Aalto University
School of Science
Master's Programme in Computer, Communication and Information Sciences

Samuli Suortti

The Role of Software Platform and Actors in Software Ecosystems:

A Case Study in Agriculture

Master's Thesis
Espoo, October 9, 2017

Supervisor: Professor Marjo Kauppinen
Advisor: Mikko Raatikainen M.Sc. (Tech.)

Aalto University
School of Science
Master's Programme in Computer, Communication and Information Sciences

ABSTRACT OF MASTER'S THESIS

Author:	Samuli Suortti		
Title: The Role of Software Platform and Actors Software Ecosystems: A Case Study in Agriculture			
Date:	October 9, 2017	**Pages:**	viii + 47
Major:	Software and Service Engineering	**Code:**	SCI3043
Supervisor:	Professor Marjo Kauppinen		
Advisor:	Mikko Raatikainen M.Sc. (Tech.)		

In today's world, companies can have difficulties in developing products that satisfy all the needs of the customers. Software ecosystems (SECOs) are emerging as a solution proposal for the problem. In SECOs different companies collaborate in order to co-innovate new business opportunities and decrease development costs. The participating companies, so-called actors, are in a critical position in the success of a SECO. Along with the actors, the software platform has a central role in SECOs. Despite its essentiality, the platform is left with little attention in previous studies.

The goal of this thesis is to investigate what is important on a platform of a software ecosystem to satisfy actors' expectations. The study was conducted as a qualitative case study in the agricultural domain. Theme interviews and document review were used as data collection techniques. The results were analysed with a whole-text coding approach.

Three different kinds of agricultural SECOs were identified. The SECOs varied from the maturity point of view and they included different types of software platforms. All of the software platforms aimed to enable actor cooperation. Also, the identified ecosystem actors were in line with the five main actor roles found in previous research. However, the actor role motivations were described only to be either monetary or non-monetary. Further, it was discovered that a software ecosystem must provide a unique value proposition to all different actor roles.

Keywords:	Software Ecosystem, Software Platform, Actors, Roles
Language:	English

Aalto-yliopisto
Perustieteiden korkeakoulu
Tieto-, tietoliikenne ja informaatiotekniikan maisteriohjelma

DIPLOMITYÖN TIIVISTELMÄ

Tekijä:	Samuli Suortti		
Työn nimi: Ohjelmistoalustan ja toimijoiden roolit ohjelmistoekosysteemeissä: Tapaustutkimus maataloudessa			
Päiväys:	9. lokakuuta 2017	**Sivumäärä:**	viii + 47
Pääaine:	Software and Service Engineering	**Koodi:**	SCI3043
Valvoja:	Professori Marjo Kauppinen		
Ohjaaja:	Diplomi-insinööri Mikko Raatikainen		

Nykypäivänä yritysten on vaikea kehittää tuotetta, joka täyttää kaikki asiakkaiden tarpeet. Ohjelmistoekosysteemit ovat nousemassa ratkaisuksi tähän ongelmaan. Ohjelmistoekosysteemeissä yritykset toimivat yhteistyössä luodakseen uusia liiketoimintamahdollisuuksia sekä alennettuja kehityskustannuksia. Yritykset, eli ekosysteemin toimijat ovat tärkeässä roolissa ohjelmistoekosysteemin menestyksen kannalta. Heidän lisäkseen myös ohjelmistoalustalla on keskeinen rooli. Sen merkityksellisyydestä huolimatta sen tutkiminen on jäänyt aikaisemmissa tutkimuksissa vähemmälle huomiolle.

Tässä tutkielmassa tarkastellaan, mikä on tärkeää ohjelmistoalustassa, jotta se täyttää ohjelmistoekosysteemin toimijoiden odotukset. Tämä tutkielma toteutettiin kvalitatiivisena tapaustutkimuksena maatalouskontekstissa. Data kerättiin teemahaastatteluiden ja dokumenttitarkastelun avulla ja tuloksia analysoitiin kokotekstikoodauksella.

Tutkimuksessa löydettiin kolme erilaista maatalousohjelmistoekosysteemiä. Ohjelmistoekosysteemit erosivat niiden kehittyneisyydessä ja niissä oli myös erilaiset ohjelmistoalustat. Eroista huolimatta kaikki löydetyt ohjelmistoalustat pyrkivät mahdollistamaan toimijoiden yhteistyön. Lisäksi löydetyt toimijat vastasivat aikaisemmissa tutkimuksissa löydettyjä toimijoita, mutta toimijoiden motivaatiot esitettiin vain rahallisiksi tai rahattomiksi. Lisäksi havaittiin, että ohjelmistoekosysteemin tulee tarjota yksilöllinen arvolupaus jokaiselle eri toimijaroolille.

Asiasanat:	Ohjelmistoekosysteemit, Ohjelmistoalustat, Toimijat, Roolit
Kieli:	Englanti

Acknowledgements

All good things come to an end! First of all, I would like to thank my supervisor Marjo Kauppinen and my instructor Mikko Raatikainen. Thanks for hunting me down in the corridors of CS building and offering me the opportunity for this thesis. Especially, I want to thank for the encouragement and the time invested throughout the process. Without your insights, feedback and guidance, this thesis would probably never have finished. I also want to thank Luke for the valuable collaboration and the possibility to get to know with the agricultural domain.

I would also like to thank C10 and SiMiLi for the unforgettable moments outside the lecture halls. I also want to thank my colleagues for providing the flexibility between studies and work.

Finally, I want to thank my family for supporting me from my first school day until today. Special thanks go to Jammi for her academic consulting. Soon you will no longer hear excuses like *"Mä teen dippaa!"* and *"Sitten kun dippa on valmis!"*.

Espoo, October 9, 2017

Samuli Suortti

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background and Motivation

Traditionally software systems have been developed by a single company together with subcontractors [29]. However, companies have realised that they can no longer fulfill all customer demands alone [50]. *Software Ecosystems* (SECOs) have emerged to address previously mentioned issue. In a SECO, a company opens its platform to external actors outside its organisational borders [11]. SECOs have been proved, not only to decrease development costs, but also to increase innovation and to create new business opportunities [8].

Software ecosystems appeared first time in the literature over a decade ago, but research around SECOs took of only seven years after the first appearance. At the time of writing this thesis, the research field has shown signs of maturity. However SECO research still lacks theories, methods, and tools that are specific to SECOs [35]. The current theories related the software platform of the ecosystem are based on *software product line* research. Also, the SECO actors are based on findings from *business ecosystem* research.

It is vital for SECO success to be able to attract and retain great enought number of actors. The SECO must not only present an interesting value proposition to its actors, but also be to capable of delivering the promised value through the software platform. Hence, it is very important to identify the ecosystem actors and their motivations. In addition, the software platform must be built in a way so it satifys the actor needs.

In this study we concentrated on SECOs in agricultural domain. In agriculture as well as in other manufacturing areas, digitalisation has transformed field for good. In the past, automation has changed how farmers operate on daily basis but in the near future we will also witness the emergence of industrial internet in agriculture. The industrial internet will enable new kind of data gathering, which can be later aggregated into different kind of services.

1.2 Research Problem and Questions

The research problem of this thesis is stated as follows:

What is important on a platform of a software ecosystem to satisfy actors' expectations?

The problem is broken down into following research questions:

- **RQ1:** *What are the typical actor roles in a software ecosystem?*
- **RQ2:** *Why do actors contribute to a software ecosystem?*
- **RQ3:** *What is the role of software platform in a software ecosystem?*

The main goal of this thesis is to enable experts from various industries to analyse example SECO. Firstly, the reader should be able to understand the benefits from SECOs. Secondly, the reader should be able to identify ecosystem actors and group them by roles. Finally, the reader should be able to determine the value proposition of the software platform of the ecosystem.

1.3 Structure of the Thesis

Figure 1.1 presents the breakdown of the structure of this thesis. In Chapter 1, we define objectives. In the first Chapter, we also provide a background and motivation to the thesis topic. In Chapter 2, we present the methodology for both literature and empirical study of this thesis. In Chapter 3, we introduce the key concepts of this study based on findings from the current research. Firstly, we define SECO and its main elements. Secondly, we define SECOs actor roles. Thirdly, we will examine software platform of the ecosystem. In Chapter 4, we present the results found from the case study. In Chapter 5, we discuss the findings from the case study and reflect the results to concepts from the literature. In Chapter 6, we will present the most significant findings in the light of the research problem.

The Figure 1.1 also illustrates how we address the research questions of this thesis. The solid line represents the primary data source, whereas dashed line presents secondary data source. The secondary data was used to validate the answers provded by the primary data.

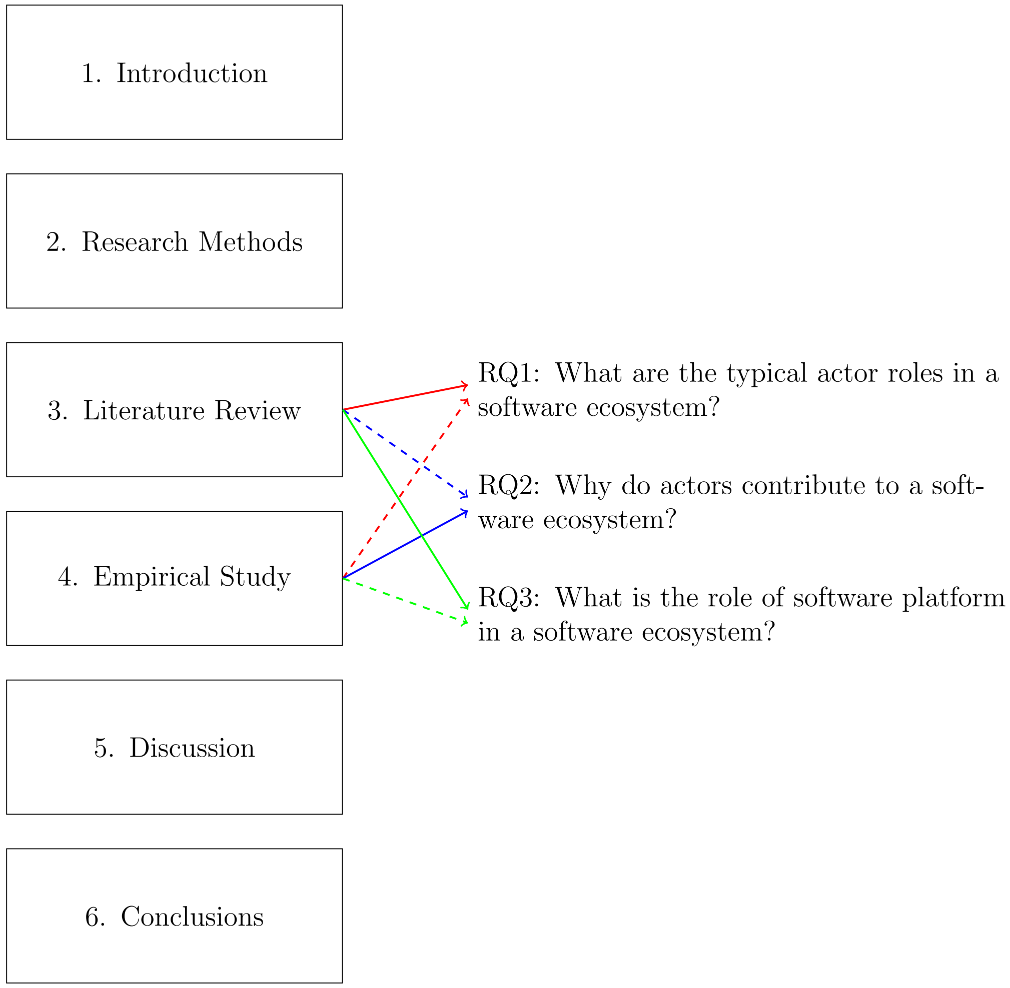

Figure 1.1: Thesis Structure

Chapter 2

Research Methods

2.1 Literature Review

The objective of the literature review of this thesis is to introduce the fundamental concepts of the thesis. The collection of material for literature review started by searching and analysing recent SECO systematic literature reviews (SLR). During the SLR analysis, we gathered relevant papers for later analysis from the references list. In addition to materials found from SLRs, we analysed research papers published in two conferences in the last five years. The former conference, The International Conference on Software Business (ICSOB), focuses on fields such as software production, software ecosystems and software product management [4]. The latter conference, International Systems and Software Product Line Conference (SPLC), concentrates on software product lines and software product family engineering [5]. During the analysis, we noticed that only a few SECO papers focused on software platforms. Due to lack of material, we extended our literature review to cover Software Product Line (SPL) text books, such as "Software product line engineering: foundations, principles and techniques" by Klaus Pohl and "Design and use of software architectures: adopting and evolving a product-line approach" by Jan Bosch [9, 44].

2.2 Empirical Study

2.2.1 Study Design

The research method used in the thesis is the *case study method*. The method allowed us to do an explanatory research and test the theories found in the literature [42]. As the primary research technique we chose, *theme interviews* and, as the secondary technique, we chose *documents analysis*. The theme interviews allowed us to ask more in-depth follow-up questions when the interviewee brought up new and relevant topics [31]. The documents, on the other hand, enabled us to create a richer image and reinforce initial findings from the interviews [42].

The subject of the case study was chosen based on a single criterion: the subject must have previous software ecosystem initiatives. Furthermore, the research group had previous collaboration with the subject selected for this study.

2.2.2 Preparation

The main themes of the interviews were based on the research questions and findings from literature review presented in Chaper 3. Also, a light website analysis was conducted to become acquainted with the subject organisation and their business. We implemented the themes, the interview questions as well as the course of the interviews to follow the practices described by Krueger and Casey in their book Focus Group [30]. After the first iteration of interview questions, the questions were reviewed by a qualitative research expert. With the help of the expert analysis, we were able to improve question neutrality, the intelligibility of the terms, and the relevance of questions.

The selection criteria for interviewees included a contract of employment with the subject company and a prior participation in a SECO project. Another criterion for the interviewees was that they should have different focuses in their daily work.

2.2.3 Data Collection

We conducted four interviews in total. All the interviews were conducted one subject at a time, and we used a single interview template in all interviews (Appendix A). However, we emphasised the themes the interviewee had the most knowledge of. The interviewees had expertise in the following areas: agricultural domain, technical, end users, agricultural domain. In the first and last interviews, we covered all interview themes, but in the second interview, we concentrated on software platforms, whereas in the third interview we focused on SECO actors.

During all the interviews, we had three interviewers present. The different interviewer roles were dealt based on prior knowledge of each theme. The author of this thesis acted as the primary interviewer in all interviews. The supervisor of this thesis acted as the secondary interviewer for ecosystem and actor themes. Finally, the advisor of this thesis acted as the secondary interviewer for the software platform theme. All interviewers took individual notes, which were used later in analysing phase. Also, the audio of all interviews was recorded and interviews were transcribed by an expert service.

We gathered the material for the document analysis from subject company's website. In addition, in case the interviewee presented us some documents during the interview, we requested access to the documents via email. In the end, the study included nine documents: two articles, four presentations and three short documents.

2.2.4 Data Analysis

In this thesis, we applied a coding approach to the data analysis. Coding is a whole-text analysis technique, which aims to reduce the size of data. In addition, coding also eases data retrieval and organisation [42]. We used ATLAS.ti qualitative analysis software to aid the analysis.

The analysis process is illustrated in Figure 2.1. The process was heavily inspired by the process steps presented by Ryan and Bernard in their article "Data management and analysis methods" [45]. Due to the limited amount of total raw data, we decided on analyse all interviews and documents. In this study, we used the same process to analyse both interview transcriptions and documents. In the second step of the process, we identified general themes. We derived the main themes form research questions and interview notes. Research questions provided us main themes such as "platform" and "ecosystem", whereas interview notes introduced us common topics such as "data" and "background". We transformed the themes into initial codebook and continued to the next step of the process. In the third phase, we marked the text with the codes from our initial codebook. The codes worked as tags to ease up the data retrieval later in the process. In the first marking iteration, the coded text blocks were rather large. Once we had marked all the raw data, we built our final codebook. In the process of creating the final codebook, we turned theme codes into code groups and reanalysed all marked texts and reassigned the text blocks with more descriptive codes. We derived the new codes from results of the literature review and initial hypotheses of the possible empirical results. For example, we transformed "actor" theme code into a code group and re-coded all "actor" code instances with codes such as "support service provider" or "end user". We decided not give the codes any descriptions, due to the limited amount of participants in this study and because the codes were based on results from literature or they were self-explanatory. After the codebook iteration, we implemented code hierarchies and relationship maps. The relationship maps enabled us to build empirical models, which were compared to findings from the literature.

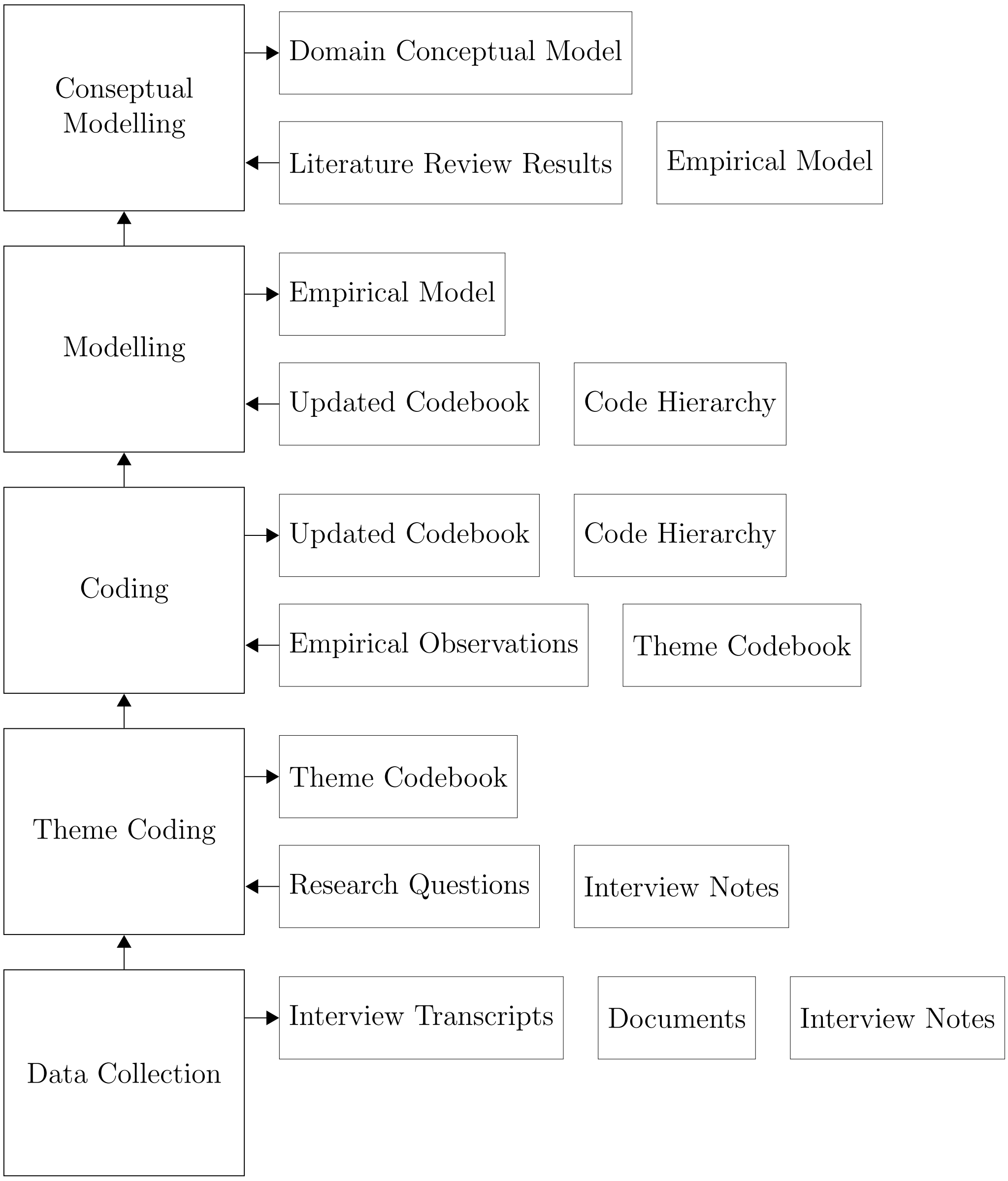

Figure 2.1: Analysis Process

Chapter 3

Literature review

3.1 Software Ecosystems

Software ecosystem (SECO) is a relatively new concept. The term "software ecosystem" was introduced in 2003 by Messerschmitt and Szypersky in their book Software Ecosystem Understanding an Indispensable Technology [39]. Manikas and Hansen conducted a systematic literature review on SECOs in the year 2013. At the time the SECO research was fairly immature but during the following years, the field has shown signs of maturity [35]. In this Section, we will use three strategies to explain the SECO concept and provide a SECO definition. First, we will analyse the evolution of the SECO concept. Then we will introduce three main SECO elements Manikas and Hansen identified in their research. Finally, we will analyse different SECO definitions from literature. In addition, we present an ecosystem typification.

The evolution of SECO concept can be divided into three stages, which are presented in Table 3.1. The word *ecosystem* originates from ecology. Merriam-Webster dictionaries define it as "a branch of science concerned with the interrelationship of organisms and their environments" [2]. Ecology is a branch of biology, whereas biological ecosystems are considered to be the base of the SECO evolution. As stated in the definition, biological ecosystems capture the idea of multiple actors working as one unit together with the environment. The ideas and concepts from biological ecosystems transferred into business context results in the next stage of the evolution, *business ecosystems* [23]. Even though the contexts are very different, the analogy between the two is applicable. Iansiti and Levien illustrate the analogy by comparing species in a biological ecosystem to companies, products and technologies in business environment. In addition, they present a parallel between the two: "Both are characterized by a large number of loosely

Type of ecosystem	Definition
Biological	”The complex of a community of organisms and its environment functioning as an ecological unit” [3]
Business	”Business ecosystems are formed by large, loosely connected networks of entities, which interact with each other in complex ways, and the health and performance of each firm is dependent on the health and performance of the whole” [22]
Software	”The software and actor interaction in relation to a software platform, that results in a set of contributions and influences directly or indirectly the ecosystem” [35]

Table 3.1: Ecosystem Evolution

interconnected participants that depend on one another for their effectiveness and survival. If the ecosystem is healthy, individual participants will thrive; if the ecosystem is unhealthy, individual participants will suffer” [24]. The final stage of the evolution is *SECO* itself. SECOs are kind of business ecosystems but the distinctive factor is the equal importance of software interaction compared to participant interaction [35].

Another way of looking into SECOs is to define the common SECO building blocks. Manikas and Hansen identified three main SECO elemets based on SECO definitions from the literature: *connecting relationships, business* and *common software* [37]. The elements can be linked to a previously presented stage of SECO evolution. The first main element, connecting relationships, captures the essence of biological ecosystems. In SECO literature and this thesis, the participants of the ecosystem are referred as actors. Actors and their roles within an ecosystem are scrutinise more in detail in Section 3.2. The authors listed *business* as the second main element of SECO. In short, the element describes the ecosystem environment. Manikas and Hansen pointed out that the element should be examined broader than just as revenue models: also non-monetary benefits should be included [37]. Due to the scope of this thesis, we will not analyse the element any further. The last main element is *common software*. The existence and importance of the software is the deciding factors that transforms business ecosystems into a SECOs. In this thesis, we will refer to the software as a *software platform*, which is described more profoundly in Section 3.3.

Currently, there exists various SECO definitions [29]. A sample of different SECO definitions are presented in Table 3.2. From the definitions, it can be clearly seen that the software is described in each definition, but the term varies. In earlier definitions, the software was referred as unstructured group with terms like "collection of software products" [39], "set of software solutions" [11] or "a collection of software projects" [34]. Additionally, in more recent definitions, the term "platform" has stabilized in the literature.

Also, in earlier definitions the business and relationships aspects are left with lesser notice, Due to the technological background of the authors [37].

In this thesis, we chose to use the latest SECO definition proposed by Manikas with a small modification, so it is in line with terms used in this thesis:

> *The software and actor interaction in relation to a software platform, that results in a set of contributions and influences directly or indirectly the ecosystem.* [35]

In this thesis, we replaced Manikas original definition of *a common technological infrastructure* with the term *software platform*.

Due to the variety of different kinds of SECOs, Kondel and Manikas acknowledged a need for an ecosystem typification. Their typification is based on differences in software platforms of the SECOs. They identified a total of four different ecosystem types: *Cornerstone ecosystems*, *Standard-based ecosystems*, *Protocol-based ecosystems* and *Infrastructure-based ecosystems*. Cornerstone ecosystems include a software platform, where the platform functionality is extended by actor contributions. The platform is in most cases run by a single strong player. Examples of such ecosystems include for example Android and iOS. Majority of previous research has concentrated on this type of ecosystems. Standard-based ecosystem platform, on the other hand, does not include software. Instead the platform is a standard specification, which describes requirements for actor contributions. While cornerstone ecosystems are often run by a single player, standard-based ecosystems are governed by a consortium. Protocol-based Ecosystem is a more flexible and restrictive version of a standard-based ecosystem. In infrastructure-based ecosystems, the software platform is replaced with a shared set of technical environment and tools between actors. [29]

3.2 Actor Roles in Ecosystems

Actor relationships and interactions are a vital part of SECOs [35]. The actors interact directly or indirectly with each other [29]. A typical example

Author	**Definition**
Messerschmitt et. al. (2005)	"Traditionally, a software ecosystem refers to a collection of software products that have some given degree of symbiotic relationships." [39]
Bosch (2009)	"A software ecosystem consists of the set of software solutions that enable, support and automate the activities and transactions by the actors in the associated social or business ecosystem and the organizations that provide these solutions" [11]
Lungu (2009)	"A software ecosystem is a collection of software projects which are developed and which co-evolve together in the same environment." [34]
Manikas and Hansen (2013)	"The interaction of a set of actors on top of a common technological platform that results in a number of software solutions or services." [37]
Sevon (2013)	"A combination of stakeholders use common technological platform to collaboratively deliver competitive value in order to satisfy stakeholders needs and expectations." [46]
Manikas (2016)	"The software and actor interaction in relation to a common technological infrastructure, that results in a set of contribu- tions and influences directly or indirectly the ecosystem." [35]

Table 3.2: Ecosystem Definitions

of an actor in an ecosystem is an individual, organisation, governmental entity, association or community [29]. Each actor implements one or more roles in an ecosystem [35]. A role is defined by a type of contribution, motive, and relationships to other actors [29].

We identified over 90 roles in 20 software ecosystem research papers and articles. Due to the immaturity of the field [37] there are no established terms for different roles. From the literature analysis, we deduced five main ecosystem actor roles. The main actor roles and their definitions are listed in Table 3.3 and found actor role synonyms are presented in Table 3.4. Also, in Figure 3.1 we illustrate actor role contributions towards the software platform of the ecosystem.

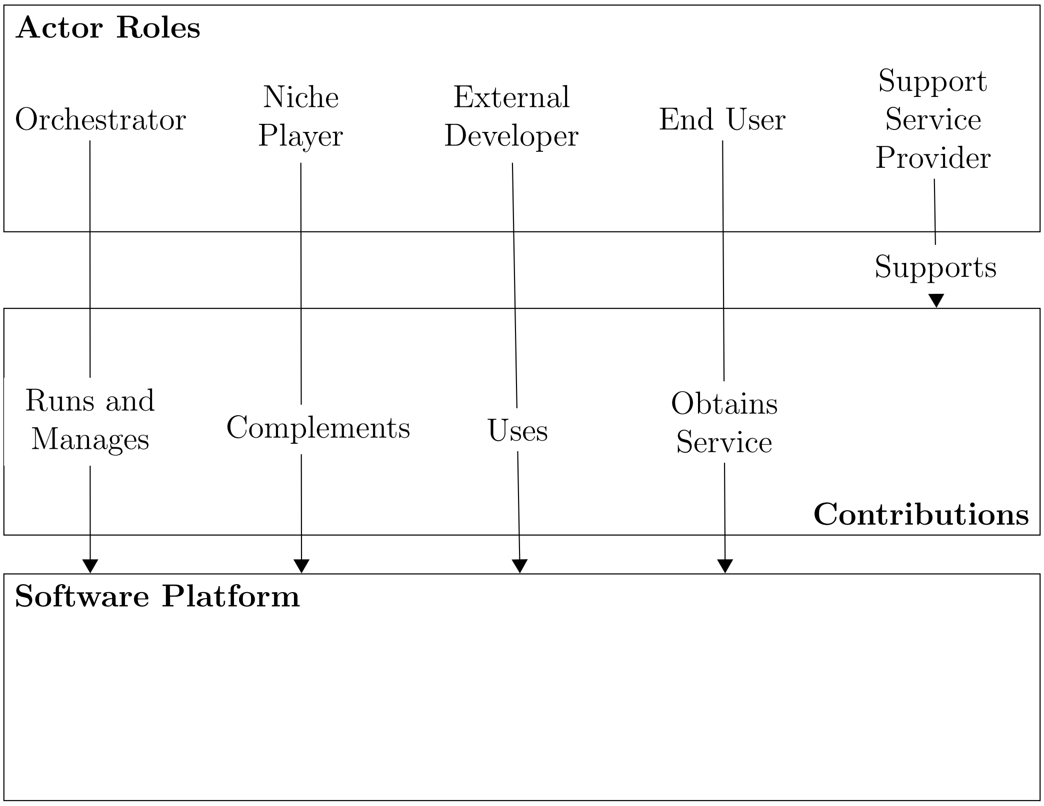

Figure 3.1: Actor Role Contributions

In early SECO research Geir Hanssen identified three key roles: *keystone*, *third-party organisations* and *end-users* [20]. According to Iansiti and Levien the keystone role descents from biological ecosystems, where it is referred as *keystone species* [23]. In a biological ecosystem, the *keystone* is

Role	Definition
Orchestrator	Governs [”Manages”] the SECO by running the platform, creating and applying rules, processes, business procedures, setting and monitoring quality standards and orchestrating the SECO actor relationships [37].
Niche Player	Complements the work of the orchestractor by providing value to the ecosystem. Contributes to the SECO by typically developing or adding components to the platform [37].
External Developer	Makes use of the possibilities the ecosystem provides and thus providing indirect value to the ecosystem. This actor is external to the SECO management and usually has an activity limited to the actor’s interest [37].
End User	Individual or company that obtains or purchases a complete or partial product of the SECO [37].
Support Service Provider	Enable or help other actors with their tasks within the SECO [32]

Table 3.3: Actor Role Definitions

Role	Synonyms
Orchestrator [12, 21, 35, 36, 47]	Keystone [12, 20, 26, 36, 37, 50], Shaper [26], Platform vendor [49], Platform owner [17], Service enabler [49], Provider [32, 49]
Niche Player [12, 26, 36, 37, 50]	Follower [26], Service providers [19], Application developer [49], Producer of complementing service [49]
External Developer [36]	Single Developer [36], Developers [17], External Developers [36]
End User [20, 32, 37, 49]	Customer [14, 19], User [28, 36], Buyer [17]
Support Service Provider	Infrastructure providers [48]

Table 3.4: Actor Role Synonyms

described as a species, which in a case of extinction would affect the whole ecosystem. They also provide an extended definition for *keystone* in a business ecosystem context. They states that *keystone* provides ecosystem-wide stable and predictable set of common assets [24]. The common assets in a business ecosystem can be viewed as the software platform in a SECO. In SECO literature the *keystone* role is also often referred as an *orchestrator*. In this thesis, we will use the term *orchestrator* because it better describes the responsibilities and tasks of the role. *Orchestrator* has a central role in the ecosystem [49]. Thus, we introduce it as the first main role. The most important responsibility of an *orchestrator* is the *ecosystem governance* [35]. *The ecosystem governance* includes tasks such as software platform management [12], actor relationship management [37] and ecosystem health monitoring [47].

To represent the second key role, Hanssen identified third-party organisations. He defined third-party organisations as "organizations that use the central technology as a platform for producing related solutions or services" [20]. Manikas and Hansen divided third-party organisations further into *niche players* and *external developers* [37]. Their division was based on differences in contributions and levels of commitment to the ecosystem. In this thesis, we consider both of them as main roles. A single *niche player* alone has no impact on other actors or the ecosystem but all niche players combined as a group form the second most important entity in an ecosystem [22]. *Niche players* are responsible for ecosystem innovation and value creation [24]. They contribute to the ecosystem by developing specialised extensions to the platform and are motivated by business opportunities [37]. *External developers*, on the other hand, provide indirect value to the ecosystem through the usage of a SECO platform [37]. Their commitment to an ecosystem is rather limited, and they are often motivated by fame and knowledge instead of monetary compensation. Their contributions to the ecosystem can include for example bug reporting and ecosystem promotion.

In addition, Hagel et al. introduce three subtypes for *niche player*: *disciple*, *influencer*, and *hedger*. *Disciples* and *influencers* contribute solely to one ecosystem. *Hedger*, on the other hand, participates in two or more ecosystems at the same time. The different contribution strategies can be analysed based on risk levels. *Disciple* and *influencer* share the risk with the ecosystem they contribute to, whereas *hedger* reduces its risk by spreading its investment across multiple ecosystems. Higher risk is followed by a bigger possible reward. The roles investing in one single ecosystem, have a better chance of gaining a better position in the market. *A disciple* is an early adaptor of an ecosystem and also works as an evangelist and spreads the word of the ecosystem. *An influencer*, on the other hand, can influence the *orchestrator* of the ecosystem. Iyer et al. introduced also a special kind of hedger, *a bridge* [25]. *A bridge* operates actively in two different ecosystems. It can transfer knowledge, people or money between the ecosystems. The third party organisations and their sub-roles are illustrated in Figure 3.2.

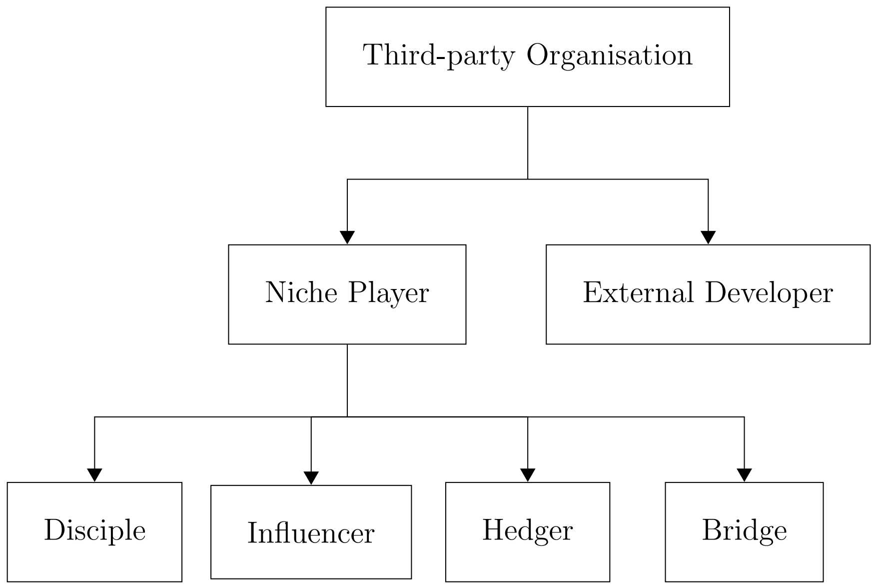

Figure 3.2: Thid-Party Sub-Role Hierarchy

The last key role Hanssen described is an *end user*. In literature, the *end user* often has no definition. However, Konstantinos Manikas defined *an end user* as follows: "person, company, an entity that either purchases or obtains a complete or partial product of the SECO or a niche player" [37].

The final main role we included is *support service provider*. , which is commonly left out in the literature. However, we want to stress that they are an important part of the network of actors. *The support service providers* help and enable other ecosystem actors to achieve their goals. Examples of such providers are for example internet service providers, research and consultancy companies and hardware manufacturers. [32]

3.3 Software platform

Often in SECO literature, the software part of the ecosystem is only referred as a platform without any proper definition. For the definition of a software platform, we need to look back to the origins of SECO. Jan Bosch describes the birth of SECOs as the following "Once the company decides to make its platform available outside the organisational boundary, the company transitions from software product line to a software ecosystem." [11]. *Software product line* (SPL) is defined as a set of software-intensive systems sharing a common, managed set of features that satisfy the specific needs of a particular market segment or mission. They are also developed from a common set of core assets in a prescribed way [13]. Also, SPL can be treated as a synonym for *software product family* (SPF), which is seen a specialisation of a product family.

Author(s)	Definition	Example
Meyer et al. [40, p.7]	A product platform is a set of common components, modules, or parts from which a stream of derivative products can be efficiently created and launched	Black & Decker, HP Inc Jet Printers
Meyer et al. [41]	A software product platform is both an architecture and an implementation of architecture that comprises core subsystems that propel a family of software products or internal corporate applications	Visio Corporation
Ommering [51]	Reuse approach separates creation of reusable components from the creation of products, where both take place under a common architecture	Philips Consumer Electrics
Bosch [9, p.312]	Platform is referred as the reusable product-line assets at the top level	Axis Communications AB, Securitas Larm AB, Symbian
Bosch [10]	Platform captures all functionality that is common to all products or applications and it typically includes a standardized inftrastructure	SymbianOS
Clements et al. [13]	Platform is a core software asset (i.e. architecture, component) base that is reused across systems in the product line	Nokia, Motorola
Pohl et al. [44, p.6]	A software platform is a set of software subsystems and interfaces that form a common structure from which a set of derivative products can be efficiently developed and produced	Ford, Symbian
Bosch [11]	Platform is referred as the product line architecture and shared components	SalesForce, eBay, Facebook
Gawer et al. [18]	Product platform as a set of assets organized in a common structure from which a company can efficiently develop and produce a stream of derivative products.	iOS, Android

Table 3.5: Software Platform Definitions

In Table 3.5, we listed different platform definitions chronologically between years 1997 and 2014. In the list, we included the definitions for the following concepts: *platform*, *product platform* and *software platform*. For example, Meyer and Lehnerd use the term *product platform* when describing product families [40], whereas Jan Bosch uses plainly the concept *platform* without any prefix or suffix in his SPL research [9–11]. After analysing the definitions, we identified three key characteristics on how *software platforms* differ from other software systems. The key characteristics are illustrated in Figure 3.3.

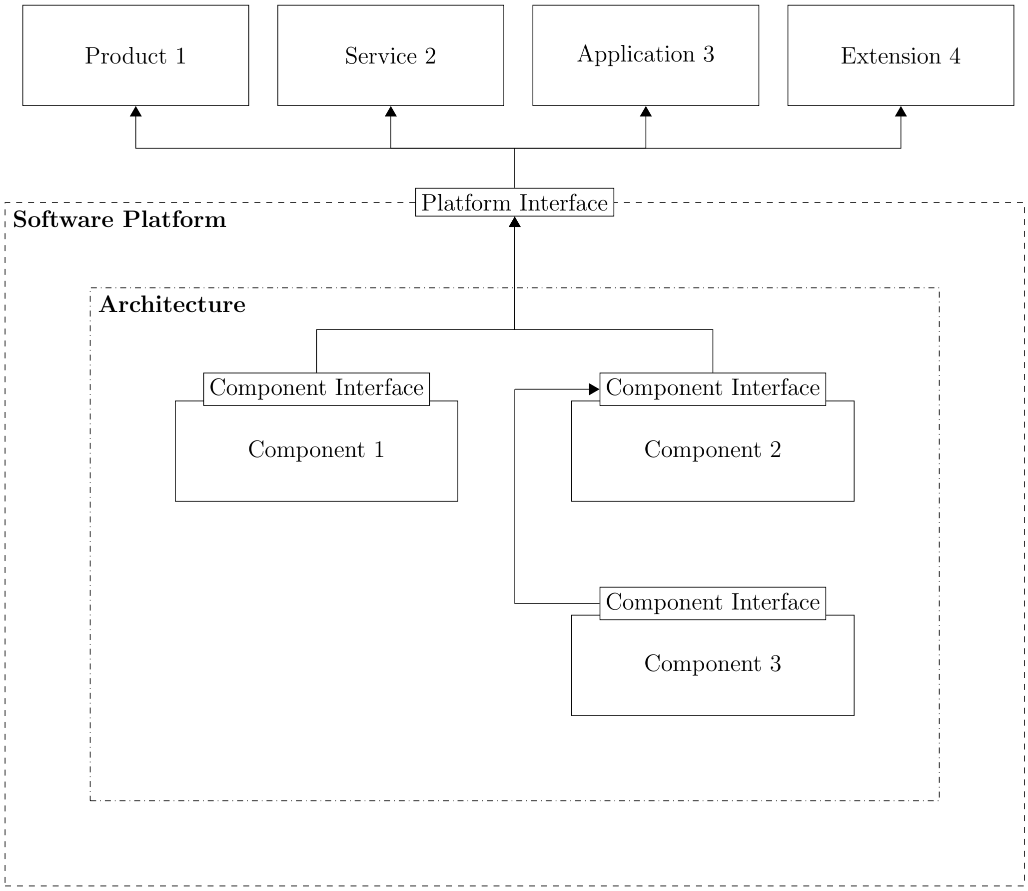

Figure 3.3: Software Platform Characteristics

The first key characteristic is the separation of the platform and the derived products. In his definition, Rob van Ommering explicitly states that the reusable components (the platform) are separated from the products [51]. Other platform definitions implicate the separation more indirectly. For example, Meyer and Lehnerd refer to the separation as following "A product platform...from which a stream of derivative products can be efficiently created and launched." [40], whereas Gawer and Cusumano refer to the separation as "Product platform...from which a company can efficiently develop and produce a stream of derivative products." [18]. The derived product is also referred as "application" [41] [27], "service" [18] and "extension" [27].

The second key characteristic is *core assets* under *architecture*. *Core assets* are described as building blocks of the platform [9, 11, 13, 18, 40, 41, 44]. Clements and Northrop define *core asset* as "A reusable artefact or resource that is used in the production of more than one product in a software product line. A core asset may be an architecture, a software component, a domain model, a requirements statement or specification, a document, a plan, a test case, a process description, or any other useful element of a software production process." [13]. From the definitions, we found the following references to *core assets*: "components" [11, 40, 51], "modules" [40], "parts" [40], "subsystems" [41, 44] and "assets" [9, 18]. Clements and Northrop include *architecture* in *core assets*, but in this thesis, we excluded it from the definition. We examine the *architecture* as a higher level concept, which surrounds the *core assets*. Software Engineering Institute at Carnegie Mellon University describes *architecture* as a structure of structures. Thus, *architecture* includes other *core assets* such as components and subsystems [33]. Gawer and Cusumano also separate *platform architecture* from *core assets* by stating: "set of assets organised in a common structure", where the common structure refers to the platform *architecture* [18].

The third and last key characteristic of a platform are *interfaces*. Meyer and Lehnerd describe *interfaces* as the key enablers and the power sources of a platform [40]. *An interface* defines the core asset functionality and its requirements from other core assets or its environment [9, p.220]. In *software platforms* there exist two types of *interfaces*. The first types of *interfaces* are found between core assets within the platform architecture and the second type of *interfaces* are located between the *software platform* and the derived products. Both interface types aim for the same benefits: they allow hiding complexity through abstraction [38], disclosing functionality and data [15] and enable platform capability reuse [40, p. 181].

We have listed platform examples related to each definition in Table 3.5. Meyer and Lehnerd present Black & Decker as case example how implementing a product platform improved their position in power tool market during 70s [40]. The platform was used internally within the company and enabled the reuse of components and processes through their product lines. More recent platforms include mobile operating systems such as Apple iOS and Google Android [18]. These platforms differ from Black & Decker not only because the platform is software based but also that the platform is available outside of the organisation boundary.

As we stated in the beginning of this Section, the literature lacks a definition for *a software platform* in SECO context. However, the definitions provided by SPL literature are in most parts applicable to SECO context as well. With some minor modifications, we can transform SPL software platform definition to suit SECO context. Firstly, in SECOs the platform include users outside the traditional organisation boundary, whereas in SPL the platform is only used by employees and contractors of a single company. Secondly, the derived products are often referred as services in SECOs. Thus we propose new definition software platforms in SECO context.

a set of assets organized in a common structure from which actors can efficiently create complementary services.

Chapter 4

Results

Natural Resources Institute Finland (Luke) is a research and expert organisation formed the in the year 2015 after mergin of four research organisations: MTT Agrifood Research Finland, Finnish Forest Research Institute, Finnish Game And Fisheries Research Institute and Information Center of the Ministry of Agriculture and Forestry. Despite the young age of the organisation, Luke brings together over 281 years of research experience. Lukes mission is to: ”work to advance the economy and the sustainable use of natural resources.” [6].

4.1 Agricultural Ecosystem

An agricultural ecosystem can be viewed as a business ecosystem, which provides the base for SECOs described in Section 4.2. One interviewee explained agricultural ecosystem by describing food production chain. The interviewee’s breakdown of food production chain is illustrated in Figure 4.1. The different phases can be linked in the main agricultural ecosystem roles: *input producer*, *farmer*, *processor*, *retailer* and *consumer*.

Figure 4.1: Food Production Chain

Input producers were mentioned in three interviews. However, none of the interviewees included a proper role description, task or responsibility list. One interviewee, on the other hand, named seed and fertiliser producers as example actors of the *input producer role*.

Farmers were mentioned in all of the interviews and most of the documents. This could be because the interviewees concentrate on primary production in their daily work. One interviewee simplified the objective of a *farmer* as following: to transform farm inputs into yields as efficiently as possible. The main tasks for achieving that object included planning, cultivation, protection and harvesting.

Processors were mentioned in three interviews. Similarly to *input producers*, there were no descriptions of common responsibilities or tasks. One interviewee illustrated *processors* with two examples: a processor can be a brewery, which can make contracts with as many as 10 000 farms about their production or a company producing baby food from pure oats. Another interviewee named protein extraction from secondary streams such as wheat and rice peels as an example of a *processor*. From examples above, we can conclude that *processor* is a role, which adds value to the input material through different kind of processes.

Retailers were brought up least in this study. *Retail* as a role was only referred directly in the food production chain description. On the other hand, the role was indirectly referred in many examples in the interviews. The examples included e-commerce vendors, brick and mortar retailers as well as traditional outdoor market traders. *Retailers* are responsible for moving the products from processors to consumers.

Consumers are the end users of the output of the food production chain. One interviewee identified three different consumer segments based on the age of consumer: children, adults and elders. Another interviewee emphasised that all *consumers* have different needs and consumption habits, which creates a need for different kinds of niche products.

In addition to the roles mentioned above related to food production chain, the interviewees mentioned *administration* as an important actor in the ecosystem: "Governments have been conventionally regulating farming actions by legislation and e.g. production subsidy schemes" [43]. One interviewee explained that all developed countries, New Zealand as the exception, subsidises primary production. In Finland Ministry of Agriculture and Forestry grants the subsidies to farmers but in return, the farmers need to document and share their plans for the coming year and also report the realisation. In addition to *administration*, *logistics* were brought up as an important supporting actor in the ecosystem. *Logistics* are responsible for example transferring good on the food production chain.

4.2 Agricultural Software Ecosystems

During the interviews, the interviewees presented three different examples of agricultural software ecosystems: *Cropinfra*, *ISOBUS* and *REKO*. The first two ecosystems are related to projects carried out by Luke. *REKO*, on the other hand, is a community-driven ecosystem, in which Luke has no part.

The *Cropinfra* concept ecosystem project took place between years 2008 and 2014 and was executed by MTT Agrifood Research Finland, one of the predecessors of Luke. The high-level goal of the project was to "design and implement an Internet-based networked crop production platform to assist farmers to operate efficiently and fulfil farming demands using present and future technologies" [43]. The project resulted in five value proposition: provide access to services; provide more high-quality food products; document all processes; environmentally friendly production and improved resource management. More concretely, requirements for the platform was to provide tools for improved farming decisions, enable interoperability between farming equipment, devices and farm information systems (FMIS) and information sharing between farmers and other interested parties.

The ISOBUS ecosystem is based on an ISO 11783 standard, which defines a communication protocol between a tractor and agricultural and forestry machinery, so-called implements [7]. The standard aims to improve interoperability between tractors and implements from different manufacturers. Luke is currently working on a project called MATYKO 2025. The goal of the project is to improve the competitiveness of Finnish agricultural machinery manufacturers in the global markets. Through the utilisation of the standard, the Finnish small and medium-sized companies can find their niche markets.

The third SECO example was community-driven *REKO*. The word "REKO" is an abbreviation for a Swedish word "Rejäl konsumtion", which translates into "fair consumption". The ecosystem aims to allow consumers to provide fair compensation to food producers and to give consumers the power to choose and verify their producers. The ecosystem also enables a marketplace for niche products, which are not available in big retail chains due to little demand. The ecosystem is based on an online marketplace combined with a pop-up market. In practice, producers display their inventory online for the consumers. Consumers can reserve their desired items in advance and fetch the chosen goods during the next pop-up market event.

Figure 4.2 presents an illustration how the three mentioned agricultural SECOs position in the food production chain. *Cropinfra* and *ISOBUS* ecosystems concentrate solely on the production phase of the chain. Both of the ecosystems are independent of one another, but the possible success of *ISOBUS* ecosystem could reflect on *Cropinfra*. *REKO*, on the other hand, skips the retail phase and connects food producers straight to consumers.

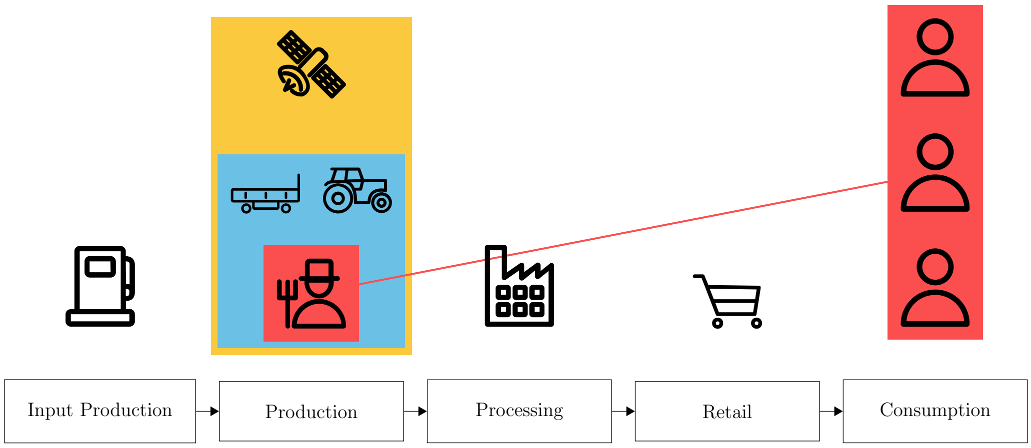

Figure 4.2: Software Ecosystems in Agriculture

4.3 Actors in Agricultural Software Ecosystems

In this Section, we illustrate a mapping between the ecosystem actors we discovered in the interviews and documents, as well as the actor roles identified in the literature. All the following illustrations comply with the following model: The coloured rectangles represent the actor roles found in the literature. The white rectangles inside them represent the actors found in the interviews and documents.

Cropinfra

Figure 4.3 shows the actor categorisation of the Cropinfra concept SECO. What stands out in the illustration is the missing orchestrator actor role, which is the most important actor role in a SECO. One interviewee also acknowledged the issue and brought up his interest to do further research on possible business models of the ecosystem. Through this kind of investigation, one could find out how money could be earned by running the software platform of the ecosystem. His concerns were mainly related to small profit margins of the agricultural domain.

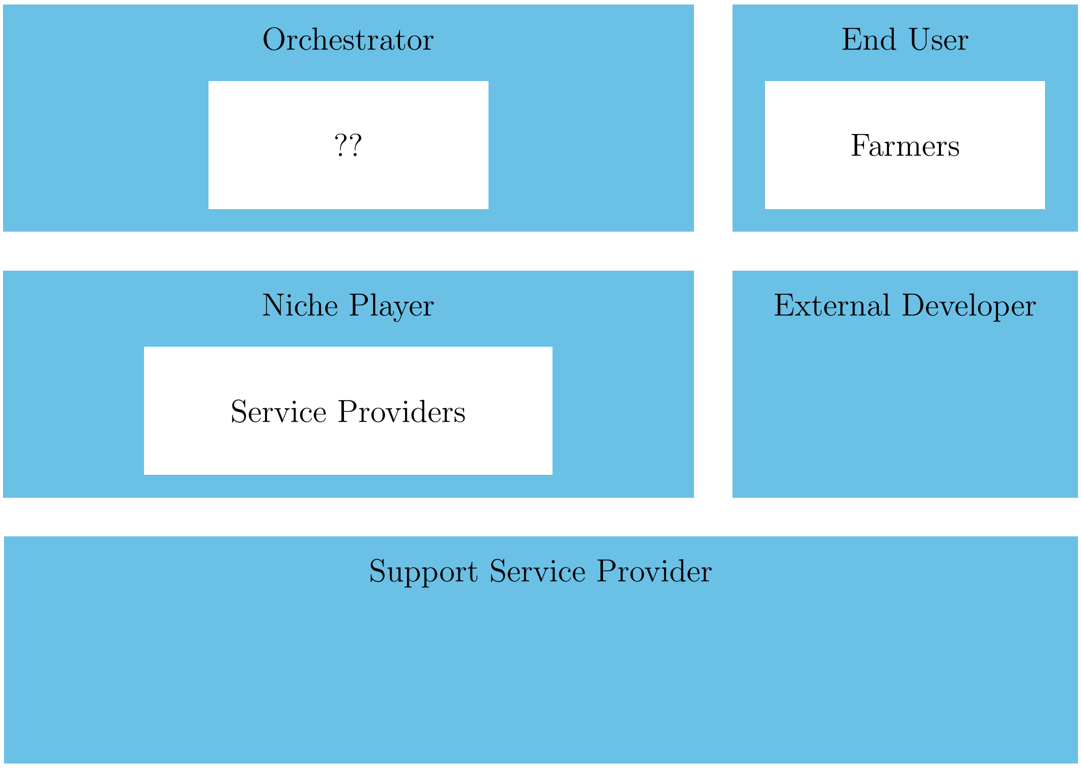

Figure 4.3: Cropinfra Actors

Niche players were mentioned multiple times both in the interviews and the documents. However, the niche players were referred as service providers. One interviewee described service providers being responsible for offering different kinds of services to end users of the ecosystem. The list of the possible service providers given by the interviewee was extensive. As an example, the interviewee presented disease forecasting as an ecosystem application. In the application, he described four different service providers: disease model service, computation service, data storage service and weather data service. The interviewee described the service provider motivation as following: ”of course the service provider wants some money for it”. Hence, we can assume that service providers are motivated by monetary compensation.

Farmers were identified as the end users of the Cropinfra SECO by one interviewee. He described the ecosystem from the farmer's perspective. Also, one document presented the Cropinfra SECO value proposition for the farmer. The proposition was summarised in the project goal as following: ”...to assist farmers to operate efficiently and fulfil farming demands...” [43]. From the goal, we can also deduct at least one motivational factor for the farmers: more efficient working. The farmer was also defined as the actor making the purchase decision of the services of the ecosystem. Support service providers and external developers were not mentioned in the interviews.

ISOBUS

Figure 4.4 provides an illustration of ISOBUS ecosystem actors.

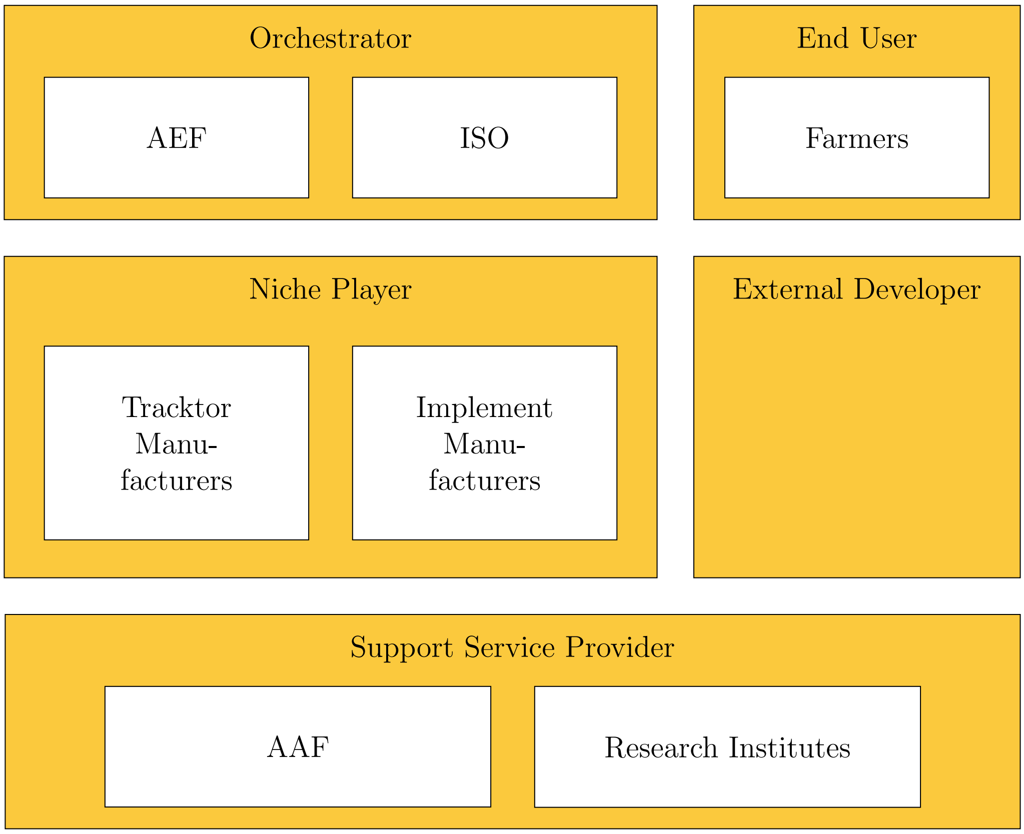

Figure 4.4: ISOBUS Actors

The orchestrator of the ecosystem was defined explicitly in a document, which presented a roadmap for digitalisation of crop production. The ecosystem is orchestrated by two organisations: ISO (International Organization for Standardization) and AEF (Agricultural Industry Electronics Foundation). The two organisations have their distinct responsibilities in the orchestration. In the document, ISO was described as an administrative organisation, whereas AEF was portrayed as dealing with more practical matters related to agriculture. AEF was described as responsible for preparing new proposals and suggesting modifications to the current standard. ISO, on the other hand, was stated being responsible in accepting change requests to the standard. The orchestrator motivation did not come up directly in either the interviews or the documents. However, according to other sources, the AEF members consist of companies, which also participate in the same ecosystem as niche players. It is possible that in addition to aiming to improve the standard, the companies are also running their interest. Hence, we can argue that the AEF member companies are indirectly motivated by monetary benefits.

The niche players of the ecosystem include currently almost all agricultural machine and device manufacturers. The companies vary from global multi-billion tractor manufacturers to small and medium-sized enterprises. The benefits of the ecosystem are similar for all the companies. However, smaller companies benefit more from the ecosystem participation than bigger companies. The ecosystem enables smaller companies to concentrate on their key competence areas which, in turn, enables niche product creation. On the other hand, the bigger companies also believe that ecosystem participation will lead to greater sales.

One interviewee described farmers as the end users of the ecosystem. She presented a business model canvas, which included an extensive list of value propositions for a farmer in the year 2025. The value propositions captured the potential benefits of the ISOBUS ecosystem, which work as the motivators for farmers. The list included three topics: easing the farming field work, improving farming processes and improving data management.

Agricultural Automation of Finland (AAF) was mentioned in both the interviews and the documents. The aim of the organisation is to "Assist in bringing ISOBUS compatible products to markets" [7]. The organisation was founded with the help of Luke and Aalto University but the member companies of AAF include solely Finnish software and electronics manufacturers. Also, Luke has a supportive role in the ecosystem. For instance, Luke offers an unofficial ISOBUS testing facility and assists in product development. As in Cropinfra ecosystem, we did not identify any external developers from the ISOBUS ecosystem.

REKO

Figure 4.5 illustrates our classification of REKO actors.

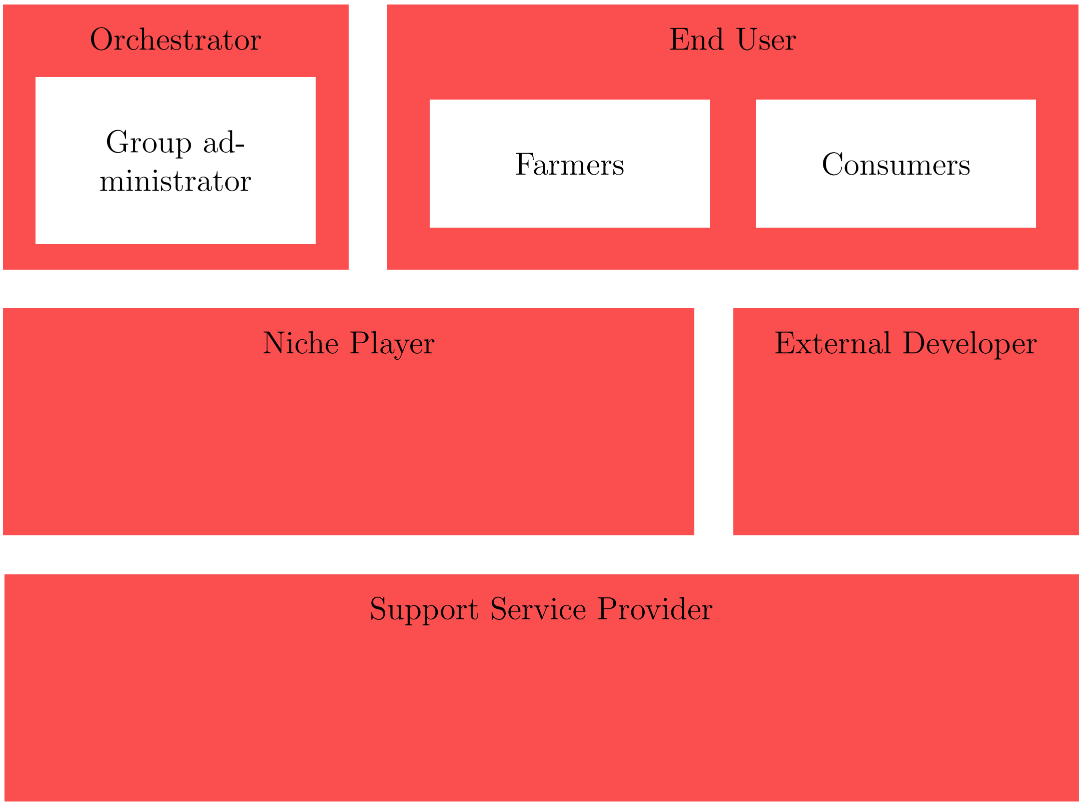

Figure 4.5: REKO Actors

One interviewee identified individual group admins as the orchestrators of the ecosystem. She described the admins being responsible for the ecosystem rule creation and supervision: The admins can accept new users to group and remove users from it if they disobey the rules. In the interviews, no motivational factors for operating as a group admin were identified.

In the interviews, we identified two actors describing the end user: the consumers and the farmers. Both of the actors have different motives for the participation. The consumers are interested in validated production processes and providing fair compensation to the producer. The farmers, on the other hand, are interested in making a profit out of niche products. Surprisingly, we did not identify niche players in any of the interviewees' descriptions of the ecosystem. In addition, we did not identify any external developers or support service providers.

4.4 Software Platforms in Agricultural Software Ecosystems

Cropinfra

The documents given by the interviewees together with their descriptions of the software platform of Cropinfra ecosystem enabled us to identify a traditional software platform from the ecosystem. One interviewee described the software platform implementation as a piece of software. After analysing the descriptions we classified the ecosystem as *a cornerstone ecosystem*.

In Figure 4.6, we present an architectural overview of Cropinfra ICT Platform concept. The Cropinfra ICT Platform describes responsibilities and relationships between the high-level components. In the figure, we have combined two illustrations contructed on the basis of personal communication with one interviewee. As in the original figure derived from the interviewee, we identified the service framework as the software platform of the ecosystem. The service framework was described to be responsible for exchanging information and user interfaces between different services. The different services are discovered from the marketplace and users are authenticated with the identity management component. The functionality and the data of the ecosystem is generated by Internet of Things enabled devices and third-party services.

The materials related Cropinfra ICT platform concept explicitly described the value proposition of the platform as: freedom to choose the most suitable bundle of services and enable easy change of service providers.

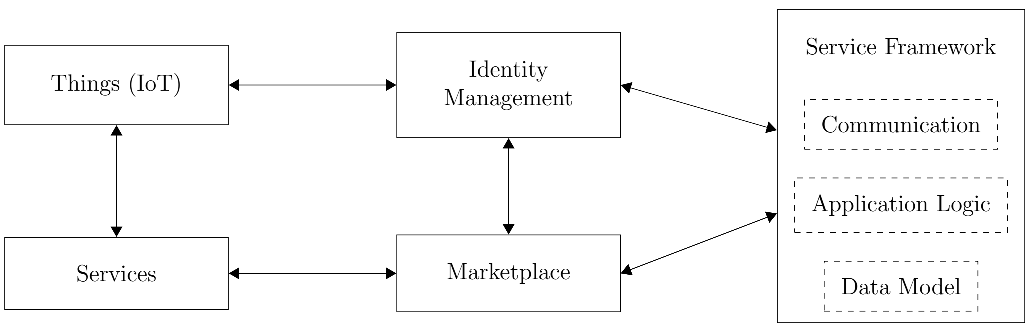

Figure 4.6: Cropinfra ICT Platform Concept

ISOBUS

One document described that *ISO 11783 standard* defines a communication between tractors and implements, on which the ISOBUS ecosystem has developed. Hence we classify the ecosystem as *a standard-based ecosystem*. At the time of writing this thesis, the first parts of the standard were released almost two decades ago, in the year 1998. Nowadays the standard to consists of 14 parts and 1400 pages. More precisely, the standard aims to improve the communication between tractors, implements and farm management information systems [1].

Figure 4.7 provides an example visualisation of ISOBUS compliant tractor and implement combination. However, both the tractor and implement include several devices, which are connected to the bus. Task controller is located in the tractor and is responsible for controlling the implement connected to it and logging the realised work. The virtual terminal is the user interface for the farmer to follow up the work realisation real time. The implement connected to the tractor and the same bus includes an electronic control unit. The unit is responsible for controlling the machine according to commands from task controller. Sensors can be found on both the tractor and machine. Sensors connected to the bus provide information to other connected devices to the same bus about the environment. The information can be used either to alter the currently planned execution or simply added to the logged. The standard does not put any constraints about how many ISOBUS compatible devices can be connected to one bus so in theory there could be multiple machines and sensors connected at once.

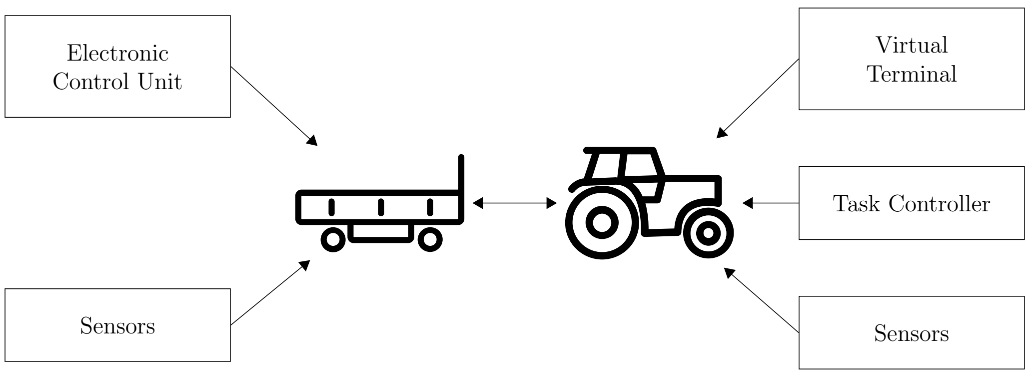

Figure 4.7: Example ISOBUS Tractor and Implement Combination

REKO

The REKO ecosystem did not include any central controller software or standards. However, all the different REKO instances we described to share the common tools. The common tools combined act as the software platform of the ecosystem. Thus, we classify REKO as an Infrastructure-based ecosystem.

Figure 4.8 illustrates the platform of the ecosystem. The black rectangles present individual REKO instances, which are not connected to each other. However, all the instances share Facebook groups as the communication tool and money transactions are done by using mobile payments systems provided by banks.

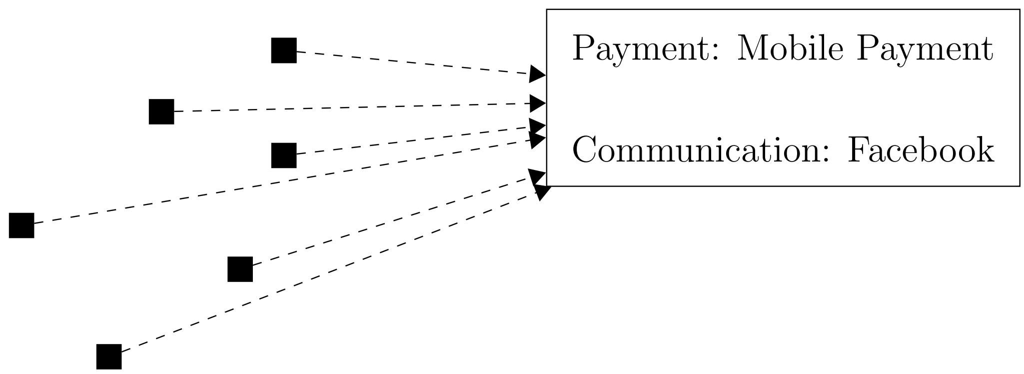

Figure 4.8: REKO Platform

4.5 Problems

The most frequently mentioned problem by the interviewees was related to data ownership. One interviewee stated that data should belong to the entity which generates it. Another interviewee explained data ownership with the term "my data". She described that the concept thrives to answer questions such as: what data I have under control?, what data I am sharing? and, to who I am sharing my data to? We identified the following parties who generate and claim authority over data: farmers, contractors, tractor manufacturers and farming equipment manufacturers. In practice, the data authority is a very complex issue. For example, John Deere offers additional services to farmers in exchange of farming data. In addition, John Deere claims ownership of data produced by farming equipment connected to the tractor. As a result, equipment manufacturers prefer only to disclose the minimum necessary data possible. Thus, companies show clear interest towards data ownership but are not willing to disclose it. Farmers and contractors, on the other hand, show less interest towards the data. One interviewee explained that this could be due to the farmers' fear of the data being used against them. As an example, he brought up having personally experienced this kind of practice. Further, another interviewee described that contractors are not willing to share their work data because they consider it as their business secret.

In addition to problems related to data ownership, the interviewees also brought up issues related to lack of agricultural equipment and systems interoperability as well as lack of interest towards software ecosystem orchestration. Growing conditions varies across the globe. Due to poor machinery and equipment customisability, farmers need to invest in unoptimized machinery, which, in turn lead to unnecessary costs. In addition, many farms have machinery from different generations, which are not compatible with other equipment and information systems. One interviewee summarised the lack of agricultural software ecosystems with a question "where is the business?". He explained that farmers already operate on very small profit margins and are not willing to invest any further without any clear benefits.

Chapter 5

Discussion

5.1 Answer to RQ1: Actor Roles

The first research question is:

RQ1: What are the typical actor roles in a software ecosystem?

We were able to synthesise five actor roles from the literature analysis: orchestrator, niche player, external developer, end user and support service provider. Orchestrators are responsible for the software platform and the rules of the ecosystem. Niche players complement the work of the orchestrator typically via creating complementary services on top of the software platform of the ecosystem. External developers, on the other hand, compared to niche players, provide more limited contributions to the ecosystem. For example, they identify bugs or promote the software platform to other developers. End users obtain products or services of the ecosystem, whereas support service providers help other ecosystem actors in their value creation tasks. In research, actor roles of ecosystems are a commonly covered topic. However, there still does not exist an established term for describing a SECO participant. In addition, there is no stabilised terminology for the actor roles. For example, we identified six different synonyms for the orchestrator actor role as well as four synonyms for the niche player role.

In this study, we identified and analysed three agricultural SECOs. The findings of the study were not as consistent as in the existing research literature. However, the findings were in line with the results of previous studies. In the most mature SECO, the ISOBUS ecosystem, we were able to identify four out of five actor roles. Only the external developer role was not mentioned. In Cropinfra SECO we identified two out of five roles: the end user and the niche player. The actor for the orchestrator role was not identified, but the need for such role was acknowledged. From REKO we only identified actors for the orchestrator and the end user roles. Surprisingly, the niche player role was not mentioned. Niche players were described to be responsible for creating the ecosystem services. Thus, we could argue that REKO does not classify as a SECO.

The following roles appear to be vital in all SECOs: the orchestrator, the niche player and the end user. In short, the orchestrator is in charge of the software platform. Niche players utilise the platform in order to provide value to the end users of the ecosystem. Hence, the three above mentioned actors are required in the value creation chain. The external developer and support service provider roles, on the other hand, are not necessarily required in every SECO. Furthermore, it is important to remember that one actor can implement multiple roles. For example, an orchestrator can also act as a niche player in a case where the same company maintains the software platform and builds services on top of it.

5.2 Answer to RQ2: Actor Motivations

The second research question is

RQ2: Why do actors contribute to a software ecosystem?

The motivations of actor roles found in this study support the findings of previous research literature. In previous studies, the ecosystem benefits have been divided into direct monetary revenues and non-monetary benefits, such as fame, knowledge and ideology [37]. In SECOs, orchestrators, niche players and support service providers are often companies that are motivated by business opportunities. External developers and end users, on the other hand, can be either companies or individuals. Non-monetary benefits were described to be the motivator for the SECO participation of external developers.

The business model for Cropinfra SECO is based on a revenue share model between the orchestrator and the niche players, which supports the results found in previous studies [37]. Also, we can argue that in ISOBUS SECO, both the orchestrator and the niche players are motivated by monetary benefits. At first, it would seem that the ecosystem orchestrator is not motivated by money because it is defined as a non-profit organisation. However, the orchestrating organisation consists of niche player companies of the same ecosystem. Hence, the niche players can push their business interests indirectly. What is surprising is that end users were the most covered actor role in the empirical study compared to previous research, where it was the least covered. Also, the found end user motivations were described in more detail compared to motivations of other actor roles. The motivations were not only possible benefits but solid and concrete value propositions. For example, the Cropinfra promises more efficient work for the farmers, whereas REKO facilitates marketplace for niche agricultural products.

These observations may support the hypothesis that actor role motivations are inherited from actor type. With the actor type we mean for example, company or individual person. For example, companies often seek financial benefits whereas individuals, associations and communities seek for non-monetary benefits. In this study, we identified farmers as the end users both in Cropinfra and ISOBUS SECO. Even though farmers may sound like individuals, they should be considered as entrepreneurs.

5.3 Answer to RQ3: Role of Software Platform

The third research question is

What is the role of software platform in a software ecosystem?

Despite the central role of software platforms, it is rarely covered in prior SECO research. However, software platforms are a common topic in SPL research, from which the idea of software platform was derived. With the assistance of platform definitions from SPL research, we were able to identify three platform characteristics: (1) separation of the software platform and the derived products; (2) core assets under architecture; and (3) interfaces [9–11, 13, 18, 40, 41, 44, 51]. Especially the first characteristic describes the role of a software platform as a foundation for all other products and services of the ecosystem. The found characteristics were in line with the results found in the empirical study. Both in Cropinfra and ISOBUS ecosystems the applications were separated from the software platform. Also, we were able to identify a structure in the software platforms of Cropinfra and ISOBUS ecosystems. ISOBUS standard was described to include 14 parts and Cropinfra software platform had clear architecture. The software platform interfaces, on the other hand, were not discovered in any of the analysed agricultural ecosystems.

Our findings support the results of the study conducted by Manikas and Hansen [37]. They described the role of the software platform to allow ecosystem involvement and contributions. In short, the role of the software platform is to enable cooperation between ecosystem actors. We identified a high-level value proposition, which crystallised the role of the software platform in each SECO. In ISOBUS the platform aims to improve interconnectability between agricultural machinery. The software platform in Cropinfra, on the other hand, tries to enable farmers to choose the most suitable bundle of services. Whereas, REKO software platform makes communication possible between the food producers and consumers.

Another interesting finding was that software platform defines the ecosystem type. Previous SECO studies mainly focus on cornerstone ecosystems, in which the platform is software-based. However, in this study, we identified three different ecosystem types. Cropinfra was identified as cornerstone ecosystem, whereas ISOBUS ecosystem was identified as a standard based ecosystem. The REKO SECO could be could be classified as an infrastructure-based ecosystem.

5.4 Limitations

This study includes a set of limitations. We interviewed only four subjects which can be considered as a small number. Additionally, the backgrounds of the interviewees varied which affected our ability to discuss certain topics. For example, only one interviewee had his background in software. Due to this, we could not elaborate aspects of software platforms with the other interviewees as profoundly. Also, the information we gained about the ecosystem actors in the interviews could be considered second-hand information. The interviewees spoke on behalf of other peoples' motivations to their best knowledge, which obligated us to rely on their perceptions.

Finally, the lack of maturity of the agricultural SECOs limited the credibility of the results. The ISOBUS ecosystem was identified as the only mature SECO. The Cropinfra SECO was described merely as a concept. Thus, the results are based on theories and assumptions instead of realised facts. In addition, the REKO SECO was not related to the projects of LUKE, but its information was based on personal findings of one interviewee.

Chapter 6

Conclusions

The objective of this thesis was to find out what is important on a platform of a software ecosystem to satisfy actors' expectations. The literature study provided main concepts such as software ecosystem, actors and software platform. The results of the literature review were compared with findings from SECOs in the agricultural domain. In addition to addressing the objective of the study, we found common problems in the SECOs.

The results of the study indicate that the software ecosystem must provide a unique value proposition to all different actor roles. It is essential for SECO success to be able to attract and retain actors. In order to achieve this, the value proposition of the ecosystem must satisfy the needs of all actors. Due to the diversity of the actors and their needs, a single value proposition for the ecosystem is not sufficient. Each actor role, on the other hand, seems to share its motivation. Hence, a specific value proposition for each actor role is probably needed.

Data ownership concerns can prevent standard based SECOs from succeeding. Standard-based ecosystems are orchestrated by consortia, which is formed by parties interested in the topic. Ecosystems orchestrated by consortia strive for neutrality. However, for example in a case concerning ISOBUS SECO, one tractor manufacturer claimed ownership of data produced by another manufacturer. This can lead to a situation where manufacturers disclose as little data as possible to the bus, which limits the potential of the whole ecosystem. This can force companies to bypass ecosystem and use unstandardized raw data from the CAN bus.

The orchestrator is an essential actor role in a SECO. The orchestrator enables value creation for other actors of the ecosystem through the software platform. Software platform management is one of the responsibilities of the orchestrator. Software platform management includes tasks such as platform maintenance, development and promotion. Furthermore, the orchestrator is also responsible for creating and maintaining actor participation rules. The rules can, for example, include a criterion for actors participation in an ecosystem. Because the above-mentioned responsibilities cover two out of three SECO main elements, we can state that ecosystem orchestrator has a central role in the ecosystem success.

The software platform lacks an established definition in SECO context. This is due to the evolution of SECO as a research field. The concept of software ecosystem has emerged both from business ecosystem and software product line research. The current SECO software platform definitions lean heavily towards definitions from software product line research. The software platform characteristics from software product lines can be transferred into SECO context, but the definition could be updated. Thus, we propose a new definition for a software platform in SECO context: *"a set of assets organized in a common structure from which actors can efficiently create complementary services"*.

For the future research, we propose two distinct research directions. First, we propose to deepen the understanding of data ownership issues in the ISOBUS ecosystem. According to the findings of this study, the data ownership seems to be a significant problem for the future of the whole ecosystem. Also, it appears to apply in all standard-based ecosystems. Secondly, we propose a systematic literature review of ecosystem case studies, which focus on ecosystem actors and their motives for participation. This would help to discover if some actor motivations could be generalised and if so, which motives are the most common. Further, this would help coining value propositions for future SECOs.

Bibliography References

[1] Aef / about isobus / what is isobus? http://www.aef-online.org/en/about-isobus/what-is-isobus.html. (Accessed on 09/21/2017).

[2] Ecology — definition of ecology by merriam-webster. https://www.merriam-webster.com/dictionary/ecology. (Accessed on 06/05/2017).

[3] ecosystem - definition of ecosystem in english — oxford dictionaries. https://en.oxforddictionaries.com/definition/ecosystem. (Accessed on 05/26/2017).

[4] Icsob 2016. http://www2.it.lut.fi/icsob2016/. (Accessed on 08/29/2016).

[5] Splc.net l software product line conferences. http://splc.net/. (Accessed on 08/29/2016).

[6] Strategy - luonnonvarakeskus. https://www.luke.fi/en/luke-3/strategy/. (Accessed on 07/06/2017).

[7] Backman, J. Kasvintuotantojärjestelmien digitalisaation tiekartta. *Luonnonvara- ja biotalouden tutkimus*, 17 (2015).

[8] Barbosa, O., and Alves, C. A systematic mapping study on software ecosystems. In *Proceedings of the Third International Workshop on Software Ecosystems, Brussels, Belgium, June 7th, 2011* (2011), S. Jansen, J. Bosch, P. R. J. Campbell, and F. Ahmed, Eds., vol. 746 of *CEUR Workshop Proceedings*, CEUR-WS.org, pp. 15–26.

[9] Bosch, J. *Design and use of software architectures - adopting and evolving a product-line approach.* Addison-Wesley, 2000.

[10] Bosch, J. Maturity and evolution in software product lines: Approaches, artefacts and organization. In *Software Product Lines, Second International Conference, SPLC 2, San Diego, CA, USA, August 19-22,*

2002, Proceedings (2002), G. J. Chastek, Ed., vol. 2379 of *Lecture Notes in Computer Science*, Springer, pp. 257–271.

[11] Bosch, J. From software product lines to software ecosystems. In *Software Product Lines, 13th International Conference, SPLC 2009, San Francisco, California, USA, August 24-28, 2009, Proceedings* (2009), D. Muthig and J. D. McGregor, Eds., vol. 446 of *ACM International Conference Proceeding Series*, ACM, pp. 111–119.

[12] Christensen, H. B., Hansen, K. M., Kyng, M., and Manikas, K. Analysis and design of software ecosystem architectures - towards the 4s telemedicine ecosystem. *Information & Software Technology 56*, 11 (2014), 1476–1492.

[13] Clements, P., and Northrop, L. *Software product lines.* Addison-Wesley,, 2002.

[14] Crooymans, W., Pradhan, P., and Jansen, S. Exploring network modelling and strategy in the dutch software business ecosystem. In Fernandes et al. [16], pp. 45–59.

[15] Cusumano, M. A., and Gawer, A. The elements of platform leadership. *MIT Sloan Management Review 43*, 3 (2002), 51.

[16] Fernandes, J. M., Machado, R. J., and Wnuk, K., Eds. *Software Business - 6th International Conference, ICSOB 2015, Braga, Portugal, June 10-12, 2015, Proceedings* (2015), vol. 210 of *Lecture Notes in Business Information Processing*, Springer.

[17] Fotrousi, F., Fricker, S. A., Fiedler, M., and Gall, F. L. Kpis for software ecosystems: A systematic mapping study. In *Software Business. Towards Continuous Value Delivery - 5th International Conference, ICSOB 2014, Paphos, Cyprus, June 16-18, 2014. Proceedings* (2014), C. Lassenius and K. Smolander, Eds., vol. 182 of *Lecture Notes in Business Information Processing*, Springer, pp. 194–211.

[18] Gawer, A., and Cusumano, M. A. Industry platforms and ecosystem innovation. *Journal of Product Innovation Management 31*, 3 (2014), 417–433.

[19] Handoyo, E., Jansen, S., and Brinkkemper, S. Software ecosystem roles classification. In *Software Business. From Physical Products to Software Services and Solutions - 4th International Conference, ICSOB 2013, Potsdam, Germany, June 11-14, 2013. Proceedings* (2013),

G. Herzwurm and T. Margaria, Eds., vol. 150 of *Lecture Notes in Business Information Processing*, Springer, pp. 212–216.

[20] Hanssen, G. K. A longitudinal case study of an emerging software ecosystem: Implications for practice and theory. *Journal of Systems and Software 85*, 7 (2012), 1455–1466.

[21] Hyrynsalmi, S., Seppänen, M., Nokkala, T., Suominen, A., and Järvi, A. Wealthy, healthy and/or happy - what does 'ecosystem health' stand for? In Fernandes et al. [16], pp. 272–287.

[22] Iansiti, M., and Levien, R. *The Keystone Advantage: What the New Dynamics of Business Ecosystems Mean for Strategy, Innovation, and Sustainability.* Harvard Business School Press, 2004.

[23] Iansiti, M., and Levien, R. Keystones and dominators: Framing operating and technology strategy in a business ecosystem. *Harvard Business School, Boston* (2004), 24–25.

[24] Iansiti, M., and Levien, R. Strategy as ecology. *Harvard business review 82*, 3 (2004), 68–81.

[25] Iyer, B., Lee, C.-H., and Venkatraman, N. Managing in a" small world ecosystem": Lessons from the software sector. *California Management Review 48*, 3 (2006), 28–47.

[26] Jansen, S., Brinkkemper, S., and Finkelstein, A. Business network management as a survival strategy: A tale of two software ecosystems. In *Proceedings of the First International Workshop on Software Ecosystems (IWSECO-2009), co-located with the 11th International Conference on Software Reuse , Falls Church, Virginia, USA, September 27, 2009.* (2009), S. Jansen, S. Brinkkemper, A. Finkelstein, and J. Bosch, Eds., vol. 505 of *CEUR Workshop Proceedings*, CEUR-WS.org.

[27] Jansen, S., and Cusumano, M. A. Defining software ecosystems: A survey of software platforms and business network governance. In *Proceedings of the Forth International Workshop on Software Ecosystems, Cambridge, MA, USA, June 18th, 2012* (2012), S. Jansen, J. Bosch, and C. F. Alves, Eds., vol. 879 of *CEUR Workshop Proceedings*, CEUR-WS.org, pp. 40–58.

[28] Jud, C. Business models for platform-based digital services: Stakeholder expectations. In Fernandes et al. [16], pp. 174–178.

[29] Knodel, J., and Manikas, K. Towards a typification of software ecosystems. In Fernandes et al. [16], pp. 60–65.

[30] Krueger, R. A., and Casey, M. A. *Focus groups: A practical guide for applied research.* Sage publications, 2014.

[31] Lazar, J., Feng, J. H., and Hochheiser, H. *Research methods in human-computer interaction.* John Wiley & Sons, 2010.

[32] Lindman, J., Kinnari, T., and Rossi, M. Business roles in the emerging open-data ecosystem. *IEEE Software 33*, 5 (2016), 54–59.

[33] LLC, M. A framework for software product line practice, version 5.0, 2017.

[34] Lungu, M. F. *Reverse engineering software ecosystems.* PhD thesis, University of Lugano, 2009.

[35] Manikas, K. Revisiting software ecosystems research: A longitudinal literature study. *Journal of Systems and Software 117* (2016), 84–103.

[36] Manikas, K., and Hansen, K. M. Characterizing the danish telemedicine ecosystem: making sense of actor relationships. In *Fifth International Conference on Management of Emergent Digital EcoSystems, MEDES '13, Luxembourg, Luxembourg, October 29-31, 2013* (2013), L. Ladid, A. Montes, P. A. Bruck, F. Ferri, and R. Chbeir, Eds., ACM, pp. 211–218.

[37] Manikas, K., and Hansen, K. M. Software ecosystems - A systematic literature review. *Journal of Systems and Software 86*, 5 (2013), 1294–1306.

[38] McLellan, S. G., Roesler, A. W., Tempest, J. T., and Spinuzzi, C. Building more usable apis. *IEEE Software 15*, 3 (1998), 78–86.

[39] Messerschmitt, D. G., and Szyperski, C. Software ecosystem: understanding an indispensable technology and industry. *MIT Press Books 1* (2003).

[40] Meyer, M. H., and Lehnerd, A. P. *The power of product platforms.* Simon and Schuster, 1997.

[41] Meyer, M. H., and Seliger, R. Product platforms in software development. *MIT Sloan Management Review 40*, 1 (1998), 61.

[42] Myers, M. D. *Qualitative research in business and management.* Sage, 2013.

[43] Pesonen, L. A., Teye, F. K.-W., Ronkainen, A. K., Koistinen, M. O., Kaivosoja, J. J., Suomi, P. F., and Linkolehto, R. O. Cropinfra–an internet-based service infrastructure to support crop production in future farms. *Biosystems Engineering 120* (2014), 92–101.

[44] Pohl, K., Böckle, G., and van der Linden, F. *Software Product Line Engineering - Foundations, Principles, and Techniques.* Springer, 2005.

[45] Ryan, G. W., and Bernard, H. R. *Handbook of Qualitative Research*, 2 ed. Sage Publications, Thousand Oaks, CA, 2000, ch. Data management and analysis methods, pp. 769–802.

[46] Sevon, P. Commitment within software ecosystem. Master's thesis, Aalto University, 1 2015.

[47] Soussi, L., Spijkerman, Z., and Jansen, S. A case study of the health of an augmented reality software ecosystem: Vuforia. In *Software Business - 7th International Conference, ICSOB 2016, Ljubljana, Slovenia, June 13-14, 2016, Proceedings* (2016), A. Maglyas and A. Lamprecht, Eds., vol. 240 of *Lecture Notes in Business Information Processing*, Springer, pp. 145–152.

[48] Tapscott, D., Ticoll, D., and Lowy, A. Digital capital: harnessing the power of business webs. *Ubiquity 2000*, May (2000), 9.

[49] Toivanen, T., Mazhelis, O., and Luoma, E. Network analysis of platform ecosystems: The case of internet of things ecosystem. In Fernandes et al. [16], pp. 30–44.

[50] van den Berk, I., Jansen, S., and Luinenburg, L. Software ecosystems: a software ecosystem strategy assessment model. In *Software Architecture, 4th European Conference, ECSA 2010, Copenhagen, Denmark, August 23-26, 2010. Companion Volume* (2010), I. Gorton, C. E. Cuesta, and M. A. Babar, Eds., ACM International Conference Proceeding Series, ACM, pp. 127–134.

[51] van Ommering, R. C. Beyond product families: Building a product population? In *Software Architectures for Product Families, International Workshop IW-SAPF-3, Las Palmas de Gran Canaria, Spain,*

March 15-17, 2000, Proceedings (2000), F. van der Linden, Ed., vol. 1951 of *Lecture Notes in Computer Science*, Springer, pp. 187–198.

Appendix A

Orchestrator interview

Johdanto
Haastattelujen pohjalta pyrimme luomaan kattavan kokonaiskuvan ekosysteemistä. Kokonaiskuvan lisäksi pyrimme löytämään ekosysteemin toimijat, sekä kartoittamaan ohjelmiston roolia ekosysteemissä. Haastattelu tullaan nauhoittamaan keskustelun ja tulosten analysoinnin helpottamiseksi. Nauhoituksen lisäksi käsittelemme vastaukset ja muut materiaalit luottamuksellisesti. Kaikki tutkimustulokset esitetään työssä anonyymisti, eikä tutkielman lukija pysty henkilöimään haastattelun vastauksia.

Haastateltavan tausta

1. Voitko kertoa hieman omasta taustastasi (koulutus, työt yms.)?
2. Mikä on sinun tämänhetkinen työnkuva luonnonvarakeskuksella?

Ekosysteemin tausta

1. Millaista ekosysteemiä olette rakentamassa?
2. Voisitko kuvailla millaisista osista ekosysteemi koostuu?
3. Milloin ekosysteemiä aloitettiin kehittämään?
4. Mikä on ekosysteemin tavoite?
5. Millaisia vaiheita ekosysteemin rakentamiseen on kuulunut tähän asti - tulevaisuudessa?
6. Millaisia haasteita ekosysteemin rakentamiseen on liittynyt?

7. Kuinka kauan olet työskennellyt ekosysteemin parissa?
8. Mikä on sinun roolisi ekosysteemiin liittyen?

RQ1 ja RQ2 - Toimijoiden osallisuminen ja motivaatiotekijät

1. Mikä on LUKEn rooli ekosysteemissä?
2. Ketkä ovat ekosysteemin loppukäyttäjiä?
3. Millainen on loppukäyttäjien rooli ekosysteemissä?
4. Millaisia tehtäviä kuuluu loppukäyttäjien rooliin?
5. Millainen on loppukäyttäjien rooli ekosysteemin rakentamisessa?
6. Mihin ekosysteemin toimijoihin loppukäyttäjät ovat suorasti yhteyksissä ?
7. Mihin ekosysteemin toimijoihin loppukäyttäjät ovat epäsuorasti yhteyksissä ?
8. Mitä muita toimijoita ekosysteemissä toimii tällä hetkellä?
9. Millaisissa rooleissa toimijat toimivat ekosysteemissä?
10. Mitä hyötyjä eri toimijat saavat ekosysteemistä?
11. Keitä toimijoita toivoisitte osallistuvan ekosysteemiin tulevaisuudessa?

RQ3 - Ohjelmistoalusta

1. Millaista teknologiaa kuuluu ekosysteemiin?
2. Voitko kuvailla ekosysteemin teknistä alustaa?
3. Mikä on teknisen alustan tehtävä ekosysteemissä?
4. Mistä tekninen alusta koostuu?
5. Käsittääkö alusta pelkän ISOBUS -standardin, vai kuuluuko muita systeemejä (maatilan tiedonhallintajärjestelmät ja muut palvelut)
6. Ovatko maatilan tiedonhallintajärjestelmät (FMIS) avoimia?
7. Mitä rooleja ISOBUS väylän eri komponenteilla on?

8. Kenen vastuulla on ISOBUS standardin kehitys ja hallinta?
9. Kenen vastuulla on palveluiden kehitys?
10. Onko teknisen alustan hallinnassa ja kehityksessä ilmennyt haasteita?
11. Mitä askeleita liittyy alustan tulevaisuuteen?

Loppu

1. Olisiko sinulla muuta materiaalia ekosysteemiin liittyen, jota voisimme analysoida tähän tutkimukseen liittyen