CHALMERS
UNIVERSITY OF TECHNOLOGY

Open Innovation Strategies for IoT Platforms

How can IoT Platforms Build and Offer Solutions using Open Innovation

Master’s thesis in Entrepreneurship and Business design.

RAJVARDHAN AJIT DESAI

DEPARTMENT OF TECHNOLOGY MANAGEMENT AND ECONOMICS
DIVISION OF ENTREPRENEURSHIP AND STRATEGY

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2022
www.chalmers.se
Report No. E2022:086

REPORT NO. E 2022:086

Open Innovation Strategies for IoT Platforms

How can IoT Platforms Build and Offer Solutions using Open Innovation

RAJVARDHAN AJIT DESAI

Department of Technology Management and Economics
Division of Entrepreneurship and Strategy
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2022

Open Innovation Strategies for IoT Platforms
How can IoT Platforms Build and Offer Solutions using Open Innovation

RAJVARDHAN AJIT DESAI

© RAJVARDHAN AJIT DESAI, 2022.

Report no. E2022:086
Department of Technology Management and Economics
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 1000

Gothenburg, Sweden 2022

Acknowledgment

This master's thesis is a part of the last semester of the Master of Science program in Entrepreneurship and Business Design at the Chalmers University of Technology, Gothenburg.

I would like to convey my sincere gratitude to my supervisor at the Chalmers University of Technology, Dr. (Prof) Bowman Heiden. It has been a challenging and immensely rewarding process writing my thesis. Thank you for your guidance and the interesting conversations.

I would also like to convey my gratitude to The Techno Creatives, and all the colleagues who have taken their time to help me navigate the field of research. I am especially grateful to my supervisor Jacob Ahrnstein. Thank you for all the learnings provided, and the interesting discussions we have had. It has been a sincere pleasure.

Lastly, I would like to thank my friends and family who have supported me along the way. It would not have been possible without your encouragement.

Rajvardhan Desai

Gothenburg, June 2022

Open Innovation Strategies for IoT Platforms
How can IoT Platforms Build and Offer Solutions using Open Innovation

RAJVARDHAN AJIT DESAI

Department of Technology Management and Economics
Chalmers University of Technology

Abstract/Summary

IoT platform market growth has exceeded expectations since 2015 and has a steady growth projected until 2026. Functionality has expanded, leading to a crowded space and intensely competitive market to operate. Most of the successful proprietary IoT platforms focus on vertical, purpose-built applications/solutions rather than universal products which leads to the lack of suitable products that fit the needs of small and medium enterprises (SMEs). The proprietary IoT platforms have a high cost and have limitations such as vendor lock-in, late responses to innovations, and customizability. This thesis aims to investigate the open innovation approaches of successful commercial open IoT platforms to devise a suitable open innovation strategy for IoT platforms to build solutions that can solve the issues present today.

The thesis is conducted using a multiple case study design. The cases are chosen from open IoT platforms that are commercially successful and are accepted as being open or use an open innovation approach in their business logic. The study also analyses the 'opensource toolbox' presented in the open platform design theory to explore how it can be used to design openness for IoT platforms.

The study concluded that open innovation in IoT platforms can solve the issues of vendor lock-in, and interoperability and enable faster innovation in the domain. A community-based or stakeholder-based open-source approach is likely to be a better solution to solve the issues, given the complex software-heavy technology underpinning the product. Although the study is not exhaustive, it presents the characteristics that need to be considered for building a commercial IoT platform using an open-source approach. An adapted version of the open-source toolbox is presented.

Keywords: IoT (Internet-of-Things), IoT platforms, Open Innovation, Opensource.

Table of Contents

1. Introduction

1.1 Background

The Internet of Things (IoT) phenomena can be summarized as a global ecosystem connecting diverse objects (devices) to the internet. By using embedded technologies, especially sensors, these objects communicate with each other and generate and exchange data without human intervention, based on a connection through the internet (Korte, et al., 2021). Internet of Things as a topic has gained significance owing to the impacts it has had on technology developments and the socio-economic lives of humans. IoT has transformed the way humans work, live, and play by connecting everyday objects to the internet and applying powerful data analytics capabilities to influence behaviors and increase productivity. Consumer goods, vehicles, and industrial utility objects are connected to the extent that the impact of IoT on the global economy is huge and projected to be more than $11 trillion by 2025 (Karen, et al., 2015). Over time, organizations have developed different taxonomies and categorizations of IoT applications. The term “Industrial IoT” is used to describe the use of IoT in manufacturing, managing supply chains, and industrial utilities. The term “consumer IoT” is used to describe personal devices such as wearables, smartphones, etc, which are further extended to describe “smart home” when home appliances are connected to be controlled via personal devices (Karen, et al., 2015). In essence, whatever may be the application of IoT, the effect is seen on human lives.

IoT Platform: “An Internet-of-Things (IoT) Platform can be defined as a multi-layer technology that enables straightforward provisioning, management, and automation of connected devices within the Internet of Things universe.” (Korte, et al., 2021). IoT platforms originated as IoT ‘middleware’ group of technologies that primarily function as a mediator between the hardware and application layers of software, UI/UX, cloud, etc. The common tasks irrespective of the use cases/application of IoT are- data collection from the devices over different communication protocols, remote device configuration and control, device management, and over-the-air firmware updates. From a higher perspective, the technical challenges that IoT platforms face are security, lack of interoperability and standards, unclear legislations, and holistic global development in terms of economics and technology deployments (Karen, et al., 2015). Out of these challenges, the lack of interoperability and standards directly affects the decisions of businesses to implement IoT platforms. Currently, the proprietary technical components in IoT implementations are of fragmented nature and pose a direct challenge to interoperability. While full interoperability may not be possible or even necessary for a business to adopt an IoT platform, the proprietary nature becomes the cause of inflexible integration with existing tools, high complexity in ownership, and vendor lock-in (Karen, et al., 2015). The standards developed by IEEE (IEEE Std 2413-2019) for IoT architectural framework are aimed at solving this problem of fragmentation. But the extent of adoption of this standard is questionable and has not gained prominence yet.

IoT platform market growth has exceeded expectations since 2015 and has a steady growth projected until 2026 (Phillipe Wegner, 2021). Use cases and the overall functionality has expanded, leading to a crowded and intensely competitive market to operate. Multiple pricing and revenue models have coupled with several revenue streams thanks to the widespread and fast adoption of IoT solutions by big industries and SMEs alike. IoT in most cases has paved the way for a digital transformation in mainstream business operations. Most of the successful IoT platforms focus on purpose-built applications and ready-to-use solutions rather than component-level products due to the lack or even difficulty in achieving interoperability with other tools that a customer is already using. Therefore, there is a general lack of coherence between the infrastructure of the services offered by different platforms (Vogel, et al., 2020).

Open IoT Platforms: In recent times, for (1) enabling better compatibility or interoperability in the technical sense, (2) efficient deployment of new technology developments in the IoT ecosystems, and (3) to avoid vendor lock-in; the term "Open IoT Platforms" has gained popularity (Vogel, et al., 2020). The common perception of the Internet-of-things phenomenon is that of being vertically integrated systems that are often too closed and fragmented in their applications, for example, IoT platforms have diverse mobile platforms and systems, different operating systems for devices, and programming languages are different for components within the technology stack, physical devices that have varying characteristics (Vogel, et al., 2020). Users and customers of IoT platforms often need to scout through different vendors that provide services in the entire value chain to select the most compatible solution. The increasing scale of applications has naturally forced an increase in devices and enabling technologies which make IoT systems development and deployment complicated. The most significant role that IoT platforms play is in the integration and development of these devices and enabling technologies, somewhat like a middleware software technology that behaves like a black box performing magic. Different types of platforms are often referred to as "IoT platforms" for example, device-to-device, cloud-based, and device-to-cloud platforms (Karen, et al., 2015). The commercial (often referred to as enterprise or proprietary) IoT platforms that consist of proprietary intellectual assets tend to promote vendor lock-in. Keeping up with the fast-developing technical landscape of IoT systems is a challenge in itself because of which proprietary platforms especially, small scale proprietary IoT platforms often fail to provide timely support for new protocols, tools, and data formats. Thereby, users suffer due to the vendor lock-in (Vogel, et al., 2020).

Open IoT Platforms, in most cases, follow the open-source approach for developing the platform in the form of contributions from the community to build the technical components, developing the documentation for different use cases and applications, or testing and providing feedback by using the IoT platform in different scenarios. However, there is a significant difference in open-source software development between IoT projects and non-IoT projects. Therefore, Open IoT platforms follow other approaches to innovate and solve issues plaguing IoT by collaborating with industry actors such as device providers, customers, and other IoT service providers by using open interfaces and technical development using open standards.

1.2 Prior Research

The literature review shows that research conducted about open IoT platforms is in the general context where only openness types were identified rather than the depths of how the openness was created. However, theories about how open platforms can be built and by defining the creation of the openness also exist in general, but not in an IoT platform context. Further, for gaining contributions in the open innovation initiatives, specifically in the form of open-source, multiple theories point out barriers, enablers, and archetypes for defining and scaling open-source projects.

The *Open Platform Design Theory* proposed by (Särefjord, 2006) gives a structured approach to how organizations can create an open innovation platform. In addition, the theory also gives insights into how openness in innovation can facilitate a business. To express the openness of an open platform, the theory divides the openness concept into intellectual property claims, availability of results, and the progress method of the open initiative. The theory is elaborated in the third chapter.

A report about *Opensource Project Archetypes* by the Mozilla Foundation (2018), an organization that champions the cause of open-source, presents a framework that can be used to set appropriate goals and methods for open-source projects. The report provides a questionnaire that can be used to evaluate intended goals and processes for making informed decisions about the governance types for building and scaling an open-source project with the required amount of community backing.

Although all the above theories evaluate openness types and open innovation methods in the general context of software development, they do not focus on understanding the said concepts in the context of IoT platforms.

The research by Vogel et. al. (2020) characterizes openness in the existing “open” IoT platforms as open-source, open standards, open APIs, open data, and open layer. According to the study, the most widely accepted openness type in open IoT platforms is open-source as it is more convenient for third-party developers to have full access to the source code to make a significant contribution to the open initiative of the IoT platform.

1.3 Problem Statement

Businesses are facing an issue of lock-ins with hyper-scalers such as AWS and Google. Especially, SMEs often find it difficult to find an IoT platform that is flexible, modular yet cost-efficient in providing services without giving in to the lock-in (Matthew Wopata, 2021). To tackle this issue, smaller IoT platform service providers can offer modular and custom-built solutions but, they find it hard to be visible in the densely populated space for IoT Platforms. An open innovation approach (especially open-source) for IoT Platform service providers may solve the issues both for themselves, as well as the customers (Matthew Wopata, 2021).

However, selecting an appropriate strategy to establish and sustain open innovation is difficult and tricky to navigate. Especially, following the open-source approach influences the development and the business model alike. A business, therefore, needs to evaluate the open-source phenomenon for software development in IoT as it is different and often more complicated than other non-IoT software development owing to the heterogeneity of technical components and their use in diverse application areas.

1.4 Research Purpose

The purpose of the thesis is to conduct an explorative study of IoT platforms that have a certain level of openness in their development and use. The study will analyze how the software assets within an IoT platform could be leveraged using open innovation to build solutions and scale in terms of providing solutions for multiple use cases (often referred to as applications). The case study is to analyze and compare commercial open IoT platforms to understand the nuances of having an open innovation approach for developing software assets.

1.5 Research Question(s)

Main Research Question

How can IoT Platforms build and offer solutions using open innovation?

Sub-research question 1

What are the different types of openness initiatives observed in IoT platforms?

Sub-research question 2

How can open platform design theory be used to design openness in IoT platforms?

Sub-research question 3

By using an open innovation approach what business models can be enabled for IoT platforms?

1.6 Scope and Delimitations

First, the report does not analyze strictly proprietary alternatives in general apart from indirectly using the background knowledge to put into context the benefits of an open IoT platform. The reason is that such an analysis has the potential for a standalone project and is not seen as a prerequisite for this project.

Second, the IoT platforms considered for the investigation are driven by a single organizational entity with a commercial motivation. The study excludes platforms that are driven by alliances and government initiatives. The reason is that main goal of this report is to conduct a study for building an open IoT platform to be driven as a commercial enterprise.

Third, the study will take a universal IoT platform perspective and not application/use case specific. It will include a high-level analysis of the main characteristics that affect building an open IoT platform- technology components, the component packaging forming the product, and their relation to the business model. An in-depth analysis of technical components is excluded due to time constraints.

Fourth, the phenomenon of open-source will only be analyzed in the context of open-source IoT platforms. The parameters for analysis will be restricted to analyzing mechanisms used for community building, contribution enablers, and the general outlook of open-source within IoT platforms.

2. Methodology

This section contains the approach and reasoning taken towards the thesis and evaluates the interpretation of data and points out the gaps that may hamper the quality of the results obtained.

2.1 Research Strategy

The research strategy adopted for the study is connected to the purpose of the thesis, that is, conducting an investigative study of how commercial Open IoT Platforms are built to offer solutions for multiple use cases using open innovation. The research questions were formed and answered with the assumption that the phenomena underpinning the subject of research are socially ontologically objective and this can be investigated by an epistemologically objective approach (Bell, et al., 2019). This essentially means that from a researcher's perspective the phenomena under study are socially constructed but widely accepted realities. Therefore, keeping this understanding as a cornerstone, the study was performed without questioning the existence of the phenomena.

The study conducted is qualitative and a mix of deductive and inductive approaches with existing theory from an abductive approach (Bell, et al., 2019). It means that the study is based on a set of preconceived theories about the subject which form the foundation to develop a theory by answering the main research question. The three sub-research questions are designed to break down the main research question into manageable subjects. The sub-research questions are meant to enable the gathering of primary data that help get an understanding of openness phenomena in open IoT platforms from a business perspective.

2.2 Research Design

The subject undertaken for the thesis affects a real-world business setting and is influenced by many variables and requires an in-depth understanding. A multiple case study fits the nature of the subject as it can accommodate the identification of variables that affect the phenomenon under study (Bell, et al., 2019).

A multiple case study research design is applied to five selected open IoT platforms. The open IoT platforms are selected based on their occurrence in other studies concerning open IoT platforms as well as their occurrence on websites for open-source software development forums such as GitHub and Stackoverflow where they are tagged as popular IoT projects.

The variables for the case study are derived from the deductive use of the theories mentioned in section 3 that form the foundation for the thesis, they are, IoT (Internet-of-Things), IoT platform, open IoT platforms, and open platform design theory, opensource archetypes, and opensource business models. Three theories, such as the theory about IoT platform reference architecture, Open-source Archetypes, and Business Archetypes are used as tools for understanding technical components, open-source logic, and business logic respectively, to be applied to the case study and derive pieces of practical information.

2.3 Data Collection and Analysis

The primary source of data for building the foundational concepts for the thesis were prominent scientific journals, publications, and reputed conferences in the field of IoT such as IEEE-Internet of things Journal, IEEE- Wireless Communications journal, IEEE-Transactions on Wireless Communications journal, MDPI Future Internet Journal, etc that were used in devising the IEEE standards for IoT. The articles and research papers are peer-reviewed and have many citations in the field. The theory about open-source business models and open-source archetypes is an exception, it was derived from corporate whitepapers, and blog articles from prominent actors in the commercial space of open-source such as venture capitals (Index Ventures and Blossom Capital) that are investors of successful opensource businesses. The open-source archetypes theory was derived from the Mozilla foundation's corporate documents. The practical nature of the subject prompted the use of such sources because they have been proven successful in other practical scenarios even if not solely in the case of IoT platforms.

The other sources of data such as podcasts, and recorded business summits, were also used to get qualitative data about opensource from a community building and community contribution perspective. Although, conducting interviews may be helpful in this context but only when the sample space is big enough. Due to the time constraints instead of interviews, other sources of data such as podcasts and recorded business summits that provide a consolidated insight about the opensource community and contribution were referred for analysis and discussion.

The variables are derived from the theory which used those variables to evaluate commercial open IoT platforms. The commercial open IoT platforms selected for the case study are successful commercially as well as have a reasonable use and possess tools for open innovation to be incorporated in developing the platform.

The source of data for evaluation of the selected open IoT platforms is the documentation given on their websites and GitHub- the platform where aspects of open source such as popularity, licenses, components, and technical characteristics are visible.

The data in the multiple case study is structured according to the variables (identified via literature review) affecting the answers to the research questions. There is a thematic approach in conducting the data analysis where the variables are compared as a theme rather than studying the causal relationships between sub-variables within the themes. The themes are open innovation approaches, business models, and tools used for constructing the open initiatives.

2.4 Research Quality

According to Bryman & Bell (2019), the quality of business research can be evaluated using three main criteria- reliability, replicability, and validity. The following paragraphs describe how the thesis study fares in these criteria.

2.4.1 Reliability

Reliability mainly relates to how repeatable are the results of a study considering the consistency of how motivated and organized the researcher and the process is (Bell, et al., 2019). It is especially important in the case of quantitative research where the measurements are not stable and differ too much. This study being qualitative, reliability may be questioned since it is the researcher's observations, and the observations of another researcher may vary slightly in terms of the degree of existence of a phenomenon. Due to time constraints, only five subjects were selected for the case study, which further makes the evaluation of reliability difficult.

2.4.2 Replicability

Replicability relates to the closeness of the findings of the research to already existing research on the same or a similar case (Bell, et al., 2019). It can be measured in a way that, if someone else does the research using the same procedure, the results must be replicable. The study may comply with high replicability since the subjects used in the case study have been used in many other research case studies, albeit in somewhat different but not entirely exclusive contexts. It may also stem from the high-level thematic approach taken towards forming and evaluating parameters without going into the depths.

2.4.3 Validity

Validity is related to the integrity of the conclusions derived from a single piece of research (Bell, et al., 2019). Validity is sub-divided into four classes Measurement validity, Internal validity, external validity, and ecological validity. The discussion about how this thesis research fares in terms of the validity of the research follows:

Measurement validity: In the context of this study, it concerns whether the variables chosen to capture the phenomena under question (open innovation) are valid i.e open IoT platforms and open-source software development in IoT. The study derives theory from research that is widely cited in the field of IoT and published in reputed journals. The specificity of the research topics in the literature review enables the selection. For example, the "IoT theory" and "IoT platform theory" used in the study were used in research concerning the performance of open IoT platforms.

Internal Validity: Internal validity is concerned with the causal relationship between the parameters of the study (Bell, et al., 2019). This thesis study has consciously tried to reason the causal relationship between the parameters but there may exist some validity concerns as the study subjects are few and deriving a causal relationship may be questioned because of lack of enough empirical evidence. However, the literature review on which the thesis relies is strong and already points out the causal relationship between the variables while the thesis only validates it using the case study. For example, the parallels drawn between the subjects in open platform design theory and components of IoT platform may be questioned because the reasons for drawing parallels are not discussed but, a reader with knowledge of the subject will be able to see the similarities.

External validity: External validity is concerned with the validity of the results being valid beyond the context of the research (Bell, et al., 2019). In this respect, the variables selected for the study if taken individually to a large extent hold valid. However, as soon as the variables are combined, they may no longer hold because they are specific to the context of open IoT platforms.

Ecological Validity: Ecological validity concerns social scientific findings and whether they are applicable to naturally occurring everyday environments (Bell, et al., 2019). The only social scientific theory that the thesis assumes is the “open-source contribution” that has inherited social barriers which are difficult to measure.

3. Theoretical Frameworks and Concepts

This section describes the theories and concepts that were used as a base for the study. The said theories and concepts are derived from academic research as well as organizational research sources.

3.1 IoT (Internet-of-Things)

The definition of IoT has evolved ever since the nomenclature of the concept, which was first used in 1997. The thesis uses the definition of IoT that is widely accepted today by academia and commercial actors alike. The said definition of IoT was coined after extensive research in the book (published by IEEE) titled, *"Towards a definition of the Internet of Things (IoT)."*

The "Internet of things," commonly referred to as "IoT" envisions a self-configuring, adaptive, complex network that interconnects physical objects (commonly referred to as 'things') to the internet using certain communication protocols (Minerva, et al., 2015). The interconnected things may be physically or virtually represented in a digital environment having sensing or actuation capabilities, and programmability and are uniquely identifiable. The representation contains information about the object's identity, functionality status, location, and any other relevant information depending on the context of use. Based on the information, services can be provided with or without human intervention for capturing data, communication, and actuation. The services are provided via the use of intelligent interfaces that can be deployable anywhere, anytime also considering the security of performing the operations (Minerva, et al., 2015).

This definition used for the study is universally applicable to systems performing the mentioned operations and helps to create a solid distinction with similar concepts such as Cyber-Physical Systems (CPS) and Wireless Sensor Networks (WSN). The distinction is that CPS is concerned with the collaborative interaction of sensors to achieve a certain goal and uses an IoT system to perform the operations of collaborating the distributed systems, basically being a part of IoT. WSN can also be seen as a part of an IoT system that could be used to form a network of sensors to achieve coordination (Minerva, et al., 2015).

The thesis achieves the base for conducting the study by understanding the conventionally defined features of IoT. They are as follows (Minerva, et al., 2015):

Interconnection of Things: It pertains to the interconnection of physical objects (referred to as things) that are relevant in the context of the application or use.

Connection of things to the Internet: Points out that the things are connected to the internet and not intranet.

Uniquely Identifiable things: Points out that objects or things under consideration can be uniquely identified and can be used as such when required.

Ubiquity: It indicates that an IoT system can be seen as a network concept that is available and deployable anywhere, anytime (not necessarily globally), when the need arises.

Sensing/Actuation capability: Sensors/actuators are connected to the things and are responsible for making the "things" smart.

Embedded intelligence: Smart things can be programmed to perform tasks and functions autonomously.

Interoperable Communication Capability: IoT systems can communicate using standard and interoperable communication protocols.

Self-configuration ability: IoT systems can perform operations involving many heterogenous devices such as sensors, actuators, storage devices, utility monitoring devices, mobile phones, network elements, and computers. Currently, the challenge arises in managing the devices remotely. Although cloud-based control has enabled efficient management of devices, as the number explodes, the difficulty magnifies. Therefore, to utilize resources efficiently for an exceptionally large number of devices, self-configuration capabilities in terms of network selection and resource provisioning are the natural alternative that is gaining prominence. It is commonly referred to as edge computing capabilities (Shi & Dustar, 2016).

Programmability: IoT systems consist of 'things' that are programmable at the simplest level that can enable changes in behavior according to a user's need without making physical changes to the things.

It is observed that in a typical IoT ecosystem, the features may be delivered by different stakeholders. Broadly, the key stakeholders are, *Platform providers*, that deliver IoT-enabling products and services including integration services with third-party applications and tools and many other features. However, there are also some cases where the entire features of IoT are provided by the platform providers. *Application Providers*, offer specialized domain-specific technical solutions in which the IoT is being deployed. *Device Providers,* offer IoT devices, sensors, actuators, and related hardware needed for the IoT application. *Network Operators,* that provide connectivity infrastructure. (Vogel, et al., 2020)

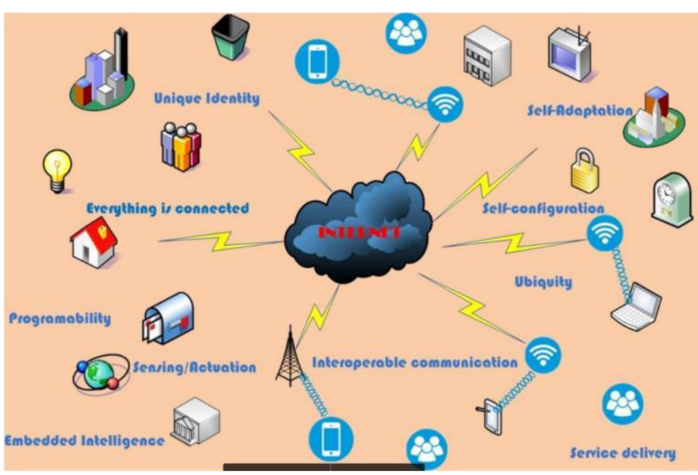

Figure 1: Features and Scope of an IoT System (Minerva, et al., 2015)

3.2 IoT Platform

"IoT Platform is a core management solution, interface between IoT devices and end users, which provides management of devices, connections, services, and data, usually through the cloud and local resources. The platform delivers data storage, management, analysis, and visualization, data security management (on users and devices), integration ability, as well as auditing and payments. On the other hand, the IoT platform offers some tools, APIs, and services for building deployment, and development of applications for IoT devices and end users." (Asemani, et al., 2019)

An IoT Platform is the element within an IoT ecosystem that enables central process management and provides vertical services to the end users by providing tools and computation capabilities for device-data lifecycle management from sensor networks to end users (Asemani, et al., 2019). The academia and the commercial actors have some differences when defining IoT platforms.

According to the research paper, *Understanding IoT Platforms,* presented by Asemani et, al. (2019) the academia, and the scientific community believe that broadly four elements- device operating systems, gateway, application development tools and platforms, central management services (which perform operations like connectivity management device management, storage, processing, analysis, and visualization) are considered as IoT platform. While there was no single definition available from a commercial actor perspective, it was stated in the research paper that there are various definitions depending on the features and functions and was derived from a survey of 13 IoT companies. The most used definition in the commercial context is that of an *"IoT platform being the central management service (with an addition of development tools) provided in the IoT ecosystem."* (Asemani, et al., 2019)

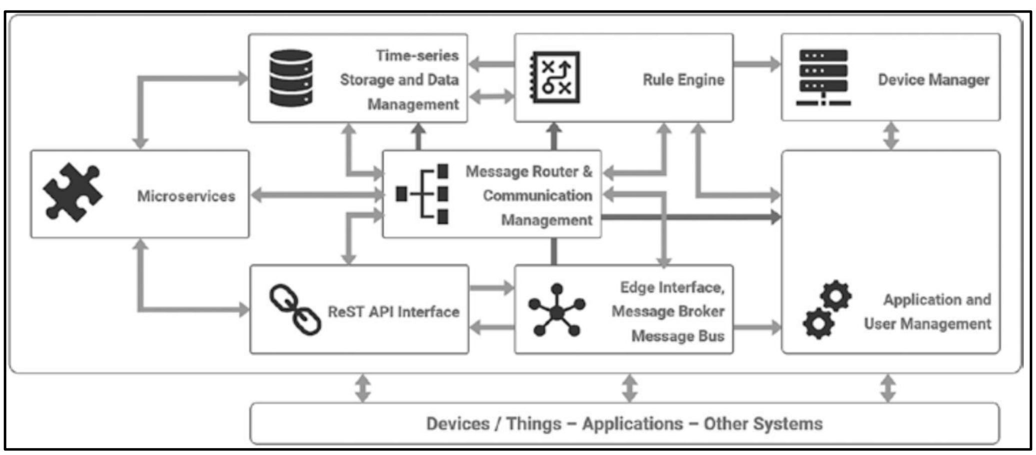

Figure 2: Block diagram of a typical IoT platform (Tamboli, 2019)

Characteristics of commercial IoT platforms as mentioned in the research paper "*Understanding IoT Platforms."* (Asemani, et al., 2019) are (1) connectivity/device management, (2) data storage and management, (3) data analysis and visualization, (4) development tools and platforms, (5) edge/local computing, (6) integration and interoperation, (7) Service management, and (8) auditing and payment. A description of the characteristics is given as follows:

Connectivity and Device Management: IoT platforms can form connections with the IoT ecosystem where they are deployed. The connections are formed using the internet, short-range communications (such as Wi-Fi, Bluetooth, etc.), cellular networks, and satellite communications. Communication is established with the devices via standard protocols (such as HTTP, MQTT, Bluetooth, etc.). The IoT platforms enable tracking and status management of the device, device configuration, and software/firmware updates, including mechanisms for error detection and self-repair.

Data storage, management, analysis, visualization: The typical application of an IoT system involves many devices. The devices are used to capture a large amount of raw data that needs to be stored, analyzed, and processed to get valuable information. This task becomes impractical to be performed on the devices as they are low-powered and would require auxiliary power sources. Alternative to this issue is the use of a centrally located storage, typically cloud storage. IoT platforms address these issues by providing cloud or local storage and analytical tools to process data to make sense of the information gathered and further enable informed decisions via visualization tools for an end user (often referred to as dashboards) such as graphs, and diagrams, etc.

Development and Deployment Tools: IoT platforms offer development tools for devices, Application Programming Interface (APIs), Software development kits (SDKs) as well as automatic application deployment on the devices that are suitable for one or more vertical IoT solutions (often referred to as use cases) such as smart home, smart energy, smart agriculture, etc. Often, IoT platforms also provide APIs for using their internal services and data to make it easier for customers/users to customize and adapt the IoT platform functions for their use case.

Service management: IoT platforms provide the ability to manage the services within a vertical solution based on specific parameters. Often services are referred to as microservices, they are smaller functions that contribute to the overall service management of the vertical solution. An example can be the ability to determine the location and control the area of operation of a vehicle on a shared/smart mobility platform.

Integration & Interoperation: IoT platforms can provide integration of other tools and platforms that an end user wants to use within the ecosystem. For example, an end user may want to use a data analytics tool to get better results from the data generated by the IoT ecosystem. Cloud integrations also enable better scalability of the platform (Asemani, et al., 2019).

Local/Edge Computing: IoT platforms improve response time in the IoT ecosystem by providing computing on the devices or even the communication gateways to the cloud. These local data processing capabilities reduce the storage requirements and resource costs for data transfer. For example, when data is generated by millions of devices, the response time for solving problems is increased, edge computing decreases this response time by deploying computation capabilities on the device or at the communication gateway of the device to the cloud.

Auditing and payment: IoT Platforms provide auditing and payment terminals for business users as well as end users. The terminals charge the end users based on certain parameters, the most common being data usage and data processing. This capability gives end users passive control over the costs of using the platform and enables cost prediction if the scale of the IoT system changes.

A visual representation of the criteria applied for the selection of an IoT platform for a business application is depicted in figure 3.

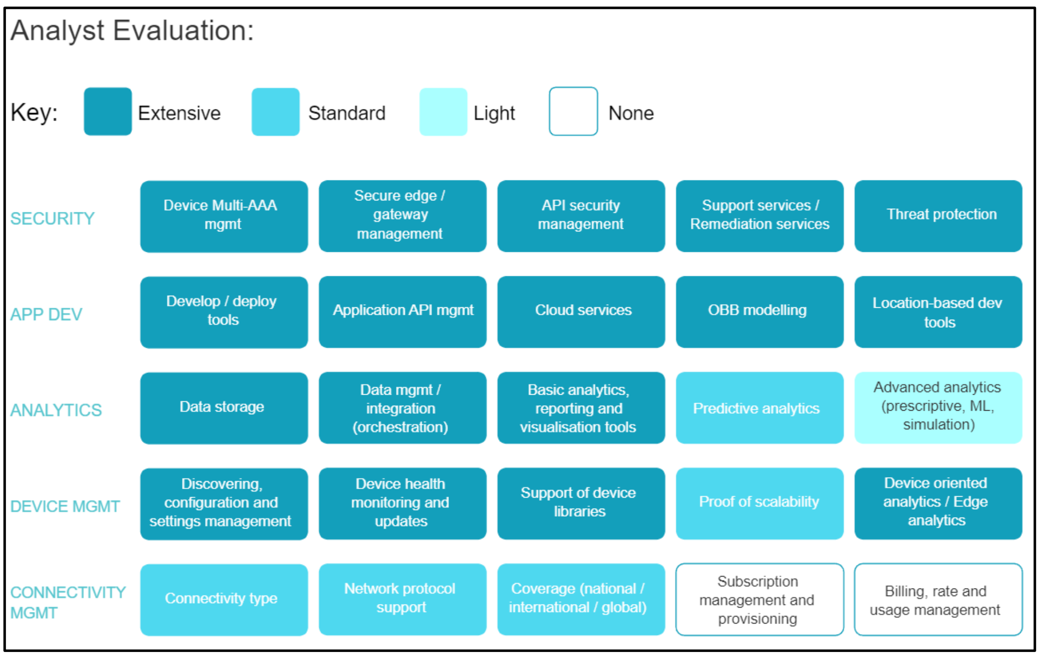

Figure 3: Selection criteria applied for platform selection in Industry (Beecham Research, 2021)

3.3 Open IoT Platform

An Open IoT Platform does not have a definition that can be used universally. The definition varies according to context. The contexts are driven by the types of openness observed in IoT Platforms. In general, it can be said that an Open IoT platform aims to solve the problems associated with closed (often referred to as proprietary) IoT platforms that typically have vertically integrated systems and are often fragmented (Vogel, et al., 2020). Open IoT platforms are associated with the ability to harmonize diverse IoT ecosystems in terms of making technical components compatible with a wide range of devices and development tools.

To understand the contexts and types of "open" IoT platforms, the thesis study relies on the gap analysis of closed (proprietary) IoT platforms with respect to open IoT platforms. The gap analysis and definitions of distinct types of openness are given by Vogel et al. (2020) in the paper titled *"What is an Open IoT Platform? Insights from a systematic mapping study."*

Openness types of IoT platforms according to the results of the study conducted by Vogel et al. (2020):

Opensource: IoT platforms that conform with the definition provided by the Opensource Initiative (OSI) to categorize themselves as open IoT platforms. The platforms comply with the ten requirements to classify themselves as being open-source.

Open Standards: IoT platforms that conform with the joint definition of open standards given by IEEE (Institute for Electrical and Electronics Engineers), ISOC (Internet Society), W3C (World Wide Web Consortium), IETF (Internet Engineering Task Force) & IAB (Internet Architecture Board). It involves five key principles that include, collective empowerment, availability, voluntary adoption, cooperation, and ability to serve different market requirements. Open standards are especially important for open IoT Platforms to offer support for heterogenous devices in an IoT ecosystem to enable better interoperability.

Open APIs: IoT platforms that offer open APIs. "Open APIs" refer to publicly available application programming interfaces that enable programmers to get access to a software application or web service. For IoT platforms, it generally means giving access to third-party developers to contribute to proprietary or even open software applications.

Open Data: IoT platforms that offer data that is freely available for everyone to use and republish without restrictions from any IP control mechanism.

Open Layer: IoT platforms that provide a software layer of their platform to integrate a third-party application.

The research by Vogel et al. (2020) also revealed that the definition of open IoT platforms depends on how the stakeholders involved in an IoT ecosystem use the platform. The study provides observations of the openness type that concerns each of the stakeholders.

- *Open-source* and *open standards* are important openness types for platform providers.
- *Open APIs* and *open standards* are important openness types for application providers.
- *Open standards* and *open-source* play a significant role for system integrators.

The most important takeaway from this open IoT platform theory is that the widely accepted openness type in open IoT platform is open-source as it is convenient for third-party developers to have full access to the source code to make a significant contribution to the open initiative of the IoT platform.

3.4 Open Platform Design Theory

The research publication *"Open Platform Design- Towards a Theoretical Framework and a Practical Toolbox"* by Daniel Särefjord (2006) reveals that openness in platforms can take various forms for Intellectual asset management to promote opportunities in value creation in commercial as well as societal aspects. Most importantly the research publication presents a conceptual framework (CAP model) to build open platforms along with exemplifying the concept by applying it to successful open platforms.

The *CAP model* is a practical tool to evaluate, compare and create open initiatives. (Särefjord, 2006)

CAP model tool is described as follows:

Claims: The parameter determines to what extent are the results of the development efforts in the platform being claimed as IPRs.

Availability: The parameter determines the route taken to get access to the results of the development efforts in the Platform. For example, how does the platform share the results of the open initiative, via, an open-source license, contracts, commercial license, etc?

Progress: The parameter determines the methods that the platform takes toward developing intellectual assets. For example, how does the platform control the direction of progress, via establishing roadmaps, raising specific issues, etc?

The CAP model also includes taxonomies that help identify the scale of openness within each parameter.

Levels to represent openness scale of *Claims:*

Public Domain: Represents the most complete openness in terms of claims and is without an owner. Information can be altered, redistributed, repackaged, and reclaimed without any limitation. Lapsed IPRs are considered information in the public domain.

Open Claims: Represents the claims that do not limit the use of results of the open initiative. The ownership can still be claimed for the assets that were created using the results of such an initiative. The open claims are imposed not for excluding others from using the assets but to ensure that the openness is maintained even when someone uses the asset.

Limiting Claims: Represent the range of claims that have an objective of guaranteeing - the openness of the asset to the exclusivity (propriety) of the asset.

Levels to represent openness scale of *Availability*:

Open: The results of the initiatives are indisputably and non-discriminatorily available to everyone on the same terms. Redistributable, irrevocably open without limiting access on economic terms or the characteristic of the user.

Share alike: The results made available by an initiative are under the conditions that the user of the results must share the results along with their contributions under the same conditions.

This type of sharing is governed by contractual agreements, often in the form of copyleft licenses in the open-source context. Share alike is an embodiment of the "limiting claims" concept stated before.

Non-commercial use: "Non-commercial use initiatives differentiate between non-commercial and commercial users, often referred to as "dual license". A common implementation is to apply "share alike" availability for non-commercial users and "non-free" availability (or even more restrictive availability) for commercial users through license agreements governing an initiative's claimed assets." (Särefjord, 2006).

Non-free: The results made available by the initiatives are made openly available on non-discriminatory terms, but a reasonable compensation is charged for usage (often in the form of royalty by licensing patents/IP.)

Levels to represent openness scale of *Progress*:

Open: Everyone and anyone is free to improve, extend as well as comment and review the output of the open initiative (Särefjord, 2006).

Semi-open/Semi-closed: This kind of initiative progress without limitations or restrictions on its user base and a formal approval process for project initiation. However, for a contribution to be included in the official project it must pass some sort of process approving its quality and suitability (Särefjord, 2006).

Controlled: *"An initiative is classified as controlled in terms of progress when there exists a process for controlling progress"* (Särefjord, 2006). The process may involve a central administration that approves the proposals for progress. The administration may also influence the formal working methods and may also restrict any progress that builds around and over the results of the initiative in question.

Driven: *"Driven progress initiatives share the project initiation, audit, and implementation process with controlled progress initiatives along with defined working methods."* (Särefjord, 2006). The difference between controlled and driven initiatives is that controlled initiatives have open and broad participation and develop solutions that fit the market requirements that all actors involved believe in. Whereas driven initiatives are closed in terms of participation and include a few participants and actors that have similar expertise and work towards developing an open solution with a biased view of what is more valuable or will be in the future.

Closed: One step further from driven, instead of being developed by a small, closed group under an open and collaborative setting, it is developed in a closed small group under a closed undemocratic setting. The progress is not open in any sense but, as long as the end result is made openly available according to any of the categories above and/or the results are claimed in accordance with the relative definition of open claims, an initiative will still be considered as "open" as a whole. (Särefjord, 2006).

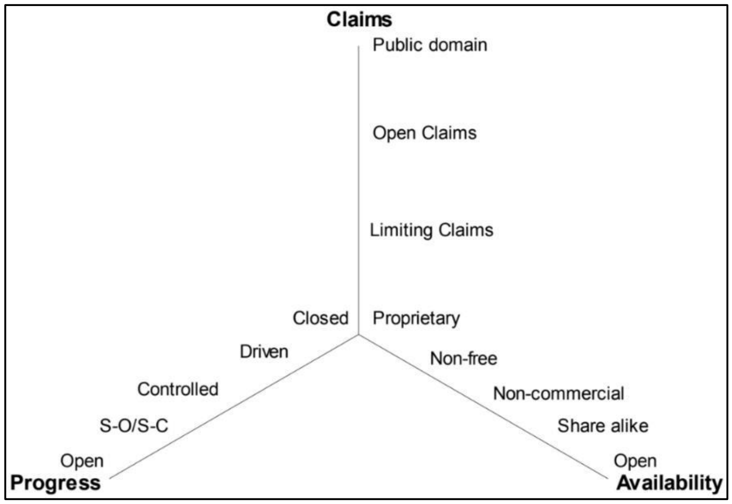

Figure 4: The CAP model to visualize the relative "openness" of open initiatives (Särefjord, 2006)

The CAP model is used to evaluate open initiatives to further describe an open-source toolbox by analyzing a few open initiatives for software development. However, the toolbox is non-exhaustive and does not include analyzing open initiatives of IoT platforms.

Table I: Opensource Toolbox (Särefjord, 2006)

Platform Concept	**Platform Construction Tool**
Management of Technical Development	Public domain technical development - Providing incentives for un-biased progress
	Modular development
	Community-based technical development
	Facilitated technical development with a coding style guide
	Error corrections are required to be given back to the community by all licensees
Management of Asset Portfolios	Public domain asset management
	IP portfolio management as a defensive measure
	Asset portfolio management by Open Interfaces – Proprietary Implementations
Management of Value Proposition	Packaging innovation in a reference implementation
	Contractual licensing structure using current assets to guarantee the freedom of future developments
	Contractual licensing structure governing freedom to choose proprietary and open
	Dual licensing structure – differentiated openness to software by type of use
	Dual licensing structure as order-winning utility
Management of Incentive Systems	Community- and recognition-based incentive structure
	Rapid response is an incentive for the community to act as beta testers.
	Open-source as a profit-oriented business model
	Open-source as support to core business

Management of Ownership Claims	Ownership clearance through participation contract
Management of Competencies	Branding as acquisition and governance of expert competence
	Merit-based hierarchy enabling governance of competences
	Open-Source community as a platform for corporate collaboration
Management of Physical and Virtual Platforms	Internet-based IT Development Platforms
	Central distribution platform for controlled innovation
Verification of Technical Solutions	Reference implementation as quality and compatibility verification tool
	Contribution evaluation process based on veteran hierarchy
	The verification process is based on hierarchy and merit
	Verification facilitation through a coding style guide
	Governing quality and cost-efficiency through using the community as beta-testers
	A corporate entity in control of verification and compatibility process
Organizational Structures and Functions	An internal formal process for conflict resolution; merit rules
	Open-source organized in a limited liability company

3.5 Open-source Archetypes

Mozilla Foundation is widely known as one of the most influential organizations that champion the cause of open-source software. With a plethora of successful open-source projects that involved a huge community contribution, the Mozilla projects have given insights into how to mobilize an open-source community and drive projects. The Mozilla Foundation released a research report-cum-guide that explains the decision-making process for an open-source project that they used internally that has been tried, tested, and proven. The guide titled *"Open-source Archetypes: A framework For Purposeful Open-source"* is used in the thesis study as a tool to identify the open-source approach in the case study as well as an element in the framework for articulating the approach that an IoT platform may adopt for achieving the desired results. Along with the approaches that one could take for their open-source project, the report also provides a questionnaire that helps to select the right approach, to articulate reasons for pursuing an open-source approach as well as the process for deployment and governance.

The report (Mozilla Foundation, 2018) provides a set of preliminary discussion points in the form of questions that the management of the open-source initiative must consider. The subject of the questions is tweaked to fit the context of IoT platforms and was used to analyze the open-source projects of the IoT platform on GitHub.

The theory provided by the Mozilla report is backed by a reasonable amount of evidence that the approaches are taken and questions to be considered stand relevant for the building and placement of an open-source project.

3.6 IoT Platform Reference Architecture

In this section, the reference technical architecture for IoT platforms is discussed and which will be used as a tool for analysing the technical components in the IoT platform chosen for the multiple case study.

The heterogeneity in the integration of IoT platform components gives rise to issues in comparison in the absence of a reference architecture. For example, different IoT platforms may provide equal functionality to be applied in each application field, but the style of implementation and the underlying technology used may differ and therefore leads to a difficulty in comparisons and selection. The difference is observed in the form of terminologies as well such as the terms 'things' and 'devices' being used interchangeably. Figure 5 gives a representation of the reference architecture which will be used as a benchmark for analyzing the IoT platform technical components for the case study.

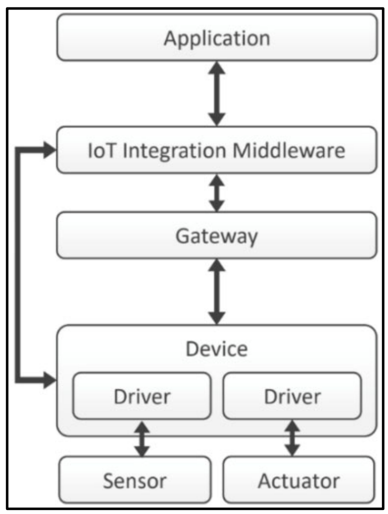

Figure 5: IoT Reference Architecture (Guth, et al., 2016)

The reference architecture is a result of a study that compares the architecture used by several IoT platforms. It was published in the paper *"Comparison of IoT Platform Architectures: A Field Study based on a Reference Architecture"* by Guth et al. (2016). The abstract architecture has been used previously in many contexts for conducting case studies about IoT platforms.

3.7 Software-as-a-Service Business Models

The business of opensource software has evolved in the past decade and there is an observation that there are four types of opensource business models that have been proven successful by most of the open source businesses in operation (Imran Ghory, 2020). IoT platform services are delivered as a software-as-a-service (SaaS) product. The successful business that has taken the open-source approach in the business models also delivers value as (software-as-a-service) SaaS product e.g., MongoDB, Elastic, Confluent, etc. Most of these businesses have been funded by investors that have repeatedly invested in opensource-based start-ups and have essentially helped these start-ups to navigate the opensource business due to their expertise (Index Ventures, 2021). The in-depth analysis of these business models is beyond the scope of the thesis study. The thesis merely uses the understanding of the business models to derive learnings and the characteristics of the business models and use them as parameters to evaluate whether the open-source IoT platforms selected for the case study follow these business models. Therefore, the thesis derives insights from the articles, corporate white papers, and conferences published by corporate authors that have been a part of investing in open-source businesses. This approach is deemed more precise and useful for the current scope of the thesis.

The business models are categorized as follows:

Open-core: The open-core business model in the past decade has emerged as the most successful model owing to the more complementary placement of the commercial product with respect to the open-core product. The open-core product usually is the main functional entity of the product that is available as a free-to-use opensource product as well as an option of a commercial product. The company's main source of income is the payment it receives from a customer that uses the commercially licensed product which is essentially the opensource product without the obligation to contribute back to the opensource project. In simple terms, the commercial product complements the open-source product rather than being conflicting. This aspect of being complementary is cited as being the motivation for the community to contribute to the project (Imran Ghory, 2020). Examples of recent, successful businesses using the open-core business model are Elastic, Confluent and GitHub.

Proserv (Professional services): A significant number of opensource businesses have achieved scale using this model, however, the insights from the industry show that the model has difficult challenges in terms of irregular revenue and high employee costs, and resource overhead. The main elements in the Proserv business model are support services and additional product features offered as paid services over the open-source product. Examples of successful businesses using the Proserv business model are Redhat, Cloudera, and Hortonworks.

Hosting: Hosting services as a source of income have become popular among open-source companies that offer SaaS products and that function around big data and infrastructure capabilities. In this case, the users of the open-source software make decisions based on the cost of availing of the hosting services by evaluating how much difference exists between building the infrastructure for self-hosting against paying for the hosting services. Hence, the pricing of the hosting services determines the willingness to pay. It is also observed that hosting services are the secondary source of income and are usually combined with the open-core and Proserv business models. Examples of successful businesses having hosting services as a major income stream are WPEngine and GitHub.

Marketplace: This model has been relatively rare in use but is gaining prominence with new start-ups and existing businesses including the marketplace approach in their businesses. The business model essentially offers multiple paid applications that add value to the open-source product. The applications may be in the form of APIs, optimized analytical tools, etc. that connect easily with the opensource product and fill the void for a user of the opensource product. Google's Android is an example of a successful opensource business that has adopted a solely marketplace-based business model.

There has been an observation of companies using a combination of the above-given business models albeit implementing the model in different proportions and according to the scale of users (Imran Ghory, 2020). However, the most common pattern is companies using the open-core model primarily with the Proserv and Hosing being added in the later stage as secondary and tertiary revenue sources. The insights from the Index venture summit (2021) where multiple founders of companies and representatives from the opensource community participated cite that the business model is affected by the goals of the founders, the perspective of the community, and engagement of the project owners and the nature of the product.

The mentioned business models are assumed as a theory to evaluate whether the selected opensource IoT platforms for the case study follow one or a combination of the above-mentioned business models

3.8 Open IoT Platform Business Archetypes

The overall understanding is about the competition in the business ecosystem of IoT. The competition exists the most between platform ecosystems rather than individual businesses. That is, the competition is mostly between the networks of complementary service providers and end users and does not necessarily include competition between the platform providers. (Smedlund, et al., 2018)

To capture value from IoT business ecosystems the open IoT platform firms may establish a presence as (i) Service System Owners, (ii) Complementors, (iii) Module Producers, or (iv) System of Systems Facilitators. The roles may evolve over time according to increase in scale and as technology progresses and the competitive advantage managed via increased specialization and value proposition targeted towards end users rather than the customer of the service. (Smedlund, et al., 2018)

The description of each archetypical business strategy observed by Smedlund et .al (2018) in open IoT platforms is as follows:

(i) **Service system owner** strategy: The business where operations are semi-closed proprietary system or controlled set of standards with specific actors bound by contractual agreements with the customer to control growth and evolution of the resulting service system. The scope of operation is centered on a specific use case. A focused strategy applicable to the early introduction and even growth. (Smedlund, et al., 2018)

(ii) **Complementor** strategy: The business in which use case specific services or service-based products are sold through systems owned by other companies or channels. The channels may be in the form of standards or a value chain specific to the use case. The business has fixed contract with customers of the platform with less control over the system and therefore provide services reactively rather than proactively and usually cannot apply the same solution to another context. The product or service developed for the customer is typically standardized and fits in the customer's system without any shared processes or visibility of the customer's systems. It is a strategy suitable for small organizations which can be scaled to module producing strategy to be shared by open interfaces. (Smedlund, et al., 2018)

(iii) **Module Producer** strategy: The technical solutions in the open IoT platform are built to be interoperable and integrable into all kinds of systems of a customer through a standard interface (e.g. via APIs). The module producer has little control over the deployment of the technical solutions in the system at the customer end. The modules are typically stand-alone functions. Revenue is generated via the sale of a license for the product and is often dependent on economies of scale and limited product lifecycle. (Smedlund, et al., 2018)

(iv) **System of Systems Facilitator** strategy: The main product offered by the open IoT platform in this case is defined interface standards, typically APIs. The platform and the customer loosely collaborate for the deployment with joint offerings but in an ad hoc manner rather than consciously planned. The customers and the actual services that the APIs connect pay rent to the IoT platform. The revenue generated is usually non-profit but network effects due to the collaboration enable the platform to gather customer interest for other services, and the ecosystem thus created benefits from emerging innovation. (Smedlund, et al., 2018)

The description of the strategies resembles the complex nature of adaptive systems that IoT-driven businesses can adapt. Through this literature review, it can be inferred that IoT platforms develop other IoT business strategies that differ from the dominant product strategy of pushing a complete IoT platform.

4. Results

4.1 Evaluation of Open initiatives in IoT platforms

IoT platforms that were chosen for evaluation:

Thingsboard: Thingsboard.io is an open-source IoT platform. The platform promotes itself as an IoT platform for rapid development, management, and scaling of IoT projects, specializing in device management, data collection, processing, and visualization for IoT solutions. The platform also engages in open innovation with businesses and partners using B2B open source and open APIs.

Thinger: Thinger is an open-source IoT platform. The platform promotes itself as a ready-to-go infrastructure for device management, storing, monitoring, and analyzing data from multiple IoT sources. The platform engages in open innovations with other IoT services and IoT hardware providers where the open-source community is also involved.

Openremote: Openremote is an open-source IoT platform. The platform promotes itself as being the only "100%" open-source IoT platform. It specializes in device control, data visualization, and creating customized applications. The platform was built using the open source community and includes customers for driving the development of the platform as an open source project.

KaaIoT: KaaIoT is an enterprise IoT platform but, built using the Kaa project- an open-source project run by the same company. It promotes itself as being the most flexible platform for any business. Although lately not focused on their open-source development project, they add features from time to time to the project to keep the community engaged.

Particle: Particle is an enterprise IoT platform. The platform promotes itself as a full-stack fully integrated platform-as-a-service that is reprogrammable and reconfigurable enabling a custom-made application possible. The platform uses open-source hardware and develops software using an open-source community for optimizing the hardware for IoT applications. In addition, it engages in open innovation by providing customers with open APIs for customizing components.

4.1.1 Thingsboard

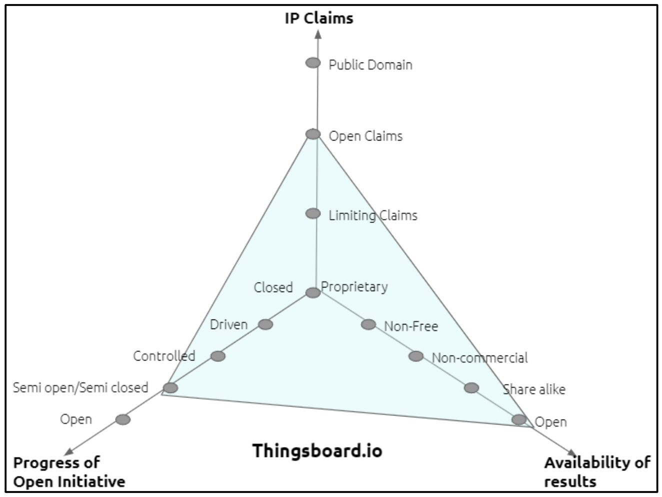

Figure 6: Relative openness of the Thingsboard IoT platform according to the CAP model.

Introduction: Thingsboard is an open-source IoT platform. It includes all the basic components to be classified as a universal IoT platform and partners with some IoT device providers to simplify the implementation of IoT solutions. The platform is organized by a limited liability company under the name of Thingsboard, Inc. It was founded in 2016 and has about 50 employees. The product is an IoT platform offered in three forms- community edition, Professional edition, and cloud edition. All three forms of the product are underpinned by the open-source project licensed under Apache 2.0 open-source software license that allows the use of the project commercially and profit-oriented. The 'community edition' product is the open-source project packaged for free users (albeit, with default Thingsboard branding) with some paid services offered for guidance in deployments. The 'professional edition' product is a forked version of the open-source project that a customer has to pay to use. It contains added functionality over the community edition and allows a customer to rebrand the product. The 'cloud edition' product allows a customer to host and deploy the IoT solution using Thingsboard's cloud hosting service. For both the paid versions of the IoT platform, Thingsboard shares open APIs for components of the IoT platform for enabling customization of the technology stack to suit the use case of a customer. The platform claims to have ready-made solutions for nine use cases and is mentioned in many articles on the web as being the best-in-class open-source IoT platform.

Business models:

Thingsboard revenue model: A combination of open-core, professional services, hosting, and marketplace.

- Sale of 'professional edition' built over the open-source project, which is essentially the core of the product.

- Professional services for the development and deployment of IoT solutions.
- Online support for cloud IoT platform subscription services sold worldwide.
- Cloud hosting and storage services.
- Marketplace for in-house developed tools and integrations for third-party tools.

***Incentives for businesses to use the Thingsboard IoT platform*:**

- Flexibility and freedom to choose the type of product and possible customizations via sharing of open APIs.
- An open-sourced core product ensures timely updates and follow-on innovations.
- A communication gateway product built using open-standards-based communication protocols and placed as a separate open-source project ensures timely updates.
- Partnerships with hardware providers allow for optimizing the device operating systems to work well with IoT platform software and further allow the development of communication gateways to connect with the devices.
- A deployment guide and rules engine enable a free user to deploy an IoT solution based on the open-source project or community edition product for their own use case before becoming a paid customer.

Incentives for the community to contribute to Thingsboard open-source project:

- Open-sourced core product is an incentive for community use and therefore contribute.
- IoT development tutorials and education, development tools, project roadmaps, software development guidelines, and timely responses on community forums attract users and developers to contribute to the open-source project.

Table 2: Identified Platform construction tools in the Thingsboard IoT platform

Concepts	**Platform Construction Tool** (Särefjord, 2006)	**Description of tools observed**
Management of Technical Development	Modular development	Separated components in the open-source project according to functions in IoT deployment.
	Community-based technical development	Project roadmaps for directing community
	Facilitated technical development with a coding style guide	Documents stating contribution guideline
Management of Asset portfolios	Public domain asset management	On GitHub platform and documentation on the website.
	IP portfolio management as a defensive measure	Only via default branding on user applications.
	Asset portfolio management by Open Interfaces for proprietary implementations	Access to proprietary tools and applications via Open APIs.
Management of Value Proposition	Packaging innovation in a reference implementation	Free prototyping is provided to attract customers.

	Contractual licensing structure using current assets to guarantee the freedom of future developments	(*Data can only be derived from interviews)
Management of Incentive systems	Contractual licensing structure governing freedom to choose proprietary and open	(*Data can only be derived from interviews)
	Open-source as a profit-oriented business model	Open-source project underpinning products paid professional service provided for users of the community edition of the product.
Management of physical and virtual platforms	Internet-based IT Development Platforms	GitHub platform used for collaborative development.
Management of	Reference implementation as a quality and compatibility verification tool	Free deployment guidance for use of IoT platform
Verification of technical solutions	A corporate entity in control of the verification and compatibility process	In-house developers act as quality testers before including community contributions to the main project.
Organizational structures and Functions	Open-source organized in a limited liability company	The GitHub project includes a commercially licensed IoT platform governed under Thingsboard, Inc.

4.1.2 Thinger

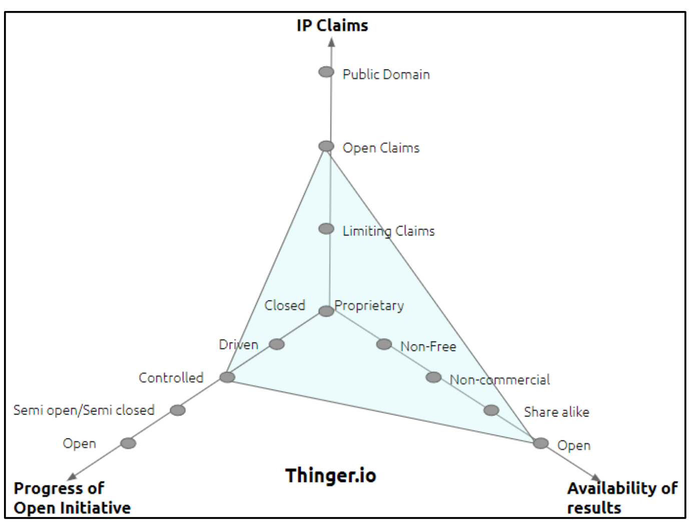

Figure 7: Relative openness of Thinger IoT platform according to the CAP model.

Introduction: Thinger is an open-source IoT platform. All the basic components of the IoT platform are placed in one single open-source project. The IoT platform is organized under a limited liability company named Internet of Thinger S.L which was founded in 2018 and has about 10-12 employees. There is no product as such, but an open-source IoT platform that is placed as an open-source project with an MIT license that allows commercial, profit-oriented use of the project. The open-source project acts as a medium for the company to win orders for consultancy projects to develop IoT solutions. The platform claims to have built expertise in IoT solutions for five use cases. Thinger has created a professional network by partnering with IoT service providers (tools and applications) and hardware providers for developing the IoT platform, these partners provide open APIs to be connected with the Thinger IoT platform.

Business model:

Thinger revenue model: A combination of open-core, professional services, and hosting.

- Possible sale of ‘professional edition’ built over the open-source project, which is essentially the core of the product.
- Professional services for the development and deployment of IoT solutions.
- Cloud hosting and storage services.

Incentives for businesses to use the Thinger IoT platform:

- Open-core ensures timely updates and follow-on innovations in the product.
- A communication gateway product built using open-standards-based communication protocols.
- IoT development tools, software development guidelines, and timely responses on community forums attract users and developers to contribute to the open-source project.
- Partnerships with other IoT service providers and hardware providers ensure up-to-date technology is included in the open-source project.

Incentives for the community to contribute to Thinger open-source project:

- Open-sourced core product is an incentive for community use and therefore contribute.
- IoT development tools, project roadmaps, software development guidelines, and timely responses on community forums attract users and developers to contribute to the open-source project.
- Partners allow the use of IoT tools when developing for the Thinger platform which is otherwise difficult or expensive to access.

Table 3: Identified Platform construction tools in the Thinger IoT platform

Concepts	Platform Construction Tool	Description of tool observed
Management of Technical Development	Modular development	Separated components in open source project according to functions in IoT deployment.
	Community-based technical development	Community allowed to contribute to the project informally, or formally by following the product development roadmap.
	Facilitated technical development with a coding style guide	Documents stating contribution guidelines.
Management of Asset Portfolios	Public domain asset management	Assets are placed and managed on the GitHub platform.
	Contractual licensing structure using current assets to guarantee the freedom of future developments	(*Data can only be derived from interviews)
	Contractual licensing structure governing freedom to choose proprietary and open	(*Data can only be derived from interviews)
Management of Value Proposition	Packaging innovation in a reference implementation	Free prototyping is provided to attract customers.
	Contractual licensing structure using current assets to guarantee the freedom of future developments	(*Data can only be derived from interviews)
	Contractual licensing structure governing freedom to choose proprietary and open	(*Data can only be derived from interviews)
Management of incentive systems	Rapid response is an incentive for the community to act as beta testers.	Self-hosted community forum enables rapid responses to issues faced by the community and free users.
	Open-source as a profit-oriented business model	Professional services are provided over the use of the open-source project.
	Open-Source community as a platform for corporate collaboration	The partners of the platform also contribute to the open-source project or provide open API to integrate with the platforms for collaborative development.
Management of physical and virtual platforms	Internet-based IT Development Platforms	GitHub is used for collaborative development.

Verification of technical solutions	Reference implementation as a quality and compatibility verification tool	Free deployment guidance for use of IoT platform
	A corporate entity in control of the verification and compatibility process	In-house developers act as quality testers before including community contributions to the main project.
Organizational Structures and functions	Open-source organized in a limited liability company	Yes, the GitHub project is governed under Thinger.io which is intern governed under the Internet of Thinger S.L

4.1.3 Openremote

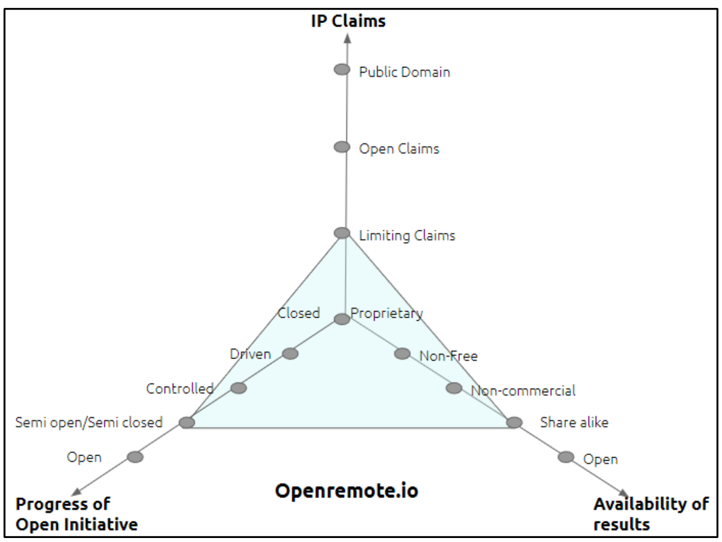

Figure 8: Relative openness of Openremote IoT platform according to the CAP model.

Introduction: Openremote is an open-source IoT platform. The open-source IoT platform is governed under a corporate entity of OpenRemote, Inc and has about 12-15 employees. The product is a free-use IoT platform with an AGPL 3.0 copyleft open-source license offered along with a commercial license. The copyleft license obligates anyone who uses the open-source licensed product to contribute back to the open-source project in case any modification and development is made over the original project. The company provides professional services for developing IoT solutions and cloud hosting services.

Business model:

Openremote revenue model: A combination of open-core, professional services, and hosting.

- Sale of a commercial license for IoT platform.
- Professional services for the development and deployment of IoT solutions.
- Cloud hosting and storage services.

Incentives for businesses to use the Openremote IoT platform:

- Open-core ensures timely updates and follow-on innovations in the product.
- Open-standards-based communication protocols.
- Open-sourced IoT development tools, software development guidelines, and timely responses on community forums attract users and developers to contribute to the open-source project.

Incentives for contributing to Openremote open-source project:

- Open-sourced core product is an incentive for the community to contribute.
- Open-sourced IoT development tools, software development guidelines, and timely responses on community forums attract users and developers to contribute to the open-source project.

Table 4: Identified Platform construction tools in the Openremote IoT platform

Concept	Platform Construction Tool	Description of tool observed
Management of Technical Development	Modular development	Separated components according to functions in IoT deployment.
	Community-based technical development	The community was allowed to contribute to the project informally.
	Facilitated technical development with a coding style guide	Documents stating contribution guidelines.
	Error corrections are required to be given back to the community by all licensees	AGPL 3.0 copyleft license governing the open-source project obligates sharing modifications.
Management of Asset portfolios	Public domain asset management	Assets are placed and managed on the GitHub platform.
	IP portfolio management as a defensive measure	Contributor License Agreement
Management of Value proposition	Contractual licensing structure using current assets to guarantee the freedom of future developments	(*Data can only be derived from interviews)
	Contractual licensing structure governing freedom to choose proprietary and open	(*Data can only be derived from interviews)

	Dual licensing structure as order-winning utility	AGPL 3.0 is a copyleft license that obligates giving back innovations therefore businesses may choose to use the commercial license for the IoT platform.
Management of incentive system	Rapid response is an incentive for the community to act as beta testers.	Self-hosted community forum enables rapid responses to issues faced by the community and free users.
	Open-source as a profit-oriented business model	Professional service is provided even for users of open-source IoT platforms.
	Ownership clearance through participation contract	Open-source license allows right for commercial use.
Management of Physical and virtual platforms	Internet-based IT Development Platforms	GitHub is used for collaborative development.
	Central distribution platform for controlled innovation	GitHub workflow tool used for assigning sub-projects for development
Verification of Technical Solutions	Reference implementation as a quality and compatibility verification tool	Free deployment guidance for use of IoT platform
	A corporate entity in control of the verification and compatibility process	In-house developers act as quality testers before including community contributions to the main project.
Organizational Structures and Functions	Open-source organized in a limited liability company	Yes, the GitHub project is governed under openremote.io which is intern governed under OpenRemote Inc.

4.1.4 KaaIoT

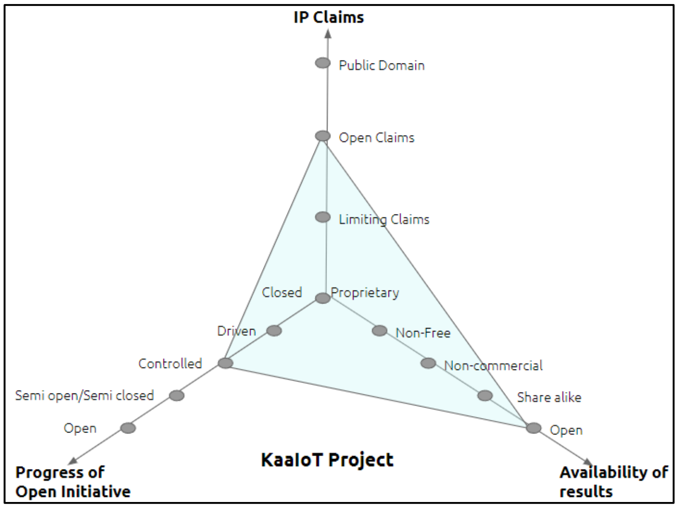

Figure 9: Relative openness of KaaIoT project according to the CAP model.

Introduction: KaaIoT is an enterprise IoT platform that started as an open-source project named Kaa project. Both, the enterprise IoT platform and the open-source project are governed under the company named Cybervision, Inc and has about 60-65 employees. The basic software that underpins the enterprise IoT platform is the open-source project which was built with an open-source community backing. The open-source project is licensed under Apache 2.0 license that permits commercial and profit-oriented use. Therefore, it enabled the owners of the open-source project to build a proprietary/enterprise version over the open-source project. The enterprise version consists of many added functionalities to the open-source project. However, the open-source project is still live and the platform owners keep updating the code on the open-source project. The open-source project is now a way to attract customers to adopt the enterprise Kaa IoT platform as it enables a customer to test the platform and see whether the technology used is suitable for their own IoT solution.

Business models:

- Sale of a commercial license for IoT Platform product.
- Professional services for the development and deployment of IoT solutions.
- Cloud hosting and storage.

Incentives for businesses to use KaaIoT (enterprise platform) :

- Complete ready-to-use IoT platform components.
- Inherent device interoperability (achieved via a community-based open-source project) complemented by open-standards-based communication gateways.
- Free, limited use trial of the product, with visible source code of the basic components of the platform in the open-source project.

- Open-source developer tools and in-built third-party integrations for developers to customize the commercially licensed product.

Incentives for the community to contribute to the Kaa project (open-source) :

- Open-core, which forms the underpinning of the enterprise platform is an incentive for free users and the community to contribute.
- Active responses on the community forum.

Table 5: Identified Platform construction tools in the Kaa IoT platform and Kaa Project

Platform Concept	**Platform Construction Tools**	Description of tool observed
Management of Technical Developments	Modular development	Separated technical components according to functions in the open-source project.
	Community-based technical development	IoT platform technical development started as an open-source project on GitHub.
	Facilitated technical development with a coding style guide	Documents stating contribution guidelines for an open-source project.
Management of Asset Portfolios	Public domain asset management	GitHub platform used for collaborative development and asset management for the open-source project
	Asset portfolio management by Open Interfaces – Proprietary Implementations	Open API (Application Programming Interface) based access for licensed users of enterprise products, enabling customization, yet control of the asset.
Management of Value Proposition	Packaging innovation as a reference implementation	Free prototyping is provided to attract customers.
	Contractual licensing structure using current assets to guarantee the freedom of future developments	(*Data can only be derived from interviews)
	Contractual licensing structure governing freedom to choose proprietary and open	(*Data can only be derived from interviews)
Management of Incentive Systems	Rapid responses as an incentive for the community to act as beta testers.	Active response on community forums- GitHub, StackOverflow and self-hosted forums.

	Open-source as support to core business	The open-source project acts as a way to get access to innovations and attract users that could turn into customers.
Management of Ownership Claims	Ownership clearance through participation contract	Ownership clearance through terms of the open-source license, and added contributor license agreements.
Management of Physical and Virtual Platforms	Internet-based IT Development Platforms	GitHub is used for collaborative development.
	Central distribution platform for controlled innovation	Self-hosted forums were used for assigning and managing workflows in the open-source project.
Verification of Technical Solutions	Reference implementation as quality and compatibility verification tools	Open-source project packaged in a reference implementation for easy deployment and prototyping.
	Corporate entity in control of verification and compatibility process	In-house developers are responsible for quality checks and the inclusion of code from community contributions.
Organizational structures and functions	Open-source project organized in a limited liability company	Both, the enterprise IoT platform and open-source project are organized under Cybervision, Inc. (LLC)

4.1.5 Particle

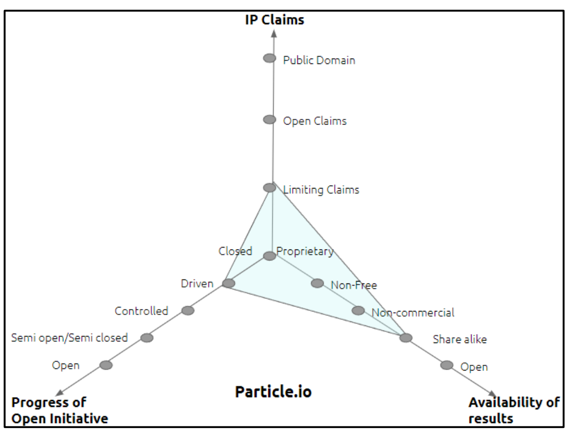

Figure 10: Relative openness of Particle's open initiative according to the CAP model.

Introduction: Particle IoT platform is an enterprise IoT platform that sells a license to a product (IoT platform and complementing tools) that is delivered in the form of software as a service. It has a big presence in the IoT platform provider space and is believed to be one of the hyper-scalers in IoT, competing with the likes of Amazon Web Services, Microsoft, and Google. The core IoT platform software is proprietary and is sold alongside some development tools and hardware that are open-source-based products. An open initiative, the platform mainly shares component-level open APIs with customers to enable customization. In addition, the software for device operating systems and tools complementing IoT platform deployments are placed as open-source projects. The IoT platform and all the open-source projects are organized under the company named Particle Industries, Inc, founded in 2012 and has about 180-185 employees.

Business models:

- Sale of a commercial license for IoT Platform product.
- Professional services for the development and deployment of IoT solutions.
- Cloud hosting and storage.
- Sale of a commercial license for packaged software and open-source-based hardware.

Incentives for contributing to Particle's open-source projects:

- Open-source projects for device operating systems are designed to adopt open-source-based hardware to be used in IoT applications. Therefore, the community is likely to contribute as an incentive of not building a completely new software themselves.
- Active responses on self-hosted community forums.

Table 6: Identified Platform construction tools in the Particle IoT platform

Platform Concept	Platform Construction Tools	Description of tool observed
Management of Technical Developments	Public domain technical development - Providing incentives for unbiased progress	Open-source projects targeted toward building device operating systems for adapting open-source hardware for IoT applications.
	Modular development	Modules in open-source projects are separated according to functions.
	Community-based technical development	The open-source project is only for developing hardware-related software and some open-source-based IoT development tools adapted for the Particle IoT platform.
	Facilitated technical development with a coding style guide	Documents stating contribution guidelines
	Error corrections are required to be given back to the community by all licensees	GPL 3.0 copyleft license obligates sharing of corrections. Getting corrections to software by default by sharing open APIs with customers.
Management of Asset Portfolios	Public domain asset management	GPL3.0 open-source license
	IP portfolio management as a	Trademarks for user applications and

	defensive measure	tools
	Asset portfolio management by Open Interfaces – Proprietary Implementations	Open API (Application Programming Interface) based access for customers.
Management of Value Proposition	Packaging innovation in a reference implementation	Free prototyping is provided to attract customers.
	Contractual licensing structure using current assets to guarantee the freedom of future developments	(*Data can only be derived from interviews)
	Contractual licensing structure governing freedom to choose proprietary and open	(*Data can only be derived from interviews)
Management of Incentive Systems	Rapid responses as an incentive for the community to act as beta testers.	Active response on self-hosted community forums. Providing free hardware for developments and experimentation.
	Open-source as support to core business	The open-source project was used to optimize prototyping and validate the IoT device.
Management of Ownership Claims	Ownership clearance through participation contract	Contributor agreements that allow commercial use of open-source projects.
Management of Competencies	Branding as acquisition and governance of expert competence	Particle has made a reputation as a provider of education and support for learning IoT development.
Management of Physical and Virtual Platforms	Internet-based IT Development Platforms	GitHub is used as a collaboration platform for open-source projects.
	Central distribution platform for controlled innovation	Self-hosted forums for assigning and managing work related to software development in community open-source projects.
Verification of Technical Solutions	Reference implementation as quality and compatibility verification tools	A prototyping kit is provided free of cost to try and test the IoT platform for a use case.
	Governing quality and cost-efficiency through using the community as beta-testers	Developers acquiring education for IoT development act as beta testers when deploying software for trying the platform for different use cases.
	A corporate entity in control of the verification and compatibility process	GitHub pull requests are checked by project owners before including code in the project.
Organizational Structures and Functions		
	Open-source organized in a limited liability company	The IoT platform and the open-source projects are organized under Particle Industries, Inc

4.2 Summary of Results

4.2.1 Observations with Platform Construction Toolbox

Table 7: Summary of observed platform construction tools

Platform Concept	**Platform Construction Tools mentioned by** (Särefjord, 2006)	**Platform construction tools as observed in selected IoT platforms**	**IoT platforms using the tool**
Management of Technical Developments	Public domain technical development - Providing incentives for unbiased progress	Open-source projects targeted toward building device operating systems (software) for adapting open-source hardware for IoT applications.	Particle
	Modular development	Separated technical components on opensource development platform (GitHub)	All platforms
	Community-based technical development	modules separated according to the product development roadmap	All, except Particle
		Opensource projects- GitHub	All, except Particle
	Facilitated technical development with a coding style guide	Prototyping projects- Self-hosted forum	Particle IoT platform
		Documents stating contribution guidelines for open-source projects.	All platforms
	Error corrections are required to be given back to the community by all licensees	Copyleft open-source license	Openremote
Management of Asset Portfolios	Public domain asset management	On GitHub, using open-source licenses.	All platforms
	IP portfolio management as a defensive measure	Default branding using trademarks for user applications and tools	All, except Openremote
		Contributor agreements are included in open-source license.	Openremote
	Asset portfolio management by Open Interfaces – Proprietary Implementations	Open API (Application Programming Interface) based access.	KaaIoT enterprise platform, Particle

Management of Value Proposition	Packaging innovation is a reference implementation	Free prototyping is provided as a way to attract customers.	All platforms
	Contractual licensing structure using current assets to guarantee the freedom of future developments	(*Data can only be derived from interviews)	
	Contractual licensing structure governing freedom to choose proprietary and open	(*Data can only be derived from interviews)	
	Dual licensing structure – differentiated openness to software by type of use	Un-conditional use, commercial use as well	All platforms allow
	Dual licensing structure as order-winning utility	The commercial license offered to stay free from the obligation to contribute to the open-source project.	Openremote
Management of Incentive Systems	Community- and recognition-based incentive structure	*Only some hints were observed on the community forums that point out contributions are rewarded.	Thingsboard, Openremote
	Rapid responses as an incentive for the community to act as beta testers.	Active response on self-hosted community forums.	Thingsboard, Thinger, KaaIoT project
	Open-source as a profit-oriented business model	An open core product, professional services	All, except Particle
	Open-source as support to core business	Hardware related opensource software development	Particle, KaaIoT
Management of Ownership Claims	Ownership clearance through participation contract	Contributor licensing agreements	Openremote
		Contract for collaborative development with customers	(*Data can only be derived from interviews)
Management of Competencies	Branding as acquisition and governance of expert competence	(*Data can only be derived from interviews)	
	Merit-based hierarchy enabling governance of competences	(*Data can only be derived from interviews)	
	Open-Source community as a platform for corporate collaboration	(*Data can only be derived from interviews)	
Management of Physical	Internet-based IT Development Platforms	GitHub platform is used for collaborative development in open-source projects.	All platforms

and Virtual Platforms		Command line interfaces (CLI) for ease in writing and deploying code on hardware.	Particle
	Central distribution platform for controlled innovation	Self-hosted forums for distribution of work.	All platforms
Verification of Technical Solutions	Reference implementation as quality and compatibility verification tools	Free use prototyping, including cloud hosting	All except Openremote
	Contribution evaluation process based on veteran hierarchy	(*Data can only be derived from interviews)	
	The verification process is based on hierarchy and merit	(*Data can only be derived from interviews)	
	Verification facilitation through a coding style guide	(*Data can only be derived from interviews)	
	Governing quality and cost-efficiency through using the community as beta-testers	Free prototyping tool kit provided for community developers. (*for other platforms, data can only be derived from interviews)	Particle
	A corporate entity in control of the verification and compatibility process	GitHub pull requests are checked by project owners before including code in the project	All platforms
Organizational Structures and Functions	An internal formal process for conflict resolution; merit rules	(*Data can only be derived from interviews)	
	Open-source organized in a limited liability company	Open-source projects are governed under a limited liability company.	All platforms

4.2.2 Observations of Variables Derived from Theoretical Framework

Table 8: IoT Platforms observed with variables identified in theory

	Thingsboard	Thinger	Openremote	Kaa IoT Project	Particle
Basic functional components of IoT platforms:	Yes	Yes	Yes	Yes	Yes
Openness types - Open-source - Open APIs - Open standards-based opensource implementations	All types	All types	All except, Open APIs	All types	Open-source Open APIs
Opensource logic (Community/stakeholder)	Community and stakeholder	Community and stakeholder	Community	Community	Community
The open-source project in the public domain contains	Core software product, developer tools	Core software product, developer tools	Core software product	Core software product	Hardware-related software products, and developer tools
Licenses for Opensource projects - Permissive (OSI approved) - Copyleft (OSI approved) - Other	Apache 2.0 (permissive)	MIT (permissive)	AGPL 3.0 (copyleft)	Apache 2.0 (permissive)	LGPL 3.0 (copyleft)
IP claiming	Open claims	Open claims	Limiting claims	Open claims	Limiting claims
Availability of results	Open	Open	Share alike	Open	Share alike
Control over progress	Semi-open/Semi-closed	Controlled	Semi-open/Semi-closed	Controlled	Driven
Dominant Business model	Professional services, Hosting	Professional services, Hosting	Selling of a commercial licensed opensource product	Professional services, Hosting	Selling commercial license of a proprietary product.

5. Analysis & Discussion

In this section, the results are discussed as per the common aspects of open innovations observed in the different IoT platforms.

5.1 Open Initiatives in IoT platforms

5.1.1 Open-source

Community-based open-source development

The most prevalent observation in the open IoT platforms was that of using open-source as the main driver of open innovation. Four out of five platforms have a very heavy influence of open source in building or contributing to the main functional components of the platforms. The platforms define themselves as being "100%" open-source IoT platforms. This is because all or most of the technical components catering to the basic functions of the platforms are released under an OSI-approved opensource license. The business of the IoT platform is also organized around the opensource projects that form a part of a limited liability company.

Two open IoT platforms- Thingsboard.io and Thinger.io are similar in most aspects. The IoT platform is a fully open-source IoT platform that has an open-source project for the entire technical stack upon which the platform is built. The platform includes all the basic technical characteristics of IoT and the functional components that make it a universal IoT platform. The platform has built customized solutions based on the existing components from the technical stack to serve as an IoT platform for a particular use case by partnering with customers for whom Thingsboard was providing services. The business is entirely based on its open-source projects and includes the open-source community for building and testing technical solutions. The community is allowed to participate either formally or informally in the open-source project depending on the type on the type and extent of the contributions made. The open-source projects are governed using a fixed product-focused approach that is communicated to the open-source community via a project roadmap. The governance also includes quality checks by the project maintainers who are employees of the platform that do a certain quality check of the contributions provided by the community before including the additions or modifications in the main software file of the project. The license governing the open-source projects being permissive gives permissions for free use, commercial use, and forking if the modification made to the project are stated even when the license does not lay an obligation on sharing the source code. The incentive for contribution lies in the guidance and education provided by the project owners for developing IoT solutions. In essence, it can be said that Thingsboard.io and Thinger.io use opensource as the openness type for development that is led by the product roadmap developed by the platform owners.

Open remote IoT platform also claims to be "100%" open-source and as seen on GitHub the company promotes openness in product development. The product is placed as an open-source project that non-discriminately allows the community to participate in the development of the platform. There is a possibility of formal participation via a community forum hosted by the project owner where guidelines and a project roadmap are issued for the future goals of the project and work towards the same is divided into the form of modules where the community can voluntarily choose the module which the want to develop. The license used for the opensource project is a strong copyleft opensource license (OSI approved) AGPL 3.0 which makes it mandatory for anyone who modifies or builds upon the project to share the source code back to the project. However, paying for a dual licensing model makes it possible for a user to avoid the obligation of sharing the source code of the modifications or the follow-on innovations made over the open-source project. Even when a user opts for a commercial license the platform has the possibility to understand the know-how of the project being used for a certain use case which gives the project owner an idea of the possible further developments that can be made to the opensource project.

Kaa IoT differs from the other platforms by using open-source in their open innovation approach. Kaa IoT platform is currently defined and promoted as an enterprise platform. However, the current enterprise IoT platforms were built using the open-source community for developing the technical stack. The IoT platform started as a "Kaa project" for a few years where the community was heavily involved in building the platform and then shifted the focus on offering an enterprise IoT platform based on the open-source project once solutions were created for multiple use cases and application areas. The open-source project was started by having a product vision where the community was guided using roadmaps and contribution guides for developing the technical stack. The expectation from the community seems to have changed from contributions for building the solutions to contributions in the form of bug fixing and improving the quality of the product. The permissive opensource license Apache 2.0 is used for the opensource project which allows the project owner or anyone to sell the opensource project commercially where the changes and adaptations made to the platform are stated.

Particle uses community based open-source mainly for developing software for open source hardware devices that Particle sells alongside it's IoT platform. However, small open-source projects do exist but they are IoT development tools that help customize the IoT platform and mainly used by the developers on the customer side.

Stakeholder based open-source development

Although all the platforms in the case study use a stakeholder open-source approach for open innovation, only four platforms- Thingsboard, Thinger, Openremote, and KaaIoT include the open-source community indiscriminately in the development of the core product. Particle IoT platform on the contrary involves a community opensource approach only for the components that aid deployment rather than the development of the core product. For example, opensource tools for developing mobile or web applications, or software development kits that could be used for deploying hardware of any brand that customers wish to use in their IoT ecosystem. Another observation is that the Particle IoT platform offers hardware with opensource properties along with opensource firmware to their customers. Customers can choose to customize the operating system and perform upgrades over the firmware of Particle's open-source hardware to adapt it to their own deployment needs. While doing this the customers share the innovations and modifications with Particle which adds to the knowledge of the platform owners about hardware deployments.

The open-source components of Particle IoT platforms on GitHub suggest that there may be contractual agreements between the platform provider and the customer for sharing the source code when Particle's services are availed for developing IoT solutions for a customer. However, this can only be speculated since an interview with key people would be necessary for validating these speculations. Another aspect observed was that Particle promotes innovation projects to developers by allowing free use of the platform for prototyping. Interested developers can join the community for learning about IoT application development while working on mini projects that the platform promotes. In return, Particle gets access to the source code of the developed prototype solution.

5.1.2 Open standards-based communication gateways

IoT platforms offer communication gateways that are built using open standards-based communication protocols such as TCP/IP, HTTP, MQTT, CoAP, Bluetooth, etc. It is observed that all IoT platforms in the case study provide open standards-based communication protocols typically used in different types of IoT applications. This provision of the solution is targeted toward promoting interoperability between devices, other platforms, and tools that are used in the IoT ecosystem. However, the method of delivery of the solutions varies. Thingsboard separates the communication gateway component from the core product and is placed as a separate project on GitHub for encouraging a focused development. But other platforms in the case study include the communication gateway components in the core project. As observed on the Thingsboard repository on GitHub the communication gateway project has a very active community involvement in the development and has a significant number of forks and is speculated as an indication of being useful for developing a collection of solutions that can serve in many use cases.

Typically, open standard communication protocols do not have an open-source code. However, the open standard communication protocols can be used by the IoT platform providers to deploy them in the IoT ecosystem in their own way, like a reference implementation. This deployment is possible by forming open-standard-based communication gateways where the source code for methods of deployment can be developed via the open-source path by involving the community or other stakeholders involved in the IoT ecosystem- device providers, customers, and third-party applications providers.

5.1.3 Open API-based open innovation

IoT solutions are typically used to collect and analyze data to make accurate decisions. This means that there is a need for tools for data analysis and applications that can visualize the data in forms that are easy to interpret. The open-source IoT platforms under study provide open APIs for the tools and applications that can be used by third-party developers to customize their solutions according to a particular use case (often referred to as the open layer). Third-party developers in open-source IoT platforms refer to both the open-source community as well as the paying customers of the IoT platforms. However, in the Particle IoT platform, the third-party developers refer only to the developers on the customer side.

Thingsboard offers some tools and applications as microservices with Open APIs. Communication gateway, edge computing, data analysis, and prediction tools, and mobile application development tools are placed as separated from the core product on GitHub. It serves a dual purpose where in one case the microservices can be used with the Thingsboard IoT platform and in a second case where the microservices can be used with any other IoT platform because of the open APIs. This enables the open-source projects of these microservices to get contributions not only from the Thingsboard IoT platform users but also from other IoT platform users where these microservices are connected using open APIs.

Additionally, each platform has created an open API for third-party opensource tools (for example node.js, Node-RED, etc.) that can be used to create commands and working logic of the IoT platform without much coding effort for any use case. The common terminology for these third-party opensource tools is "rules engine".

Open initiatives in IoT platforms are created primarily for achieving (i) interoperability in the use of the technical stack for different use cases (ii) compatibility with heterogeneous hardware devices (iii) standardization and uniformity.

Opensource addresses all the above issues because of the involvement of the developer community in the building of the technical stack. The diversity brought in by the community enables the construction of a technical solution/product with multiple perspectives yet with a certain consensus. However, the consensus is best achieved by the active involvement of the open-source project owner in managing the open-source development. Opensource project owners act like benevolent dictators by forcing the developer community to follow certain rules such as compliance with focused roadmaps, contribution guides, quality checks for inclusion code, and formal participation in the community forum. A stakeholder-based open-source approach may also solve the issues if the developers on the customer side are treated as an open-source community where the best practices for development are dictated by the platform owner to achieve a focused product development.

Open APIs are provided by IoT platforms primarily to promote interoperability for the platform tools and applications to work in sync with other third-party tools customers or users may include in their IoT ecosystem. The Open API openness approach may not reap the same benefits as open-source because the tools and applications are just a bonus utility offered by the platform. Therefore, the open API approach is mostly used to attract users to the open-source platform.

5.2 Learnings from the open platform design theory

The open platform design theory analyses different types of open-source and open standards initiatives to create an open-source and open standards toolbox. The most relevant analysis of the open platform design theory for this study is the open-source toolbox. The open-source toolbox is created by including learnings about platform construction of different opensource initiatives such as TCP/IP, Apache HTTP server project, Linux, MySQL, and Sun OpenSPARC. While the toolbox specifically does not analyze IoT platforms, the initiatives analyzed can be seen as equivalent to smaller parts of an open IoT platform. Parallels can be drawn from the observations in the case study to each of the initiatives analyzed. For example, open-source implementations of the TCP/IP standard are equivalent to communication gateways in IoT platforms which are, in essence, open-source implementations of multiple open-standard protocols; opensource software development kits for device operating systems provided by the IoT platforms to achieve hardware agnostic properties in IoT deployments are equivalent to the Linux project; the open-source platform logic of Thingsboard and Kaa IoT project is equivalent to Apache HTTP server project and are governed by the same license; Openremote IoT platform is equivalent to MySQL in terms of licensing and business model; Particle IoT platform's opensource IoT hardware project is equivalent to Sun OpenSPARC opensource initiative.

This equivalence observed has verified the suitability to use the open-source toolbox for designing open-source IoT platforms. However, it raises the concern whether a uniform, single objective strategy is suitable for all the technical components of the platform. The case study has revealed the use of separate open-source strategies for components depending on the role of the component in the business model. For example, Thingsboard and KaaIoT maintain a separate open-source project for the communication, while the other components of the platform form a part of a single big opensource project for the core product. By doing so they offer an open API for the communication gateway to be used with any other IoT platform. This strategy helps the platform get contributions from the community as well as free users, and in-turn allowing them to sell the component as a product and offer services on the product separately.

5.3 Business archetypes and business models enabling open innovation

The business models vary in the IoT platforms included in the study but, a common feature observed is the use of Professional services and Hosting services business models. Thingsboard, Thinger, and Openremote are based on the open-core business model where the other business models are combined. Open-core is in the sense that the core product is developed with the opensource community and placed as an opensource project on GitHub as a public domain repository. KaaIoT was developed with an open-core product for three years then was forked by the company to build an enterprise product over the open-source project. Particle is the only IoT platform in the study that has a proprietary core product with elements of open-source for the software to be deployed on IoT hardware devices and other applications and tools for third-party developers.

The business models are dependent on the business archetype, which in turn are dependent on the resources of the business. For example, Thinger and Openremote have 10-15 employees, Thingsboard, and KaaIoT have more than 50 employees, and Particle has about 160 employees. The business archetypes are speculated to vary between four strategies of delivering the product, depending on the resources in the company and the role the platform plays in developing a customer's IoT solutions - (i) Service system owner, (ii) Complementor, (iii) Module producer (iv) System of systems facilitator.

Thingsboard, Particle, and KaaIoT have a considerably large number of components in the IoT platform that have different versions of the components that are readily deployable for different use cases. Moreover, the components are mature to fit any type of use case and therefore offer open APIs to the customer for all components of the proprietary version where the platform and customer can collaborate towards the intended IoT applications, in this way the IoT platforms also gain knowledge about that particular use case. Therefore, Thingsboard, Particle, and KaaIoT can be seen as assuming the role of a 'module producer' and a 'system of systems facilitator'.

Openremote and Thinger have relatively less mature technical components and are therefore speculated to use a 'service system owner' strategy where the innovation process is semi-open, and semi-closed when a customer chooses to use the commercially licensed version of the open-source project. The platform providers and customers work together on building a use case-specific IoT platform. In this case, the platform providers may only have access to the know-how of building the use case-specific software rather than getting the source code to be included in the open-source product.

The 'complementor' strategy for business delivery may not reap benefits in the open innovation context as the platform provider may have a fixed contract with the customer with less control over the product being used, and the services provided by the platform are usually reactive than proactive. This trait is observed in the Particle and enterprise KaaIoT.

6. Conclusion

This section presents the conclusions about the open innovation strategies that IoT platforms can use to build solutions for multiple use cases. The answers to research questions help in understanding the nuances when designing an open innovation strategy.

6.1 Answers to sub-research questions

What are the different types of openness initiatives observed in IoT platforms?

The openness initiative depends on the goal of the business owners. Community-based opensource is the most prominent openness type observed when the goal of the business owner is to build a universal IoT platform with an inherent trait of interoperability and as a cost-effective way to develop the IoT platform. An open API-based open innovation is the second most prominent where open APIs are used to collaborate on building solutions with the customers of the IoT platform. This enables the platform owners to have a focused approach to building a use case-specific solution. Open APIs enable access to the know-how even if the customer is not willing to share the source code. Lastly, developing open standards-based solutions and further using an open-source approach for a reference implementation of the solution is another open initiative that promotes interoperability of the IoT platform. This type of open initiative seems to increase interoperability in terms of compatibility of the platform with third-party applications and with different types/brands of IoT hardware.

How can open platform design theory be used to design openness in IoT platforms?

The open-source toolbox given by Daniel Särefjord (2006) is adapted to aid the construction of an open initiative for IoT platforms.

Platform Concept	**IoT Platform Construction Tools**
Management of Technical Developments	- For community-based open innovation (Opensource) GitHub repositories, a roadmap for the product, and contribution/coding guidelines. Tutorial videos and education for developers.
	- For stakeholder-based open Innovation (Using Opensource, Open APIs) Collaboration contracts for sharing knowledge, and development guidelines for the style of writing code.
Management of Asset Portfolios	- Public domain asset management using the GitHub platform.
	- Default branding on user applications (UI/UX) that are used with the free and open-source version of the IoT platform.
	- Open-source licensing - Commercial licensing of open-source software by including a contributor agreement to limit claims on assets. - Contracts for development collaborations with stakeholders especially when done via Open APIs for guaranteeing freedom to pursue future developments.

Management of Value Proposition	- Reference implementation by packaging product and allowing free use for prototyping and validation. - Product add-ons for paid users of the open-source-based product. - White labeling for user applications for paid users. - Contracts with customers to guarantee freedom to pursue future developments
Management of Incentive Systems	- Core product as an open-source project attracts community contribution. - Education for IoT development - Rapid response on community forums - rewards for commitment to the open-source project
Management of Ownership Claims	Contracts/agreements with contributors and collaborators for claims over the exclusive ownership of the results of the open initiatives.
Management of Competencies	A mechanism for distinction without discrimination for managing: - Formal participants of the open-source community. - Stakeholder based opensource with customers.
Management of Physical and Virtual Platforms	Separate platforms for promoting unbiased progress: - For opensource community collaborations: GitHub - For stakeholder collaborations: GitLab-like platforms (or placing projects separately for community and stakeholders on GitHub.)
Verification of Technical Solutions	Providing incentives to the open-source community for acting as a tester. Free prototyping software development kits are provided to interested developers on the condition that feedback is shared with the platform owner.
	Making available free-of-charge prototyping for customers. No monetary charge, but compulsory feedback sharing for the product.
Organizational Structures and Functions	Benevolent dictator for the open-source project. Using a “controlled” progress approach in the open-source project.
	Opensource project to be included in the limited liability company.

By using an open innovation approach what business models can be enabled for IoT platforms?

Offering an open-source product or offering a proprietary product via open APIs can enable a professional services business model where the platform owners collaborate with customers for developing use case-specific IoT solutions. An open-core product can enable dual licensing by using appropriate IP claiming mechanisms for the open-source project such as a copyleft open-source license combined with a contributor licensing agreement. Cloud hosting services for customers of the open-source product are also possible for all types of open innovation approaches. A combination of open-core, professional services, hosting, and a marketplace (for additional tools for IoT deployment) is also possible when using an open innovation approach by implementing a freemium product (or stages of freemium) that gives an option for a customer to pay as per the scale of deployment. The freemium model enables free to use for a customer for trial application and prototyping where the platform owner can limit the extent of free use and promote paid usage.

6.2 General considerations for designing open innovation strategy for IoT platforms

IoT platforms indulge in open innovation to achieve interoperability and low-effort scalability to cater to multiple use cases and applications. For a customer to adopt a strictly proprietary platform, there is an issue of vendor lock-in in terms of hardware compatibility and interoperability with other tools and platforms that they use for their business operations and for their IoT ecosystem. Having an open innovation approach for building the IoT platform's software assets, can help tackle these issues that benefit both the platform provider and the customer.

An IoT platform is a complex software-based product that requires different types of expertise for development. IoT platform providers can therefore extract many benefits by using the open-source approach. A controlled, community-based opensource project suits the objective of open innovation as the unintentional bias created by the in-house developers of the IoT platforms is broken by the opensource community while the IoT platform is still built with the same goals. The bias which is a hindrance to interoperability is tackled by the opensource community and free users of the platform as they work on developing the software to suit their own objectives.

Therefore, open-source, either community-based, or stakeholder-based is recommended as a suitable open innovation approach for building an IoT platform that provides solutions to multiple use cases and in the process also solves the issues of interoperability, scalability, and vendor lock-in.

However, building a product using open-source logic has its share of challenges. The recommendations are stated as follows:

- To achieve a successful open innovation via community-based open-source, the community needs to be incentivized and motivated for contributing to the open-source project,
  - Having the core product as the open-source project means all characteristics of the IoT platform product must be a part of the open-source project.
  - Separated components on the opensource development platform (e.g., GitHub) according to the function in the technical stack like developer tools, developer libraries, application layer software (mobile applications, web applications, etc.), IoT middleware, Communications gateway, device OS software development kits (SDKs), etc. This is believed to reduce the developer specialization required for significant contributions to the open-source project.
  - Providing education and training to develop IoT solutions.
  - Active response on community forums by the project owners.
  - Maintaining product documentation, product development roadmaps, contribution guidelines, etc.
  - Appropriate OSI-approved licensing for the open-source project and avoiding exploitative contributor licensing agreements (CLAs) if used.
  - Formal onboarding, acknowledgments for

- To achieve successful open innovation via stakeholder based opensource (usually, a business-to-business opensource development in the case of IoT),
    - Open APIs should be created for the main components for the technical stack of the IoT platforms, for example, separate APIs for tools and applications layer software, IoT middleware, and communications gateways.
    - If an open-source license is not used, there should be contracts that state the software assets are shared between the stakeholders and guarantees freedom for future development over the shared assets.
    - A standard set of developer tools and software development kits (SDKs) can enable maintain a uniformity even when developers with different skill sets are involved.

7. Limitations, Strength, and Future Scope

This study was conducted to present an overview of the possibilities of open innovation strategies that can be used for building and scaling an IoT platform for multiple use cases. Open innovations can solve the issues of interoperability, scalability, and vendor lock-ins that are a signature trait of strictly proprietary IoT platforms. The study provides a toolbox for taking an open-source approach for building solutions specifically for IoT platforms, that also serves as a checklist to refer to when devising an open innovation strategy. However, the limitation lies in the evaluation of an appropriate strategy for each technical component of the IoT platform that performs a specific function in the IoT ecosystem. The characteristics of an IoT platform are delivered via different technical components that work in syncs such as device management, communication components, data storage, analytics and management, security developer tools, and applications. It is observed that every technical component can be evaluated as a separate subject for open innovation due to the diversity of engineering aspects underpinning each technical component. Another limitation is that this study has analyzed the qualitative aspects of open IoT platforms, a detailed objective and quantitative study of each platform’s performance is beyond the scope of this study but may affect the results of the analysis.

7.1 Recommendations for further study

The results obtained from this study are reliable to an extent that further studies can take forward the learnings to solidify the results by conducting a wider case study on more platforms. The parameters concerning IoT platform phenomena such as basic capabilities, functional characteristics, and functional elements have the scope to be studied objectively by doing an independent analysis with respect to the open-source context as some unobvious interactions were observed during the study. For example, the open innovation strategy for each technical component can vary according to the complexity of the engineering and the criticality of maintaining a competitive advantage of the IoT platform. This objective study of individual components of IoT platforms will enable gathering enough evidence to support the reasoning behind a potential framework that could be created to devise an open innovation strategy for IoT Platforms.

References

Al-Faqaha, A. et al., 2015. Internet of Things: A Survey on Enabling Technologies, Protocols and Applications. *IEEE Communications Surveys and Tutorials,* 17(4), pp. 2347-2376.

Asemani, M., Abdollahei, F. & Jabbari, F., 2019. *Understanding IoT Platforms: Towards a comprehensive definition and main characteristics description.* Tehran, IEEE.

Balali, S., Steinmacher, I., Annamalai, U. & Sarma, A., 2018. Newcomers' Barriers. . . Is That All? An Analysis of Mentors' and Newcomers' Barriers in OSS Projects. *Computer Supported Cooperative Work (CSCW),* Volume 27, pp. 679-714.

Beecham Research, 2021. *The IoT Platform finder.* [Online]
Available at: https://www.iotglobalnetwork.com/iotpilot
[Accessed 15 May 2022].

Bell, E., Bryman, A. & Harley, B., 2019. *Business Research Methods.* 5 ed. Great Clarendon Street, Oxford, UK: Oxford University Press.

Corno, F., Russis, L. D. & Saenz, J. P., 2020. How Is Open Source Software Development different in popular IoT projects. *IEEE Access.*

Gupta , A., Christie, R. & Manjula, R., 2017. Scalability in Internet of Things: Features, Techniques, and Research Challenges. *International Journal of Computational Intelligence Research,* 13(7), pp. 1617-1627.

Guth, J. et al., 2016. *Comparison of IoT Platform Architectures: A Field Study based on Reference Architecture.* Paris, IEEE Xplore.

Hejazi, H., Rajab, H., Clinker, T. & Lengyel, L., 2018. *Survey of Platforms for Massive IoT.* Eger, s.n.

Imran Ghory, 2020. *The Secrets of Successful Open Source Business Models.* [Online]
Available at: https://medium.com/blossom-capital/successful-open-source-business-models-2709e831e38a
[Accessed 11 May 2021].

Index Ventures, 2021. *Our Journey so Far.* s.l.:Index ventures.

Ismail, A., Hamza, H. & Kotb, A., 2018. *Performance Evaluation of Open Source IoT.* Alexandria, Egypt, IEEE.

Karen, R., Eldridge, S. & Chapin, L., 2015. *The Internet of Things: An Overview,* s.l.: The Internet Society (ISOC).

Kondratenko, Y., Kondratenko, G. & Sidenko, L., 2018. *Multi-Criteria Desicion Making for selecting a Rational IoT Platform.* Kyiv, Ukraine, IEEE.

Korte, A., Tiberius, V. & Alexander, B., 2021. Internet of Things (IoT) Technology Research in Business and Management Literature: Results from a Co-Citation Analysis. *Journal of Theoretical and Applied Electronic Commerce Research,* p. 2073–2090.

Matthew Wopata, 2021. *5 Things to Know About the IoT Platforms Market.* [Online]
Available at: https://iot-analytics.com/5-things-to-know-about-iot-platforms-market/

Minerva, R., Biru, A. & Rotondi, D., 2015. *Towards a Definition of the Internet of Things (IoT).* Turin: IEEE.

Mozilla Foundation, 2018. *Open Source Archetypes: A framework for Purposeful Open Source,* Mountain View, California: Mozilla Foundation, Open Tech Strategies.

Nakhuva, B. & Champaneria, T., 2015. Study of Various Internet of Things Platforms. *International Journal of Computer Science & Engineering Survey (IJCSES),* 6(6), pp. 61-74.

openremote.io, 2022. *Homepage- openremote.io.* [Online]
Available at:
https://openremote.io/?gclid=CjwKCAjwj42UBhAAEiwACIhADvCeg8EXIGP4sRAxVR_2C5QRKKWcv2LCd864HSttebnmwshpHYanURoCTcgQAvD_BwE
[Accessed 17 May 2022].

Ottolini, D., Zyrianoff, I. & Kameinski, C., 2022. *Interoperability and Scalability Trade-offs in Open IoT Platforms.* Las Vegas, IEEE.

Pandora FMS, 2021. *IoT open source – what are its advantages and disadvantages?.* [Online]
Available at: https://pandorafms.com/blog/iot-open-source/
[Accessed 15 May 2022].

Panduman, Y., Sukaridhoto, S. & Tjahonoo, A., 2019. *A survey of IoT Platform Comparison for Building Cyber-Physical System ArchitectureMechkeed.* Yogyakarta, Indonesia, IEEE.

Phillipe Wegner, 2021. *IoT Platform Companies Landscape 2021/2022: Market consolidation has started.* [Online]
Available at: https://iot-analytics.com/iot-platform-companies-landscape/?utm_source=IoT+Analytics+Master+People+List&utm_campaign=23ee816740-Most-read-articles-2021&utm_medium=email&utm_term=0_3069fbcae4-23ee816740-346051628

Salazar Ch, G. D. et al., 2018. *Open Middleware proposal for IoT focused on Industry 4.0.* Barranquilla, Colombia, IEEE.

Särefjord, D., 2006. *Open Platform Design- Towards a Theoretical Framework and a Practical Toolbox,* Gothenburg: Center for Intellectual Property Studies (CIP).

Shi, W. & Dustar, S., 2016. The Promise of Edge Computing. *IEEE Cyber Security*, 5 May, pp. 78-81.

Smedlund, A., Ikävalko, H. & Turkama, P., 2018. *Firm Strategies in Open Internet of Things Business Ecosystems: Framework and Case Study.* Hawaii, HICSS.

Tamboli, A., 2019. *Building Your Own IoT Platform.* 2019 ed. Sydney, Australia: Apress.

Thinger.io, 2020. *Homepage- Thinger.io.* [Online]
Available at: https://thinger.io/
[Accessed 17 May 2022].

Thingsboard.io, 2022. *Homepage-Thingsboard.io.* [Online]
Available at: https://thingsboard.io/
[Accessed 17 May 2022].

Vogel, B., Dong, Y., Emruli, B. & Davidsson, P., 2020. What Is an Open IoT Platform?. *Future Internet 2020, MDPI.*

Zeenat Rehena, 2021. Internet of Things: Challenges and Its Applications. In: P. K. S. D. Monideepa Roy, ed. *Interoperability in IoT for Smart Systems.* s.l.: CRC Press, pp. 7-8.

DEPARTMENT OF TECHNOLOGY MANAGEMENT AND ECONOMICS
DIVISION OF ENTREPRENEURSHIP AND STRATEGY
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY