select and modify the most appropriate method for gathering input from stakeholders. These methods may include techniques such as interviews, workshops, and surveys, among others. It is recommended to integrate a variety of existing methods and tailor them specifically to address the target SECO goal or sub-goal. Furthermore, it is of paramount importance that the resultant findings are presented in a consistent format, whether it is through goal models, scenarios, or user stories. This consistency ensures clarity and promotes effective communication among stakeholders, which is essential to provide a thorough understanding of the solution that is intended to be developed.

5.2.2 Incorporate stakeholders requirements with SECO goals

This action ensures that the perspectives, needs, and constraints of all stakeholder groups are properly aligned with the overall goals defined for the SECO. Since SECO requirements originate from diverse sources — such as platform providers, software developers, end users, domain experts, and regulatory bodies — it is essential to systematically map each requirement to one or more existing SECO-level objectives. This mapping helps to identify potential conflicts, redundancies, or gaps early in the engineering process.

We recommend performing this integration using a traceable goal-requirement model, which allows verification of how each stakeholder requirement supports or impacts strategic, tactical, or operational SECO goals. This step is also useful for communicating the rationale behind decisions to stakeholders. Regular review of this mapping is encouraged, especially in dynamic SECOs, where goals and stakeholder roles may evolve over time. Ultimately, this activity helps maintain internal coherence, supports transparency, and ensures that the SECO evolves in line with both stakeholder expectations and strategic direction.

5.2.3 SECO requirements analysis

The main objective during this activity is to*categorise requirements*, *prioritise*, and *validate* them as illustrated in Figure 15.

Requirements classification: Besides addressing fundamental classification types as *functional* and *non-functional*, within SECO the classification of requirements should

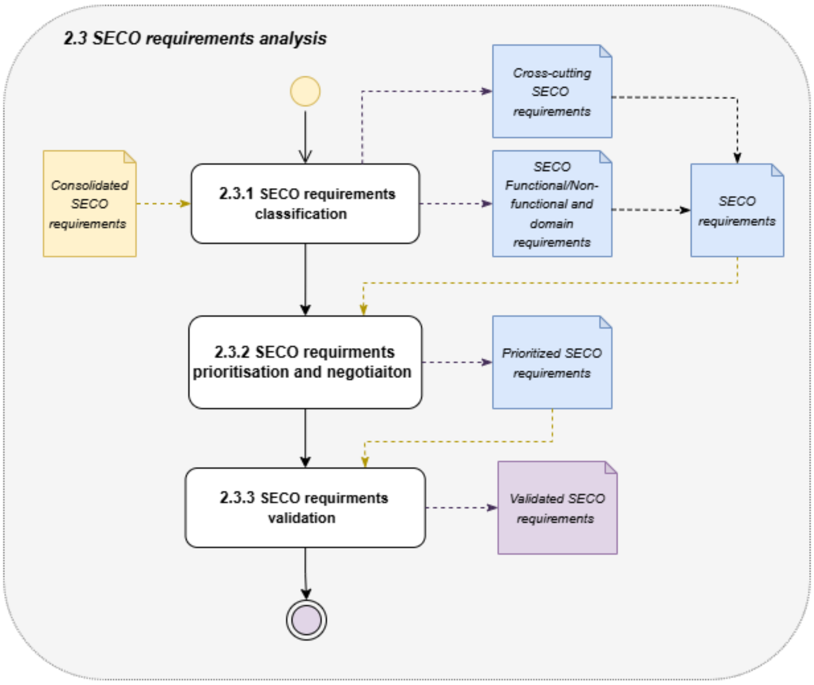

Figure 15. Analysis of SECO requirements

also consider the following categories such as *cross-cutting requirements*, and *specific SECO requirements*.

Functional requirements in SECO must specify essential services, business logic, APIs, data flows, and integration mechanisms that allow seamless collaboration between stakeholders.

Non-functional requirements in SECO must establish quality attributes such as niche creation, performance, scalability, reliability and maintainability. In SECO, non-functional requirements must address the distributed and evolving nature of the SECO, ensuring that components remain adaptable and resilient.

Cross-cutting requirements. Cross-cutting requirements influence all aspects of the SECO, shaping its overall structure and functionality. Among all, *security*, *privacy*, and *interoperability* stand out as the most critical and commonly addressed cross-cutting requirements. The main difference between functional/non-functional requirements and cross-cutting requirements is that cross-cutting requirements influence the overall SECO, affecting all components and stakeholders. In contrast, non-functional requirements, while crucial, may only apply to specific components or services rather than the entire SECO.

- *Security and privacy*. To effectively manage security within an SECO multiple stakeholders, it is crucial to implement a robust *cybersecurity framework*. This framework should integrate best practices tailored to SECOs, which differ from traditional software environments by requiring *federated security measures*. Such measures ensure cross-platform interoperability while safeguarding data confidentiality and integrity. Continuous threat assessment is essential to address emerging risks, especially in SECOs that include third-party components. Furthermore, employing a *Privacy by Design approach* is vital, incorporating data minimisation, pseudonymization, and strong consent management mechanisms to improve user trust and compliance among participants.

- *Interoperability*. Given that SECOs facilitate interactions between multiple components, typically centred on a platform, *interoperability* is a key requirement that affects all components and stakeholders. When defining interoperability requirements, an emphasis should be placed on maintaining flexibility across applications. SECO platforms should be designed for continuous enhancement, integrating new features that support standardisation and using open-source technologies. In addition, explicit requirements must be set to accommodate various communication protocols, semantic data exchanges, and alignment with interoperability solutions.

Specific SECO requirements. In addition to cross-cutting requirements, SECOs must address specific requirements that emerge due to their unique SECO structure and stakeholder dynamics. These requirements often extend beyond traditional software needs and include aspects such as emotional requirements that serve as a catalyst for building trust between organisational boundaries, cultivating a sense of community, and encouraging collaboration.

Requirements Prioritisation: Prioritisation in SECO involves ranking requirements according to their significance, feasibility, and impact on the overall goals of SECO. In an SECO setting, efficiently prioritising requirements entails examining requirements that had been associated with SECO goals, focussing on assessing their feasibility, risks, and overall impact.

Given the distributed nature of SECO, prioritisation must consider factors beyond individual stakeholder influence, incorporating the impact of SECO, interdependency between components, and long-term sustainability. For example, keystone players might prioritise interoperability and security, while niche creators may focus on API stability and monetization opportunities, leading to potential conflicts. If disagreements persist, a sub-step, such as *requirements negotiation* may be used to resolve differences. Using structured frameworks (such as EasyWinWin and StakeSource2.0), along with iterative techniques, helps align stakeholder expectations with technical feasibility.

Requirements Validation: In addition to common *validity*, *consistency*, *completeness*, *feasibility* checks, validation of requirements in SECO must also confirm that the requirements are aligned with the complex network of different stakeholders and platform ensuring interoperability. In contrast to conventional software development, SECO validation requires consideration of *compatibility*, external contributions, and changing platform limitations. To improve the validation process within SECO, it is vital to utilise automated tools to identify inconsistencies, embrace common standards for seamless interaction, and employ iterative validation strategies to keep up with ongoing refinements in SECO requirements. Validation should involve iterative methods, such as stakeholder walk-through, prototyping, and tests, along with user acceptance tests to assess both functionality and usability in real-world scenarios. This process ensures that any gaps or issues are quickly identified and addressed, allowing for refinement of requirements and supporting the sustainable development of the SECO.

5.2.4 SECO requirements consolidation

The primary objective of this step is to iteratively refine, verify, and enhance SECO requirements based on continuous feedback from stakeholders. This iterative process ensures that the final requirements are precise, well-aligned with stakeholder expectations, and structured for formalisation into technical specifications for subsequent development. This final consolidated list serves as the foundation for the development phase, ensuring precision, feasibility, and alignment with the general goals of the SECO.

5.3 Develop SECO design solution

With an initial list of requirements in place, the SECO design stage should begin. This stage bridges the gap between the requirements and a solution designed to satisfy them by systematically translating these requirements into a structured design framework. The result of this stage should be a comprehensive reference model of the SECO architecture, which describes the organisation of the SECO. This model will serve as the foundation for the further implementation and refinement of the SECO.

The proposed methodology, as illustrated in Figure16, begins with the development of the SECO reference architecture. Next, during validation, the SECO reference architecture is verified that the architecture is consistent with the initial requirements and adheres to the quality standards.

5.3.1 Design SECO reference architecture

Initially, common reference architectures within the target SECO domain should be studied to identify the best fit. Modifications or combinations of models can also be applied, aiming to develop a framework that effectively represents SECO and its components for diverse stakeholders. It is recommended to include next layers into SECO reference architecture:

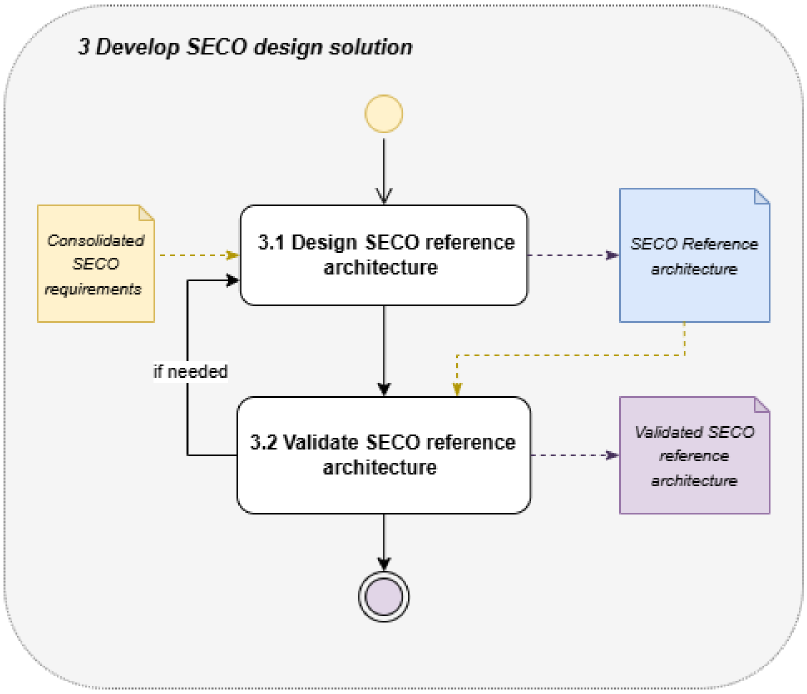

Figure 16. SECO design development process

- **Collaboration Layer**: This layer provides tools and resources for SECO participants, including a marketplace for extensions, documentation for niche creators, developer portals, and forums for knowledge sharing. It fosters engagement and supports the co-creation of new services within the SECO.
- **Cross-Cutting Concerns Layer**: This layer addresses critical aspects such as *interoperability*, ensuring seamless data exchange between components, *security*, protecting data and system integrity, and *data privacy*, ensuring compliance with regulations and safeguarding user information. These concerns impact all parts of the SECO and ensure its long-term stability and trustworthiness.
- **Project-Specific Layers**: These layers accommodate domain-specific needs, such as additional services, specialised devices, and custom functionalities. Depending on the SECO's focus, this could include IoT integrations, domain-specific AI models, or real-time processing engines.

By structuring the SECO architecture in this way, it becomes more adaptable, allowing different stakeholders to contribute while maintaining a stable and secure foundation. In addition, the defined reference architecture model must be as technology-agnostic as possible to adapt to evolving requirements and external constraints. In addition, considering that SECO involves external developers during design of the reference architecture interfaces, it must be stable and well documented.

5.3.2 Validate SECO reference architecture

This step involves checking the design reference developed against the requirements. During this step, we recommend iterative validation of architecture design where technical requirements and architectural components are refined step by step and validated in realistic scenarios. Throughout, focus on interoperable standards and modularity to accommodate the dynamic nature of SECOs, and utilise evaluation metrics to guide architecture updates.

5.4 SECO implementation

In this section, we examine the steps for the implementation of the SECO project in a collaborative manner. As illustrated in Figure 17, the recommended approach starts with the planning and alignment of tasks between contributors, continues with the development or adaptation of the platform and components, and finishes with the deployment of incremental releases.

5.4.1 Map reference architecture to implementation components and technical design

This activity focusses on translating the abstract SECO reference architecture into concrete implementation elements. Although the reference architecture defines the high-level structure, coordination style, and component responsibilities, this step ensures that each architectural element is connected to an actual system component, interface, or integration pattern in the technical design.

A well-defined mapping from architecture to implementation is crucial for directing distributed development and minimising team ambiguities. This requires a detailed description of the technologies, communication methods, and deployment settings for each component explicitly guided by the requirements obtained previously. In addition, cross-cutting requirements, such as interoperability or security, should be addressed and connected to relevant design aspects.

To reduce the risk of under-specification, we recommend explicitly linking architecture views to API, module, and data format implementations, ensuring traceability. This transition from design to development improves team coordination, maintains architectural consistency, and ensures that the behaviour of the system is in accordance with the design goals and the structure of the SECO.

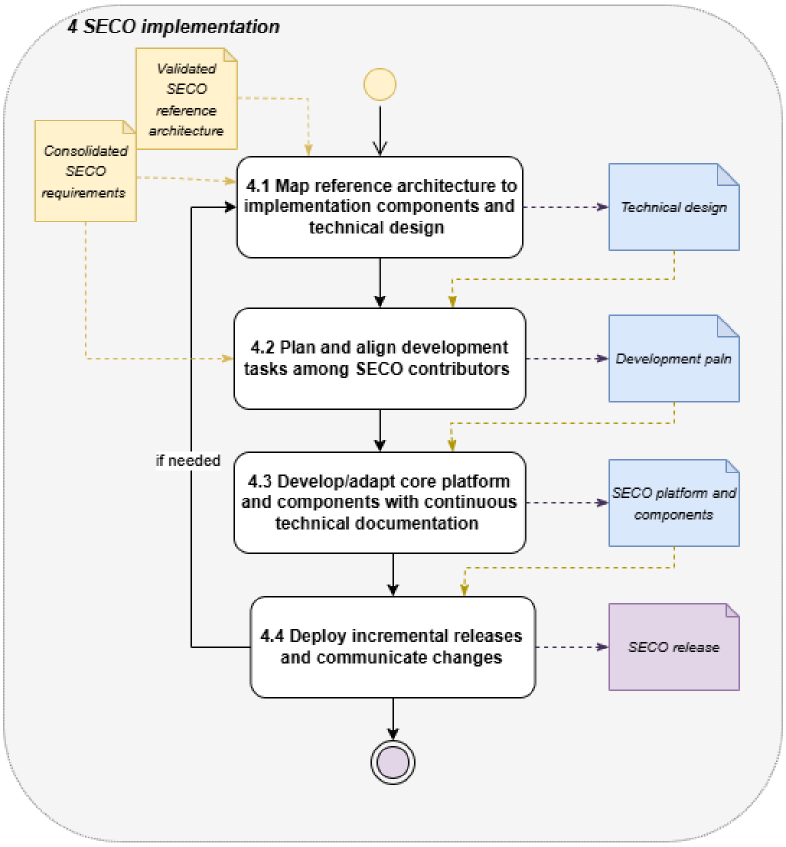

Figure 17. SECO implementation process

5.4.2 Plan and align development tasks among SECO contributors

During development planning, we recommend adapting short, iterative planning cycles with frequent feedback to remain responsive to stakeholders' needs and feedback. In some cases, it is also recommended to maintain a transparent and shared repository for real-time updates, preventing effort duplication while promoting efficient collaboration. Work packages must be assigned to teams with distinct expertise, but keep coordination tight under common SECO goals. Generally, it is important to prioritise tasks with significant impact and remain adaptable in your comprehensive strategy. This approach aligns with the typical role of a keystone stakeholder, who is responsible for maintaining the essential platform. Meanwhile, other niche stakeholders focus on the development of additional optional modules.

It is essential to establish well-defined tasks for each group while simultaneously providing accessible channels for open collaboration. To preserve uniformity and build trust within SECO, it is equally important to articulate clear quality guidelines along with precise acceptance criteria for those who contribute code. Moreover, from the outset, it is essential to implement robust review mechanisms specifically designed to assess adherence to security standards and conformity with interoperability guidelines. This ensures that high-quality solutions are delivered consistently.

5.4.3 Develop/adapt core platform and components with continuous technical documentation

This task emphasises the practical advancement and modification of the SECO core platform along with its components. Throughout this process, an importance is placed on keeping the technical documentation current, guided by an array of requirements. These include broad considerations such as ensuring security and promoting interoperability. The core platform typically includes shared infrastructure such as governance mechanisms, privacy and security enforcement layers, and reusable domain models that support the integration of distributed SECO contributions.

To support efficient development, we recommend using existing and proven tools wherever possible and extending them with targeted components to cover SECO-specific needs. This strategy helps reduce development overhead and accelerates implementation. Standardised APIs and data models should be prioritised to facilitate integration between systems and contributors.

Central repositories should be used to store open source modules, SECO-specific libraries, and configuration templates. This enables flexible reuse, supports consistent deployment, and improves interoperability between local adaptations and central platform services.

Continuous technical documentation, including API specifications, configuration guides, and usage examples, must accompany all platform development activities. This documentation is essential to ensure that new contributors can be incorporated efficiently, that integration points are clearly understood, and that the SECO can evolve without introducing unnecessary technical debt.

5.4.4 Deploy incremental releases and communicate changes

In this activity, the emphasis is placed on the methodical deployment of incremental updates to the SECO platform and its individual components. It is essential to ensure that all modifications are transparently communicated throughout the SECO. To facilitate regular, secure, and reliable releases in a multi-organisational framework, it is recommended to implement a continuous integration and delivery (CI/CD) pipeline. This pipeline should be based on DevSecOps principles, aligning with the security requirements that were initially established.

To accommodate the diversity of technologies and partners in an SECO, the release infrastructure should support both centralised and distributed CI/CD configurations. This includes pipelines per technology for building and verifying specific components, and pipelines per partner for partial integration and testing across organisational boundaries. This flexibility allows contributors to align release processes with their own technical environments while maintaining SECO-wide consistency and quality standards.

Incremental deployment allows new features, fixes, or platform updates to be delivered in manageable steps, reducing the risk of integration issues and facilitating continuous feedback. Equally important is the communication of changes: each release should be accompanied by clear release notes, changelogs, and integration instructions that are made accessible to all stakeholders. Transparent communication ensures that all contributors are aware of modifications that may affect interoperability, shared services, or local deployments.

Together, these practices facilitate efficient partnership collaboration, reduce barriers to product release, and foster SECO development.

5.5 SECO evaluation

A structured SECO evaluation ensures sustainability, growth, and value creation. This section defines the evaluation framework, key performance indicators (KPIs), and analysis methods. As illustrated in Figure 18, it begins by specifying a standardised evaluation method and essential KPIs to assess SECO performance. Next, it introduces a centralised KPI analysis using a dashboard to aggregate and interpret data between organisations. Finally, it outlines the evaluation process, focussing on continuous assessment, stakeholder feedback, and transparent reporting to drive improvements and foster trust.

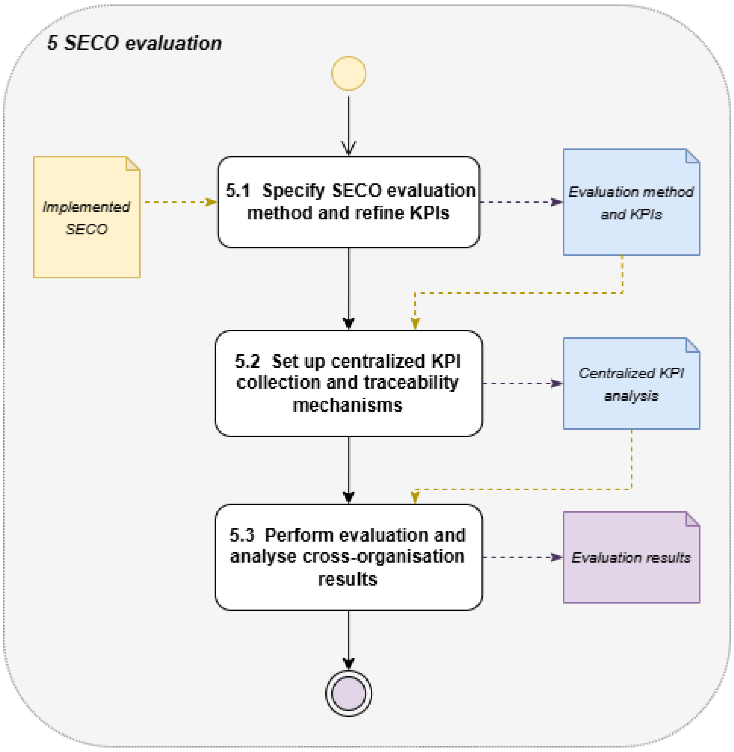

Figure 18. SECO evaluation process

5.5.1 Specify SECO evaluation method and refine KPIs

To effectively evaluate SECO, it is essential to implement a clear and systematic evaluation methodology. This approach should assess key attributes, including robustness, productivity, and niche creation, while accounting for domain-specific characteristics and quality attributes.

Initially, KPIs were defined per each SECO goal, establishing a foundational baseline for evaluation. In this step, these KPIs must be revisited and refined based on the insights gained during the implementation phase. Each KPI should clearly reflect specific aspects of SECO performance: growth and size through metrics such as active developers, number of solutions, and participation levels; health and sustainability through indicators such as platform adoption speed, churn rate, and defect density; business and value creation assessed by revenue streams, partnership diversity, and time-to-market; and finally, user satisfaction and engagement determined by usage rate, retention, and experience ratings [FFFLG14].

Furthermore, refined KPIs should remain closely aligned with the predefined common SECO goals. It is crucial to map each KPI explicitly to an appropriate data collection method and clearly defined evaluation periods, ensuring accurate and meaningful ongoing analysis.

5.5.2 Set up centralised KPI collection and traceability mechanisms

This activity focusses on establishing a centralised infrastructure for collecting, aggregating, and analysing key performance indicators (KPIs) between SECO participants, while ensuring traceability and comparability of evaluation data. A consistent and coordinated evaluation setup enables meaningful cross-organisational insights and supports evidence-based improvement.

To ensure data consistency, semantic data models should be defined and adopted in all SECO. These models help interpret the evaluation results uniformly, even when data is gathered from diverse sources. In addition, advanced analytics methods, such as data mining, can be used to identify influencing factors and forecast trends based on KPI outcomes.

A central dashboard is recommended as the primary interface for collecting and visualising performance and usage metrics. It should support both component-level and system-level views, tailored to the needs of different stakeholder groups. The dashboard should also incorporate mechanisms for capturing structured feedback, enabling stakeholders to comment on evaluation results and identify improvement areas.

Traceability should be embedded throughout the evaluation process, linking KPIs to specific goals, requirements, and SECO components. This ensures that changes in performance can be meaningfully interpreted and acted upon. At the same time, great attention must be paid to data privacy and security. Aggregation, access control, and anonymisation strategies must be used to ensure compliance with legal and ethical standards, particularly in SECOs that involve sensitive or personal data.

5.5.3 Perform evaluation and analyse cross-organisation results

For a thorough assessment and examination of the results involving several SECO participants, it is crucial to consistently track specific KPIs. Use multiple imputation methods to address absent data and maintain statistical soundness when comparing across organisations. Employ an iterative process where results are reviewed both locally and globally, allowing for the gradual improvement of SECO strategies.

Regular evaluations are necessary. It is important to refine the SECO's implementation, design, or requirements by incorporating updated policies, integration guidelines, developer incentives, and architectural constraints based on the results of these evaluations. In addition, it is recommended to publicly share the evaluation results to build trust and promote greater partner participation.

6 Validation

6.1 Methodology

In Design Science Research, evaluation demonstrates an usefulness of proposed approach by measuring how effectively it addresses the target problem [AAS$^+$04]. In this section, we examine whether the proposed approach to the engineering of SECOs is usable (comprehensible and easy to follow) and useful (perceived as valuable for SECO engineering activities). An ex post evaluation in a real-world setting was chosen, since the SECO engineering method was established prior to validation, and practical input from professionals is prioritised over laboratory results in the context of SECO, aligning with human-centred design principles[VPHB12]. The main risks are related to human factors such as understanding, ease of use, and perceived value.

Experts with experience in SECO work, covering areas such as requirements engineering, SECO architectural design, implementation management, and SECO evaluation, were invited to participate. Each interview was required between 30 to 60 minutes. The evaluation followed a semi-structured interview format, focussing on open-ended questions. A more detailed interview plan can be found in the Appendix A. Participants were introduced to the SECO engineering process, including its five stages and supporting diagrams. The evaluation was structured around the following criteria:

- *Clarity and completeness*: Are the process steps understandable and sufficient?
- *Alignment with SECO practice*: Does the process reflect real-world engineering experience?
- *Perceived usefulness*: Would the approach help in practical SECO projects?
- *Adoption Barriers*: What challenges may arise during implementation?

During the participant search, it became evident that professionals with direct SECO development expertise are scarcely available. This limitation reflects a known challenge in the SECO domain, where engineering roles are often distributed across diverse organisations and job functions.

6.2 Results and Discussion

In the end, four experts (P1 - P4) agreed to participate in the interview. All of them have more than five years of experience working with SECO in various areas. Despite the small sample, the experts provided valuable and in-depth feedback. Key insights are summarised below.

Clarity and completeness: The process was generally considered clear and thorough. Each participant offered minor suggestions to improve clarity. P1 stressed the need to refine stakeholder classification and requirements prioritisation, proposing additional tools -such as stakeholder maps and MoSCoW matrices - to support transparent negotiation and prioritisation between partners. P2 appreciated the decomposition from strategy to operational goals and recommended formalising goals and requirements to allow traceability and validation between levels. P3 welcomed the dedicated requirements-negotiation step. P4 confirmed that the staged flow aligns with the work, but called for a more explicit link between goals and requirements from the outset and a clearer influence of requirements on subsequent design and implementation stages. P4 also advised viewing the reference architecture from three complementary perspectives - information, interaction, behaviour - to cope with SECO complexity.

Alignment with practice: All participants (P1, P2, P3, and P4) recognised the structure of the process in their previous projects. P2 related it to orchestrated SECOs and noted that the staged flow fits projects where top-down goal decomposition is required. P4 confirmed that the staged flow mirrors common practice, but stressed that GDPR and cybersecurity requirements must remain visible throughout design and validation. P4 also underlined that the initial KPIs should be set during the context-of-use stage and refined during evaluation.

Perceived usefulness: P1 and P3 highlighted that the structured approach helps align requirements and communicate between teams. P2 observed that the approach enables alignment of the SECO development with its objectives. P4 saw the method as transferable to SECO projects in other domains and stressed that such a method is urgently needed because demand for systematic SECO engineering is increasing.

Barriers to adoption: The experts noted that they did not observe significant barriers to adopting the approach, in addition to comments drawn from their practical experience. All emphasised agreeing early on a shared glossary of architectural and coordination terms. P1 warned about the risks related to uncertainty in partner responsibilities that must be addressed early on in the context of use definition stage and emphasised the importance of supporting teams in distributed SECOs through better documentation practices. P2 raised the need to put more emphasis on the mapping step between high-level architecture and concrete implementation and recommended that future iterations of the approach consider how unmet requirements at the operational level could trigger a re-evaluation of tactical or strategic goals. P3 elaborated that the approach itself does not pose a major adoption barrier: its stages are clear and can be followed 'without major difficulties', but any serious obstacles are likely to appear later during the concrete implementation work. P4 cautioned that effective adoption hinges on clearly defining what SECO is as misunderstandings may arise otherwise.

In general, the evaluation confirmed the practical relevance of the proposed framework and highlighted specific areas where additional support could improve its adoption in real-life SECO projects. The experts viewed the approach as a solid starting point for structuring the SECO engineering tasks and accurately reflecting the activities carried out in practice. All feedback from P1–P4 was systematically analysed and incorporated into the final version of the approach to better match the day-to-day needs of SECO engineers and to facilitate smoother adoption across diverse SECOs.

7 Limitations and future work

Despite the contributions of this thesis, several limitations should be acknowledged. First, the validation of the SECO engineering approach was conducted through expert interviews with a small number of respondents. Although the insights were valuable, the limited sample size restricts the generalisation of the findings. Second, the development of the approach was based on experience and lessons learnt in the context of a single industrial case, the PHArA-ON project. Although the project involved diverse technologies and stakeholders, the results may not fully reflect the challenges in other SECO domains. Third, the engineering process of SECO remains complex. It involves iterative requirement elicitation, architectural alignment, and stakeholder coordination. Although the proposed approach aims to structure this process, practical application may still require a great deal of effort and experience. To overcome these limitations, several directions are planned in future work:

- Extend the validation by conducting structured evaluations with a broader group of SECO experts and practitioners from multiple domains.
- Apply the approach to additional projects to evaluate its generalisation beyond the PHArA-ON project domain.
- Promote adoption of the approach in real-world SECO engineering processes.
- Incorporate gathered feedback for further enhancement of the proposed SECO engineering approach.

This thesis contributes to SECO research by proposing a structured, human-centred approach to their engineering, rooted in real-world industrial experience. The approach emphasises the development of SECO through the full life cycle, a key gap in current research. However, SECOs remain highly complex systems, and the findings presented here represent a first step toward a generalisable and scalable engineering methodology.

Preliminary feedback from expert interviews was found, but emphasised the need for stronger evaluations, such as in-depth interviews, structured experiments, and multiple case studies. Addressing these gaps is essential to boost the method's credibility and applicability.

8 Conclusions

The goal of this thesis was to develop a comprehensive approach for the engineering of SECOs that supports the full development life cycle. To address the complexity and dynamic nature of SECOs, the proposed approach is structured into five main stages: SECO context-of-use specification, SECO requirements specification, SECO design solution development, SECO implementation, and SECO evaluation. Each stage integrates the principles of ISO 9241-210[Int10] and extends them to address SECO-specific challenges.

The approach was developed iteratively based on studies, existing practices, and insights gained from the PHArA-ON project. The PHArA-ON experience was used as a baseline to inform and guide the definition of the method. This included analysing how SECO-related processes were handled in a real-world project and identifying key lessons to generalise into a reusable engineering framework.

In result each stage includes clearly defined sub-activities, inputs, and outputs, aiming to support developers in systematically addressing both technical and organisational aspects of SECOs. The process begins with the context of use stage that includes the definition of the SECO goals, identifying the stakeholders, and understanding their tasks and needs. It continues with structured elicitation, analysis, and consolidation of SECO-specific requirements, distinguishing between functional, non-functional, and cross-cutting concerns such as security, interoperability, and privacy. The architecture phase guides the development and validation of the SECO reference architectures, followed by practical strategies for the implementation planning, platform adaptation, component integration, and incremental release management. Finally, the evaluation phase defines SECO-specific KPIs and describes how to collect, analyse, and use results between organisations for continuous improvement.

To validate the proposed method, interviews with SECO experts were conducted. Their feedback confirmed the relevance and usefulness of the approach. According to experts, it helps to bring structure to SECO development, improves clarity in roles and requirements, and supports decision-making in phases. The results demonstrate that the proposed method provides clear guidance for SECO developers, promotes stakeholder alignment, and facilitates iterative development and evaluation. Thus, the thesis contributes a reusable and practical engineering method that enhances the quality, sustainability, and user focus of SECOs across domains.

Future work should focus on strengthening and generalising the approach. This includes conducting structured evaluations with a broader group of SECO experts from different domains to assess its completeness and relevance in various contexts. Furthermore, the approach should be applied in more SECO projects to evaluate its adaptability and utility outside the PHArA-ON context. Finally, efforts should be made to support the adoption of the developed approach in real-world SECO engineering processes.

References

[AAS+04] Hevner Alan, R Alan, March Salvatore, T Salvatore, Jinsoo Par, Ram, and Sudha. Design science in information systems research. *Management Information Systems Quarterly*, 28:75–, 03 2004.

[AG19] Memoona J. Anwar and Asif Q. Gill. A review of the seven modelling approaches for digital ecosystem architecture. In *2019 IEEE 21st Conference on Business Informatics (CBI)*, volume 01, pages 94–103, 2019.

[AMAGC22] Simone da Silva Amorim, John D. Mcgregor, Eduardo Santana de Almeida, and Christina von Flach Garcia Chavez. Connecting non-functional requirements to open source ecosystems health. In *Proceedings of the 16th Brazilian Symposium on Software Components, Architectures, and Reuse*, SBCARS '22, page 76–80, New York, NY, USA, 2022. Association for Computing Machinery.

[AS16] Jakob Axelsson and Mats Skoglund. Quality assurance in software ecosystems: A systematic literature mapping and research agenda. *Journal of Systems and Software*, 114:69–81, 2016.

[BdCI+07] David Bell, Sergio de Cesare, Nicola Iacovelli, et al. A framework for deriving semantic web services. *Information Systems Frontiers*, 9(1):69–84, March 2007.

[Bos09] Jan Bosch. From software product lines to software ecosystem. In *Proceedings of the 13th International Software Product Line Conference*, pages 111–119, 01 2009.

[Bos10] Jan Bosch. Architecture challenges for software ecosystems. In *Proceedings of the Fourth European Conference on Software Architecture: Companion Volume*, ECSA '10, page 93–95, New York, NY, USA, 2010. Association for Computing Machinery.

[CHKM14] Henrik Bærbak Christensen, Klaus Marius Hansen, Morten Kyng, and Konstantinos Manikas. Analysis and design of software ecosystem architectures – towards the 4s telemedicine ecosystem. *Information and Software Technology*, 56(11):1476–1492, 2014. Special issue on Software Ecosystems.

[CNDGS18] Aparna Chirumamilla, Anh Nguyen Duc, Shang Gao, and Guttorm Sindre. A systematic mapping study on requirements engineering in software ecosystems. *Journal of Information Technology Research*, 11:49–69, 01 2018.

[CPBS21] Emilia Cioroaica, Akanksha Purohit, Barbora Buhnova, and Daniel Schneider. Goals within trust-based digital ecosystems. In *2021 IEEE/ACM Joint 9th International Workshop on Software Engineering for Systems-of-Systems and 15th Workshop on Distributed Software Development, Software Ecosystems and Systems-of-Systems (SESoS/WDES)*, pages 1–7, 2021.

[DGV$^+$22] Stefania D'Agostini, Gabriele Giammatteo, Pasquale Vitale, Paolo Fabriani, et al. Pharaon ecosystem – Initial release. Technical Report D5.1, PHArA-ON Project, Horizon 2020 (Grant Agreement No. 857188), March 2022. Version 1.0, Actual date of delivery: 11/03/2022.

[DLJ$^+$21] Daniela Damian, Johan Linåker, David Johnson, Tony Clear, and Kelly Blincoe. Challenges and strategies for managing requirements selection in software ecosystems. *IEEE Software*, 38(6):76–87, 2021.

[ET13] Åsa Ericson and Peter Törlind. A deep dive into creative thinking: The now-wow-how framework. *Proceedings of the International Conference on Engineering Design, ICED*, 7:337–346, 01 2013.

[FdMF$^+$19] Laura Fiorini, Marleen de Mul, Isabelle Fabbricotti, Raffaele Limosani, Alessandra Vitanza, Grazia D'Onofrio, Michael Tsui, Daniele Sancarlo, Francesco Giuliani, Antonio Greco, Denis Guiot, Eloise Senges, and Filippo Cavallo. Assistive robots to improve the independent living of older persons: results from a needs study. *Disability and Rehabilitation Assistive Technology*, 07 2019.

[FFFLG14] Farnaz Fotrousi, Samuel Fricker, Markus Fiedler, and Franck Le Gall. Kpis for software ecosystems: A systematic mapping study. volume 182, 06 2014.

[Fri10] Samuel Fricker. Requirements value chains: Stakeholder management and requirements engineering in software ecosystems. In Roel Wieringa and Anne Persson, editors, *Requirements Engineering: Foundation for Software Quality*, pages 60–66, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[FSOC15] Bruna Ferreira, Williamson Silva, Edson Oliveira, and Tayana Conte. Designing personas with empathy map. 05 2015.

[Gas15] Urs Gasser. Interoperability in the digital ecosystem. *SSRN Electronic Journal*, 01 2015.

[GCN+21] Marco Di Girolamo, Giuseppe Coffano, Carlo Nava, Christian Sestu, Chiara Bonferini, Andrea Diarena, Vadim Denissiouk, Christian Temporale, Miran Mošmondor, Andrej Grgurić, José María Barriga García, Francisco José Melero, Rafael Maestre, Ana Perandrés Gómez, Carina Dantas, and Sofia Ortet. Pharaon initial ecosystem architecture. Technical Report D2.2, PHArA-ON Horizon 2020, EU Horizon 2020 Research Programme, October 2021. Grant Agreement No. 857188.

[GFN+24] Mohamad Gharib, Mariana Falco, Femke Nijboer, Angelique Tinga, Stefania D'Agostini, Erika Rovini, Laura Fiorini, Filippo Cavallo, and Kuldar Taveter. *Dealing with Emotional Requirements for Software Ecosystems: Findings and Lessons Learned in the PHArA-ON Project*, pages 99–114. 05 2024.

[GJM+24] Andrej Grguric, Luka Jelic, Miran Mosmondor, Dimitrios Tsolovos, Miguel Ángel Beteta Medina, José Barriga, Carolin Hammer, María Victoria Bueno Delgado, Boris van Schooten, Alex McDonald, João Ribeiro, Danny Jansen, Fabio Guasconi, et al. Cybersecurity verification and recommendations. Technical Report D10.7, PHArA-ON Horizon 2020, November 2024. This document reports on cybersecurity verification and recommendations for the Pharaon project, focusing on governance frameworks, risk-based approaches, and compliance with cybersecurity standards.

[HD12] Geir Hanssen and Tore Dybå. Theoretical foundations of software ecosystems. 879:6–17, 01 2012.

[HF18a] James Hedges and Andrei Furda. *The Emerging Role of the Ecosystems Architect*, chapter 6. Taylor & Francis Group, LLC/CRC Press, 04 2018.

[HF18b] James Hedges and Andrei Furda. *The Emerging Role of the Ecosystems Architect*. 04 2018.

[HJB13] Eko Handoyo, Slinger Jansen, and Sjaak Brinkkemper. Software ecosystem roles classification. volume 150, pages 212–216, 06 2013.

[Int10] International Organization for Standardization. *ISO 9241-210: Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems*. Geneva, Switzerland, first edition edition, 03 2010.

[ISO22] Information technology – Security Techniques — Information security management systems — Requirements, 2022.

[KBKD12] Alessia Knauss, W.A. Boriçi, Eric Knauss, and Daniela Damian. Towards understanding requirements engineering in it ecosystems. pages 33–36, 09 2012.

[KDKB14] Eric Knauss, Daniela Damian, Alessia Knauss, and W.A. Boriçi. Openness and requirements: Opportunities and tradeoffs in software ecosystems. 08 2014.

[KM16] Jens Knodel and Konstantinos Manikas. Towards reference architectures as an enabler for software ecosystems. pages 1–4, 11 2016.

[KWS+16] J.W. Kruize, J. Wolfert, H. Scholten, C.N. Verdouw, A. Kassahun, and A.J.M. Beulens. A reference architecture for farm software ecosystems. *Computers and Electronics in Agriculture*, 125:12–28, 2016.

[KYB+18] Eric Knauss, Aminah Yussuf, Kelly Blincoe, Daniela Damian, and Alessia Knauss. Continuous clarification and emergent requirements flows in open-commercial software ecosystems. *Requirements Engineering*, 23, 03 2018.

[LLBS18] Antonio Lopez Lorca, Rachel Burrows, and Leon Sterling. Teaching motivational models in agile requirements engineering. In *2018 IEEE 8th International Workshop on Requirements Engineering Education and Training (REET)*, pages 30–39, 2018.

[Mai12] Neil Maiden. Exactly how are requirements written? *IEEE Software*, 29:26–27, 01 2012.

[MH13] Konstantinos Manikas and Klaus Marius Hansen. Software ecosystems – a systematic literature review. *Journal of Systems and Software*, 86(5):1294–1306, 2013.

[MPLL+15] Tim Miller, Sonja Pedell, Antonio Lopez-Lorca, Antonette Mendoza, Leon Sterling, and Alen Keirnan. Emotion-led modelling for people-oriented requirements engineering: the case study of emergency systems, 09 2015.

[oTUFS+21] University of Tartu (UTARTU), Adriano Fernandes, Henrique Silva, Carina Dantas, Sofia Ortet, Flávia Rodrigues, Margarida Nery, Sandra Teixeira, Paulo Correia, Laura Fiorini, Grazia D'Onofrio, Femke Nijboer, Christiane Grünloh, Mike Burnard, Ana Perandres, María

Sánchez, Francisco José Melero, Javier Moreno, Miran Mosmondor, Kerli Mooses, Tahira Iqbal, and Kuldar Taveter. D2.1 user and pilot requirements. Deliverable D2.1, PHArA-ON Consortium, Horizon 2020, Grant Agreement No 857188, 2021. Confidential Report.

[PRRT14] Birgit Penzenstadler, Ankita Raturi, Debra Richardson, and Bill Tomlinson. Safety, security, now sustainability: The nonfunctional requirement for the 21st century. *IEEE Software*, 31(3):40–47, 2014.

[RB22] Pekka Ruotsalainen and Bernd Blobel. Transformed health ecosystems—challenges for security, privacy, and trust. *Frontiers in Medicine*, 9:827253, 03 2022.

[RB+23] Jorge Rodríguez, José Barriga, et al. Pharaon d3.2 - interoperability platforms descriptions - final. Report D3.2, PHArA-ON Consortium, November 2023. This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No 857188.

[RCL14] Reza Rezaei, Thiam Chiew, and Sai Lee. A review on e-business interoperability frameworks. *Journal of Systems and Software*, 93, 07 2014.

[SdSANM+17] Simone Simone da Silva Amorim, Félix Simas S. Neto, John D. McGregor, Eduardo Santana de Almeida, and Christina von Flach G. Chavez. How has the health of software ecosystems been evaluated? a systematic review. In *Proceedings of the XXXI Brazilian Symposium on Software Engineering*, SBES '17, page 14–23, New York, NY, USA, 2017. Association for Computing Machinery.

[SM15] Alexander Serebrenik and Tom Mens. Challenges in software ecosystems research. In *Proceedings of the 2015 European Conference on Software Architecture Workshops*, ECSAW '15, New York, NY, USA, 2015. Association for Computing Machinery.

[SMMM98] A.G. Sutcliffe, N.A.M. Maiden, S. Minocha, and D. Manuel. Supporting scenario-based requirements engineering. *IEEE Transactions on Software Engineering*, 24(12):1072–1088, 1998.

[Som15] Ian Sommerville. *Software Engineering*. Pearson, 10th edition, 2015.

[TPRP24] Dimitrios Tsolovos, Matthias Pocs, Lorenzo Rettori, and Jasmine Pani. Privacy by design and standardisation contribution. Technical Report D10.8, PHArA-ON Horizon 2020, November 2024. This document presents an assessment of Pharaon's adherence to data protection regulations, privacy by design implementation, and standardisation efforts within the digital health domain.

[TSP+19] Kuldar Taveter, Leon Sterling, Sonja Pedell, Rachel Burrows, and Eliise Marie Taveter. A method for eliciting and representing emotional requirements: Two case studies in e-healthcare. In *2019 IEEE 27th International Requirements Engineering Conference Workshops (REW)*, pages 100–105, 2019.

[Val13] George Valen¸a. Requirements negotiation model: A social oriented approach for software ecosystems evolution. In *2013 21st IEEE International Requirements Engineering Conference (RE)*, pages 393–396, 2013.

[vL01] A. van Lamsweerde. Goal-oriented requirements engineering: a guided tour. In *Proceedings Fifth IEEE International Symposium on Requirements Engineering*, pages 249–262, 2001.

[VPHB12] John Venable, Jan Pries-Heje, and Richard Baskerville. A comprehensive framework for evaluation in design science research. In Ken Peffers, Marcus Rothenberger, and Bill Kuechler, editors, *Design Science Research in Information Systems. Advances in Theory and Practice*, pages 423–438, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[WBSE15] Asanka Wasala, Jim Buckley, Reinhard Schäler, and Chris Exton. An empirical framework for evaluating interoperability of data exchange standards based on their actual usage: A case study on xliff. *Computer Standards & Interfaces*, 42:157–170, 2015.

[WWS+22] Tobias Wulfert, Robert Woroch, Gero Strobel, Sarah Seufert, and Frederik Möller. Developing design principles to standardize e-commerce ecosystems. *Electronic Markets*, 32:1–30, 07 2022.

[ZCM22] Stan Zajdel, Diego Elias Costa, and Hafedh Mili. Open source software: an approach to controlling usage and risk in application ecosystems. In *Proceedings of the 26th ACM International Systems and Software Product Line Conference - Volume A*, SPLC '22, page 154–163, New York, NY, USA, 2022. Association for Computing Machinery.

Appendix

A Semi-Structured Interview Plan

Purpose To collect expert feedback on the proposed human-centred engineering approach for Software Ecosystems SECOs across the full life cycle: context definition, requirements, design, implementation and evaluation.

Participants Interviewees should have practical experience with SECOs (minimum 3 years). Record area of expertise in SECO development and years of SECO experience.

Interview Format

- 30-45 minutes, online
- Audio-record the session (with consent).

Section 1. Introduction

Briefly present the purpose and structure of the proposed SECO engineering approach, then confirm the background of the interviewee.

- **Years of SECO work.** How many years have you worked with SECOs?
- **Primary expertise areas.** Which life-cycle phases - context, requirements, design, implementation, evaluation - are your main areas of expertise?

After the introductory experience questions, the interviewer determined which SECO life-cycle phase(s) the participant knew best. The interview then prioritised the matching thematic block: context, requirements, design, implementation, or evaluation, looking more deeply and collecting concrete examples. Sections outside of the participant's core expertise were covered in more detail. This adaptive focus kept the total interview within the planned window, while ensuring rich data in the areas where each expert could contribute most.

Section 2. Context of Use

Briefly present the *Context of Use* and its activities.

- Does the proposed set of context-specification activities cover the SECO goals, stakeholder roles, needs, and tasks? What important context factors are missing?

Section 3. Requirements

Briefly present the *Requirements specification* stage and it's activities.

- In the requirements specification phase, do you find the described activities aligned with practices from your experience? Are there additional techniques or practices that should be considered?
- How did you handle conflicting requirements between stakeholder groups?
- Which cross-cutting requirements (e.g. security, privacy, interoperability) were the most challenging, and how did you address them?

Section 4. Design

Briefly present the stage *development of the design solution* and its activities.

- Do the SECO design activities listed capture the necessary work? What is missing or mismatched?
- If you could redesign the last SECO project, which design choices would you change and why?

Section 5. Implementation

Briefly present the *Implementation* stage and its activities.

- Does the proposed implementation set reflect typical SECO practice? What would you add or adjust?

Section 6. Evaluation

Briefly present the *Evaluation* stage and its activities.

- Considering the evaluation phase, do you think the evaluation activities are sufficient to evaluate a SECO? Are there activities you think were not addressed properly or should be added?

Section 7. Overall approach

- What difficulties do you see in applying the proposed activities in real-world SECO environments?
- Is the overall approach applicable? What advice would you give others who want to use it?

Section 8. Closing

- Invite any additional comments

II. Licence

Non-exclusive licence to reproduce thesis and make thesis public

I, **Daryna Pyshchyk**,
(author's name)

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to

   reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

   An Approach for Engineering Software Ecosystems Based on an Industrial Case Study,
   (title of thesis)

   supervised by Mohamad Gharib.
   (supervisor's name)

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Daryna Pyshchyk
15/05/2025