

Received December 17, 2020, accepted January 4, 2021, date of publication January 8, 2021, date of current version January 20, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3050255

Heroku-Based Innovative Platform for Web-Based Deployment in Product Development at Axis

PATRIK DANIELSSON¹, TOM POSTEMA², AND HUSSAN MUNIR³ 

¹Mpya Sci & Tech, 41114 Gothenburg, Sweden

²Axis Communications, 22369 Lund, Sweden

³Department of Computer Science and Media Technology, Malmö University, 20506 Malmö, Sweden

Corresponding author: Hussan Munir (hussan.munir@mau.se)

ABSTRACT The introduction of cloud technology has reduced the feedback time in software development for many companies. However, companies require significant resources to run applications in an Infrastructure as a Service. The study aims at developing an innovation platform which enables faster deployment of web-based applications. The innovation platform is an add-on for a cloud service, its purpose is to allow developers to set up and use multiple cameras in a cloud environment. Furthermore, the study investigates the drivers and value created by the platform for Axis. We developed a proof of concept built on the innovation platform, using the design science methodology and evaluated it in a focus group. The prototype was a mock version of a grocery store's internal website, used to show the potential of the innovation platform. Moreover, we conducted semi-structured interviews to investigate the drivers for implementing the platform and the value it created. Results showed that the implementation of the prototype may make the deployment of web-based applications easier. A majority of the interviewees and the focus group participants agreed that the development of the innovation platform should continue and be tested to adopt the platform as a complement to AWS. However, the degree to which the platform should be open source needs a more clear management strategy. The prototype is seen as an innovation platform which may be used to quickly experiment with new ideas. The key drivers for implementing a prototype entails reduced cost, faster time to market and reduced complexity of web-based deployments.

INDEX TERMS Heroku, SaaS, PaaS, IaaS, deployment, cloud, open innovation.

I. INTRODUCTION

The increased demand for delivering products with faster time to market and increasing development costs has put pressure on many software companies to find new strategies for developing software products [1]. Companies struggle to remain competitive using the existing closed model of innovation and need to combine the internal ideas with external ideas. These strategies entail harnessing external ideas while leveraging their in-house R&D outside their current operations [2]. One of the possible ways to reduce the development costs and shorter the time to market is to use Open Innovation (OI), coined by Chesbrough, to harvest the external ideas. OI is defined as “a distributed innovation process based on purposively managed knowledge flows across organizational

boundaries, using pecuniary and non-pecuniary mechanisms in line with the organization's business model” [3].

Companies are constantly reevaluating their strategies to generate new ideas and create value by bringing these ideas to market. Gassmann *et al.* [4] described three forms of open innovation processes: (1) The outside-in process: Enriching a company's own knowledge base by using external knowledge sourcing, which may increase a company's innovativeness. (2) The inside-out process: The external exploitation of ideas by opening up the development to the external environment. (3) The coupled process: The combination of outside-in and inside-out by working in alliances with other companies using Open Source Software (OSS) ecosystems. OSS ecosystems are introduced as one of the most attractive domains in the software ecosystem field which covers a wide range of software projects in the software engineering domain and is not limited to a few projects [5], [6]. For instance, some common


The associate editor coordinating the review of this manuscript and approving it for publication was Xiaobing Sun .

TABLE 1. Cloud services model.

Service Model	Description
Software as a service (SaaS)	The user uses the software applications running on a cloud infrastructure (e.g., Dropbox, Google Docs).
Platform as a service (PaaS)	The user deploys applications on the cloud infrastructure with all the libraries, services, and tools supported by the cloud solution provider (e.g., MS Azure, AWS Elastic Beanstalk, Heroku.)
Infrastructure as a service (IaaS)	This refers to providing the user with the computing infrastructure such as virtual machines, block and file-based storage, firewalls, load balancers, IP addresses, virtual local area networks etc. (e.g., Amazon EC2, Rackspace, Google Compute Engine.).

examples are platforms, libraries, operating systems, modules and components, and tools [7]–[12].

The strategies of developing software have changed over time in which companies have focused on software ecosystems instead of having a software product line in every category [13], [14]. Therefore, software development is not an in-house development in a single organization anymore. Companies co-develop their products in collaboration with users and developers in OSS communities. OSS is the most notable example of switching from a conventional Closed Innovation model to an Open Innovation model and has characteristics that resemble a software ecosystem [5]. For example, IBM has supported Linux by investing \$100 million a year for more than a decade and also donated hundreds of patents to challenge the dominance of Microsoft Windows [15]. One of the biggest advantages of using the Open Innovation model is to share the risk and development cost of the platform [16]. Other firms such as Nokia, Intel, and Hitachi made substantial investments in development of Linux along with IBM and the overall investment in Linux was more than \$1 billion a year [15]. The Linux platform development allowed companies to build their applications and services on top of it. Specifically, IBM's investment in Linux provided them a platform to experiment with other companies to strengthen its own business model by running its own software solutions on top of Linux [17].

In this study, we created an innovation platform prototype for the case company which can be used by the company's developers to build, deploy and run their own applications. Currently, the case company is using Amazon Web Services (AWS) in their context for running applications, which requires significant skills, resources and incurs licensing costs. The objective behind building an innovation platform prototype for Axis is to be able to create fast prototypes and quickly get customer feedback and to reduce the development time and cost.

This paper is structured as follows. Section II explains the related work and section III outlines the research methodology. Section IV describes the implementation and section V presents the evaluation of the prototype from the focus group and interviews. Finally, sections VI and VII present a discussion followed by conclusions in relation to the research questions respectively.

II. RELATED WORK

This section discusses related work in the areas of cloud computing, continuous delivery, security, software value and

the level of openness when it comes to OSS. It also discusses the three layers for product innovation [18].

There are many services that can be provided over the internet with the help of cloud computing. The three main different types of cloud computing are Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). The differences between these services are described by Goyal [19] in terms of what a provider has implemented and what the customer still needs to implement (see Table 1). An IaaS offers full control to clients by providing them with all the necessary infrastructure. SaaS takes this concept even further by deploying the already developed application in the cloud infrastructure. It allows clients to access the application using a web interface. PaaS is somewhere in between, providing developers with readily available development stacks to handle security, backups and recovery so that developers can focus on implementing the application. PaaS enables an increased speed of the application deployment [20]. Continuous delivery cycles of software applications in cloud systems can be improved by using existing functionality in PaaS platforms. These cycles do not have to be fixed, instead automated deployment to production is made possible when implementing continuous delivery. Continuous delivery refers to ensuring the release and deployment of software at any time. It allows faster time to market, reduced development costs, a more flexible development environment for developers and faster software updates to clients which creates value for the company [21], [22]. In other words, short release cycles are essential in continuous delivery. Security is another very important aspect of almost any application. There are numerous studies discussing cloud security issues related to data leakage, data relocation, security policies, privileged admittance, transparency, vulnerabilities in virtualization, legal and regularity issues [23], [23]–[26].

Some of the most notable examples of PaaS are AWS Elastic Beanstalk, MS Azure, Heroku, Google App Engine and AppScale. First, AWS Elastic Beanstalk is a cloud service for deploying and scaling web applications. It allows users to automatically handle the deployments on scale while retaining full control over the AWS resources required to power the deployed applications. Second, MS Azure is a cloud computing service created by Microsoft for building, testing, deploying, and managing applications and services using MS managed data centers. Third, the Heroku network runs applications in virtual containers which execute on a reliable run-time environment. Heroku calls these

containers Dynos. Heroku lets the developer scale the app instantly just by either increasing the number of dynos or by changing the type of dyno the app runs in. Fourth, Google App Engine enables developers to build scalable web and mobile backends in any language on a fully managed platform. Finally, AppScale is an Open source software infrastructure which offers automatic deployment, configuration, and scalability [27].

The use of OSS in the development of proprietary products to create value is a widely accepted practice in which companies try to use the existing code to speed up the development process instead of reinventing the wheel [10], [12], [15], [28]–[31]. The degree of innovation is increased by developing software together with OSS communities [21]. However, the degree of openness may vary between completely open, partially open and completely closed [28], [32]–[35]. There are many studies that relate to fostering innovation and how it can be achieved using an OSS ecosystem [21], [32], [36]–[42]. However, the literature has a limited number of studies on the implementation of new innovation platforms [43] and the validation of these newly built platforms.

III. RESEARCH METHOD

This section describes the case company. Furthermore, we present the research questions and the research design is presented followed by validity threats.

A. CASE COMPANY

Axis Communications AB was founded in 1984 and launched the world's first network camera in 1996. Axis is a market leader in network surveillance video cameras. Axis has about 3200 employees and has partners in a number of countries around the world. The company has been using and contributing to open source software (OSS) communities (e.g., Jenkins, GStreamer etc.) and is also involved in discussions with other companies about open source strategies. Whenever possible, the preferred option is usually an open source alternative. Axis has well defined OSS processes and strategies when it comes down to using or releasing code to OSS communities. Axis uses open source code of Linux and has been a member of the Linux Foundation since January 2013.¹

B. RESEARCH QUESTIONS (RQs)

Brad *et al.*, [43] developed a platform for open innovation that brings together companies of different sizes that contribute information. The platform is only supportive in the development of software and not usable on its own. In this paper, we introduced an independent platform which can deploy and run applications. The platform helps spur innovation by enabling faster prototyping.

The research questions were identified as a mean to fulfill and achieve a platform suitable to Axis. The research

¹Linux Foundation Announces New Members: <https://www.linuxfoundation.org/press-release/2013/01/linux-foundation-announces-new-members/>, retrieved 30 May 2019

questions were improved through early interviews with different stakeholders at the company in order to ensure their relevance.

There is a need to understand the potential drivers (RQ1) to create the innovation platform prototype and what value it offers for Axis. The initial reason for this study was related to making it easy to add new functionality to the products that Axis offers but what additional drivers exist at Axis to pursue this? Furthermore, there is a need to understand what potentials such a platform prototype could bring, resulting in the following three research questions:

RQ1 What are the drivers for Axis to create an innovation platform prototype?

RQ2 How can an innovation platform be created and evaluated to facilitate web-based deployments at Axis?

RQ3 What value does the innovation platform create for Axis?

RQ2 deals with the creation of an innovation platform prototype based on the requirements stated in section III-C in order to deploy web-based applications. Furthermore, the developed proof of concept for the innovation platform was evaluated in the focus group. Finally, RQ3 addresses the value created by the new prototype for Axis, using the software value map [44].

C. DESIGN SCIENCE BASED IMPLEMENTATION

This section presents the design science approach based on the guidelines by Hevner [45]. Rigor and relevance are two important criteria in empirical research settings [46]. There is a risk that one of the criteria becomes more susceptible to validity threats in order to strengthen the other. For example, if a study is too focused on rigor and the theoretical contributions, the design of the artifact and its relevance to the industry may become less significant. On the other hand, if the research design is too pragmatic and flexible, rigor and quality of the artifact's underlying design knowledge may become subjective. Hence, we choose a design science research approach to balance the needs of industry and academics. This study balances both rigor and relevance when design cycles are conducted. An overview of the research methodology can be seen in Figure 1. The design science framework has three major components: 1) **environment**, 2) **research** and 3) **knowledge base**. First, the **environment** defines the problem space in which the artifact is relevant. In this case, the study elicited the requirements from experts at Axis to improve the deployment of web-based projects by developing an innovation platform based on Heroku. The goal of the relevance cycle is to identify the business needs of Axis in order to develop an artifact (innovation platform prototype) which is in line with Axis' business needs. The details of the experts involved in the relevance cycle are mentioned in Table 2. Second, the **research** process entails the implementation and evaluation of an innovation platform in a design cycle. The purpose of the design cycle is to iteratively

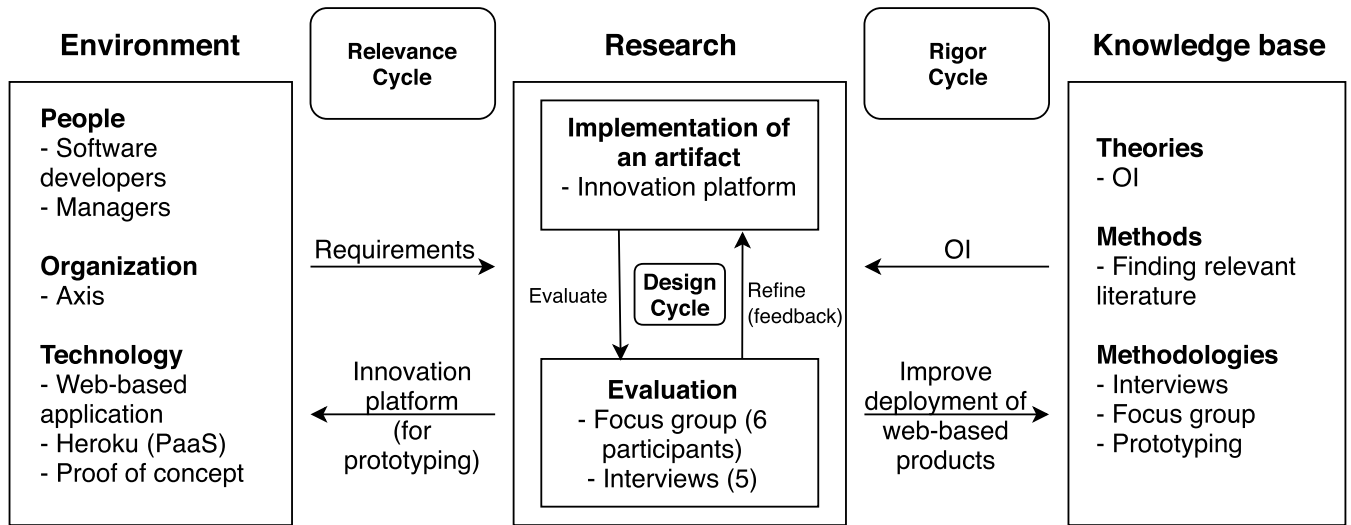


FIGURE 1. Overview of the research method.

develop and evaluate the prototype based on the refined needs of an innovation platform. A detailed assessment of the prototype was performed in a focus group at Axis. The details of the focus group participants and evaluation of the innovation platform can be seen in Table 3 and section V respectively. Finally, **knowledge base** represents the underlying theory used for the development of an innovation platform. This study uses Open Innovation by extracting the knowledge from outside Axis using Heroku to develop the innovation platform and combine it with the internal knowledge of experts to reduce the complexity of web deployment in their development process. The rigor cycle was achieved by using the existing literature on PaaS cloud and challenges associated with it. The related work outlines the knowledge base for the development of the innovation platform. Moreover, the study performed semi-structured interviews to refine the requirements and identify the value created by the implementation of the innovation platform.

Designing an artifact: Discussions with experts at Axis lead to designing an artifact, which refers to the implementation of an innovation platform prototype for web-based deployments. The purpose of the innovation platform is to provide easy to use functionality to developers who want to create cloud services that use Axis cameras. A relevant community of developers and managers was identified at Axis to better understand the requirements for implementing the innovation platform. The community mentioned several requirements of the innovation platform that should be fulfilled in order for the prototype to be considered useful by Axis’ standards. The following requirements are identified for the implementation of the artifact:

Req 1 It should be fit for larger systems. Applications built upon the platform should be able to handle a large number of cameras and/or users.

- Req 2** The build and deploy should be fast. It should be possible to quickly set up an application that can be deployed to receive prompt customer feedback.
- Req 3** Deploying and installing applications using the prototype shall be easy.
- Req 4** There should be a system for receiving payments in place so that one does not have to implement this.
- Req 5** It should be possible to migrate an application to another cloud service if necessary.
- Req 6** It should be possible to have support for older cameras using the prototype thus it should be backward compatible.
- Req 7** The platform should work as intended and should not have bugs or unexpected behavior, i.e. it should be reliable.
- Req 8** The functionality of the innovation platform should be comparable to similar alternatives, not less.
- Req 9** The prototype should be cost competitive compared to other solutions.

The detailed implementation of the artifact can be found in section V.

D. SELECTION OF THE INTERVIEWEES AND THE FOCUS GROUP PARTICIPANTS

This section provides the demographics of the interviewees and the focus group participants to answer the research questions.

1) INTERVIEWS

We conducted five semi-structured interviews in order to investigate web-based deployment in product development for Axis. This resulted in a better understanding of the drivers (RQ1) and value for Axis (RQ3). The rationale for the selection of interviewees was to select targets with different

TABLE 2. Demographics of the interviewees.

Interviewee(s)	Title	Years of experience
A	Software developer	5
B	Senior Web Engineer	11-12
C	Director	6
D	Engineering Manager	10
E	Senior Software Engineer	4

roles to understand both technical and managerial aspects of the implemented prototype, including targets from the cloud service department. The engineers were all from different departments of the company. Table 2 shows the demographics of the interviewees. An initial set of questions was formulated and these questions were continuously improved and validated by all authors, as specified in section III-E3. The semi-structured interviews were conducted based on the following pointers.

- 1) Demographics
- 2) Drivers for developing an innovation platform
- 3) Challenges of using an existing platform
- 4) Value in creating a new innovation platform

All interviews were recorded, transcribed and then summarized. Furthermore, the summaries of the transcriptions were validated by multiple researchers. We performed a thematic analysis on the qualitative data collected through interviews, in which five distinct themes were identified (see Table 5). Later, the statements were labeled with the set of themes. The results of this analysis are presented in section V-A.

TABLE 3. Demographics of the focus group participants.

Participant (s)	Title	Years of experience
1	Software Engineer	10
2	Senior Software Engineer	9
3	Software Engineer	2
4	Software Engineer	10
5	Senior Software Engineer	10
6	Senior Software Engineer	15

2) FOCUS GROUP

A focus group meeting with six people was arranged to evaluate the developed prototype. The demographics of the focus group can be found in Table 3. The reason for the composition of the focus group was to have engineers from different departments give feedback on the viability and feasibility of the prototype. The focus group started off with a short presentation which included a demo of the proof of concept. Afterwards, the participants were given a link to a survey to evaluate the prototype. Finally, a discussion was held among the focus group participants based on the responses. The discussion was recorded and transcribed and then validated after which statements from the discussion were labeled with the themes found during the interviews. The results of the innovation platform prototype can be seen in section V.

E. VALIDITY THREATS

This section explains the validity threats pertaining to the study. We addressed four types of validity threats [47] with their mitigation strategies.

1) INTERNAL VALIDITY

Internal validity concerns causal relationships and the introduction of potential confounding factors [47]. The motives behind the implementation of the prototype were understood and documented in the review protocol document to avoid misunderstanding the objectives of the study. The review protocol document was also verified for its objectives in a meeting with Axis. Furthermore, authors have conducted multiple interviews before and after the implementation of the prototype to ensure the right drivers for the implementation of the prototype. This helped us in reducing the risk of introducing the possible confounding factors which could affect the outcome of the study. Furthermore, we used data triangulation and observer triangulation during the study. First, we collected data from multiple sources namely interviews and a focus group. Second, we involved multiple researchers in analyzing the data independently in order to avoid the subjective interpretation of the data. Finally, the focus group participants are composed of mostly software engineers and lacked a business manager to evaluate the business value more objectively. Therefore, the aforementioned limitation may be seen as a threat to the internal validity of the study.

2) EXTERNAL VALIDITY

This refers to what extent the findings of a study can be generalized to other contexts [47]. The scope of this study is limited to companies using software development as their key strategic business area, to accelerate their internal innovation process using open innovation. The selected case company focuses on software development for security cameras with the software development as a key strategic area. Therefore, the results of this study may be generalized to companies involved in the development of embedded devices. More specifically, results may be useful for the companies trying to innovate tools used for prototyping of new ideas.

3) CONSTRUCT VALIDITY

This refers to what extent the operational measures investigated in the study really represent what a researcher has in mind, and what is investigated according to the research questions [47]. First, informal meetings were conducted with two technical developers from Axis before implementing the prototype and understanding the motives behind the implementation of a prototype to try out new ideas for the camera. In order to understand the business and technical benefits of the prototype implementation, Axis gave access to the right set of interviewees (e.g., software developers and managers) and participants for the interviews and focus groups. This enabled the researchers to investigate the technical and business aspects of the prototype implementation as per

TABLE 4. Comparison of different PaaS solutions for implementing the prototype. '+' is considered the best option and '-' is the worst option while '0' is neutral.

Facets	Amazon Elastic Beanstalk	Dokku	Heroku	OpenShift
Faster time to market	+	+	+	+
Decrease complexity	+	-	+	+
Extendability	-	-	+	+
Open Innovation	+	+	+	+
Payment	-	-	+	0
Ease to integrate	+	+	+	+
Portability	+	+	+	+
Capabilities	0	+	+	-
Costs	+	+	-	0
Per-application users	0	-	+	-

research questions. Second, to avoid interviewees guessing the research questions, an email prior to the interview was sent to them to familiarize them with the context of the interview instead of sending the exact interview questionnaire. Furthermore, the names of all the interviewees were kept anonymous to ensure their privacy.

4) RELIABILITY

The reliability refers to what extent the data and the analysis are dependent on the specific researcher, and the ability to replicate the study [47]. In the study, the first two authors individually transcribed the interviews, sent back to interviewees for validation, analyzed the data and later on used that data for the thematic analysis. The first two authors kept track of all the qualitative and quantitative data collected from interviews and the workshop in a systematic way to be able to go back for validation. The third author was involved in designing the interviews and the workshop. Furthermore, validation of the data collected from the interviews and the workshop was also performed by the third author in order to make the findings more reliable. The study design and findings are kept as transparent as possible between the researchers and Axis. Finally, results were shared and presented at Axis in a seminar prior to writing a paper. However, the study has a limitation in terms of disclosing the management's decision regarding the future direction of the prototype.

IV. IMPLEMENTATION OF THE ARTIFACT

We considered several available PaaS solutions based on the facets mentioned in Table 4 to select a suitable platform for the implementation of an innovation platform prototype. The considered PaaS platforms included Amazon Elastic Beanstalk, Dokku, Heroku and OpenShift. In Table 4, '+' is considered the best option and '-' is the worst option while '0' is neutral. After the comparison, Heroku turned out to be the best option. Therefore, we chose Heroku to implement the prototype because it offers scalability, decreased complexity, security and payment integration which is more in line with the needs of Axis. Heroku is a container-based PaaS and these containers are known as 'dynos' containing the application and all necessary dependencies and run on a shared host. Whenever it is needed, the number of dynos an application

runs on can be scaled up or down. Heroku has a wide variety of add-ons that eases the development of an application.² One example of this is that there are a number of data stores to choose from. Instead of spending time on the implementation of your own data store, you can simply choose one of the available data stores. As a Heroku user, you can also sign up to become an add-on partner and contribute with your own add-ons.

Next, an initial architectural overview was set up. Here, the prototype agent enables the usage of Connect APIs and also handles authentication while our prototype communicates with Connect. Connect is a cloud service provided by Axis which runs on AWS and allows remote access to cameras. Therefore, multiple APIs may be used, for instance, upgrading firmware or viewing snapshots, that would otherwise not be possible unless having local network access. Go was chosen as the programming language for the prototype since it is both supported by Heroku and widely used by Axis in cloud development. Furthermore, we ensured that the prototype would be portable to avoid vendor lock-in. This included an investigation into the usage of a Docker image.

The actual implementation of the prototype started with a skeleton where endpoints specific to Heroku were implemented, for instance, one endpoint that was needed for a user to start using the prototype or add-on as it is called in Heroku. Next, in order to communicate with Connect, it was necessary to get a session ID from Connect. This first required a client ID and a client secret that was sent to Axis Open ID Connect (OIDC), generating a token that could be sent to Connect. The token would then be verified by Connect after which a session ID would be sent back. The implementation of the authentication sequence can be seen in Figure 3.

In the next step, we investigated how to communicate with Connect. This could be done by a direct translation of Connect's API in the prototype but this would require the implementation of every single possible API call that can be made to Connect. A second option was to create a POST handler which simply received the type of the request and the URL. A third option, which was chosen to be the preferred

²Add-ons: <https://elements.heroku.com/addons>, retrieved 9 February 2020

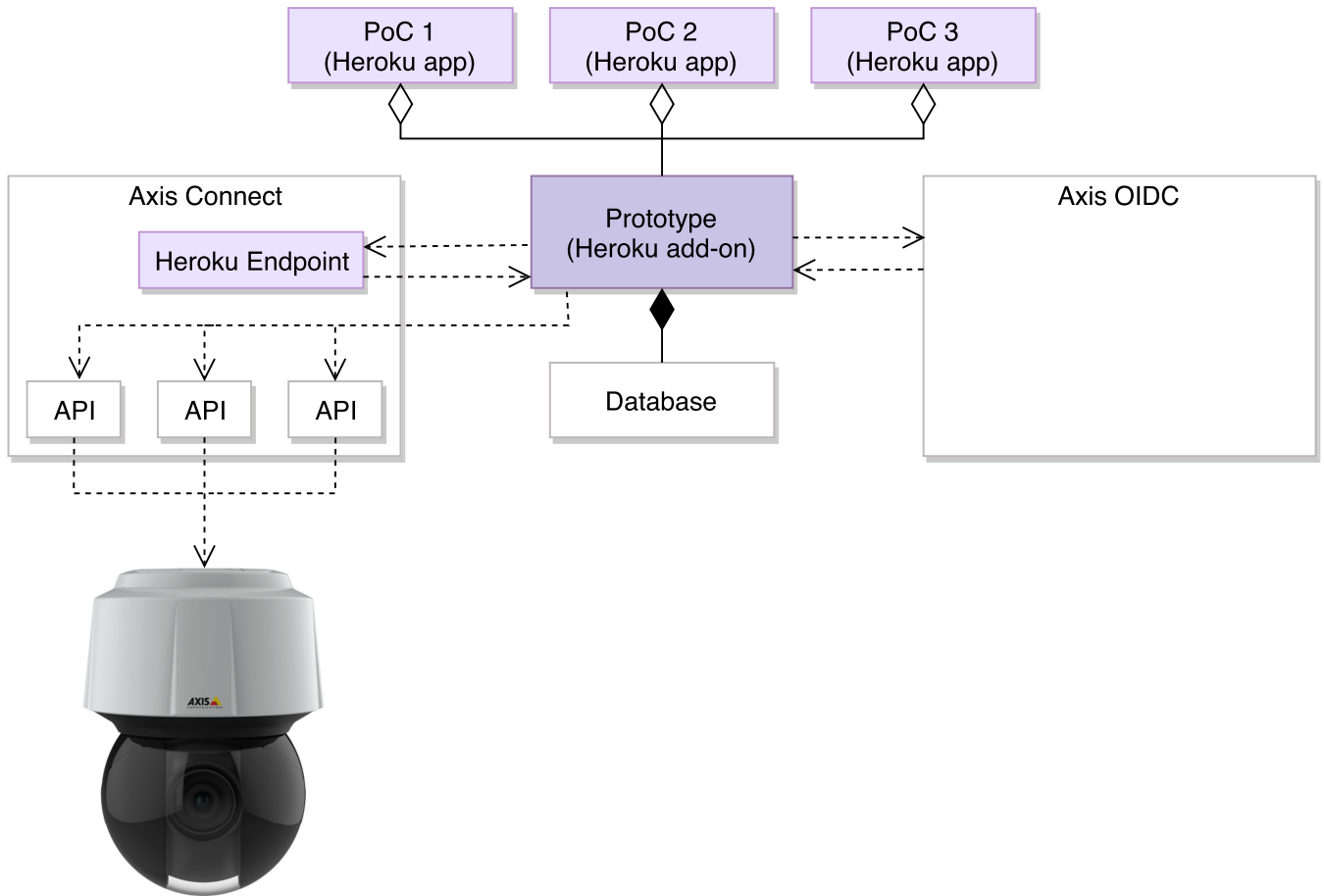


FIGURE 2. Updated architectural overview.

option, was letting all calls go to /connect/url, extract the URL and send it to Connect with the same type of HTTP request. This was the preferred option as it allowed a proof of concept to make any available Connect API call to the prototype just as if it had been towards Connect, thus also reducing overhead.

An important part during the implementation was to make sure that the prototype is secure. Therefore, a lot of time was spent on authentication and ensuring that all communication with Axis Connect was done using HTTPS. To authenticate users, a cookie is used. If no such cookie exists, a user is redirected to a login page whenever a call on the add-on is made.

One way to create additional value with our prototype was to chain API calls available in Axis Connect. One example of such a chained API call was implemented in order to refresh all snapshots of the cameras in a specific folder. A discussion about how API calls may be chained and the additional value of such solutions can be found in section VI. While implementing the prototype, some new aspects came to light, especially regarding authentication and the need for a database. This is reflected in an updated architectural

overview, shown in Figure 2. Here, the prototype runs on Heroku and Axis Connect runs on AWS.

After setting up the prototype, a proof of concept (PoC) using the prototype was created. This is denoted as PoC in Figure 2. The proof of concept is used to demonstrate how the innovation platform could be used to develop applications quickly and to easily demonstrate a proof of concept. The theme of the proof of concept was the website of a grocery store. For instance, in the proof of concept, a revenue counter showing the fictional daily revenue for the grocery store was added. It was also displayed which day of the week the next “delivery” was going to come on. Another proof of concept could be a large parking garage showing the different exits in combination with information about number of free spots and tickets issued. A third is to use AI to analyze facial expressions in real time of window shoppers in a shopping mall.

A question that could be raised is what value the prototype provides since API calls sent are simply redirected to Axis Connect. However, the prototype provides authentication of users and enables the possibility to chain API calls, extending functionality without the need to modify Connect. This is

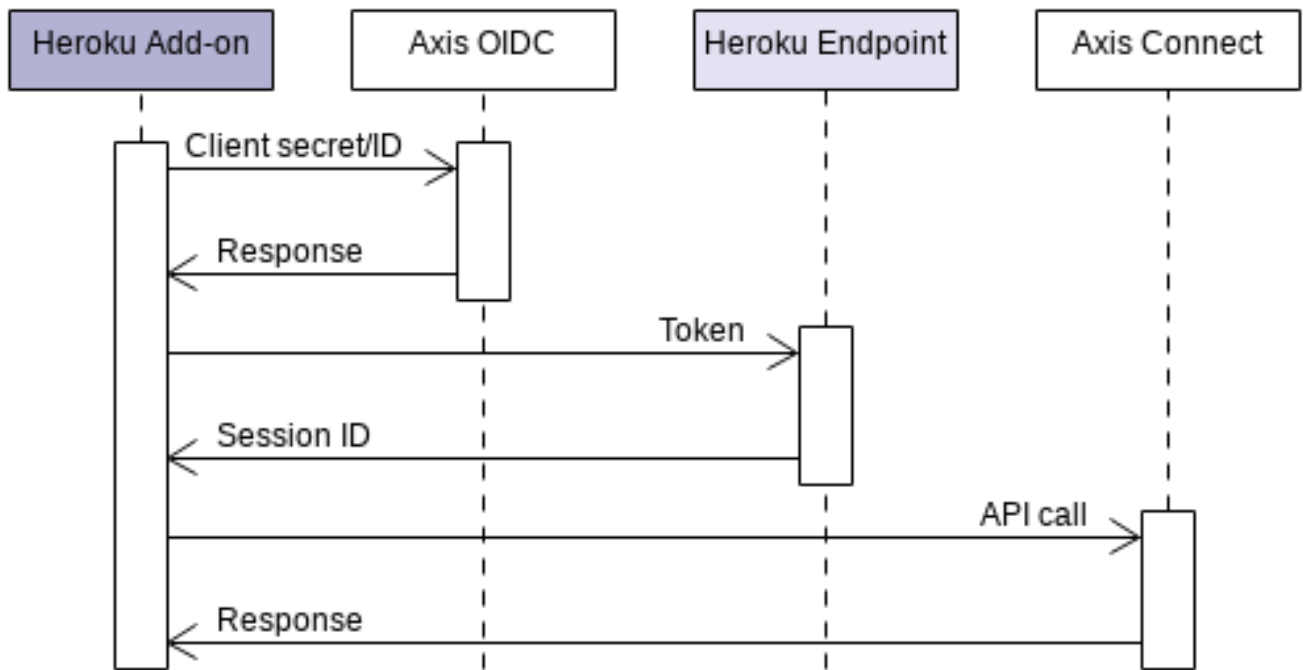


FIGURE 3. Sequence diagram.

especially interesting if the prototype would be open source, enabling many more programmers to add new functionality. Another benefit of the prototype is that Heroku has a number of other add-ons that could be used in combination with the implemented prototype to further extend the functionality. In the discussion part of this paper, section VI, there is a discussion about the many possibilities created by the prototype and what benefits this may entail for the developer.

V. ARTIFACT EVALUATION

This section includes the results from the focus group conducted to validate the implemented prototype. The details of the results can be seen in Figure 4.

The participants of the focus group regard the prototype as an innovation platform for trying out new ideas. As shown in Figure 4, question 1a, all participants agree (three agree and three strongly agree) that the prototype provides an experimentation layer to test innovative ideas. Furthermore, five of the participants agree or strongly agree that the prototype reduces the deployment complexities (Question 1c) of web-based applications. One participant even stated that prototyping is a lot faster when comparing with the current setup, “*you can just push your application and it is built and released.*” Another participant is of the opinion that “*it is a really good solution for trying out new ideas but using it in production is a completely different thing. However, it seems only positive for prototyping.*” It is also worth noting that no one thinks that development of the platform should not continue, instead four agree on and two were neutral about further development (Question 11). Looking at business aspects, four of

the participants find the Heroku add-on practical for projects in the future (Question 2, Figure 4). The payment model that was mentioned in several interviews was not a topic of discussion during the focus group, possibly due to the fact that no managers were present.

Furthermore, it is very important that the prototype is thoroughly tested before using it for other projects. As one participant puts it, “*For me it is an unknown. So it is something that we need to. . . Either contain really strictly and test or understand perfectly and then maintain our own branch of it and so on. So, I see the risk as well.*” Later, the same participant also remarks that “*for actual production you really have to understand how secure is it and it is really important because all the data we handle is really sensitive. So, it is big concern for us.*” A limitation mentioned during the focus group is that the prototype is dependent on Heroku and that Axis has no control over that platform. Another participant stated that “*we might end up adding another platform dependency*” This was also a limitation mentioned during the interviews but with regard to Axis Connect instead. How dependent this project is on other projects is therefore something that needs to be investigated further.

In relation to drivers of implementing the prototype, three participants of the focus group agree that the innovation platform can very well be seen as a viable complement to Amazon Web Services since it is easier to deploy projects with it. On the other hand, three of the participants took a neutral stance (Question 9, Figure 4). As one of the participants stated, “*it is viable because it is easier and that is mostly I think where its strength lies.*” There was a discussion on

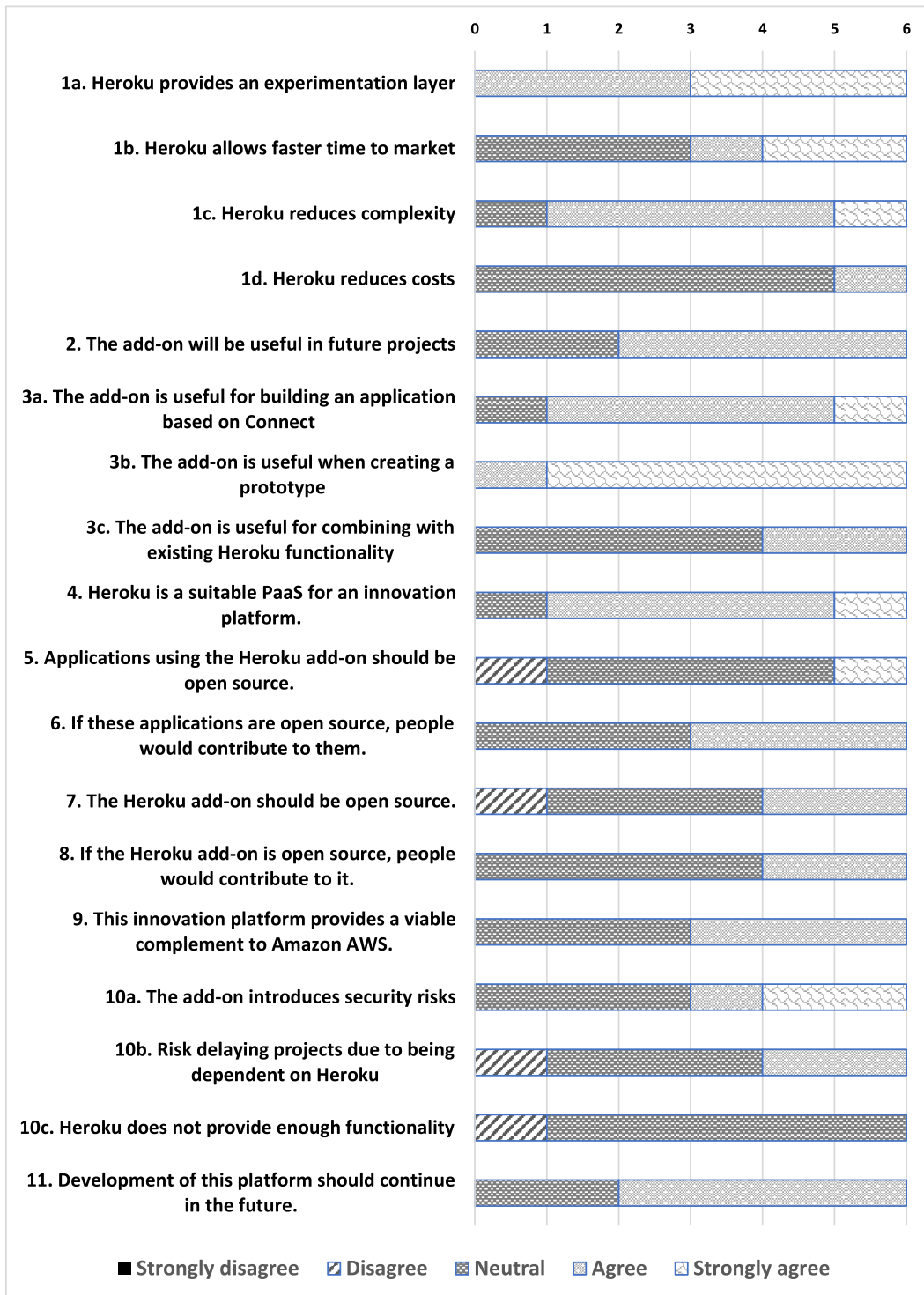


FIGURE 4. Focus group results.

making this prototype open source in which the participants had different opinions regarding the level of openness. Four of the participants were neutral to the idea of the prototype being open source, while one disagreed and one agreed. A participant stated that “It all depended on the application. If Axis

wants to profit from it, the application should not be open source.” Here, the participant refers to an application built upon the prototype. There was also a concern that the add-on encroaches too much of Connect. “The add-on integrates a lot with Connect and Connect isn’t open source. So it

TABLE 5. Themes found during the thematic analysis.

Theme	Definition	Research Question
Value	The refers to the value (see Table 6) created by the prototype for Axis.	RQ2, RQ3
Business value and Openness	This refers to business value and to what extent the implemented prototype can be made open source	RQ2, RQ3
Challenges	This deals with the obstacles that Axis sees with this prototype. These might be considered as constraints on the system.	RQ2, RQ3
Drivers	This refers to the reasons for Axis to create a prototype to facilitate the deployments of web-based applications.	RQ1

TABLE 6. Value aspects and its perspectives.

Value	Perspective	Subcategory
Functionality	Internal business	Production value: Physical value wrt. Quality: product architecture value
Completeness	Internal business	Production value: Market requirements value
Performance	Customer	Perceived value: Pragmatic value
Reliability	Customer	Perceived value: Intrinsic value
Usability	Customer	Perceived value: Pragmatic value
Fit with organization	Internal business	Production value: Physical value wrt. Quality
Economical aspects	Financial	Shareholder value

might be an example of just... We don't want open source to encroach too much of Connect." Comparing these results to the results from the interviews, the interviewees had some mixed opinions as well. Looking at the results in Figure 4, four were neutral towards open sourcing proof of concepts (Question 5) while when asked whether to open source the prototype (Question 7), the participants were more positive.

A. QUALITATIVE ANALYSIS FROM INTERVIEWS

We performed a thematic analysis [48], [49] on the qualitative data collected through semi-structured interviews. The thematic analysis leads to identifying the recurring patterns in the data collected using semi-structured interviews. The study presents four distinct themes shown in Table 5. The following steps were performed for the thematic analysis.

- 1) Transcribe data from interviewees (see Table 2)
- 2) Identify the distinct themes in the data (see Table 5)
- 3) Classify the statements based on the themes
- 4) Answer RQ1, RQ2 and RQ3

1) VALUE

The study uses a software value map [44] to identify the value for Axis when it comes to implementing the prototype. According to the software value map, the value aspects can be divided in four perspectives: 1) Financial, 2) Customer 3) Internal business process and 4) Innovation and learning perspective. These perspectives can then in turn be categorized even further. The aspects and their respective categorizations can be seen in Table 6. It is to be noted that two of the value aspects, *Completeness* and *Performance*, are not addressed in the software value map [44]. However, we categorized them in a subcategory that seemed appropriate.

The value aspects are clarified with the help of the interviews and the focus group in the following list:

- **Functionality:** One aspect of this is "to get features out to the users in a faster way which means that we should be able to develop things much faster... ", as interviewee B puts it. It is also important to provide functionality not already present in Axis Camera Companion ACC) according to almost all of the interviewees.
- **Reliability:** As interviewee C puts it "It would need to work seamlessly [...] it would need to install itself." For the product to be reliable it also has to be secure and on par with current solutions already at the company. This is in line with what interviewee A said "I suppose we would need to make sure that it is secure and reliable, we've found Amazon to be very reliable, very secure..."
- **Usability:** This is about how easy it is to use. As one participant of the focus group puts it, "you should be able to just push your application and it is built and released." Interviewee A in turn stated that "it would be worth it if it is cheaper and easy to develop."
- **Fit with organization:** As explained by interviewee C, "[...] if it's something that could easily be utilized by many to try things. I mean building prototypes or proof of concepts first [...] absolutely, it could be an innovation tool."
- **Economical aspects:** As interviewee C puts it, this is about having a "[...] low cost for Axis, high value for our partners, if you could demonstrate that and demonstrate a pull effect from our partners then that is probably what is needed to sell to Axis that this is something we should continue using."

All interviewees regard the platform as an innovation platform for quickly developing applications. As interviewee B puts it, the platform can be used to "get it out, get some experience, re-factor and continuously improve." Axis imagines a short release cycle and a simple way to add new functionality to Axis' products, thereby making it possible for customers to gain value quickly without too much effort. With a PaaS,

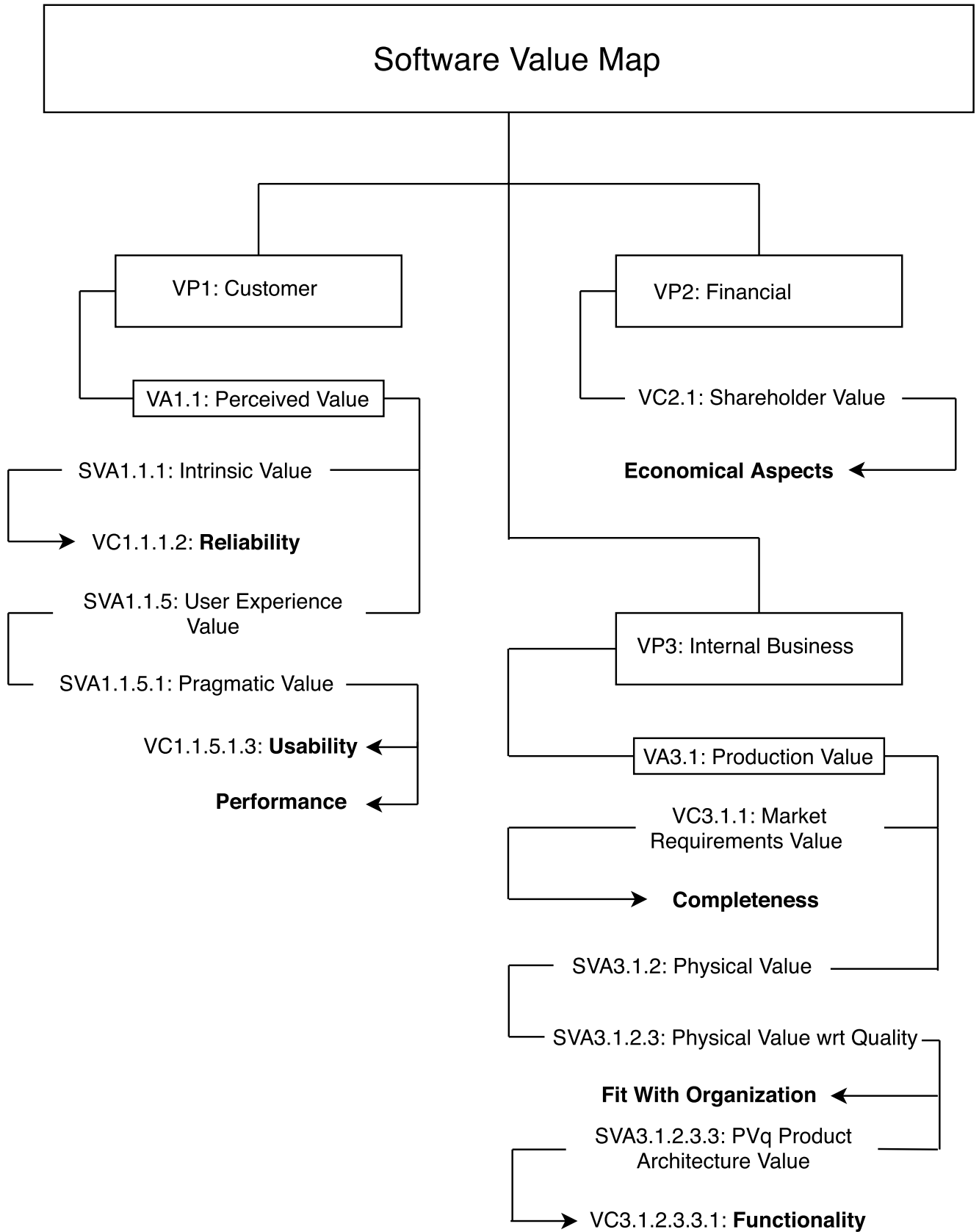


FIGURE 5. Software Value Map.

the burden on the developer is reduced because they don't need to do things such as setting up virtual machines. The platform may also be seen as a manual giving examples on how to do things. Interviewee D pointed out that this solution enables backwards compatibility by stating *"when building products, it can happen that you have to break your old APIs and then the cloud service can still be a way to have the support for older products. [...] if it works on this product, you can be sure that this functionality works on all products."* New ideas have been created, which may enable a faster time to market and reduced complexity, as discussed in section II.

2) BUSINESS

When it comes to the business aspects, almost all interviewees pointed out that the platform needs to offer something new. Another reoccurring point is that Axis is trying to be more of a solution-based service company. Interviewee D imagines something similar to the App-store/Google Play community: *"I don't think it is a problem that there are several other applications in App Store that does the same thing. As long as we get the best one. [...]. One makes it possible to build many services that do similar things and, in the end, you have the key to satisfying customer needs in the best way."* In relation to the three-layer product model by Bosch [18], the innovation platform prototype is on the innovation layer, enabling Axis to experiment with ideas before adopting them for the deployment of all web-based applications. It creates value by making it easy for Axis to build services. Furthermore, the important business factors include increased speed of innovation and the fact that payment could be received through Heroku. Interviewee B suggested *"[...] having a mature Amazon AWS-based implementation and then below on the innovation layer you have something in Heroku which allows you to quickly push out things."* This is in line with something that Bosch mentions in his paper [18], the so-called 'New-product transition interface', allowing products to move from the innovation layer into the differentiation layer. If a product is first built on the innovation platform it can then move into the differentiation layer if successful.

To what extent the implemented artifact can be made open source is still an on-going debate and interviewees had mixed opinions related to the openness of the prototype. The majority thinks that the implemented artifact should be open source as much as possible while a few think that the artifact should not be open sourced at all. Some were more neutral leaving it up to the management to decide. The reason for this is whether or not Axis wants to charge for the prototype. This is also discussed by Munir et al. [21] highlighting the risk of losing competitive advantage by open-sourcing. Interviewee D argues that, if the proof of concepts were open source, other developers could see examples of how it could be done. *"All of a sudden you have your own application and you haven't spent that many hours getting it working. You don't need to understand everything. Copy-paste programming."* This can create an entire ecosystem of applications in which

you can copy the code needed for the basic functionality and then apply the changes needed to fulfill the needs of your application.

3) CHALLENGES

The implementation of the prototype was not without the challenges for Axis. Challenges mentioned during the interviews for Axis includes that *"the company has a very limited experience with the cloud to begin with"* and another challenge is *"setting things up in a cyber-secure way"*, according to interviewee C. The platform might also be limited by Connect. The interviewees explained that it takes time to set up Amazon Web Services but after setting up the service, it is simple since a lot of time and money has already been invested in AWS. Concerns that came up during the interviews included delaying other projects, unexpected high costs, (cyber) security risks and the risk of opening backdoors into Connect. These risks need to be addressed before using the implemented prototype for the deployment of all projects.

4) DRIVERS

The key drivers for the implementation of the prototype seems to be to enable faster prototyping, spur innovation and reduce complexity during deployment. Interviewee D is talking about *"Making it easy to add new functionality to our products, abstracting away things that are hard to do [...] quickly and easily gaining quite a lot of customer value without putting a lot of working hours into it"*. Another driver for Axis is that customers might not want to use Amazon AWS because they *"could potentially view Amazon as a competitor"* according to interviewee C who is a director at Axis. Also, another reason to investigate alternatives is that Axis *"would not want to be entirely dependent on one supplier, especially another supplier as big and fast-moving as one of the giants within cloud"*, as interviewee C puts it. Looking at reasons not to continue with the project, one reason that was given during the interview with interviewee E was that a lot of time and money has already been invested in their current setup. However, this should not be seen as a hindrance to developing a new innovation platform to improve the deployment process.

VI. DISCUSSION

In this section the research questions defined in section III-B are discussed based on the results and analysis in section V-A. The research questions are also discussed in relation to existing literature.

There were many drivers identified during the interviews and focus group to answer **RQ1**. Some of the key drivers for building a prototype, as mentioned in V-A4, are 1) enabling faster prototyping and spurring innovation, 2) achieving higher customer satisfaction and 3) reducing the complexity of the deployment process. The main business drivers mentioned seemed to hail from a need to spend less time on aspects of the development process that were deemed non-essential. When showing the deployment process of Heroku

TABLE 7. Requirements fulfilled by the prototype implementation.

ID	The prototype implementation	Fulfilled	Value
Req 1	Enabled by Heroku, achieved through deploying more dynos as the application grows	Yes	Functionality, Fit with organization
Req 2	Achieved by the easy setup of an application that can be achieved on Heroku. The deployment process to Heroku is also a factor, the response from the focus group related to the deployment speed was very positive	Yes	Usability, Performance
Req 3	Partially enabled by Heroku and our implementation	Partially	Usability
Req 4	Made possible through the use of pricing plans on Heroku	Yes	Economical Aspects
Req 5	As Docker images can be deployed to Heroku this helps mitigate some of the portability issues one might face because one can be independent from Heroku's stack	Yes	Completeness
Req 6	Enabled by Axis Connect	Yes	Completeness
Req 7	Partially enabled by Heroku and our implementation	Partially	Reliability
Req 8	Several features, see the discussion about RQ2 later in this section.	Yes	Functionality, Completeness
Req 9	Cost of the add-on application can be covered by pricing plans imposed on the user.	Yes	Fit with organization, Economical Aspects

to the participants of the focus group the reactions were positive, thus confirming the idea of an easy deployment process being beneficial. The drivers speak to a need for faster development and only focusing on the code, which may lead to faster time to market. If the interviewers were to guess how the drivers would change in the future, it would have to be that as technology accelerates, the need for tools to keep up with the pace becomes an even more relevant aspect.

To address **RQ2**, requirements were elicited from experts to implement the prototype which are defined in section III-C. Both the implementation of the prototype and the analysis from the interviews made it possible to understand whether the requirements are fulfilled and how they are achieved. Through the interviews, four additional requirements were added. Those additional four requirements were enable backward compatibility (Req 6), reliable (Req 7), offer new functionality (Req 8) and cost competitive (Req 9). The first two added requirements were mainly requested from interviewees who had worked with similar platforms before and who valued stability. The last two, new functionality and cost competitive are both business-related suggesting that, if it is supposed to be an alternative, it should be able to compete with similar solutions. Table 7 discusses to what degree all requirements are fulfilled. Regarding the implementation of the prototype, two of the requirements in Table 7 namely Req 3 (easy to use) and Req 7 (Reliable) are marked as partially fulfilled. The participants were positive to the deployment on Heroku and agreed that the add-on could be used to develop prototypes, which indicates it is easy to use. The participants were also positive to the idea of the low downtime of Heroku and the way developers can deploy to Heroku without the target application being inaccessible for users, which both seem to indicate that they find it reliable. The implemented prototype is a Heroku add-on that developers may install. There is an apparent need at Axis to offer new functionality compared to other solutions.

Authentication towards Connect is an important one as it is not trivial to implement and having an easy API to use for it makes developing an application much faster. To redirect calls to Axis Connect helps with authentication as that is handled by the prototype instead of Axis Connect. To combine API calls to Axis Connect is especially interesting, both for the fact that one may set up large chains of commands into one, thus reducing the amount of programming required. This can be extended to allow for more specific API calls that return the exact value for what has been queried, instead of having to chain many API calls and only use half of the returned information. Another interesting aspect is when considering that you can add computationally heavy operations such as machine learning that could be used in conjunction with API calls. Another great aspect with the prototype is the possibility to add other Heroku add-ons, for instance, there are add-ons for video recording,³ encoding,⁴ and uploading⁵ that would provide great functionality when used in conjunction with video from the camera. Both the prototype and the proof of concept was demoed during the focus group meeting. It became apparent that both the interviewees and the focus group members were positive to the concept of the platform.

Concerning **RQ3**, Axis sees the developed prototype as an innovation platform which enabled faster innovation and a shorter release cycle to achieve faster feedback from customers. More specifically, the value proposition becomes more attractive to Axis as it is easier to quickly develop an application to test a new idea. Furthermore, it allows Axis to be less dependent on Amazon for all web-service needs. It might also be attractive for Axis to have the possibility to combine Heroku functionality and other Heroku add-ons in conjunction with the developed cloud application. Looking at

³<https://elements.heroku.com/addons/cameratag>

⁴<https://elements.heroku.com/addons/pandastream>

⁵<https://elements.heroku.com/addons/transloadit>

the value aspects that were found and presented in Table 6, all of these aspects have been fulfilled by the prototype as shown in the Value column of Table 7. However, we received mixed responses about whether or not the prototype and/or proof of concept should be open source. Many participants of the focus group were neutral to the idea of the prototype and/or proof of concept being open source. We interpreted it as the participants are primarily not negative to the idea but there may be more management-related factors at play as to why they feel this way. However, at least for internal use, the participants seem to agree on that it is a very useful platform. There are multiple alternatives as to what parts of the platform could be made open source. The results of the focus group survey whether the add-on or the applications built using the add-on should be open source were very similar. Despite this, one could imagine a scenario in which the more back-end part of the platform, the add-on was closed source and the example application together with any coming applications be made open source to help aid the development of new applications. When it comes to which open source license should be used, Axis has a Github page with several of its open source projects, the licensing across projects varies from MIT to BSD-3-Clause and GPL-2.0.⁶

VII. CONCLUSION

In this study, we have used the design science approach to develop an innovation platform prototype to explore the faster deployment of web-based applications at Axis. The main contribution of this study is the implemented prototype which has the potential to become an add-on for a cloud service that enables developers to set up and use multiple cameras in a cloud environment. A proof of concept was developed to demonstrate the practical application of the prototype. The proof of concept was a mock version of a grocery store website that used the implemented prototype to help show its feasibility. The key drivers for implementing a prototype entails reduced cost, faster time to market and reduced complexity of web-based deployments. The focus group results showed that the participants were positive about using the prototype to quickly experiment and test the business potential of new ideas at Axis. However, there was no consensus regarding whether or not the platform should be open source.

VIII. FUTURE WORK

As for future work, the innovation platform for web-based deployments can be further developed depending upon the business needs and the management decision. There are several more features of the prototype which could be implemented. For example, the implementation of video streaming from Axis Connect and different pricing plans on the add-on depending upon the usage. Furthermore, we need more validation studies on the innovation platform for future work. There is also a possibility to make more quantitative

comparisons with the existing cloud solutions as the platform matures and gains traction internally and among the partner companies.

ACKNOWLEDGMENT

The authors would like to thank Axis Communications AB for letting us conduct the study. They also would like to thank Prof. Dr. Per Runeson for taking the time to give us valuable feedback to improve the article.

REFERENCES

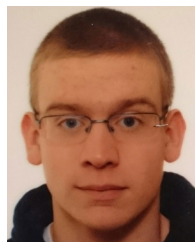
- [1] H. Chesbrough, "Why companies should have open business models," *MIT Sloan Manage. Rev.*, vol. 48, no. 2, p. 22, 2012.
- [2] H. W. Chesbrough, *Open Innovation: The New Imperative for Creating and Profiting From Technology*. Boston, MA, USA: Harvard Business School, 2003.
- [3] H. Chesbrough, W. Vanhaverbeke, and J. West, Eds., *New Frontiers in Open Innovation*. Oxford, U.K.: Oxford Univ. Press, Nov. 2014.
- [4] O. Gassmann and E. Enkel, "Towards a theory of open innovation: Three core process archetypes," in *Proc. R D Manage. Conf.*, 2004, pp. 1–18.
- [5] K. Manikas and K. M. Hansen, "Software ecosystems—A systematic literature review," *J. Syst. Softw.*, vol. 86, no. 5, pp. 1294–1306, 2013.
- [6] K. Wnuk, K. Manikas, P. Runeson, M. Lantz, O. Weijden, and H. Munir, "Evaluating the governance model of hardware-dependent software ecosystems—A case study of the axis ecosystem," in *Proc. Int. Conf. Softw. Bus.*, 2014, pp. 212–226.
- [7] A. Orucevic-Alagic and M. Host, "Network analysis of a large scale open source project," in *Proc. 40th EUROMICRO Conf. Softw. Eng. Adv. Appl. (SEAA)*, Aug. 2014, pp. 25–29.
- [8] C. Jensen and W. Scacchi, "Role migration and advancement processes in OSSD projects: A comparative case study," in *Proc. 29th Int. Conf. Softw. Eng. (ICSE)*, May 2007, pp. 364–374.
- [9] S. Jansen, S. Brinkkemper, and A. Finkelstein, "Business network management as a survival strategy: A tale of two software ecosystems," in *Proc. 1st Int. Workshop Softw. Ecosyst.*, 2009, pp. 34–48.
- [10] J. Henkel, "Selective revealing in open innovation processes: The case of embedded Linux," *Res. Policy*, vol. 35, no. 7, pp. 953–969, Sep. 2006.
- [11] M. Mukadam, C. Bird, and P. C. Rigby, "Gerrit software code review data from android," in *Proc. 10th Work. Conf. Mining Softw. Repositories (MSR)*, May 2013, pp. 45–48.
- [12] H. Munir, J. Linäker, K. Wnuk, P. Runeson, and B. Regnell, "Open innovation using open source tools: A case study at Sony Mobile," *Empirical Softw. Eng.*, vol. 23, no. 1, pp. 186–223, Feb. 2018.
- [13] J. Bosch, "From software product lines to software ecosystems," in *Proc. 13th Int. Softw. Product Line Conf.*, 2009, pp. 111–119.
- [14] J. Bosch, "Software ecosystems: Taking software development beyond the boundaries of the organization," *J. Syst. Softw.*, vol. 85, no. 7, pp. 1453–1454, Jul. 2012.
- [15] H. Munir, K. Wnuk, and P. Runeson, "Open innovation in software engineering: A systematic mapping study," *Empirical Softw. Eng.*, vol. 21, no. 2, pp. 684–723, Apr. 2016.
- [16] J. West and S. Gallagher, "Challenges of open innovation: The paradox of firm investment in open-source software," *R D Manage.*, vol. 36, no. 3, pp. 319–331, 2006.
- [17] J. West and J. Dedrick, "Open source standardization: The rise of Linux in the network era," *Knowl., Technol. Policy*, vol. 14, no. 2, pp. 88–112, 2001.
- [18] J. Bosch, "Achieving simplicity with the three-layer product model," *Computer*, vol. 46, no. 11, pp. 34–39, Nov. 2013.
- [19] S. Goyal, "Software as a service, platform as a service, infrastructure as a service—A review," *Int. J. Comput. Sci. Netw. Solutions*, vol. 1, no. 3, pp. 53–67, 2013.
- [20] S. Costache, D. Dib, N. Parlavantzas, and C. Morin, "Resource management in cloud platform as a service systems: Analysis and opportunities," *J. Syst. Softw.*, vol. 132, pp. 98–118, Oct. 2017.
- [21] H. Munir, J. Linäker, K. Wnuk, P. Runeson, and B. Regnell, "Open innovation using open source tools: A case study at Sony Mobile," *Empirical Softw. Eng.*, vol. 23, no. 1, pp. 1–38, 2017.
- [22] S. Neely and S. Stolt, "Continuous delivery? Easy! just change everything (well, maybe it is not that easy)," in *Proc. Agile Conf.*, Aug. 2013, pp. 121–128.

⁶Axis Communications, Github <https://github.com/AxisCommunications/>, retrieved 31 May 2019

- [23] S. A. Hussain, M. Fatima, A. Saeed, I. Raza, and R. K. Shahzad, "Multi-level classification of security concerns in cloud computing," *Appl. Comput. Informat.*, vol. 13, no. 1, pp. 57–65, Jan. 2017.
- [24] D. Dave, N. Meruliya, D. T. Gajjar, T. G. Ghoda, H. D. Parekh, and R. Sridaran, "Cloud security issues and challenges," *Big Data Analytics*, vol. 654, pp. 499–514, Oct. 2017.
- [25] K. Popović and Ž. Hocenski, "Cloud computing security issues and challenges," in *Proc. 33rd Int. Conv. (MIPRO)*, 2010, pp. 344–349.
- [26] S. Subashini and V. Kavitha, "A survey on security issues in service delivery models of cloud computing," *J. Netw. Comput. Appl.*, vol. 34, no. 1, pp. 1–11, Jan. 2011.
- [27] C. Krintz, "The AppScale cloud platform: Enabling portable, scalable Web application deployment," *IEEE Internet Comput.*, vol. 17, no. 2, pp. 72–75, Mar. 2013.
- [28] H. Munir, P. Runeson, and K. Wnuk, "A theory of openness for software engineering tools in software organizations," *Inf. Softw. Technol.*, vol. 97, pp. 26–45, May 2018.
- [29] A. Oručević-Alagić and M. Höst, *A Case Study on the Transformation From Proprietary to Open Source Software*. Berlin, Germany: Springer, 2010, pp. 367–372.
- [30] M. Höst and A. Oručević-Alagić, "A systematic review of research on open source software in commercial software product development," *Inf. Softw. Technol.*, vol. 53, no. 6, pp. 616–624, Jun. 2011.
- [31] S. Jansen, S. Brinkkemper, J. Souer, and L. Luinenburg, "Shades of gray: Opening up a software producing organization with the open software enterprise model," *J. Syst. Softw.*, vol. 85, no. 7, pp. 1495–1510, Jul. 2012.
- [32] J. Linåker, H. Munir, K. Wnuk, and C. E. Mols, "Motivating the contributions: An open innovation perspective on what to share as open source software," *J. Syst. Softw.*, vol. 135, pp. 17–36, Jan. 2018.
- [33] L. Dahlander and M. David Gann, "How open is innovation?" *Res. Policy*, vol. 39, no. 6, pp. 699–709, 2010.
- [34] M. Mashilo and T. Iyamu, "The openness of the concept of technology open innovation," in *Proc. IEEE Int. Conf. Manage. Innov. Technol. (ICMIT)*, Jun. 2012, pp. 487–492.
- [35] E. Knauss, D. Damian, A. Knauss, and A. Borici, "Openness and requirements: Opportunities and tradeoffs in software ecosystems," in *Proc. IEEE 22nd Int. Requirements Eng. Conf. (RE)*, Aug. 2014, pp. 213–222.
- [36] B. Rolandsson, M. Bergquist, and J. Ljungberg, "Open source in the firm: Opening up professional practices of software development," *Res. Policy*, vol. 40, no. 4, pp. 576–587, May 2011.
- [37] P. Runeson, H. Munir, and K. Wnuk, "It is more blessed to give than to receive—open software tools enable open innovation," *Tiny Trans. Comput. Sci.*, vol. 4, 2015.
- [38] J. Bosch, "Speed, data, and ecosystems: The future of software engineering," *IEEE Softw.*, vol. 33, no. 1, pp. 82–88, Jan. 2016.
- [39] U. Lichtenthaler and H. Ernst, "Opening up the innovation process: The role of technology aggressiveness," *R D Manage.*, vol. 39, no. 1, pp. 38–54, Jan. 2009.
- [40] G. Ahuja, "Collaboration networks, structural holes, and innovation: A longitudinal study," *Administ. Sci. Quart.*, vol. 45, no. 3, pp. 425–455, 2000.
- [41] A. Mockus and D. J. Herbsleb, "Why not improve coordination in distributed software development by stealing good ideas from open source," in *Proc. Meeting Challenges Surviving Success, 2nd Workshop Open Source Softw. Eng.*, 2002, pp. 19–25.
- [42] O. Alexy, J. Henkel, and M. W. Wallin, "From closed to open: Job role changes, individual predispositions, and the adoption of commercial open source software development," *Res. Policy*, vol. 42, no. 8, pp. 1325–1340, Sep. 2013.
- [43] S. Brad, M. Fulea, B. Mocan, A. Duca, and E. Brad, "Software platform for supporting open innovation," in *Proc. IEEE Int. Conf. Autom., Qual. Test., Robot.*, vol. 3, May 2008, pp. 224–229.
- [44] M. Khurum, T. Gorschek, and M. Wilson, "The software value map—An exhaustive collection of value aspects for the development of software intensive products," *J. Softw., Evol. Process*, vol. 25, no. 7, pp. 711–741, Jul. 2013.
- [45] R. H. Von Alan, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *MIS Quart.*, vol. 28, no. 1, pp. 75–105, Mar. 2004.
- [46] M.-A. Storey, E. Engstrom, M. Host, P. Runeson, and E. Bjarnason, "Using a visual abstract as a lens for communicating and promoting design science research in software engineering," in *Proc. ACM/IEEE Int. Symp. Empirical Softw. Eng. Meas. (ESEM)*, Nov. 2017, pp. 181–186.
- [47] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical Softw. Eng.*, vol. 14, no. 2, pp. 131–164, Apr. 2009.
- [48] D. S. Cruzes, T. Dybå, P. Runeson, and M. Höst, "Case studies synthesis: A thematic, cross-case, and narrative synthesis worked example," *Empirical Softw. Eng.*, vol. 20, no. 6, pp. 1634–1665, Dec. 2015.
- [49] D. S. Cruzes and T. Dybå, "Research synthesis in software engineering: A tertiary study," *Inf. Softw. Technol.*, vol. 53, no. 5, pp. 440–455, May 2011.



PATRIK DANIELSSON received the master's degree in computer science and engineering from Lund University. He worked as an Embedded Developer with Axis Communications. He is currently working as a Software Consultant with Mpya Sci & Tech. His research interests include embedded development in C, React, Go, the IoT, and cloud computing.



TOM POSTEMA received the master's degree in computer science and engineering from Lund University. He is currently working as a Software Engineer with Axis Communications in camera firmware. Together with Patrik, he has been involved in developing a system using cloud services that allow for feedback from customers more quickly when it comes to new features while aiming for minimal complexity.



HUSSAN MUNIR received the Ph.D. degree in software engineering from Lund University. He has worked as a Software Developer with Ericsson and also a Former Developer Advocate with Axis Communication, where he was involved in setting up the open-source community (Unimatrix) under the Linux Foundation. He was also a Contributor and Maintainer of the Axis Camera applications platform on GitHub. He is currently working as an Assistant Professor with Malmö University. He has collaborated with Sony Mobile, Axis Communications, Scania, and Volvo Cars for his research work. His research interests include open innovation, open-source, digitalization, business models, and ecosystem.

• • •