DiVA
http://www.diva-portal.org

This is the published version of a paper published in *Information and Software Technology*.

Citation for the original published paper (version of record):

Axelsson, J., Papatheocharous, E., Andersson, J. (2014)
Characteristics of software ecosystems for Federated Embedded Systems: A case study
*Information and Software Technology*, 56(11): 1457-1475
https://doi.org/10.1016/j.infsof.2014.03.011

Access to the published version may require subscription.

N.B. When citing this work, cite the original published paper.

Permanent link to this version:
http://urn.kb.se/resolve?urn=urn:nbn:se:lnu:diva-37846

Information and Software Technology 56 (2014) 1457–1475

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

# Characteristics of software ecosystems for Federated Embedded Systems: A case study

Jakob Axelsson [a,b,*], Efi Papatheocharous [a], Jesper Andersson [c]

[a] Swedish Institute of Computer Science (SICS), SE-164 29 Kista, Sweden
[b] School of Innovation, Design and Engineering, Mälardalen University, SE-721 23 Västerås, Sweden
[c] Department of Computer Science, Linneaus University, SE-351 95 Växjö, Sweden

ARTICLE INFO

*Article history:*
Received 3 July 2013
Received in revised form 21 March 2014
Accepted 25 March 2014
Available online 4 April 2014

*Keywords:*
Software ecosystems
Embedded Systems
Systems-of-Systems
Architecture
Case study

ABSTRACT

*Context:* Traditionally, Embedded Systems (ES) are tightly linked to physical products, and closed both for communication to the surrounding world and to additions or modifications by third parties. New technical solutions are however emerging that allow addition of plug-in software, as well as external communication for both software installation and data exchange. These mechanisms in combination will allow for the construction of Federated Embedded Systems (FES). Expected benefits include the possibility of third-party actors developing add-on functionality; a shorter time to market for new functions; and the ability to upgrade existing products in the field. This will however require not only new technical solutions, but also a transformation of the software ecosystems for ES.

*Objective:* This paper aims at providing an initial characterization of the mechanisms that need to be present to make a FES ecosystem successful. This includes identification of the actors, the possible business models, the effects on product development processes, methods and tools, as well as on the product architecture.

*Method:* The research was carried out as an explorative case study based on interviews with 15 senior staff members at 9 companies related to ES that represent different roles in a future ecosystem for FES. The interview data was analyzed and the findings were mapped according to the Business Model Canvas (BMC).

*Results:* The findings from the study describe the main characteristics of a FES ecosystem, and identify the challenges for future research and practice.

*Conclusions:* The case study indicates that new actors exist in the FES ecosystem compared to a traditional supply chain, and that their roles and relations are redefined. The business models include new revenue streams and services, but also create the need for trade-offs between, e.g., openness and dependability in the architecture, as well as new ways of working.

© 2014 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/3.0/).

## 1. Introduction

In many industries developing technical products, such as automotive, aerospace, or process automation, Embedded Systems (ES) and software play an increasingly important role [1]. Traditionally, the ES has been an integral part of the physical product and being in charge of controlling and monitoring the product during its operation. Some of the key characteristics of many ES are that they have to be cost-efficient, reliable, robust, safe, and secure, and hence they are typically tailored for a certain product.

The software of the ES is usually monolithic in the sense that it is handled in the product as one piece, without the possibility to replace only parts of it, or add new parts on top of the present ones. The software is installed during production, and upgrades normally require physical connection to the ES. Often, the ES is developed by an external supplier based on a specification from the manufacturer responsible for integrating the final product.

With the arrival of affordable communication technologies, in particular wireless, it is becoming feasible to provide external communication capabilities to the ES. This eventually allows updates of software to be carried out remotely even though the flexibility of such upgrades would be limited due to the monolithic nature of the ES software. Recent research [2] has however demonstrated how a software plug-in mechanism can be utilized, that allows

* Corresponding author at: Swedish Institute of Computer Science (SICS), SE-164 29 Kista, Sweden. Tel.: +46 72 734 29 52.
*E-mail address:* jakob.axelsson@sics.se (J. Axelsson).

http://dx.doi.org/10.1016/j.infsof.2014.03.011
0950-5849/© 2014 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/3.0/).

the installation of add-on functionality in the ES and changes its monolithic nature. Such a plug-in mechanism, in combination with external communication, opens up many new possibilities for ES in particular and product development in general:

- It shortens dramatically the time to market for introducing new or extended features, and allows addition of new features into products that are already deployed, something which was not feasible before.
- It allows the creation of Systems-of-Systems (SoS), by letting plug-ins in different products connect to each other and exchange data through communication channels, thus forming a Federated Embedded System (FES), in ways that were not thought of at the time of design of the individual systems.
- It provides opportunities for third-party developers to supply add-ons, and thereby fosters open innovation formed by openness in collaborations in a way that was never seen before in the area of ES.

As an illustrative example of a FES, let us consider the case of car-to-car and car-to-infrastructure communication. In such a scenario, one could imagine a service where information is passed to cars about the recommended speed to be able to pass road intersections without having to stop at red lights. The cars and the servers in the road infrastructure form a federation to achieve this service. The recommended speed could be fed into the car's cruise control system to automatically adapt the car's speed. Thus, the service could decrease fuel consumption and travel time, and improve traffic flow. The information gathered by this service could further be used in real-time services that estimate the travel time, but also in aggregated services that show traffic patterns for route guidance, or even for long-term road network planning improvements. It is in this combination of direct improvements to the individual ES products and the creation of novel services that the real potential of FES resides.

These possibilities lead to many challenges on a technical level. In addition, they also have large implications on how the development process is instantiated and executed, and changes the relations between the different parties involved. The supply chain model traditionally used for ES, with a manufacturer integrating parts ordered from suppliers, will need to expand into a much more dynamic software ecosystem [3], where many new actors enter the ecosystem and interact. This perspective involves expansion beyond and across organizational boundaries, exposing platforms and opening access to reusable assets, increasing collaboration in the software ecosystem, but also involving the need of management of an economic, business and social environment [4].

The contribution of this paper is an empirical study of how such an ecosystem for FES should be set up and supported, resulting in a characterization of the business aspects, development issues, and architectural strategies. This ecosystem includes the ES with its hardware and built-in software; the software plug-ins that are added dynamically to the ES; the software running on servers that is needed to create services in the FES; and also to some extent, the mechanical products on which the ES are installed. Ecosystems in the ES domain are different than typical software ecosystems because the processes carried out need to support building multiple complex units of distributed functionality with real-time properties and constraints.

### 1.1. Research questions

The possibility of creating FES is currently emerging, and many companies in different industries are struggling with how to position themselves in this new ecosystem. Often, it will require the companies to make major revisions to their core business strategies, as well as their technology platforms, to remain competitive. The *goal* of this research is to provide support for such decisions by providing a more complete picture of how an ecosystem for FES should be organized. The underlying hypothesis of the research is:

**H.** The business related aspects are just as important as technology to the industry actors in order to make FES successful.

Therefore, two primary *Research Questions (RQs)* have been studied:

1. *What actors are needed in an ecosystem of Federated Embedded Systems (FES), and what are their relations?*
2. *What are the key elements of (a) the business models, (b) the product architecture, and (c) the processes, methods, and tools, needed to make the ecosystem effective?*

As a side effect of studying these RQs, we expected that various other relevant areas related to software ecosystems for FES would be discovered. Even though the development of FES technology is at a relatively early stage, we find it essential to have an early look at the non-technical aspects around it, since business models and development processes must be in place to make the technology effective. The way actors in the ecosystem interact will most likely also influence technology, by requiring different mechanisms to support development and operation of the system, and this research aims at discovering those mechanisms up front in order to ensure that they are properly reflected in future technical solutions.

### 1.2. Research method

Ecosystems for FES have only recently started to emerge, and therefore it is hard to find good examples of mature constellations to study empirically. Therefore, our research is highly explorative in its nature, and the best we can do to gain empirical data is to study similar ecosystems in related areas and try to extract and extrapolate information from there. The goal is thus to formulate a conceptual model which can be thought of as a set of hypotheses describing the important elements and relations in an ecosystem for FES. Naturally, these hypotheses have to be validated in future research.

There are several candidate research methods that could suit this kind of research, including case studies, grounded theory, and ethnographic studies. Grounded theory is useful when there is a need to generate new theories, but in this case there are already established theories for software systems, and it did not appear likely that the extension to FES would require a whole new set of concepts. As for ethnographic methods, it was difficult due to practical reasons. It would have required us to follow a certain ecosystem within it for some time, and we did not have access to a suitable object of study to allow this.

The research method we have chosen to apply is therefore an explorative case study, details of which can be found in [5]. In this research, the context is the relevant industry developing technical products and services in general. The focus is on the ecosystem as a whole, rather than on an individual company. The study includes a number of different companies, and is thus, according to the terminology of [5], an embedded case study with multiple units of analysis.

Since the study is not based on a pre-defined theory and it is performed for exploratory purposes, it is important to be open-minded and open-ended in the data collection process, and therefore a research method based on semi-structured interviews was considered suitable. In this way, an a priori idea of the most important

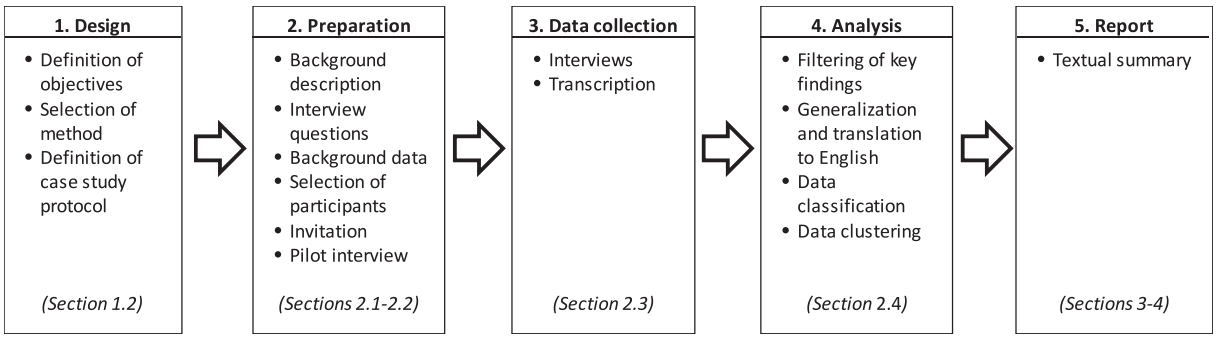


**Fig. 1.** Overview of the study.

aspects can be complemented with the spontaneous input from interviewees. This also makes the study qualitative in nature, and it has not been a target to collect substantial quantitative data.

In addition, the research method of the study serves the purpose of descriptive research, portraying the situation or phenomenon, based on Robson's classification [6]. It involves information from a specific population or some sample from one. The method is closely related to a case study and is primarily used for exploratory purposes, but also for descriptive purposes in the cases where the generality of the situation or phenomenon is or can be of secondary importance. The basic process steps in conducting case study research proposed in [7] were applied, particularly, (1) case study design, (2) preparation for data collection, (3) collecting evidence, (4) analysis of collected data, and, (5) reporting. The adaptation of these generic steps to the concrete setting was defined in a case study protocol during the design of the study. Fig. 1 shows an overview of these steps in the study, with indications on where in this paper the different parts are covered.

### 1.3. Overview of paper

The remainder of this paper is structured as follows. In the next section, the design of the case study is presented in more detail, including information on the interviews conducted and the analysis framework used. In Section 3, the analysis of the collected data is presented, and Section 4 discusses these findings with a focus on the areas described in the RQs. In Section 5, the validity of the findings is analyzed, and Section 6 provides a review of related research. Finally, in Section 7, the conclusions are summarized together with indications of future research.

## 2. Case study

In this section, the design of the case study is presented. The description includes what kind of preparation was done, how participants were selected and contacted, how the interviews took place, and the design of the analysis framework used to extract the relevant information from the interview data in order to address the RQs.

### 2.1. Preparation

In order to ensure that all relevant aspects were covered during the interviews, a set of interview questions were prepared in advance. The questions (Appendix A) served as a checklist, and also drove the interview forward in case this did not happen naturally. There were approximately 20 interview questions prepared beforehand divided in five categories: (1) Preliminary data about the interviewee and the organization; (2) roles and relations in the ecosystem (RQ1); (3) business models (RQ2a); (4) development processes, methods, and tools including product architecture (RQ2b-c); and (5) wrap-up questions, including an open question if anything relevant should be added to the discussion.

In addition, a short text of approximately half a page was prepared which was used to frame the background and objectives of the interview. The text also explained how the data would be used and included ensuring the confidentiality and anonymity of the participants and their organizations. The first interview conducted was used as a pilot, with the intention of updating the interview questions and background text for the remaining interviews. However, the pilot did not reveal any need to change the interview questions, and all interviews were conducted based on the original material prepared. Prior to the interviews, some background data was also gathered from public sources about the companies, their business and the interviewees, in the cases where they were not known.

### 2.2. Participants

The selection of the participants in the interviews can be best described as a stratified convenience sampling. The stratification was in the sense that we strived to include companies that we could beforehand understand would be most likely to play different roles in a future ecosystem. Therefore, we included participants from different domains, namely manufacturers, traditional suppliers, service operators, and add-on software developers. Also, we included participants that play different roles in their organization, including both engineers and business people, and both managers and non-managers. In addition, we included participants from several industrial domains, namely automotive, off-road machinery, and automation.

The sampling was convenience-based, since most participants were, in one way or another, acquainted with the researchers before. This included old colleagues, previous research partners, or just more remote and recent contacts of the researchers. We considered this necessary for two reasons: it allowed us to find suitable people in the organizations to interview, thus ensuring high quality data collected from the interviews; and it ensured that the participants trusted the researchers enough to be able to speak openly without the fear of confidential information being disclosed.

The participants were initially contacted via e-mail, receiving a short personal message, which differed depending on the participant's relation with the researchers. Attached to the e-mail was also the short text describing the background and objectives of the research, mentioned in the previous section. After a positive response had been received, an interview time was decided. In several cases, the participant asked if they could bring a colleague to the interview, and this was accepted.

**Table 1**
Overview of interviews, companies, and participants.

| Interview | | | Company | | | | Participant | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Id | Duration (hh:mm) | No. of statements | Id | Industry | Role | Size (S, M, L) | Id | Years employed | Years since MSc | Gender (M, F) |
| 1 | 01:10 | 39 | A | Automotive | Service operator | L | i | 3 | 12 | F |
| 2 | 01:10 | 27 | B | Automotive | Supplier | M | ii | 5 | 40 | M |
| | | | | | | | iii | 1 | 11 | M |
| 3 | 00:55 | 26 | A | Automotive | Service operator | L | iv | 10 | 18 | M |
| | | | | | | | v | 20 | 25 | M |
| 4 | 00:50 | 20 | C | Automotive | Manufacturer | L | vi | 20 | 26 | F |
| 5 | 02:00 | 62 | C | Automotive | Manufacturer | L | vii | 26 | N/A | M |
| 6 | 01:10 | 32 | C | Automotive | Manufacturer | L | viii | 14 | 20 | M |
| 7 | 00:50 | 19 | D | Automotive | Manufacturer | L | ix | 27 | 35 | M |
| | | | | | | | x | 10 | 8 | M |
| 8 | 01:20 | 41 | E | Software Consultancy | Add-on developer | S | xi | 1 | 19 | M |
| 9 | 00:45 | 21 | F | Embedded Systems | Supplier | M | xii | 5 | 16 | M |
| 10 | 00:57 | 24 | G | Embedded Systems | Supplier | L | xiii | 21 | 22 | M |
| 11 | 01:10 | 44 | H | Off-road Machinery | Manufacturer | M | xiv | 5 | 18 | M |
| 12 | 01:15 | 15 | I | Automation | Manufacturer | M | xv | 6 | 16 | M |
| Avg | 01:07 | 30.8 | | | | | | 11.6 | 20.4 | |

We performed in total 12 interviews with 15 persons from 9 companies. Table 1 shows an overview of the interviews, companies, and participants. All of the participants but one had a university education, and the average time from graduation was over 20 years, which indicates that the selection of participants involved highly knowledgeable people. The subjects were also considered to be experienced within their companies, with an average of over 10 years with their current employer. All participants were Swedish, and act in leadership roles (i.e., line managers, project managers, technical experts and system architects). The order of the interviews in Table 1 follows the order in which they were carried out. Regarding the size of companies, it was a bit tricky to define exactly, since many companies form part of a larger group, and the size can either represent the unit or the group. In the table, the companies are classified in sizes according to the unit at the site in question, and using the scales of Small (S) (<10 employees), Medium (M) (10–1000), and Large (L) (>1000).

### 2.3. Data collection

The interviews started by the researcher shortly recapitulating the background of the interview, based on the short text that was included in the e-mail sent to the participant. In most cases, this was sufficient for the respondents to start discussing, and there were only limited need for the researchers to pose direct questions. However, towards the end of each interview, the set of prepared interview questions (see Appendix A) was used as a checklist by the researcher to ensure that no major area intended to be discussed was left uncovered. Often, the respondents illustrated their points by giving concrete examples from their companies. All interviews were conducted in Swedish. Most interviews were carried out face-to-face, except interview 9, which was over telephone. The first author conducted ten of the interviews, and the last author the remaining two (interviews 10 and 12). One researcher thus participated in each interview and usually took handwritten notes. This method of data capturing was chosen to make the interviews as informal in nature as possible and to encourage openness from the respondents. The handwritten notes were transcribed on a computer shortly after each interview (usually directly after, and in all cases within 24 h) to minimize the risk that they would be difficult to fully understand and interpret afterwards. In two cases (interviews 10 and 12), a recording equipment was also used, since there was a risk entailed in those interviews to lose some of the information as the transcription had to wait for a few days. The result was thus one computer file from each interview, and each file contained a list of statements collected from that interview. In a few cases, the respondent drew pictures on a whiteboard, and these drawings were also captured in the notes. In some cases also presentation material was briefly used by the respondent in order to explain some things, and copies of these presentations were collected by the researchers. As shown in Table 1 above, the duration of the interviews varied between 45 and 120 min, with an average of 67 min.

### 2.4. Analysis framework

The processed result of each interview was a list of statements, in a Microsoft Office Excel spreadsheet, and each statement was tagged with a unique identity indicating the interview it originated from and the sequence number in that interview, to allow tracing back to the source. Some of the statements contained key findings that could bring light to the RQs, whereas others were more contextual, e.g., gave concrete examples of a specific situation in a certain company, but did not directly address the topics of interest. The latter were categorized as background information, and were removed from the structured analysis. The classification of which statements constituted key findings was done in the spreadsheet by two researchers independently, and then their analyses were compared. The key findings were then reformulated to become more general (e.g., the names of companies and people, and other specific information, scenarios were removed), and simultaneously translated from Swedish to English. The reformulation was reviewed by two researchers.

The statements considered as key findings were further classified into a number of areas, using columns in the spreadsheet, and the categories used for this were the building blocks identified in the Business Model Canvas (BMC) [8]. The reason for basing the analysis of business models on the BMC was that it is an established method for business analysis with clearly structured categories, where several categories focus on relationships between actors (RQ1), which are of particular interest to this study. The method is designed for the purpose of analyzing new business models, which is also highly relevant and applicable in the case of FES ecosystems (RQ2a). Also, after the BMC creation, the extraction of the product architectures, processes, methods and tools (RQ2b-c) was carried out as an additional step to complete this study.

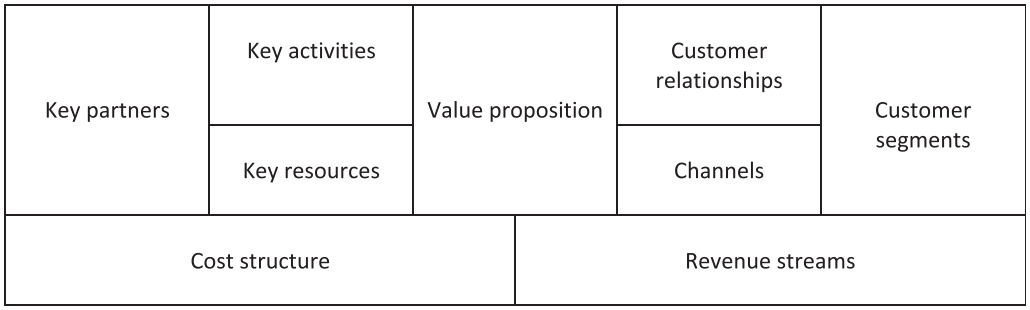


**Fig. 2.** The Business Model Canvas (BMC).

The categories of the BMC are illustrated in Fig. 2, and are further described next:

- *Customer Segments* define the different groups of people or organizations an enterprise aims to reach and serve.
- *Value Proposition* identifies the products and services that create value for a specific customer segment.
- *Channels* describe how a company communicates with and reaches its customer segments to deliver a value proposition.
- *Customer Relationships* define the types of relationships a company establishes with specific customer segments.
- *Revenue Streams* represent the cash a company generates from each customer segment.
- *Key Resources* describe the most important assets required to make a business model work.
- *Key Activities* are the most important things a company must do to make its business model work.
- *Key Partners* describe the network of suppliers and partners that make the business model work.
- *Cost Structure* describes all costs incurred to operate a business model.

By analyzing the interview data based on the BMC categories, it becomes possible to also extract elements to answer the different RQs. In particular, the actors and their relations (RQ1) can be found primarily by studying the Customer Segment and Key Partners categories; the product architecture (RQ2b) appears primarily as a Key Resource; and the processes, methods, and tools (RQ2c) are found as Key Resources and Key Activities. Information on the business model (RQ2a) are found in most categories, and especially in Value Proposition, Revenue Streams, and Cost Structure.

In our analysis, we chose to apply the BMC for the ecosystem as a whole, whereas a more refined analysis could have included one canvas for each actor type, or even for each individual company. Each interview statement was classified as belonging to several of the BMC categories. The classification was performed initially by the researcher that conducted each interview, and then a second researcher repeated the classification activity independently. In the end, all key findings that at least one of the researchers assigned to be included in the analysis of that category was used.

To illustrate the process used in the analysis, a concrete and representative example will now be provided. In one of the interview transcripts, the following statement was included (shown in an English translation, although in reality it was in Swedish, and with the information that could reveal the company made anonymous):

> *"[Name of company] delivers all [type of product] with a telematics unit, and contracts have been made so that the cost for data traffic is included in the service agreement."*

**Table 2**
Number of key findings assigned to each category.

| Category | No. key findings |
|---|---|
| Customer Segments | 21 |
| Value Proposition | 106 |
| Channels | 36 |
| Customer Relationships | 82 |
| Revenue Streams | 67 |
| Key Resources | 100 |
| Key Activities | 110 |
| Key Partners | 86 |
| Cost Structure | 52 |
| Total number of category assignments | 660 |

The statement was reviewed by two researchers, who agreed that this had sufficient relevant information in it to qualify as a key finding. However, it contained information about the company that had to be made anonymous, and therefore it was reformulated, while at the same time also being translated to English, resulting in:

> *"Telematics units could be included in the base product, and the data traffic fees could be included in a service agreement."*

Then, one researcher made the classification based on BMC, and decided that this key finding related to the categories Revenue Streams (since it deals with service agreements) and Cost Structure (since it deals with data traffic fees). A second researcher reviewed the first classification, and proposed that the key finding also related to Value Proposition (since the service agreement is a value proposition) and Key Resources (since the telematics unit is part of the hardware architecture, which is a key resource). The two researchers discussed their views, and agreed that this key finding should be classified in all four categories.

The same process was followed for all interviews and statements. The 12 interviews conducted resulted in 370 statements, of which 231 were considered to be key findings (62%). On an average, each key finding was assigned to 2.9 categories from the BMC. Table 2 shows how the various key findings were assigned to each of the categories.

After classifying all key findings to categories, the statements assigned to each category were analyzed separately, by grouping similar items into related topics for which a common conclusion could be drawn. These topics are discussed in the next section.

## 3. Analysis

We now present the analysis of the interview data based on how the different aspects of BMC are reflected in ecosystems for FES, as identified in our interviews. Firstly, we provide a section with a short description of the key actors that have been identified

from the interviews to address RQ1, and this is because the actors appear frequently in the discussion of the remaining analyses.

In the text below, quotations are sometimes used. Note that these refer to the generalized descriptions of key findings formulated in Step 4 of Fig. 1, and are not direct quotations from the interviews, but still reflect a direct link to the raw data.

### 3.1. Actors

Based on the interview data, we have identified a number of different actors that play important roles in the ecosystem for FES:

- *Manufacturer:* an organization that manufactures and sells products which are ready for final use. The manufacturer is responsible for integrating products and their components.
- *Supplier:* an organization that provides components to a manufacturer (or to another supplier) based on their requirements.
- *Content provider:* an organization that provides data which can be used in the operation of a product.
- *Communication provider:* an organization that manages and maintains the communication networks between actors and products.
- *Service operator:* an organization that operates different services related to the products, usually based on information obtained from the ES in those products.
- *Add-on developer:* an organization or individual that develops plug-in software that can be integrated into a product after production.
- *Owner:* an organization or individual that purchases the product offered by the manufacturer and the services offered by the service operator.
- *End user:* an individual who operates the final product.
- *Regulatory agency:* a public authority that puts restrictions on the technical solutions and operation of the products.
- *Information broker:* an organization that trades information, such as operational data from ES.

The above actors are role descriptions, and in a real ecosystem, the same organization could be playing several roles. Comparing these actors with the abstract categories identified by Manikas and Hansen [9], the manufacturer is usually the "orchestrator" and the "vendor", and the suppliers are "niche players". Content providers, service operators, add-on developers, and information brokers are all "external actors", and the owners and end users are "customers".

Within many organizations, such as manufacturers and suppliers, there are also important internal actors, e.g., engineering and production. For example one of the subjects described that the engineering department could be responsible for developing the product of the company, tailoring it to specific customer needs, and for providing advanced consultancy services.

### 3.2. Customer Segments

The Customer Segments building block of the BMC includes the identification of the most significant customer for whom the products create value. In the actors description in the previous section, the customers are primarily represented by the Owner and End user actors. To a large extent, the concrete Customer Segments are specific to the applications, and the interview subjects also did not discuss much about customer segments (as can be seen in Table 2). However, some general observations were made from the interviews, relating to broadening customer reach, improving customer feedback, and offering customer differentiation.

#### 3.2.1. Broader customer reach

The customers are individuals or organizations that buy the physical products where the ES is integrated. The possibility of having add-on software for these products and developing new services, introduce new features and extended functionality in the products. Thus, as the products are extended, the market is also extended to include niche segments that were previously ignored since they were considered unprofitable. The main reason why customer reach is currently not developed to the full extent for services, according to one participant of the interviews, is: "Manufacturers are good at mapping products to different customer segments, but not so good at doing the mapping for services. It is not always the case that the same customer segments are relevant for services as for products." It would be beneficial to find ways to facilitate for manufacturers to identify and target different customer segments, based on the products, as well as based on the services offering something which is not currently successfully performed to the extent it could be.

#### 3.2.2. Improved customer feedback

Manufacturers require market and end user feedback in different ways to better understand the customer segments. As one of the subjects mentioned in the interview, the engineering department is responsible for developing the product as well as for tailoring it to specific customer needs. However, planned new product generations are troublesome as the engineering department often lacks the ability to efficiently provide appropriate configurations and differentiated products. "It is interesting to get feedback from the field regarding how products are being used, in particular by early adopters." Thus, building a close relation between the development organization, collaborating parties and customer segments gives opportunities for obtaining immediate feedback from the field. Such feedback is significant, especially if it comes from early adopters of new products and services, since it can be expected to eventually reduce lead time and increase quality as well as provide an improved understanding of the relevant customer segments and their needs. However, feedback poses another challenge, related to privacy. Specifically, the information that is obtained from the use of products in various contexts and by various types of customers, needs to be protected, so sensitive information is secured and anonymous. Thus, finding ways to enable privacy, trust and confidentiality in the communications between the parties is a fundamental challenge. We will return to this issue in Section 3.5.3 below.

#### 3.2.3. Customer differentiation

A platform that is flexible and provides the means for feature adaptation could be proven particularly useful when trying to reach new and diversified customer segments for services and products. More flexible architectures will be overall more easily adapted to changes, even after initial deployment, which creates new opportunities for all actors.

As mentioned in the interviews, organizations must differentiate their product and service offerings to satisfy different actors and customer segments. A platform that supports adaptability at different levels in different parts is a prerequisite for that to happen. In order to extend the market, the key customer segments need to be identified and collaborations and partnerships must be established. This can result in the definition of new products based on the market needs. Moreover, new features will be tested and evolved based on the observed markets' reaction.

### 3.3. Value Proposition

The Value Proposition building block of the BMC describes the bundle of products and services that create value for a specific

Customer Segment. In this section, the focus will be on defining different service types, discussing the value of openness, and what are the most important quality attributes for FES.

#### 3.3.1. Service types

A number of examples of products and services related to FES have been provided during the interviews, and based on our analysis, we can broadly classify them into the following categories:

- *Product services:* that improve the direct operation of a product in isolation. An example could be a connected navigation system in a car that not only shows maps but also traffic information. These services are often directed towards the end user of the product.
- *Process services:* that improve the operation of a product as part of a larger process (i.e., as part of a SoS). As an example, a truck fleet operator could gather data from all vehicles in a central system and improve navigation directions to drivers. These are often oriented towards the product owners, and mainly fetch data, with only limited control of individual products.
- *Life cycle services:* that use data from products to improve the associated life cycle processes. This includes predictive and preventive maintenance, but also feedback to product development on the usage of the individual products. These services are often directed towards the manufacturer or associated organizations (see Section 3.2.2 above).
- *Extended services:* that use data from a product to improve the operation of totally unrelated products or services. For instance, vehicles could collect data about the status of the roads (e.g., detecting pot holes), and this data could be used by road administrators to improve road maintenance. They could also supply such data to an information broker, who would then sell it to other actors in the FES.

#### 3.3.2. Open interfaces

Open interfaces, once in place, provide a value on their own for products, since they would make it feasible to integrate products with other products, forming a SoS. However, the add-on software would enhance these interfaces considerably, since it would allow much more extensive tailoring than, for example, an open serial bus interface. For process services, it is a common practice that products from different manufacturers are used within the same process. Hence, a product that is adaptable to, and is efficient in, such a SoS would be of value and particularly desirable to the owner. There are however, two contradictory trends: on the one hand, process operators wish more support in creating a complete system solution, and on the other hand, they strive to cut costs by purchasing equipment from competing manufacturers. Finally, being able to offer open interfaces of ES to IT systems, e.g., for business administration, could also improve considerably the owner's overall business processes.

#### 3.3.3. Quality attributes

It was discussed during the interviews that having a vast amount of different add-on software for a product is important for ES, but not as critical as it is for smart phones. Nevertheless, it is crucial to offer a clear value to the specific products. Quality attributes, such as a usability, reliability and safety, were discussed and are expected to be important for ES, and assign specific restrictions on just how open the development could possibly get. The information stored in the system can also have different characteristics, based on the defined quality of the data itself, the acceptable delays, and so on. These characteristics depend on how the information will be used by a specific function and in different contexts.

### 3.4. Channels

The Channels building block in the BMC describes how a company communicates with and reaches to its Customer Segments to deliver a Value Proposition. In relation to FES, the interview subjects discuss four types of channels, namely sales channels, distribution channels, configuration channels, and information channels. They also highlight the importance of who is controlling these channels. The main difference with the FES setup is that manufacturers will be able to handle sales, distribution, and configuration directly, reaching their customer anywhere in the world over the Internet, and thereby reducing the need for certain special service providers (such as workshops) that exist today.

#### 3.4.1. Sales channel

In traditional ES, the sales channel for software has been the same as for the overall product, and software has been delivered initially as pre-loaded into that system. Any configuration or distribution of upgrades has been required to be handled by service personnel, e.g. at a service station. The online customers' reach, through the Internet, anytime, from and to anywhere in the world, allows the possibility to manufacturers to have direct contact with the customer, with the potential of conducting additional sales of their products and services. However, manufacturers currently find it hard to quantify the value of this relationship in their business cases for connectivity, probably due to a lack of evidence concerning the magnitude of the additional sales.

#### 3.4.2. Distribution channel

The distribution channel ensures that add-on software that has been made available to a certain system instance gets installed into that system. Much of the functionality of this channel is probably generic, and hence some of the manufacturers mention the possibility to outsource the operation of this communication channel to a third party, e.g., a telecom provider. It should be noted that there is also a need expressed of a distribution channel for testing purposes, where add-on developers can make their prototype software available to specific users.

#### 3.4.3. Configuration channel

The configuration channel gives the possibility for the users to configure what kind of add-on software they have installed in their system, and modify it, either manually from a distance, through a desktop computer, or automatically, update to the latest version. By keeping track of what software customers have installed in their system, the manufacturer can also tailor their offerings to increase overall sales.

#### 3.4.4. Information channel

Although the focus of the discussions in the interviews was on channels for add-on software, it is also recognized that similar channels are needed for information exchange. Hence, there could be a need for an information broker, whose main operation is to sell basic data to different service providers. Access to data over a public network would also provide a possibility for tuning and for individualizing equipment maintenance.

#### 3.4.5. Channel ownership

In many practical situations, the above channels could all be implemented by the same IT system, but other possibilities exist. If channel responsibilities are allocated to several IT systems, channel ownership and control become critical factors, and to many, it seems natural that the manufacturer is responsible for the channels of their products. However, for add-on developers who provide software to products from many manufacturers, it might be desirable to provide their own channel, in particular if the end

users purchase products of different brands and want to include the same plug-ins in all of them. There is also a fear that the manufacturers would take a large part of the revenue if all sales go through their channel (as can be seen in some channels used for apps in smart phones), and that an add-on developer's brand would be less visible.

### 3.5. Customer Relationships

The connectivity provided in FES gives the actors new possibilities to build relationships with their customers. By tying the ES into the business IT infrastructure systems of the customer, it becomes more difficult later on to switch to the product of a competitor. However, the technology also raises issues related to handling the services over the product life cycle, liability concerns, and dealing with the ownership of information, including the related privacy concerns.

#### 3.5.1. Product life cycle

Many industrial products, that may be the target of FES, have very long lifetime, whereas the add-on developers and service operators can be expected to often be smaller and younger companies. Most likely, many customers will turn to the manufacturers to provide guarantees of the services associated with their products. However, in our interviews, there are clear indications that many manufacturers intend to make no commitment towards what services will be provided over time, and if such commitments should be made, there are no specific guidelines for what compensation should be offered, in case the service nevertheless ceases to exist. For many products, it is quite common that they also are re-sold second hand, and this leads to further issues related to the ownership of the installed add-ons, and service commitments. The considerations include whether they should be automatically transferred to the new owner, or if there are other possibilities that should better take place.

#### 3.5.2. Liability

In general, product liability becomes complex when working in an ecosystem where different actors can add functionality. Some would claim that the manufacturer always has a liability for their product vis-à-vis the customer, but it is unclear how to handle this when third-party software is added, since it can often be hard to sort out if a problem was caused by the base product or the add-on. Most likely, the manufacturer will only take a limited responsibility for the add-on software. Many ES also have to meet legal requirements regarding safety, and are subject to certification by regulatory agencies. Today, in the FES domain these requirements do not often foresee the existence of add-on software.

#### 3.5.3. Ownership of information and privacy

Networked systems like FES have enormous possibilities for collecting data, and this information is valuable to many actors. Therefore, some manufacturers and service operators require the users to grant them the right to access certain data in the products for other purposes than to provide a service to the same customer. In our interviews, contradictory opinions were voiced regarding if this constitutes a problem with respect to the privacy of the users. Some respondents see this as a major problem that has led to vivid debate within their companies, whereas others claim that as long as the identity of individuals is kept confidential it is not a problem. Possibly, the difference in perspective of each respondent depends on if the product is consumer- or business-oriented, where privacy of individuals is a larger concern for consumers and less so for companies, where employees expect to be subject to a certain degree of surveillance. Also, it is noted that the importance of privacy is to some extent situation dependent, e.g., in an emergency situation the location of a missing person might be very important to share, whereas under normal circumstances it should be kept confidential.

### 3.6. Revenue Streams

The Revenue Streams building block of the BMC represents the cash a company generates from each Customer Segment. In this section, different ways of generating revenue are discussed. From the interviews, it is apparent that "there are not any established models for paying for services and data." It is unlikely to see one model to fit all, but different models will most likely be needed in different situations.

#### 3.6.1. Volume increases

In current ES-based industries, there is no specific revenue stream defined for software, and the prospect of FES could thereby improve the revenue for manufacturers in several ways. One way is to make their products more attractive with the added functionality and increased flexibility, thus increasing earnings from goods through increasing the volume of the offering, e.g., by reaching niche segments as discussed in Section 3.2.1 above.

#### 3.6.2. Recurring sales

FES also offers possibilities for recurring sales of services during the lifetime of the product, and not obtaining profit just from the initial selling of the product. In many industries, as products mature they are commoditized, and the manufacturers need to offer new values, such as overall business and system solutions and services, to maintain their profitability.

#### 3.6.3. Direct sales of add-on software

One reason for manufacturers to open up their products to third-party software suppliers in FES is that they can improve innovation processes that can provide an extended customer value. The add-on developers can receive revenues through either direct sales of their software to users, or by developing the add-ons based on a contract with the manufacturer or owner. Whichever way is used, it is important to find financial solutions that give a sustainable profit to both sides. Different pricing schemes are possible, including offerings of a complete package with all available add-ons, without any additional usage cost; providing an entry-level of functionality for free, and extended features at a premium price; providing trial-versions for a limited time or with limited functionality, and then requiring payment to continue using the offering afterwards.

#### 3.6.4. Subscription fees

Several interviews mention the possibility of using subscription fees as an alternative to selling an asset. "Subscription could be an interesting business model. However, manufacturers do not have the experience of handling this." Some consider it would be better to give unlimited access to information services based on an overall service agreement rather than having to deal with a costly management of small revenue streams from many singular services. Anyhow, it is important to find ways for getting paid for information, to ensure that the services can be maintained over time and hence allow users to build confidence for the subscription scheme.

#### 3.6.5. Revenue sharing

"Revenue sharing is an interesting business model. For instance, if the service promises to achieve x% efficiency improvement, the profit the customer makes from this improvement could be shared with the actor offering the service." Some information service operators are also interested in revenue-sharing models with manufacturers, since having their service available in the

manufacturer's product gives them another channel to the customer, which could motivate sharing the revenues with the manufacturer.

### 3.7. Key Resources

The subjects in the study span the range from enterprises with worldwide operations, to small and medium enterprises with niche products and small organizations. The main resources used by the partners of the ecosystem involve the products, services and their add-on components; key people involved in the principal activities; the platforms used for the requirements definition, development, integration, testing and maintenance; the distribution channels as well as the assets owned by the organization taking the central role in the development of the products and services (i.e., the manufacturer).

Our study revealed other issues which are more related with the intellectual properties and human resources, such as, the hardware and software constructed within the ES domain and the know-how around them. However, in the envisioned scenario of FES that motivates our study, additional intellectual resources, such as, brands, patents, and standards are of key importance. The principal challenge is best illustrated with the following statement from one subject: "How should one be able to protect intellectual property of the add-on software that is installed?"

#### 3.7.1. Development environment

Currently, the manufacturers in our study provide ES packages containing both hardware and software, where the two are closely linked to each other. Most manufacturers open up their products to some degree for their customers, providing opportunities for black-box configuration and programming. However, they rarely open up for third-party extensions, for example for add-ons. One of the problems they describe, may be illustrated by the statement: "an add-on software developer needs access to some kind of representative hardware for testing". In the IT domain, systems mainly comprise of software, which is easy and inexpensive to produce and transport. It is also easy to provide connectivity using, for instance, a network to connect to another remote system. However, for ES the custom hardware and even mechanical parts add some complexity and are also important when it comes to considering development, integration, testing and maintenance. In particular, it is required for testing by a third party, and is expensive to reproduce and transport. This is considered an impediment for opening up to small niche players, as the up-front investments are quite high, especially when considering scenarios where they would like to provide integration of products from several manufacturers.

#### 3.7.2. Human capital

Several companies in the case study include services in their value propositions. This implies that human resources and assets, such as the experience and know-how, are key resources for a company. A company's collective experience from deploying their products in a specific domain is transferred to clients. From the company perspective, their solution, including products and services, is applied on a particular problem. Opening up products would pave the way for niche players that approach the problem from the opposite perspective, i.e., from the problem domain. They can integrate and combine hardware and software from several players targeting an optimal solution for a specific niche problem. This will also increase the market competition, so that all players would benefit from it in the long-term.

However, open ecosystems may pose threats to players having as asset highly skilled staff. It would create several opportunities for key employees to set up their own startups or leave the ecosystem for collaborating with other external parties. The consequence is a flight of human resources (brain drain) that may constitute a severe risk to a company. Another risk is the flexibility induced by the openness that may cause quicker and more frequent shifts in market trends. As highlighted in one of the interviews "it is a challenge to try new business models and adapt the organization". The players must invest to keep up with the pace if new products and add-ons appear more frequently on the market. Activities related to knowledge management and training become key for the ecosystem actors.

#### 3.7.3. Branding

Another important issue regarding the key resources is branding. Consider a situation today where one company provides both hardware and software. The company uses displays and possibly the hardware for branding, for instance, providing a company specific user interface and logotypes. This causes several challenging problems in an open-market scenario. As expressed by one of the niche players in the study: "would it be allowed to show our logotype in the display of the product"? In some situations the answer is no, in others it is yes. Sometimes, "if a third-party developer has a very strong brand, they could be allowed to show their logotype in the display of a manufacturer's product".

#### 3.7.4. Standards

Things like branding and patents are important to protect resources of a certain actor, but an alternative route is to standardize different parts of the system instead. This can be either carried out through established standardization bodies, or using de facto industry standards based on one manufacturer's (or a group of actors') solutions that are adopted by other actors. The need for standards is expressed by representatives of many actors in our interviews, including manufacturers, suppliers, add-on suppliers, and also indirectly through customers. "Customers are complaining that they need to invest in expensive add-on equipment with uncertain value, and they are pushing for more standards." However, it is far from trivial to develop standards, in particular in an area which is emerging and has not yet matured. The innovation potential in FES is high, but standards can easily become conservative and restrictive, and the process of developing a standard can take very long time.

A conclusive statement from the analysis regarding key resources is that the challenges lie in finding a balance between openness and protecting intellectual properties. The overall value of the key resources may be increased if they are integrated with other resources. However, the common viewpoint is that the associated risks are more prominent and difficult to deal with.

### 3.8. Key Activities

Key activities in the BMC capture the most important things a company does to make its business models work. The main activities that need to be carried out include defining the product and its architecture, and developing, integrating, delivering and maintaining the product and the services around it. With respect to the business model other activities include building a stable collaboration network, offering services such as consulting, data trafficking, security, marketing, sales, and supply network support. In this section, we will focus on production related activities, and activities for maintaining product line components to preserve their competiveness.

#### 3.8.1. Production

The manufacturers in the study typically own the product from the phase of conception until production. They design a product and manage the production and target to optimize both accordingly. In the envisioned scenario of FES, hardware and software

may be offered by third parties, which could cause some implications, as raised in the interviews. However, "add-on equipment with physical components could be fitted on the production line of the manufacturer, or at a separate company". In the more mature domains in this study, like the automotive, that is often common practice. Vehicles are equipped during production or before delivery. For example a truck-mounted crane is often ordered together with the truck, delivered by a third party, and installed by the vehicle manufacturer during production. This implies that the product architecture and production activities are prepared for the add-ons. One such example is when a truck provides an interface that an add-on may utilize to enable the truck driver to operate the add-on from the truck cab.

The manufacturers in the study represent a mature domain with respect to production. Activities for integrating hardware add-ons on key resources are commonplace practice. However, opening up the activities in respect to software resources is considered challenging. Companies are not using software to the same extent and have instead relying more on electro-mechanical and hydraulic interfaces. Software integration is new to these companies and thus it poses several challenging activities that they have not yet addressed.

#### 3.8.2. Product line management

The interviews depict a current landscape where key resources are proprietary and manufacturers have full control over all aspects, including management and evolution. "The market department almost always defines new products and features based on customer contacts." When forming ecosystems for FES, activities related to evolution require more extensive coordination among the actors. Changing or removing Application Programming Interfaces (API) and similar activities may have dramatic effects on other parties in the ecosystem and could even result in niche players being forced off the market. This is the technical perspective on the discussion included in Section 3.5.1. However, open ecosystems would introduce a possibility for manufacturers to reduce risks concerned with evolution, given that third parties will take on that responsibility to some extent. This will reduce evolution-related costs and could have an indirect effect on revenues.

A question raised in the interviews is how open the evolution will be. "How difficult will it be to get permission from the manufacturer to add software?" This implies that some manufacturers hold a reluctant position that may possibly hinder a flourishing ecosystem. It is clear that management and evolution activities, to some extent, must be opened up and should be appropriately coordinated among the actors. As a consequence, actors may create new value propositions that generate new revenue streams, for instance, developer licenses that provide guarantees with respect to support.

A recurring concern in the interviews is the issue of assurances. Verification and validation (V&V) within the process are considered key activities. The interviews have described several scenarios and models for this, ranging from offering V&V as a service to third parties, and third parties performing V&V themselves. It is clear that this area requires a more in-depth analysis, given the complexity of the activities and the degree of V&V criticality for product lines.

Manufacturers and third parties also perform brand-related activities, including market analysis, campaigns and rebranding. In mature domains, these activities are already well-known and well-coordinated, with respect to hardware. However, when opening up products and markets to other key actors and niche players that enter the ecosystem, the landscape is bound to change dramatically, and thus, all of these activities will need some degree of support and re-definition.

### 3.9. Key Partners

The Key Partners building block of the BMC represents the network of actors that make the business model work. The key partners in the ES ecosystem have already been mentioned in Section 3.1 above. In this section, we will mostly focus on their relations and issues associated with this.

#### 3.9.1. Traditional supply chain

Just as for traditional ES, FES will also use a basic supply chain where the primary actor is the *manufacturer*, responsible to manufacture and sell products. Often, manufacturers are also responsible for the product integration. The main partnerships of the manufacturer are formed with *suppliers* providing components to be integrated in products, and with *content providers* providing data to different actors. The content provider may be the one responsible for also distributing the data, or might have another partner to provide the required connectivity (e.g., a telecommunications operator), for transmitting the data. In general, the core partner network composed from these four partner types (manufacturer, supplier, content provider and communications provider) is typically controlled by the manufacturer.

#### 3.9.2. New actors

In a more open business model, new actors appear and the traditional supply chain is not sufficient, but further partnerships can be formed dynamically. The analysis of case studies from large enterprises is an emerging need, while also transferring lessons to small and medium enterprises is a challenge in the field. The operation details of such a business model are complicated, because it expands all current operations of actors and also new actors are entering in the model. Adapting the organization to becoming able to receive them is challenging for several reasons. First of all, the span of the model, human capital and operational management difficulty increases. This brings in challenges of who is in control and what are the architectural and infrastructural needs to support the business model. Secondly, accumulating efforts to direct services and their revenues, especially when they are entailing many international or transnational organizations can be quite complicated.

In this model, new services are developed and also new actors emerge. One new partner is the *service operator* with the ability to operate new services related to the products. The products might even not be originally offered by the base product, but emerge or are extended later on, with new features. In addition, other new actors are *add-on developers* who develop software and hardware that can be integrated into the base product. They are typically younger and smaller organizations than manufacturers or suppliers, and are concentrated around ideas to satisfy a new market segment, create a new innovative product or component, create a better value, extend collaborations, offer a new service, and utilize latest technologies.

Add-on developers play a secondary role in the ES ecosystem today; however, they have a prospect of rapid growth by attracting new customers and partners. These organizations face the following challenges: identifying what is not currently offered by large manufacturers; deciding where to invest (e.g., in products, components or services with relatively low profitability for the manufacturers, where the competition and risk is lower); and facing high cost barriers for entering the ecosystem. Manufacturers that participated in our study mentioned that sometimes they decide to leave a market segment to other players. This is usually because the cost of maintaining expertise in a particular domain may be too high compared to the revenue it creates. The manufacturer thus sometimes instead focuses on supporting players that could

act in that segment. They trade the risks related to market presence for a reduced share of the revenue stream.

### 3.10. Cost Structure

The Cost Structure component of the BMC describes all costs incurred to operate a business model. In the interviews, four general groups of costs were discussed, namely development costs, product costs, operating costs, and information costs.

#### 3.10.1. Development cost

For manufacturers, the development costs do not appear to be radically different for FES compared to traditional ES, since it is basically a matter of providing some additional capabilities in the platform, which is anyway something carried out once. However, for add-on developers, development cost is a larger issue. In many situations, add-on developers see the need to adapt the same add-on to different platforms of several manufacturers in order to amortize cost on a larger user base. "The effort to adapt a solution to the base system of different manufacturers is as large as developing the functionality." In particular, this is an issue when there is a Human Machine Interface (HMI) involved. It also leads to an enormous problem with variant handling, and evidence of this can already be seen in the smart phone industry, where certain apps exist in many hundred variants for different phones. Therefore, it is essential to provide reuse possibilities for the software components that are used to build add-ons. In addition, there might be sometimes a need for rather expensive hardware to stand as a representative test environment (see also Section 3.7.1), and therefore, manufacturers need to consider how they can better provide shared facilities for testing purposes in the ecosystem. On the other hand, by having a shared infrastructure for the different channels, economies of scales in the server side systems outside the physical products can be achieved.

#### 3.10.2. Product cost

Even though manufacturers are not so concerned about the development costs for FES, they are the more worried about the product costs. Flexible solutions, like add-on software, will certainly require surplus capability in the hardware (e.g., processing power, memory, input/output), especially if graphic displays are involved since they require large volumes of data. Many of the manufacturers are traditionally very cost-focused organizations, and they find it difficult to discuss adding this flexibility in their business cases. This may considerably slow down their transformation from product-centric towards service-centric companies. For the user, on the other hand, physical add-on equipment is usually much more expensive than an integrated software solution that uses sensors and actuators already present in the product.

#### 3.10.3. Operating cost

The most evident operating cost is the cost required for data traffic, which is in particular a concern for mobile ES that need to rely on wireless telecom services, sometimes involving coverage costs and expensive roaming when traveling to different countries. Different models for sharing these costs between the actors are discussed in the interviews. One possibility is to have all costs included in some higher-level service agreement between the manufacturer and the owner. Another possibility is to have users pay for data traffic directly related to the user features, whereas the manufacturer pays for traffic related to services, e.g., diagnostics and field data follow-up. A third possibility is that the owner covers all costs directly by providing their own telecom subscription. There is also a direct relationship with the operating costs and the quality of service, since high-quality services will require more frequent and accurate data updates, leading to higher bandwidth consumption.

#### 3.10.4. Information cost

Another category of costs discussed in the interviews is the information costs. In the future, users will probably expect something in return for letting anyone take advantage of data originating from the systems they operate, either monetary revenue or new or improved service offerings. This could result in some cost for example to a manufacturer who is using such data for obtaining feedback about their products and services in the field, or an

<table>
<tr>
<td rowspan="2">Key partners<br>• Manufacturer<br>• Supplier<br>• Content provider<br>• Communication provider<br>• Service operator<br>• Add-on developer<br>• Owner<br>• End user<br>• Regulatory agency<br>• Information broker</td>
<td>Key activities<br>• Production<br>• Product line management</td>
<td rowspan="2">Value proposition<br>• Service types<br>○ Product<br>○ Process<br>○ Life cycle<br>○ Extended<br>• Open interfaces<br>• Quality attributes</td>
<td>Customer relationships<br>• Product life cycle<br>• Liability<br>• Ownership of information and privacy</td>
<td rowspan="2">Customer segments<br>• Broader customer reach<br>• Improved customer feedback<br>• Customer differentiation</td>
</tr>
<tr>
<td>Key resources<br>• Development environment<br>• Human capital<br>• Branding<br>• Standards</td>
<td>Channels<br>• Sales<br>• Distribution<br>• Configuration<br>• Information<br>• Channel ownership</td>
</tr>
<tr>
<td colspan="3">Cost structure<br>• Development cost<br>• Product cost<br>• Operating cost<br>• Information cost</td>
<td colspan="2">Revenue streams<br>• Volume increases<br>• Recurring sales<br>• Direct sales of software<br>• Subscription fees<br>• Revenue sharing</td>
</tr>
</table>

**Fig. 3.** BMC summarizing the key elements of the analysis.

independent service operator who is aggregating data from many users to provide a new service.

### 3.11. Summary

To summarize the analysis using the BMC, the key elements of each category are shown in Fig. 3.

## 4. Discussion

The analysis based on the BMC provides sufficient information to address our business-oriented RQs, and especially the RQ1 and RQ2a related to actors and business models. In our research, we are also interested in what are the particular implications of FES on architectures, processes, methods, and tools, as indicated in RQ2b-c. In this section, we revisit our RQs to discuss and summarize our findings and identify a number of critical challenges. We also discuss some overarching challenges that cut across the key elements of the ecosystem.

### 4.1. Ecosystem actors and relations

Our first RQ is concerned with the ecosystem's actors and their relationships, which constitute the business architecture backbone. Sections 3.1 and 3.9 discuss actors and to some extent their relationships. One observation from the interviews is that a traditional supply chain with manufacturers and suppliers will still be at place as the core. However, we have identified developments involving existing and new actors that are likely to extend this structure and introduce new challenges to the ecosystem. These actors are the add-on developers, the information brokers, and the regulatory agencies (presented in Section 3.1).

#### 4.1.1. Challenge – Explicit and implicit relationships

The *manufacturers* and *suppliers* are key actors in the traditional supply-chain based ecosystem. In such ecosystems the relationships are well defined and explicit, and therefore relatively easy to trace and understand. For instance, when the suppliers and a manufacturer have agreed to a particular standard all parties are aware of that agreement. However, the introduction of add-on developers and information brokers in the ecosystem would unavoidably shift this balance. A typical scenario for an add-on developer is "providing a new product through integration of two or more existing products". In the scenario, interactions with other actors use well-established, explicit relationships. However, some explicit relationships may become implicit for the add-on developer, and as such, they are more difficult to identify and therefore they constitute a potential risk, if they are not properly managed. One such risk may occur when an actor integrates systems without direct involvement with the sub systems' original manufacturer or suppliers. The subsystems may for instance have dependencies on artifacts provided by a regulatory agency. These artifacts provide specific rules and regulations that may be violated by the integration if their results are not known by the integrating actor. The challenge is to make all critical relationships explicit for the actors in the ecosystem.

#### 4.1.2. Challenge – Flexible authority

In the interviews, several example scenarios were described where the ecosystem's authority was granted to several actors, or even distributed across several actors, and this was changing over time. Authority in this case, refers to the form and responsibility of control for the ecosystem products and information artifacts, channels, etc. The add-on developer and information broker actors are two concrete examples that take over authority from manufacturers and suppliers; add-on developers take over control when they are integrating systems or adding functionality, and information brokers when they are aggregating and redistributing information. The challenges are among the developing models and protocols managing these phenomena. This requires a better understanding of the underlying rules that govern these processes and how the models should be instantiated for a specific ecosystem.

### 4.2. Business and technical architecture

With our second RQ we target additional key elements of the business (RQ2a) and technical (RQ2b) architectures and their relationships. The case study participants described their current architecture, and discussed issues and concerns from different points-of-view and at different levels of abstraction. A general observation is that the concept of architecture spans the range from high-level business architecture involving components, such as business models, to low-level components that represent a block in a control algorithm.

The analysis of the collected material identified a number of key elements as reported above in Section 3. We have also identified four key challenges related to elements in the architectures and their inter-relationships:

1. Business and technical architecture alignment, which is concerned with a comprehensive architecture that covers all aspects of the company's business and key resources.
2. Openness, which deals with the extensibility of products and processes, including management and the technical aspects involved.
3. Assurances are to a large extent cross-cutting the former categories. They are concerned with how product, service and process qualities are achieved and what type of guarantees regarding the respective qualities are derived and how.
4. Configuration is concerned with product instantiation and configuration that involves multiple stakeholders.

#### 4.2.1. Challenge – Business and technical architecture alignment

Several case study participants expressed that the architecture was the key success factor for their business. However, their mapping of business to processes and products posed a great challenge to them. A direct quote from one subject captures the essence: "What should the architecture and infrastructure be like to support the business model?" This enterprise architecture challenge is well known and has been addressed in research and practice for many years now. However, in the envisioned landscape which introduces more openness and integration, add-on developers add to the complexity of the problem and impose required re-factorings of the enterprise architecture. Statements like "the complete offering only considers the company's products. No specific support is included for third-party products and services" illustrate this. The business and product architectures are not prepared for open business models.

#### 4.2.2. Challenge – Openness

The challenges when moving towards more open processes and products that integrate with others were highlighted in several interviews. Subjects described it as: "Many ES are closed products under the full control of the manufacturer. How can they be opened up?" The subjects describe their current architectures as mature and the software architecture's component view is a direct mapping of the deployment architecture. The architectures define concrete subsystems and the interactions in-between subsystems are kept at a minimum. The primary cause is that it reduces the V&V effort.

Most subjects discussed openness at the technical architecture level. The subjects also talked about several risks and mitigation strategies in relation to this. When investigating the rationale for some of the risks we discovered that most were related to control and trust. A majority of the subjects are active in closed and regulated domains, where system behaviors and interactions are negotiated and stated in contracts or are regulated by standards and laws. In this landscape, it is understandable that actors are reluctant to open up, most likely due to the overhead it will introduce when the control over the complete supply-chain is decentralized and also the uncertainty with respect to conformance. Several subjects expressed concerns with respect to control and governance risks. However, they explained how they consider most of them as technical risks and described strategies for mitigating some of them. One strategy was to introduce a classification of architecture subsystems that explicitly expresses their degree of openness. This would reduce the technical issues, and as a consequence, also mitigate risks connected to trust and control. Several participants also advocated standards to regulate openness: ''there is a need for industry standards when it comes to handling add-on software and other additions''.

#### 4.2.3. Challenge – Assurances

The concerns expressed with respect to openness are instantiations of the principal challenge in the domains where the case study subjects are active, namely safety. This challenge permeates the companies' architectures and explains the rationale for their closed processes and technical architectures. In these domains, owners and regulatory agencies require guarantees that the subjects' products behave according to specification. The manufacturers and add-on developers address this with determinism. Specific output as a result of specific input to processes, products, and their constituents must be predictable. However, opening up would introduce an element of non-determinism that reduces their ability to verify and ensure certain behavior, i.e., reducing or sometimes removing predictability.

One such example is mixed criticality, where some sub-system should behave according to one profile while another profile applies to other sub-systems. One subject stated, ''real-time properties are also important but more easily managed''. They mitigate risks related to real-time constraints by local control. This reduces the problem to provide assurances. However, if systems open up, then that same component may be used in another context, according to another profile where it is controlled externally. This introduces uncertainties related to communication, and predicting real-time behavior becomes a much harder problem since the assurances provided are no longer valid. The subjects discussed some strategies to mitigate these risks. Most of them were stating the same principles, i.e., more standards, more negotiation and more governance. One statement illustrates the subjects' views on this: ''software for embedded applications requires safety, and can never be as open as it is for smart phones.''

#### 4.2.4. Challenge – Configuration

Several subjects describe that their products allow end-user configuration, including integration with add-on developer products and programming or configuration of specific components. A more open setting would provide end users and integrators with even more powerful means. For instance, the possibility to develop add-ons would improve the current situation considerably by introducing more flexibility. It would ''be useful to adapt their system to the particular context and be able to communicate with the other systems in that context''. This illustrates that manufacturers and add-on developers would benefit from more open architectures and improved configuration support, as it will improve the products' extensibility and flexibility.

### 4.3. Processes, methods, and tools

The focus of RQ2c is processes, methods, and tools, which form the infrastructure for how products are managed during the entire product life cycle. It captures practices, guidelines, and standards in an accessible format that is used to coordinate and perform work within an organization. In a federation of ES, coordination of processes poses a great challenge. Coordination is required in-between different actors and the actors' processes. The possibility to add hardware and software to a product increases the complexity of the product's life cycle as ''there are [at least] three cycle times to handle: One for the physical product, one for the software platform, and one for add-ons [...]''. In an ecosystem, we will also have dependencies between different products and services, and their life cycles.

#### 4.3.1. Challenge – Quality assurance

Connecting with other actors in the design and production processes will require processes to be integrated to a large extent. Such integration might be implicitly or explicitly defined and described, and address internal or external integration. In the latter case, process standards must play a prominent role. For example, ''manufacturers claim they need to develop add-ons themselves to ensure quality and a get a consistent HMI''. This illustrates a situation where manufacturers and add-on developer processes should be integrated and possibly governed by standards. The motivation for such integration is product quality assurance, which is critical in the domains under study.

V&V is more complicated in a FES when compared to pure software ecosystems. Products consist of a mix of the manufacturers' hardware and software in combination with add-on hardware and software from possibly several add-on developers. They also might have interactions with other external and internal systems, forming a SoS. The safety–critical nature of most applications is the key contributor to the problem. The Directive on Machinery issued by the European Union [10] is an example of how to regulate development and testing of safety–critical applications. This directive (or similar documents) forms a basis in most domains for governing integration of add-on developer's parts in products or add-ons. Some subjects expressed that ''some sort of certification is needed for add-on software''. There were also examples of how to integrate processes and certify add-on developers' parts. For example, ''developers of [add-on] software are given an API, a test environment, and then the manufacturer certifies the software''.

''An add-on software developer needs access to some kind of representative hardware for testing'' and at least ''a simulation environment is necessary for third-party developers to test their plug-in modules''. The challenge of testing software on dedicated, possibly expensive hardware is addressed by simulation and emulation. However, several subjects describe a situation where the manufacturer may use hardware or test environments to control the market and decide if a market segment is open or closed.

#### 4.3.2. Challenge – Updates

The second area that several subjects discussed is updates. Updates also require extensive coordination, especially with the V&V processes, and this must be reflected in processes, methods and tools. ''There is a need for an updating mechanism for installed add-on software''. However, this update mechanism is not just about distributing and installing software updates. It must also make sure that the software being deployed is verified and validated, possibly certified to function with the installed hardware and software, which may include a set of software add-ons that have been developed and added to the system after its initial deployment. In addition, it must deliver similar assurances regarding reliability, stability, and performance as the ES functions.

This exemplifies the complex web of developing process interactions that is required due to various quality assurance standards to guarantee safety. Managing these interactions is a key, and risks are mitigated by standards. This is probably why "things that are standard tend to survive".

### 4.4. Ecosystem challenges

In the analysis and discussion we have presented a number of challenges, most of which relate to opportunities and problems of the introduction of FES. We will now summarize what we consider to be novel overarching challenges at the FES ecosystem level that need to be addressed both by research and practitioners.

#### 4.4.1. Challenge – Ecosystem management

A thriving ecosystem will most likely not emerge by itself, it will require stimulus from key actors, either manufacturers that create ecosystems around their own products, or possibly industry consortia that agree on certain common rules and standards. In FES it may be more challenging to identify key actors and the role as key actor may shift, for instance, from manufacturer to supplier for one value proposition, or from manufacturer to information broker for another proposition. To achieve *sustainability* in the ecosystem, finding ways of sharing both revenue and cost is essential, and this includes cost for products, services, and add-on software, but also for data where the functioning markets need to be organized. Finding a stable *architecture* is also critical, and this is not limited to the technical architecture, but also adheres to the enterprise architectures and all actors involved. The *open interfaces* play a central role. However, openness that can lead to new functions and services must be balanced against the dependability of the base system. Good interfaces are necessary, but not sufficient, to stimulate *open innovation*, and other means also need to be installed, for instance, support channels, meeting places for the actor communities, etc.

There appears to be significant differences in focus between consumer-oriented products and business-to-business products. For the former, customer features are mostly important, whereas for the latter, ways of improving efficiency matter more. Possibly, this could affect how ecosystems are managed. Another critical force that has an indirect effect on how the ecosystem may be managed is the regulatory agency actor. This actor provides rules and regulations that will affect parts of or the entire ecosystem, how actors behave, their relationships and obligations to other actors, etc.

#### 4.4.2. Challenge – Product development

Once the ecosystem is in place, support is needed to develop the products. The challenges we identified include several that stand out as unique for the FES domain. The product domains of FES typically impose more strict requirements on processes, methods and tools. These requirements are often regulated and controlled by an outside authority. This has an effect on several actors, for instance, add-on developers need to have a *development environment* for the products of a particular manufacturer, and ways must be found where such an environment of sufficient quality can be created without imposing heavy costs for the manufacturer and in compliance with adequate regulations. In many cases, add-on developers will make versions of their software to fit several manufacturers' products, and this means that strategies for managing product lines are needed, both by the manufacturer and by the add-on developer, to minimize the adaptation cost. This again poses novel challenges with respect to relationships to different regulatory agencies.

Another challenge that contains unique novel aspects is quality assurance. In FES ecosystems the responsibility for *V&V* is in some way shared by manufacturers and add-on developers, but the exact distribution of work between the two needs to be sorted out in each case. The same holds for the manufacturer-information broker relationship. V&V also include issues of *liability*, i.e., who is in the end to blame if something goes wrong. In a *mixed criticality* environment, where safety–critical applications share a platform with other software, this is particularly important and constitutes an even more complex challenge as it varies from one product to another.

#### 4.4.3. Challenge – Ownership and control

The *channels* for add-ons and information will be central in the ecosystem, and the ownership and control of those channels will give power to the actors who possess them. An open environment will make it harder to protect the *intellectual properties* of different parties, and either mechanisms have to be in place for this, or the actors have to accept the fact that this risk will not cease to exist, and assume that the benefits from participating in the ecosystem outweighs this drawback, as many actors within the open source software domain have already chosen to do so. Finally, there will be sensitive information about individuals and companies flowing in these systems, and adequate protection of this information, in respect to the *privacy* of individuals, anonymity and confidentiality, needed to gain *trust* in FES.

#### 4.4.4. Challenge – Support for open innovation

One long-term goal for our research is to investigate and understand how to best cater for open innovation in the ecosystem of FES. Several of the challenges described above include problems and opportunities that require further investigations in this respect, as also highlighted in open software innovation [11,12]. Open innovation requires openness at the business and technical architecture levels. The emergence of new open business models needs to be supported with new roles and patterns of collaborations to cater for innovation and transparency [11]. Software ecosystems, conceived as networked organizations, are considered to contribute in understanding this new community aspect, extending across many (traditional) organizations [11]. The notions of openness with the adjustment of the smallest amount of innovation are enough for offering the competitive advantage to organizations that can effectively harvest its benefit [12]. However, several impediments that are unique to the FES exist. These impediments are a consequence of the products' domains and the requirements imposed by these domains, for instance, safety requirements and certification of system behavior by external bodies. The principal challenge to establish provisioning for open innovation in FES ecosystems translates into finding the right balance of management, development, ownership, and control in the ecosystem. A balance that makes it feasible for new and existing actors to extend the ecosystem with new value propositions and establishing sustainable revenue streams while fulfilling domain-specific requirements.

## 5. Validity

We will now discuss the validity of the findings presented in this paper. A number of validity threats exist in case studies, and following the categories of [5], the threats in this case study are reported below, including some more detailed threats based on [13].

### 5.1. Construct validity

The construct validity ensures that the construction of the study actually relates to the problems stated in the RQs, and that the chosen sources of information are relevant. A specific threat to construct validity is the use of unclear terminology, which is

certainly an issue when dealing with emerging concepts like FES. To reduce this threat, we have used the established BMC framework for structuring the analysis of the interview data and also, we have made sure throughout the interviews that a common understanding of the meaning of the terms used existed. Another possible threat is that the respondents might have guessed what hypothesis the researchers had in mind, and adapted their answers accordingly, for instance by exaggerating their opinions in an attempt to try to influence the outcome of the study. We tried to reduce this threat by using open-ended questions in the interviews conducted and by not explicitly discussing the hypotheses with the interviewees.

There is also a risk that the analysis is influenced by the researcher's expectations, so that the researcher only sees what he or she already believes. To reduce this risk, we tried not to invent our own structures of classification, but instead use an established framework (the BMC) and use open-ended questions. Also, the statements and findings were cross-checked by at least two independent researchers. Finally, respondents could be hesitant to express their views if they could later be affected by the responses, and to handle this risk, the anonymity of both respondents and the participating companies was guaranteed.

None of the respondents had dependencies on any of the researchers or the research outcomes. However, as pointed out in Section 2.2 above, there were established relations between the researchers and interview subjects beforehand. This is thus a potential source of bias, but also a facilitator for finding the right people to talk to, offering the appropriate level of trust that allowed an open discussion to be carried out, and being able to interpret the results correctly due to an extensive contextual knowledge already in place. One interview subject even stated explicitly that "since I know you, I can speak freely, trusting that you will understand what is confidential information and keep that to yourself."

### 5.2. Internal and conclusion validity

Internal and conclusion validity concern the possibility to ensure that the actual conclusions drawn are true. In [5], it is stated that "internal validity is only a concern for causal (or explanatory) case studies". This case study is highly explorative, and hence less sensitive to this threat. However, there are still issues that are relevant to mention.

One issue is the selection of respondents, which was made to ensure that they have the right knowledge about the context of the research and the challenges around it. By relying on already established contacts and having some prior knowledge about these individuals, this risk was minimized, and the long experience in the business area and in the specific company of all participants provided the possibility for gathering a rich variety of data.

The sample size of the individuals involved was relatively small, and does not allow for any deep quantitative analysis, but with a stratified selection, we ensured that many of the actors in the ecosystem were covered. The mortality was reasonable, with two persons opting not to participate in the interviews due to time restrictions.

Another risk is related to "fishing", i.e., that the researchers consciously or unconsciously search for certain answers. We tried to avoid this by having as open questions as possible in the interviews, and by finishing each interview by asking if the respondent felt that there was anything else that should be brought up and discussed.

### 5.3. External validity

External validity concerns how the results can be generalized. This is a specific concern for a case study, where it can always be discussed to what extent the observations are particular to a certain environment, or whether they are examples of general phenomena.

The primary type of external validity is whether the conclusions can be generalized to a different organization, either within the same or in a different industry. In our work, we did include representatives from several business sectors and companies and we believe to have captured many characteristics that are general to the problem domain. Still, we cannot with certainty say that this is the case, and to enable us to draw such conclusions, further studies are needed.

### 5.4. Reliability

Reliability relates to the ability of others to replicate the study and arrive at the same results. A basis for replication is to have a well-documented study design and well-structured data collection processes, and we believe that this is the case for the study presented here. Assuming that the study was replicated and resulted in roughly the same transcripts of the interviews, it would still not guarantee that the resulting issues would be the same. There are different ways of interpreting the textual material, and in some cases there could be several ways of grouping statements resulting in different sets of abstractions. To reduce this risk, we had two researchers doing the classification of statements collected independently.

A particular threat to validity in this study is how the data was collected during the interviews. In 10 out of 12 interviews, handwritten notes were used, and in the remaining two, a recording equipment was used. There are pros and cons of both approaches. Clearly, there is a risk of missing important information when not recording interviews. This risk was reduced in the non-recorded interviews by transcribing the handwritten notes directly after the meetings, and one of the reasons for actually recording the remaining two interviews was to handle the risk that the period between interview and transcription could become a little longer in those two cases. However, a drawback of recording equipment is that some respondents may become hesitant to speak freely. In general, in this study, the topic is very central to the future business strategies of the companies, and thus the risk was that some information would be held back and thus reducing the value of the whole study (as discussed in Section 5.1). In our work, we tackled this risk by making the interviews as informal as possible through appropriate ways of communication and interaction with the participants.

## 6. Related work

This section describes related work on software ecosystems and initially provides descriptions of previous attempts in defining the terminology emerging for the domain as it is highly relevant to FES ecosystems. Secondly it provides an analysis of related case studies to this work that emphasize challenges and characteristics of software ecosystems and in particular the evolving domain of collaborative software development processes, methods and business models. The contribution and relation of the prior work to ours is to identify the background material and definitions found in the discipline of software ecosystems and highlight their related characteristics and benefits.

### 6.1. Ecosystem concepts

In an attempt to clarify definitions in the problem domain we present several definitions found in the related literature including software ecosystems, business models and SoS. Messerschmidt and

Szyperski [3] were among the first researchers to use the terms "software" and "ecosystem" together, in their attempt to expand our current understanding of software and its industry. They refer to a software ecosystem as a "collection of software products that have some degree of symbiotic relationships" and describe a number of factors that they consider to drive software industry's future. They discuss implications on new strategies and business models of existing actors, as well as the possibility of major changes caused on the infrastructure of the industry and other supporting industries, like telecommunications, manufacturing and computing. Other works related to ours, looking at the non-technical aspects of software ecosystems, include the following: Popp [14] introduces the business and revenue models of software ecosystems in the case of embedded hybrid commercial open source business models and discusses how they can create value both for customers and software vendors. However, the lack of sufficient modeling for the domain, the techniques reported by [15] for software ecosystems inhibit existing actors and potential new actors to conceptualize the potentials of an ecosystem approach. Further studies, from an empirical aspect are needed to analyze strategic dependencies between various actors mentioned in [15], i.e., vendors, third-party developers and end users.

Software ecosystems are also viewed as collections of projects that are developed and evolved together in the context of an organization or a development community. As the network and collaborations strengthen and increase, more and more complex interactions are found to be formed among the related players. Also, in situations of high complexity, interoperability between the organizations, external developers and other actors needs to be ensured, and alliances need to be formed to share information, resources and technology efficiently to keep up with this evolution of the traditional software supply network. Software ecosystems offer complementary organizational views to SoS development, introducing between the systems comprising the SoS synergistic capabilities, collaboration and interaction, while new roles and rules are introduced [16]. In the related previous work of the authors [16] formulation of the research of ecosystems for FES as SoS, as well as both the challenges and benefits have been introduced to provide a basis for the on-going research activities of this work.

In the related literature, many researchers expressed increasing interest in this domain and provide detailed analyses. Bosch [17] explains the extension of software product lines to ecosystems, and recommends opening up product line platforms to external third parties. The same author proposed examining software ecosystems from the commercial and the social aspect, while a taxonomy of operating-system-centric ecosystems and application-centric ecosystems is discussed. The definitions of software ecosystems introduce apart from the technical perspective, the social and business aspect, i.e., mentioning that a software ecosystem consists of a software platform, a set of internal and external developers and a community of users that compose relevant solution elements to satisfy their needs [18].

A software ecosystem according to Boucharas et al. [19] consists of a set of actors, acting as a group, interacting and managing a market of shared resources (including software and services). In [20] the notion "ecosystem" is presented in two mutually interdependent units, i.e., one unit consisting of a set of actors (e.g., platform owner, third-party developers, platform partners and platform users) and another unit consisting of a set of technology artefacts (e.g., software platform, boundary resources). Jansen also [21] stresses the interconnection between these two units, the efficient exchange of information, resources and artefacts, and this is confirmed in our study.

Recent literature reviews in the field of software ecosystems [9] provide a list of common items found in patterns in the various definitions in the literature. One of the results of the review is that the wider definitions, referenced by the majority of papers, include mainly the elements of common software development, business in the ecosystem and connecting relationships within the community and actors of domain experts. The systematic mapping in [22] identified characteristics of software ecosystems linked mainly to the business perspective and open source development models, but not all the characteristics cover the ones included in our analysis. A three-tier perspective is analyzed [23], including the organizational scope level (actors and their relationships in the context of an organization), software ecosystems scope level (software supply networks) and software ecosystems' scope level (software ecosystems themselves and their relationships). The main challenges refer to the architecture and metrics measuring software ecosystems' health.

Even though several recent works exist that suggest terminologies and describe the conceptualization of emergent software ecosystems, agreement on the exact relations between the concepts has not been established [11,24]. Moreover, we see examples of researchers trying to understand the interconnections within this field of research by studying cases where, for example, the manufacturer is the keystone player who is responsible for development, integration and maintenance of products [25]. In other cases, alternative dependencies between add-on components and the related players are formed and product development is coordinated by both the team within the company and also by external developers and teams [26]. Finally, there are cases where an open infrastructure is promoted [2,27]. These scenarios of operation of software ecosystems result in complex communities interacting that share resources and interests but also have conflicting benefits to protect.

Comparing our work to other related works several concepts are revealed that are unique in the sense that they have not appeared in other cases. Even though most relationships identified allow actors to contribute to the ecosystem following various models, our analysis includes several forms of sharing and granting the ownership and control, by having regulatory bodies involved in the processes of buying, selling and supporting new forms of collaborations and federations.

### 6.2. Case studies

Substantial related work includes the analysis of case studies of software ecosystems which in general aim to understand the domain and manage the complexity of large-scale product development. An analysis of some of the most recent and interesting case studies will now follow.

Hunink et al. [28] present three case studies of large software ecosystem orchestrators and each of the studies reports the model structure used, benefits, requirements, entry barriers and goals strived for by the model owner. The goals related to the network are emphasized for all three organizations, while the different legal forms, types of platform and their business model are the main reasons to differentiate each respective case. In addition, differences in structure between business models are identified and within the model the participants are found to develop in different ways over time. The membership models found were layered and the partnership model role-based. The main entry barriers identified are related to financial issues and the need to devote considerable amount of resources, participation or entrance fees, a certification process, and a high level of involvement with another existing platform. However, alliances opt to stimulate members' long-term engagement. Similar barriers as the ones described in this study were also discovered in our work.

Bosch and Sijtsema [26] discuss the challenges entailed in constructing a software ecosystem and they identify that problems

primarily stem from the fact that organizations follow an overly integration-centric approach to software development instead of a component-centric approach. The authors also present architectural, procedural, organizational and applicability problems they discovered through their analysis of three companies. The challenges identified relate to coordinating with the teams, life cycle processes, prioritizing, and architecture. The validation of their conclusions is based on several case study companies.

van de Berk et al. [29] present a model to describe software ecosystem characteristics, starting from a business perspective and mapping concepts to a software engineering perspective, in four dimensions: biology, lifestyle, environment and health care. A strategy assessment model is proposed which is also evaluated through a case study. Some issues identified relate to how open design alliances affect insights about the ecosystem in the particular domain, what its main quantitative characteristics are and how stakeholders evaluate the assessment model.

Jansen et al. [30] present two case studies of organizations dealing with software ecosystems and receiving the benefits. In the first case, some of the benefits mentioned are the growing partner network, partner-friendly environment, successful product, business opportunities, more customers satisfied and savings on development costs through reuse. In the second case, the ecosystem created multiple business opportunities to its members (including third parties). In addition, the collaborations formed are expected to survive through technological breakthroughs, internal competition, or other forms of severe competition from other ecosystems. Various viewpoints are described that consist of analysis of the software ecosystem and its relationships from the software vendor level, the software supply network level, and the ecosystem level. According to the authors, relationship definition and identification are both essential to face the challenge of creating a successful software supply network.

Jansen et al. [31] used a software supply networks modeling technique to analyze coalitions formed between participants in a supply network and through a case study they propose a way to perform strategy and risk evaluation in the business of software.

Iyer et al. [32] examined networks of competitions and alliances between organizations of the software industry during the years 1990–2001. Specifically their interconnected activities in marketing, licensing, consulting, data processing, and research and development were analyzed and their findings could serve further exploration of resource exchanges, central roles and structures, such as hubs, brokers and bridges. The series of different software ecosystem models created and analyzed showed that their development over time is worth investigating and could provide interesting insights about the health, life and death of certain technologies, platforms and ecosystems.

Iansiti and Levien [25] mention that in ecosystems, organizations different than the traditional value chains of suppliers and distributors are often used. They propose, through the examples of Wal-Mart and Microsoft, that in collaborative ecosystems, a keystone player emerges, becomes a driving force and provides stability in the environment. The authors emphasize the business aspect of ecosystems and that knowledge is required of the ecosystem, the actors and their roles for the ecosystem to function effectively.

Currently, a range of research attempts are taking place in collecting and analyzing case studies from large enterprises, while transferring research to small and medium enterprises is a major challenge. Another challenge is also to investigate the effects and evolution of community-based software development partnerships in the context of software ecosystems. However, software ecosystems have not matured enough to be fully understood, taking into consideration existing parameters, actors, behaviors, information flows, models, processes, methods, technologies and tools. Thus, in our work, instead of analyzing a selection of organizations that have a specific relation with each other (such as a partnership), we complement previous related work, through an explorative case study based on interviews and investigate from a holistic perspective both the business and software engineering perspective of software ecosystems and their evolution. Primary aim is to keep an open-ended and open-minded approach and study empirically the emergence of ecosystems for FES.

Most of the related work described above focus on software-only systems, and our work distinguishes itself in its focus on ES. Such systems are increasingly dominated by software, but also highly dependent on hardware, and contain other characteristics, such as safety requirements, that lead to further challenges for an ecosystem.

## 7. Conclusions

In this paper, we have described a case study of the ecosystems for FES. The basis for the analysis was a series of interviews with representatives of the key actors in such an ecosystem. In all our interviews the discussion led to a clear indication that the benefit from letting ES connect and allowing dynamic software add-ons is very interesting for the industries included in the case study, especially when it comes to providing new functionality and services, reducing time to market, adapting to niche segments, and improving innovation through working with external partners. However, the interviews also clearly confirmed our hypothesis that it is not sufficient to only solve the technical challenges with FES, but that business related issues are at least as important and demanding. A more comprehensive view of our findings relates to our hypothesis and is thus considered of special interest.

### 7.1. Summary of findings

In the paper, two RQs were posed and we will now briefly summarize the answers to them.

1. *What actors are needed in an ecosystem of Federated Embedded Systems (FES), and what are their relations?*

As it was presented in Section 3.1, a large number of actors are present in the ecosystems of FES. The actors cover the traditional supply chain of ES, including suppliers, manufacturers, owners, and end users. However, many new actors are added, such as content providers, communication providers, service operators, add-on developers, and information brokers. The roles and responsibilities of each actor also changes. As a striking example, the manufacturer is no longer responsible for the final integration of the product. Instead, the owner or user adds additional software to the product, from third-party add-on developers that might not even have formal relations to the manufacturer. This leads to many challenges related to development methods, liability, etc.

2. *What are the key elements of (a) the business models, (b) the product architecture, and (c) the processes, methods, and tools, needed to make the ecosystem effective?*

The business model of a FES ecosystem was analyzed using the BMC, and the analysis showed many new ways of doing business compared to what is traditional for ES, including both new value propositions from manufacturers and other actors, as well as adapting to new customer segments. The revenue streams for non-traditional ES particularly include many new ways of selling software products and information. All in all, the application of FES could in many ways reshape the ES industry.

It is clear from the study that the architecture is a key success factor for FES, and it is not limited to the ES architecture of the product, but to the overall enterprise architectures of the actors in the ecosystem. The architecture also reflects some of the important trade-offs that are a consequence of the FES concept, such as openness vs. dependability, which is due to the safety–critical nature of many ES, and the need for hardware cost efficiency. All these are factors that differentiate the FES ecosystem from pure software ecosystems.

Finally, the processes, methods, and tools for developing ES need to be revised as well, and in particular the interplay between manufacturers and add-on developers must be defined. This includes both what support an add-on developer can expect from the manufacturer, and what the responsibilities are for ensuring quality of the end product. To some extent, manufacturers will try to retain control, but this will again be a trade-off against the openness of the ecosystem.

### 7.2. Future work

From the data collected in this study, many important challenges can be identified that provide topics for further research, including the following:

- Development of the technical mechanisms for handling plug-ins, in accordance with the requirements identified in this study, such as dependability, cost, etc.
- Conduct further case studies based on the conceptual model generated in this research, to assert to what extent it can be generalized to other industry segments and to understand important limitations.
- Characterization of an appropriate software architecture, e.g., how to structure the API, how to deal with conflicts between plug-ins, and how to efficiently create software systems on the federation level.
- Experimental verification of the open innovation aspects, by development of add-on software in a controlled or semi-controlled environment.
- Refinement of the BMC by creating an individual BMC for each actor type, and make the links between them more explicit.

## Acknowledgements

The authors would like to express their thanks to all participants in this study. The research was funded by VINNOVA, the Swedish Agency for Innovation Systems (Grant No. 2012-03782). The work of Dr. Papatheocharous was carried out during the tenure of an ERCIM "Alain Bensoussan" Fellowship Programme. The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under Grant Agreement No. 246016.

## Appendix A. Interview checklist

This appendix presents the checklist used during the interviews. Note that it was not used as a questionnaire, and thus it does not reflect the flow of the interviews conducted. Rather, interviewers used it as a way of ensuring that all important topics were covered at some degree and at some point during the interviews. The interviews were mainly carried out as open dialogues.

A. **Preliminary data**
1. Briefly describe the background of this research and purpose of the interview.
2. Discuss the interviewee's role at the company.
3. What are the total number of years working in the industry and number of years in the company?

B. **Roles and relations in ecosystem**
4. What type of external organizations do you interact with (i.e., Original Equipment Manufacturers (OEM), suppliers, add-on developers, service providers, others)?
5. What is your motivation for working with these partners?
6. Who typically initiates the interactions around an idea of a new product? You? A partner?
7. Have you experienced any conflicts as a consequence of working with external partners?
8. How do you structure your products to work efficiently with partners? Product lines?

C. **Business models**
9. What value do you offer to your customers, e.g. what need is satisfied, or problem solved?
10. Do you consider satisfying any particular customer segments, or types of customers?
11. In what form do you get revenue from the customers? Per installation, per use, for free, etc.?
12. What are the business models used in the interaction with these organizations, i.e., how do you share revenues and costs with them?
13. How are products and add-ons sold and distributed to the customers (end-users)? Are joint or separate channels used?
14. How do you maintain a relation over time with customers, e.g., for making updates, providing associated services, etc.?

D. **Development processes, methods and tools**
15. What type of information do you give to, or receive from, these organizations during development?
16. How does working with external partners influence the way you structure your technology (product architecture, interfaces, etc.)?
17. Who is responsible for the quality of the total solution, and how are liability issues resolved?

E. **Wrap-up**
18. Is there anything else that we should have discussed?

## References

[1] U. Eklund, J. Bosch, Introducing software ecosystems for mass-produced embedded systems, Softw. Business (2012) 248–254.
[2] J. Axelsson, A. Kobetski, On the conceptual design of a dynamic component model for reconfigurable AUTOSAR systems, in: Proceedings of the 5th Workshop on Adaptive and Reconfigurable Embedded Systems (APRES'13), Philadelphia, USA, April 8–13, 2013.
[3] D.G. Messerschmidt, C. Szyperski, Software Ecosystems: Understanding an Indispensable Technology and Industry, MIT Press, 2005.
[4] R.P. Dos Santos, C.M.L. Werner, Revisiting the concept of components in software engineering from a software ecosystem perspective, in: Proceedings of the Fourth European Conference on Software Architecture: Companion Volume, 135–142, ECSA'10, ACM, New York, NY, USA, 2010. doi:http://dx.doi.org/10.1145/1842752.1842782.
[5] R.K. Yin, Case Study Research: Design and Methods, fifth ed., SAGE Publications, 2013.
[6] C. Robson, Real World Research, second ed., Blackwell, 2002.
[7] P. Runeson, M. Höst, Guidelines for conducting and reporting case study research in software engineering, Empir. Softw. Eng. 14 (2) (2009) 131–164.
[8] A. Osterwalder, Y. Pigneur, Business Model Generation, John Wiley, 2010.
[9] K. Manikas, K.M. Hansen, Software ecosystems – a systematic literature review, J. Syst. Softw. 86 (5) (2013) 1294–1306, http://dx.doi.org/10.1016/j.jss.2012.12.026.
[10] European Union, Directive 2006/42/EC, Official Journal of the European Union, pp. L157/24-86, 2006.

[11] G.K. Hanssen, T. Dybå, Theoretical foundations of software ecosystems, in: Proceedings of the 4th Third International Workshop on Software Ecosystems (IWSECO'11), Boston, Massachusetts, 2011, pp. 6–17.
[12] R. Goldman, R.P. Gabriel, Innovation happens elsewhere open source as business strategy, Morgan Kaufmann, Amsterdam; Boston, 2005.
[13] C. Wohlin, P. Runeson, M. Höst, M. Ohlsson, B. Regnell, A. Wesslén, Experimentation in Software Engineering: An Introduction, Springer, 2000.
[14] K. Popp, Goals of software vendors for partner ecosystems – a practitioner's view, Softw. Business (2010) 181–186.
[15] E. Yu, S. Deng, Understanding software ecosystems: a strategic modeling approach, in: Proceedings of the 3rd International Workshop on Software Ecosystems (IWSECO'11), Brussels, Belgium, CEUR: 65–76, 2010. <http://ceur-ws.org/Vol-746/IWSECO2011-6-DengYu.pdf>.
[16] E. Papatheocharous, J. Axelsson, J. Andrersson, Issues and challenges in ecosystems for federated embedded systems, in: Proceedings of the First International Workshop on Software Engineering for Systems-of-Systems (SESoS'13), Montpellier, Paris, 21–24, ACM, New York, NY, USA, 2013. doi:http://dx.doi.org/10.1145/2489850.2489854.
[17] J. Bosch, From software product lines to software ecosystems, in: Proceedings of 13th International Software Product Line Conference (SPLC'09), IEEE Computer Society, San Francisco, USA, 2009, pp. 111–119.
[18] J. Bosch, P. Bosch-Sijtsema, From integration to composition. On the impact of software product lines global development and ecosystems, J. Syst. Softw. 83 (2010) 67–76.
[19] V. Boucharas, S. Jansen, S. Brinkkemper, Formalizing software ecosystem modeling, in: Proceedings of 1st International Conference on Open Component Ecosystems (IWOCE'09), ACM, Amsterdam, 2009, pp. 41–50.
[20] A. Ghazawneh, O. Henfridsson, Micro-strategizing in platform ecosystems: a multiple case study, in: Proceedings of the 32nd International Conference on Information Systems (ICIS 2011), Shanghai, China, December 4–7, 2011.
[21] S. Jansen, A. Finkelstein, S. Brinkkemper, A sense of community: a research agenda for software ecosystems, in: Proceedings of the 31st International Conference on Software Engineering, New and Emerging Research Track, 2009, pp. 187–190.
[22] O. Barbosa, C. Alves, A systematic mapping study on software ecosystems, in: Proceedings of the 3rd International Workshop on Software Ecosystems (IWSECO'11), Brussels, Belgium, 2011. <http://ceur-ws.org/Vol-746/IWSECO2011-2-BarbosaAlves.pdf>.
[23] S. Jansen, M.A. Cusumano, S. Brinkkemper, Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry, Edward Elgar Publishing, 2013.
[24] G.K. Hanssen, A longitudinal case study of an emerging software ecosystem: implications for practice and theory, J. Syst. Softw. 85 (7) (2012) 1455–1466.
[25] M. Iansiti, R. Levien, Strategy as ecology, Harvard Bus. Rev. 82 (3) (2004) 68–81.
[26] J. Bosch, P.B. Sijtsema, From integration to composition: on the impact of software product lines, global development and ecosystems, J. Syst. Softw. (2010) 67–76.
[27] T. Kilamo, I. Hammouda, T. Mikkonen, T. Aaltonen, From proprietary to open source—growing an open source ecosystem, J. Syst. Softw. 85 (7) (2012) 1467–1478.
[28] I. Hunink, R. Van Erk, S. Jansen, S. Brinkkemper, Industry taxonomy engineering: the case of the European software ecosystem, in: Proceedings of the 4th European Conference on Software Architecture (ECSA 2010), ACM, Copenhagen, Denmark, August 23–26, 2010, pp. 111–118.
[29] I.V.D. van de Berk, S. Jansen, L. Luinenburg, Software ecosystems: a software ecosystem strategy assessment model, in: 2nd International Workshop on Software Ecosystems (EcoSys'10), Proceedings of the 4th European Conference on Software Architecture (ECSA'10), Companion Volume, Copenhagen, Denmark, August 23–26, 2010, pp. 127–134.
[30] S. Jansen, S. Brinkkemper, A. Finkelstein, Business network management as a survival strategy: a tale of two software ecosystems, in: Proceedings of the First Workshop on Software Ecosystems, 2009, pp. 34–48.
[31] S. Jansen, S. Brinkkemper, A. Finkelstein, Providing transparency in the business of software: a modeling technique for software supply networks, in: 8th IFIP Working Conference on Virtual Enterprises (Pro-VE'07), Guimarães, Portugal, September 10–12, 2007.
[32] B. Iyer, C.-H. Lee, N. Venkatraman, Managing in a "small world ecosystem": lessons from the software sector, Calif. Manage. Rev. 48 (3) (2006) 28–47.