



# A SECO Meta-model

## A Common Vocabulary of the SECO Research Domain

J. Wouters<sup>1</sup>, J. R. Ritmeester<sup>1</sup>, A. W. Carlsen<sup>1</sup>, Slinger Jansen<sup>1</sup>,  
and Krzysztof Wnuk<sup>2</sup>(✉)

<sup>1</sup> Department of Information and Computing Sciences,  
Utrecht University, 3584 CC Utrecht, The Netherlands  
j.wouters1@uu.nl

<sup>2</sup> Department of Software Engineering,  
Blekinge Institute of Technology, Karlskrona, Sweden  
krw@bth.se

**Abstract.** Software development companies are venturing towards collaborative approach and software ecosystems (SECO) participation. Over the years, many papers have been written and different modelling languages were proposed to capture the interactions between the SECO participants. What is missing, however, is a comprehensive meta-model describing possible entities and relationships that constitute a SECO. The goal of this paper is to create a common language for academic researchers for software ecosystems by creating such a meta-model. We constructed the meta-model by extracting and grouping entities and relationships from research papers. The meta-model consists of 5 themes: actors and roles, products and platforms, boundaries, ecosystem health and strategy. We advocate that our meta-model allows for easy sharing and comparing of case studies and the generalization of results across studies. We present the results from initial expert evaluation of the meta-model.

**Keywords:** Software ecosystems · Meta-model · Grounded theory · Literature review · Research synthesis

## 1 Introduction

Software development companies are venturing towards collaborative approach and software ecosystems participation [15]. The SECO literature is quite rich and offers case studies, experiences and models. A recent systematic literature review on the subject by Manikas [17], summarized 90 papers. They noted the following: “(a) *there is little consensus on what constitutes a software ecosystem and (b) few analytical models of software ecosystems exist*”. A SECO meta-model could be a useful tool for creating consensus and for creating consistent analytical models. Similarly, one of the main challenges in the SECO research agenda was the characterization and modelling of the emerging SECOs [16]. Since 2009, various SECO models [4, 24, 25] were created, but the domain lacks a meta-model which captures the SECO landscape as a whole.

Accurate context descriptions support generalization and research synthesis across various cases and studies. Briand et al. [5] argue that context-driven research is needed in general since it makes clear working assumptions, and helps to achieve practicality and scalability. Dybået al. [8] propose to use a set of questions (what, who, where, when, why) to characterize the contexts of empirical research results. Petersen and Wohlin [20] provided a context checklist for industrial software engineering research. Ghaisas et al. [9] provide a proposal for reasoning on how to generalize findings across cases. To the best of our knowledge, no study has yet made an attempt to create a unified context description schema for SECO research. Throughout academic literature different terminology is used, which makes it difficult to link different scientific contributions within the domain together.

The goal for this research is *to develop a common language for the Software Ecosystem domain*. This is done by combining models and theories developed earlier by other research partners, utilizing the design science methodology suggested by Wieringa [27]. The main research question is formulated as follows:

*What main elements a meta-model for SECO domain should contain?*

The rest of this paper is structured as follows. In section two, we present the results from the literature study on SECOs and meta-modelling. In section three, the method used to develop the meta-model is described. In section four, the developed meta-model is presented. In the last section, research opportunities are identified and conclusions are drawn.

## 2 Related Work in Classification and Meta-models

Bosch developed a software ecosystem taxonomy, where SECOs would be placed on a three-by-three grid based on the category (end-user programming, application and operating system) and platform (desktop, web or mobile) [3]. This taxonomy, however, is not able to show all the aspects of the SECO, it merely positions them based on the output of the SECO. Jansen et al. also developed a classification model, where ecosystems would be classified based on four factors: Base technology, coordinators, extension markets and accessibility [15]. Both these models merely classify a SECO based on some aspects of it.

As the focus of this research is the creation of a SECO meta-model, defined as: *A meta-model provides a unifying framework in which to ensure and check consistency, while at the same time providing the means to distinguish between valid and invalid models, that is, conformance* [19].

The aim is to create a unifying framework by combining entities from existing models. A meta-model can aid in the creation of consistent and unambiguous models, making it easier to reason about SECOs [4].

## 3 Research Method

We used a design cycle methodology to develop the SECO meta-model [27]. The treatment under design is the SECO meta-model, designed in the four steps

outlines in the following text. The following requirements are taken into account for the meta-model:

1. Requirement 1: The meta-model should be applicable to every SECO.
2. Requirement 2: The entities included into the meta-model have to be derived from scientific sources.
3. Requirement 3: The meta-model should be easy to use and understand.
4. Requirement 4: The meta-model should provide an extensive list of universally used terms to make it easier for researchers to discuss SECOs.

### 3.1 Step 1: Systematic Literature Review

We utilized the recent SLR of Mankikas [17] that contains 90 academic papers relevant for software ecosystems. We also applied the snowballing methodology [28] to identify other relevant papers. The selection criteria include:

1. The paper has to have SECOs as its main object of study.
2. The paper has to present some form of modelling technique, classification technique or characterization about SECOs.
3. The model or theory presented has to be somehow verified using academic techniques, e.g. proof of concept or case study.

Each paper was reviewed by one researcher of our research team. When in doubt, a second researcher was consulted until a consensus was found. All considered papers and inclusion decisions are available from the researchers upon request.

### 3.2 Step 2: Meta-model Relevant Entity Selection

To extract entities from the selected papers, we utilized open coding procedure in the grounded theory method [6]. The text that described or contained SECO entities was identified and coded. We used the following guidelines for coding:

1. The coding will be done by each researcher individually.
2. All codes need to be related to software ecosystems.
3. For every identified entity, the entire sentence in which this entity is found will be recorded.
4. For every code, a definition will be written, describing what it is and why it was recorded.

The coding process happened in weekly iterations, concluded with a synchronization meeting. In a final meeting, the separate coding files were combined and categorized. To eliminate unnecessary or incorrect codes, we applied the following selection criteria:

1. The entity should be unique (so no synonym), unambiguous and clear, and should not be included in another definition or definitions.
2. The entity should be applicable to all Software Ecosystems.
3. The entity is defined and described in academic literature.
4. The entity is without a doubt important to describe SECOs.

### 3.3 Step 3: Model Development

The identified entities were clustered according to the similarities and overlaps between them. We also modelled the relationships between the entities. We used the Class Diagram Language as specified by the Object Management Group [18] to depict the SECO meta-model. We described the following relationships:

- Aggregation: entity A consists of entity B. For example; a SECO contains actors.
- Generalization: entity B is a logical generalization of entity a. For example; A bridge is an example of a role.
- Navigated association: entity A is somehow related to entity B. For example; An ecosystem type is based on an extension market.

### 3.4 Step 4: Expert Review

We reviewed the developed meta-model with an expert in the field of software ecosystems and software business, using a semi-structured interview, in which we discussed the following:

- The completeness of the entities (Req 4).
- The categorization of the entities and the determined themes (Reqs 1, 4).
- The relationships identified and their significance (Req 4).
- The applicability of the meta-model to different kinds of SECOs (Req 1).
- The understandability of the meta-model (Req 3).

The expert review suggestions helped to further improve the meta-model.

## 4 Results and Analysis

### 4.1 Step 1: Paper Selection

We reviewed 181 papers (90 papers from the SLR by Manikas et al. [17] and 91 additional papers from the snowballing on the SLR). Based on the exclusion criteria, we included 36 papers and after the full read we left 33 papers<sup>1</sup>.

### 4.2 Step 2: Entity Selection and Theme Construction

Open coding resulted in the identification of 218 entities, each checked on the four criteria defined in Sect. 3.2. After applying the criteria, 114 entities remained. Table 1 shows the number of entities which passed or failed the different criteria<sup>2</sup>. The, entities could have been excluded based on multiple selection criteria.

<sup>1</sup> The final list of selected papers is available online at <https://drive.google.com/open?id=1ZzUHA7H22jPm7IfcHSv1AxSRd2hTQ5ye>.

<sup>2</sup> The full list of codes and sources for building entities is available at online at <https://drive.google.com/open?id=1ZzUHA7H22jPm7IfcHSv1AxSRd2hTQ5ye>.

We have clustered the entities into five themes, described below. The full list of entities is available in Table 2.

**Table 1.** Included and excluded entities per criterion.

Criterion	Passed	Failed
Initial coded	218	–
Criterion 1: Unambiguous, unique, clear and no synonym	142	76
Criterion 2: Applicable to (almost) all SECOs	201	17
Criterion 3: Defined and described in academic literature	197	21
Criterion 4: Without a doubt important to describe SECOs	169	49
<u>Incorporated into final meta-model</u>	<u>114</u>	<u>105</u>

**Theme 1: Actors and roles.** We identified 46 actor entities, outlined in Table 2. Each actor can have one or more roles in a SECO. This theme consists of all the actors and the roles they can take in a SECO. The term actor is not identified in the SLR [17], but can be seen as an overlapping term for all legal entities in a SECO who are taking part in the SECO in some form. The actors were split into two categories: Individuals and organizations. Individuals can be for example customers who are buying something from the SECO [15, 25]. Also, hobbyists can be seen as individuals providing extensions to a SECO [7]. Another important individual actor is researcher, as identified by Alves et al. [1]. Organizations can also be part of a SECO. For example, independent software vendors can produce extensions for a base product (discussed below), enabling the keystone (discussed below) to focus on the core products [10]. Also, developer communities are active at SECOs, in certain ecosystems, all users are potential developers and form their own community (for example, in the Android ecosystem) [12].

The role of an actor can be derived from the relationship it has with other actors. A lot of these roles are derived from the work of Jansen et al. [14], e.g. niche player, keystone and dominator, disciple and hedger. The creation of the hub as an overlapping role has been discussed by Dos Santos et al. [23]. Several papers discussed roles which were linked to the roles a keystone fulfils in the ecosystem. For example, a keystone can be seen as an orchestrator, as discussed by Jansen et al. [14], van den Berk et al. [25]. Viljanen et al. also point out that “orchestration is mainly a keystone players’ task” [26], which justifies the placement of the orchestrator entity under keystone. Also, a keystone most of the time is a platform leader, as a platform leader is “controlling large parts of an ecosystem”, thereby making a platform leader a keystone [15]. As a keystone can fulfill one or multiple of these ‘sub-roles’, these sub-roles will be modelled as an aggregation. Actor categorization and description helps to map the ecosystem players and their influence on SECO.

**Theme 2: Products and platforms.** We identified 27 entities in this theme. Each SECO can have one or multiple products and platforms. A product is

**Table 2.** All entities categorized into five themes.

<b>Theme 1: Actors and roles</b>	<b>Theme 2: Products and platforms</b>	<b>Theme 3: Strategy</b>
Actor (17, 19)	API (2, 9, 12, 16, 34)	Acquisition strategy (16)
Advertiser (34)	Architecture (2, 3, 7, 8)	Affiliate model (30)
Banks and investors (23, 31)	Base technology (16)	Business strategy (3)
Bridge (14, 16)	Community driven (16)	Cost of development (3)
Broker (14)	Documentation (1, 2, 28)	Ecosystem openness (2, 15, 31, 36)
Buyer (34)	Executable components (2)	Ecosystem strategy (16)
Community leader (1, 28)	Extension market (16)	Entry barrier (14, 16, 30, 31)
Competitor (4, 16, 31)	Hardware product (4)	Licensing (2, 4, 8, 36)
Complementor (34)	Intellectual property (15)	Membership model (30)
Consultancy firms (31)	Open-source component (12)	Orchestration techniques (2, 4, 14, 16, 31, 34)
Customer (4, 10, 15, 16, 17)	Operating system (2)	Partnership model (30)
Decision maker (23)	Ownership model (Self-developed)	Platform strategy (16, 31, 34)
Developer communities (32)	Platform (2, 4, 6, 7, 8, 14, 16, 17, 19, 31, 34)	Product lifecycle strategy (16)
Disciple (8, 14)	Platform planning (31)	Risk management (34)
Domain expert (31)	Privately owned (16)	Revenue model (24)
Dominator (8, 6, 14, 16, 31)	Product (Self-developed)	Strategic planning (4)
Governments (1, 23, 31)	Product quality (1)	Technology asset management (34)
Hardware vendor (32)	Programming language (7)	Technology scouting (34)
Hedger (8, 14)	Requirement (11, 27, 34)	Time to market (3)
Hobbyist (9)	SDK (12)	Tipping strategy (34)
Hub (8, 16, 31)	Software artifact (2, 7, 11, 23, 27)	Transaction costs (13)
Ind software vendor (13, 15)	Software concept (16)	
Individual (Self-developed)	Software library (31)	
Influencer (8, 14)	Software product (4, 16, 17, 19, 29, 31)	
Keystone (8, 14, 15, 29, 30, 31, 34)	Software source code component (2)	
Legal entity (6, 12, 32)	Software standard (16)	
Legal firms (31)	Use case (19)	
Manufacturer (9)		
Niche player (8, 14, 16, 29, 31, 32, 34)		
Open source organization (15)		
Orchestrator (14, 30)		
Organization (Self-developed)		
Platform leader (16)		
Platform provider (32)		
Product manager (15)		
Quality assesor (19)		
Regulatory bodies (7, 31)		
Relationship (Self-developed)		
Researcher (1)	<b>Theme 4: Ecosystem health</b>	<b>Theme 5: Boundaries</b>
Research group (19)	Growth (2)	Abstraction level (17, 19)
Role (8, 14, 23, 30, 34)	Health factors (6, 32)	Ecosystem type (6, 12, 13, 19)
Seller (34)	Maturity (1, 28)	Environment (4)
Supply chain partner (15)	Niche creation (6, 8, 14, 16)	Inputs (4)
Technology provider (15)	Popularity (11)	Legal framework (5)
Third-party developer (9, 15)	Productivity (6, 8, 14, 16)	Market (14, 17, 19)
Value-added reseller (15, 17, 30)	Robustness (6, 8, 14, 16)	Output (4)

defined in this research as either a software standard, hardware product or software artefact that is part of the SECO. A software product is mentioned many times as a part of a SECO by Jansen et al. [15] and Boucheras et al. [4]. A software concept is also discussed by Jansen et al. [15], even naming it a way to

discuss software ecosystem types. Other software artefacts were also discussed in different SECO papers. A full list is included in the entity overview in Table 2.

The platform in an ecosystem is a central entity on which different actors can provide their products. Most of the time the platform itself is some form of product, which can be extended by apps or extensions produced by different actors [14, 15]. The link between products and platforms can be defined by the entity “base technology”, called by Jansen et al. [15] as “the technology underpinning the SECO”. In this research, it is argued that the base technology itself is always some form of product, linking the two concepts together. Some key elements about a platform are also identified from the literature, such as a platform planning [2] and an ownership model [15].

**Theme 3: Strategy.** We identified 21 entities in this theme. The third theme identified is Strategy. Every actor within the ecosystem has certain strategies regarding their products, position in the ecosystem and revenue generations. A lot of these strategies are described by different papers. For example, Dos Santos et al. discussed seven papers which describe strategy in some form [23], Jansen et al. discussed making strategy (platform strategy, acquisition strategy, product lifecycle strategy) explicit [15] and Popp described revenue models for the hybrid software industry [21]. All of these strategies are part of the strategy of an actor within the SECO, and therefore indirectly influence the SECO.

A special kind of strategy is the Ecosystem strategy, which directly influences the ecosystem. Jansen et al. identified developing SECO strategy as one of the main research challenges in the SECO domain [16]. Several papers discussed parts of SECO strategy. For example, the different orchestration techniques discussed by Jansen et al. [14] are part of the SECO strategy, as are the entry barriers set by the keystone(s) of a SECO. Van den Berk et al. discussed a model to measure the SECO strategy in their paper [25], which can be used to formalize the SECO strategy, as well as its underlying entities.

**Theme 4: Ecosystem health.** We identified 7 entities in this theme. Different papers discuss the health of a certain ecosystem. The health of a SECO is a main theme as it is the main way to measure the current status of an ecosystem. It, therefore, is the main operationalization of the success of an ecosystem thus far. The main factors of measuring health in an ecosystem are identified by Iansiti and Levien [11] to be Niche creation, robustness and productivity. These three factors are further operationalized by Jansen [13]. This research also found some entities which are directly influenced by ecosystem health, such as the growth, maturity and popularity of the SECO.

**Theme 5: Boundaries.** We identified 7 entities in this theme. The boundaries of a SECO are defined in this research are the defining properties which describe what is part of the SECO, and what is not. The paper of Jansen et al. [14] described some initial boundaries, such as the market the SECO operates in and the ecosystem type the ecosystem at hand has, which is determined based on four factors: base technology, keystone, extension market and accessibility. It also discusses abstraction levels (SECOs level, SECO level or Software Supply

Network level) of SECOs which can be used to further define a SECO. A SECO is further defined by the output it generates and the input it uses to do so [4].

### 4.3 The SECO Meta-model

The SECO meta-model has gone through four iterations before it reached its current final state. The final version of the meta-model consists of 114 entities and 120 defined relations between them. Figure 1 shows the relationships between a SECO and the main themes. The most important design decisions and relationships are described in the paragraphs that follow.

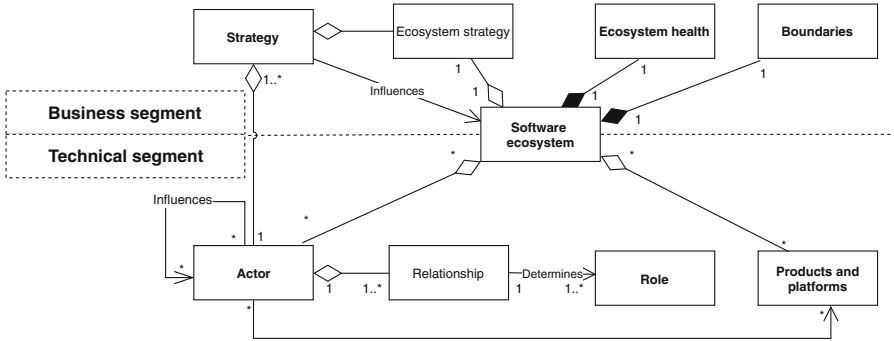


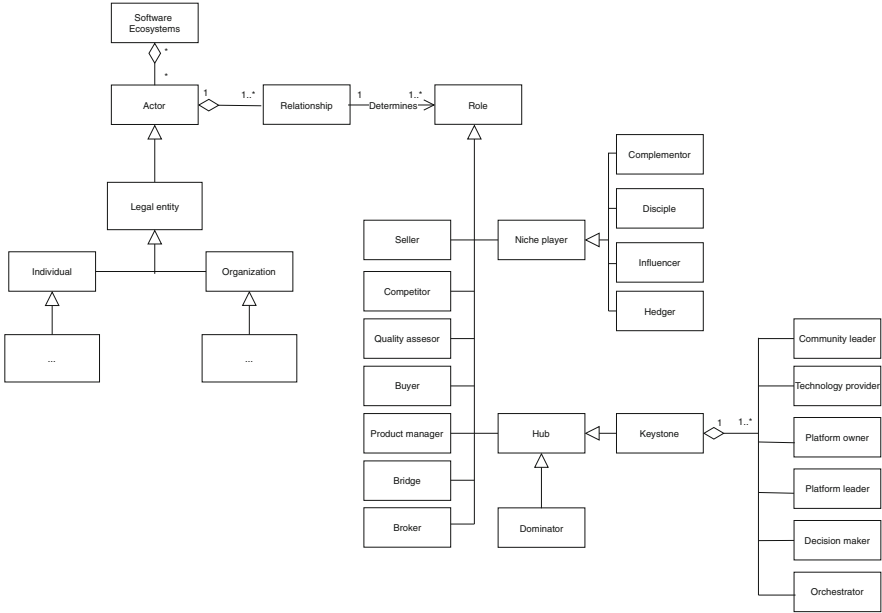
Fig. 1. Part of the model showing the relationships between SECO and main themes

**Two segments: Business (strategic) and technical (operational):** In the final iteration of the model, it became apparent that two ‘segments’ of the model can be identified. The first segment is a technical segment. Here, the model describes the actors, products, and platforms the SECO consists of. The other segment, the business segment, describes the SECO on a higher level: it describes the SECOs boundaries, strategy, and health. This will be useful when analyzing the business segment of the SECO in academic situations. Together, these two segments will give a complete overview of the SECO being modelled. If the main aim is to research a SECO in the most complete way, both segments have to be used. Both segments should be used to describe SECO research in the most complete way. Of course, this is not recommended as it glances over the business decisions made about the SECO.

**Split between actors and their roles:** As identified above, each SECO consists of a number of actors, which can have one or multiple roles. This split has also been made explicit in the meta-model. The ‘relationship’ entity connects the actor and the roles, as a relationship between actors determines the role a certain actor can have. Special roles are that of the ‘niche player’ and the ‘keystone’. A keystone fulfils certain sub-roles in the SECO, for example; that of platform owner, decision maker, and orchestrator, as discussed above. A keystone does



not have to fulfil all of these sub-roles, but when it has at least one of these sub-roles, the actor can be considered as a keystone. See Fig. 2 for a schematic overview of this split.



**Fig. 2.** The part of the model showing the relationships between actors and roles

**Strategy and ecosystem strategy:** In the business segment, we define a relationship is defined between Strategy and ecosystem strategy. As discussed, different strategies followed by different actors might influence the SECO. Therefore, this relationship is defined as “Strategy influences SECO”. To make the link between strategy and the actors having these strategies more explicit, an aggregation relationship from strategy to actor has been defined. The ecosystem strategy, which can be seen as a special kind of strategy, is directly related to the SECO, as each SECO has a SECO strategy determined by the keystone players. Therefore, an aggregation relationship is defined between SECO and SECO strategy. This relationship is also shown in Fig. 1

**Links between business and technical segment:** The model developed has three links between the business and technical segment. Firstly, the output of a SECO (which is part of the ‘boundaries’ theme) is linked to the products and platforms theme, as the output generated by SECO can always be described as a product or a platform. Secondly, the ecosystem type (in the ‘boundaries’ theme), is also linked to two different products and platforms entities and one role entity. This relationship is defined based on the paper of Jansen and Cusumano [15].

This further indicates that both segments are connected to each other, and both have to be modelled to fully represent a SECO. Finally, the relationship between strategy and actor is already described above, but is also a relationship between the two segments of the meta-model.

**Other relationships:** There are more relationships defined in the meta-model, most of which are generalization relationships derived from the literature. Some notable other relationships in the model are:

- The relationship between actor and product and platforms: Every product or platform in a SECO is developed and/or maintained by one or multiple actors. This relationship is made explicit in the meta-model. The relationship does not specify ownership of a product: This is determined by the strategy a SECO follows about licensing and its affiliation model.
- The relationship between keystone and platform: As described in the section about the identified actors and themes, a keystone controls a platform in some form. This relationship has been made explicit in the meta-model.
- Also, the relationship between platform, product and base technology (as described in the theme identification part of this paper) has been made explicit. This relationship is modelled such that a base technology can be seen as a type of product, and each platform has an aggregation relationship with at least one base product.
- The relationship between output and products and platform: Output is described by Jansen et al. [4] as one of the general characteristics which define a SECO, therefore making it part of the boundaries theme. A relationship can be defined between output and products and platform as the output in the SECO are products and/or platforms and their supporting documents. Therefore, the output entity is modelled to have a generalization relationship with products and platforms.
- Finally, relationships are been made between ecosystem health and three factors which are influenced by the health of an ecosystem: Growth, Maturity and Popularity. As of now, no research has been done on these relationships. Therefore, the direction of this relationship is not made explicit: further research in this subject is needed.

The full meta-model is included in Appendix A.

#### 4.4 Expert Review

The selected expert wrote several papers about Software Ecosystems and is considered an expert in the field of the SECO domain. Because of travel constraints, the interviews were held using Skype. The expert has 15 years of research experiences and 20 years of industrial experience. The results of this expert review are discussed below.

The expert concluded that the technical section was very complete and its entities were all known to him, validating that the entities derived can be seen as common vocabulary in the domain of SECO research. The expert argued that

most of the concepts were clearly defined in the literature, and were therefore well-placed in the model.

On the business segment, the experts argued that the strategy entity should be better related to the actor, as the actor follows a certain strategy, and the ecosystem is merely impacted by that strategy but does not contain it (except for the ecosystem strategy). The layout of the meta-model in this stage of the research suggested that all strategies were part of the SECO. In response, we created a new relationship between strategy and actor in the meta-model. The expert also argued that the “boundaries” entity was somewhat unclear. This resulted in the formulation of a definition for the boundaries term, which is described above. The experts also argued that the terms ‘transparency level’ and ‘ecosystem openness’ were meaning somewhat the same which resulted in a re-evaluation of the two terms and the dropping of the transparency level entity.

A final remark was to restructure the business segment of the meta-model in three parts: one part about starting the SECO, one part about monetizing on a SECO and one part about the boundaries of a SECO (already included). After consideration, the researchers decided not to include this in the meta-model, as this split was not found in the literature considered and therefore including this split was in contradiction with the third entity-selection criterium.

## 5 Discussion and Conclusion

### 5.1 Limitations

There are some limitations to the study at hand. First of all, the researchers are unable to claim completeness: because of time constraints not all papers in the SECO domain could be identified and analyzed. Therefore, the decision was made to use an existing SLR study, and adapt it to the requirements of this research. This could be a threat to the generalizability of this study.

A second limitation is that the model has yet to be evaluated by several case studies. Because of time constraints, only an expert review has been executed, but the meta-model will be further strengthened when there are plenty of case-studies modelled using the technique. Case studies could provide the research field with additional data about the operationalizability of the SECO meta-model developed in this paper. Performing case studies to validate the meta-model is subject to further research.

Another limitation is that we could have missed some papers even after running a snowballing search on the 90 academic from the SLR. Still, snowballing appears to be the most suitable method for following up on previous literature reviews and its nature gives high probability of complete results [28].

Moreover, we are aware that the current meta-model is more of a vocabulary rather than a meta-model and therefore we are planning to apply more sophisticated conceptual modeling and modeling approaches to further develop the main concepts and relationships within the meta-model.

Finally, we are working on creating guidelines for reporting SECO research (similar to [22]) based on the meta-model structure. These guidelines are a necessary element for popularizing the meta-model and unified SECO description that will increase the utility of the meta-model.

## 5.2 Conclusion

In this paper, we present a SECO meta-model that should help researchers to describe and structure the SECOS they are investigating. The meta-model should also improve communication about SECOS research, making it easier to share case studies or to compare SECOS and research about SECOS. Therefore, the research goal of developing a common language for the SECO domain has been fulfilled.

The meta-model which has been created can be used for different aspects within the SECO domain. Researchers are now able to link their work to the meta-model. By linking their work with a certain entity within the model, it becomes clear to every researcher how their work links within the SECO research domain. The model can, therefore, be used as a basis to link future research about SECOS with the knowledge already available in the field.

In addition, researchers can now make generalizations of certain types of SECOS, as modelling different cases of a SECO type now becomes possible with the meta-model. Shared elements in these models can then be identified, which makes it easier to formalize a SECO type or theory. These types can then be researched more in depth, deriving more general theories about SECOS. This helps to formalize the domain, as most research is now done on a case-by-case basis, which makes it hard to generalize from the results.

## 5.3 Future Work

We plan to further develop the concepts included into the SECO meta-model. We also plan to create a knowledge repository where, some example actors and software products can be included, to ensure fast modelling possibilities for a SECO. The meta-model created in this paper can be used as a base in developing an extensive reporting technique which enables to report on the structure and policies of a SECO. Additionally, an algorithm can be developed which might be able to support the modelling of a SECO. The algorithm can populate some simple derivable entities from a SECO, like the actors in a SECO based on an app store or the products in an extension market.

## Appendix A: The Full Meta-model



## References

1. Alves, A.M., Pessoa, M., Salviano, C.F.: Towards a systemic maturity model for public software ecosystems. In: O'Connor, R.V., Rout, T., McCaffery, F., Dorling, A. (eds.) *SPICE 2011*. CCIS, vol. 155, pp. 145–156. Springer, Heidelberg (2011). [https://doi.org/10.1007/978-3-642-21233-8\\_13](https://doi.org/10.1007/978-3-642-21233-8_13)
2. van den Berk, I., Jansen, S., Luinenburg, L.: Software ecosystems: a software ecosystem strategy assessment model, pp. 127–134, January 2010
3. Bosch, J.: From software product lines to software ecosystems. In: *Proceedings of the 13th International Software Product Line Conference*, pp. 111–119. Carnegie Mellon University (2009)
4. Boucharas, V., Jansen, S., Brinkkemper, S.: Formalizing software ecosystem modeling. In: *Proceedings of the 1st International Workshop on Open Component Ecosystems, IWOCE 2009*, pp. 41–50, August 2009
5. Briand, L., Bianculli, D., Nejati, S., Pastore, F., Sabetzadeh, M.: The case for context-driven software engineering research: generalizability is overrated. *IEEE Softw.* **34**(5), 72–75 (2017)
6. Charmaz, K.: The search for meanings-grounded theory. In: *Rethinking Methods in Psychology*, pp. 27–49 (1996)
7. Draxler, S., Jung, A., Boden, A., Stevens, G.: Workplace warriors: identifying team practices of appropriation in software ecosystems. In: *4th International Workshop on Cooperative and Human Aspects of Software Engineering*, pp. 57–60. ACM (2011)
8. Dybå, T., Sjøberg, D.I., Cruzes, D.S.: What works for whom, where, when, and why? On the role of context in empirical software engineering. In: *Proceedings of the ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, ESEM 2012*, pp. 19–28. ACM, New York (2012)
9. Ghaisas, S., Rose, P., Rose, P., Daneva, M., Sikkel, N., Wieringa, R.: Generalizing by similarity: lessons learnt from industrial case studies. In: *Proceedings of the 1st International Workshop on Conducting Empirical Studies in Industry, CESI 2013*, pp. 37–42. IEEE Computer Society, United States, May 2013
10. Hilkert, D., Wolf, C.M., Benlian, A., Hess, T.: The “As-a-Service”-paradigm and its implications for the software industry – insights from a comparative case study in CRM software ecosystems. In: Tyrväinen, P., Jansen, S., Cusumano, M.A. (eds.) *ICSOB 2010*. LNBIP, vol. 51, pp. 125–137. Springer, Heidelberg (2010). [https://doi.org/10.1007/978-3-642-13633-7\\_11](https://doi.org/10.1007/978-3-642-13633-7_11)
11. Iansiti, M., Levien, R.: *Keystones and dominators: framing the operational dynamics of business ecosystems*. The operational dynamics of business ecosystems (2002)
12. van Ingen, K., van Ommen, J., Jansen, S.: Improving activity in communities of practice through software release management. In: *International Conference on Management of Emergent Digital EcoSystems*, pp. 94–98. ACM (2011)
13. Jansen, S.: Measuring the health of open source software ecosystems: beyond the scope of project health. *Inf. Softw. Technol.* **56**(11), 1508–1519 (2014)
14. Jansen, S., Brinkkemper, S., Finkelstein, A.: Business network management as a survival strategy: a tale of two software ecosystems. In: *IWSECO@ICSR 2009* (2009)
15. Jansen, S., Cusumano, M.A.: Defining software ecosystems: a survey of software platforms and business network governance. In: *Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry*, vol. 13 (2013)

16. Jansen, S., Finkelstein, A., Brinkkemper, S.: A sense of community: a research agenda for software ecosystems. In: ICSE-Companion 2009, pp. 187–190. IEEE (2009)
17. Manikas, K., Hansen, K.M.: Software ecosystems—a systematic literature review. *J. Syst. Softw.* **86**(5), 1294–1306 (2013)
18. Object Management Group (OMG): Unified Modeling Language (UML) Specification, Version 2.5.1. OMG Document Number formal/17-12-05 (2017). <https://www.omg.org/spec/UML/2.5.1/>
19. Paige, R.F., Brooke, P.J., Ostroff, J.S.: Metamodel-based model conformance and multiview consistency checking. *ACM Trans. Softw. Eng. Methodol. (TOSEM)* **16**(3), 11 (2007)
20. Petersen, K., Wohlin, C.: Context in industrial software engineering research. In: 3rd International Symposium on Empirical Software Engineering and Measurement, ESEM 2009, pp. 401–404. IEEE Computer Society, Washington, DC (2009)
21. Popp, K.M.: Hybrid revenue models of software companies and their relationship to hybrid business models. In: IWSECO@ ICSOB Confernece, pp. 77–88 (2011)
22. Runeson, P., Höst, M.: Guidelines for conducting and reporting case study research in software engineering. *Empirical Softw. Engg.* **14**(2), 131–164 (2009)
23. dos Santos, R.P., Werner, C.M.L.: A proposal for software ecosystems engineering. In: IWSECO@ ICSOB, pp. 40–51 (2011)
24. Van Angeren, J., Kabbedijk, J., Jansen, S., Popp, K.M.: A survey of associate models used within large software ecosystems. *Computing* **746**, 27–39 (2011)
25. Van Den Berk, I., Jansen, S., Luinenburg, L.: Software ecosystems: a software ecosystem strategy assessment model. In: Proceedings of the Fourth European Conference on Software Architecture: Companion Volume, pp. 127–134. ACM (2010)
26. Viljainen, M., Kauppinen, M.: Software ecosystems: a set of management practices for platform integrators in the telecom industry. In: Regnell, B., van de Weerd, I., De Troyer, O. (eds.) ICSOB 2011. LNBIP, vol. 80, pp. 32–43. Springer, Heidelberg (2011). [https://doi.org/10.1007/978-3-642-21544-5\\_4](https://doi.org/10.1007/978-3-642-21544-5_4)
27. Wieringa, R.J.: Design Science Methodology for Information Systems and Software Engineering. Springer, Heidelberg (2014). <https://doi.org/10.1007/978-3-662-43839-8>
28. Wohlin, C.: Guidelines for snowballing in systematic literature studies and a replication in software engineering. In: 18th International Conference on Evaluation and Assessment in Software Engineering, EASE 2014, pp. 38:1–38:10. ACM, New York (2014)