

Review

Architecting Software for the Internet of Thing Based Systems

Abdulrahman Alreshidi  and Aakash Ahmad *

College of Computer Science and Engineering, University of Ha'il, Ha'il 21955, Saudi Arabia

* Correspondence: a.abbasi@uoh.edu.sa

Received: 23 April 2019; Accepted: 17 June 2019; Published: 10 July 2019



Abstract: *Context*—Internet of Thing (IoT) based systems support any-time/place computations by interconnecting humans, systems, services, devices, and things that enabling autonomous systems to shape-up digitized societies. Software architecture, as the blue-print of software-intensive systems, abstracts the complexities of modeling, design, development, and evolution phases of a software to engineer complex IoT driven systems effectively and efficiently. *Objectives and Method*—Research and development efforts are required to exploit architectural principle and practices to design and develop IoT systems to go beyond the state-of-the-art for IoTs. The objectives of this research are to empirically investigate and systematically classify the state-of-the-art on architecting IoT based software. We have used the Evidence Based Software Engineering (EBSE) method to conduct a mapping study of the existing IoT solutions by investigating 88 qualitatively selected studies. *Results and Implications*—The results of the mapping study highlight various research themes that exploit software architecture models to develop IoT systems. The identified research themes include, but are not limited to, cloud-based software ecosystems, software defined networking, autonomous, and adaptive software and agent-based systems that IoTs drive. The mapping study suggests that futuristic research on architecting IoT software is focused on architectural languages and patterns that support reusability, automation, and human decision support to develop and dynamically adapt IoT software. The mapping study represents a concentrated knowledge regarding architectural principle and practices to facilitate knowledge transfer—benefiting researchers and practitioners—on the role of software architecture for IoT systems.

Keywords: Internet of Things; software architecture; mobile cloud computing; evidence based software engineering; systematic mapping study

1. Introduction

Internet of Things (IoT) are being increasingly adopted as a technology and an enabling platform that interconnects humans, systems, devices, and things to ensure a connected world that is based on autonomous systems that are fundamental to digitized societies [1,2]. IoT based systems represent a combination of (i) hardware (component and sensor) that is controlled and manipulated by (ii) software (source code and programs) that are interconnected through the use of (iii) network (protocols and connectivity) to enable things that collect, process, and exchange useful data [2]. For example, a connected home service robot (i.e., hardware) that is available for control and manipulation through a mobile app (i.e., software) empowers a user to exploit the 'robotic thing' for cleaning, surveillance and other housekeeping activities [3]. A recent study on the current state of the IoT has highlighted that approximately 25 billion devices will be connected to the IoTs by 2020 [4]. In principle, IoT systems envision the 'anytime, anyplace connectivity' of all the things to create technical and socio-economic opportunities with direct integration between the physical world and computer-based software systems. However, in the context of software-driven IoTs, a number of challenges exist, such as engineering

efforts, safety, security, and privacy issues, along with regulatory and governance aspects that needs to be addressed while modeling, developing, evolving, and operating IoT based systems effectively and efficiently [5].

Software architecture as per IEEE standard 1471-2000, represents a system's blueprint to model, develop, and evolve software-intensive systems by specifying computational entities or data-stores of software as architectural components and connectors [6]. Software architecture models have been successfully utilized to effectively engineer medium as well as complex industrial scale systems in the past [7]. Software architecture for the IoT aims to abstract the complexities of heterogeneous hardware components and network protocols to ensure smooth operations as well as the quality of computational things in IoT systems [8,9]. For example, architectural components (as modules of executable code) can provide programmable and standardized interfaces for home service robots and home appliances that can seamlessly coordinate with each other by means of architectural connectors (as message passing between code modules). This means that the existing research, best practices, and principle of software architecture can be exploited to model, develop, execute, and evolve complex IoT systems that satisfy the desired functionality as well as the required quality [7,8]. However, there is a need to go beyond existing research to develop architectural solutions for the next generation of IoT based systems to support the emerging and futuristic challenges for software-driven IoTs [10–12].

Scope and Contributions: In the current decade, there is a steady growth of research and development regarding architectural solutions to address various challenges that range from engineering, operations, governance, and security specific aspects for IoT software [10]. Additionally, a number of reference architectures have been proposed to unify the standards for developing IoT based software [8,9]. However, no research currently exists empirically investigating the existing solutions and the impacts of research on architecting IoT software. We propose using the Evidence Based Software Engineering (EBSE) method to conduct a systematic mapping study [5]. The proposed mapping study qualitatively investigates 88 studies (published from 2011 to 2018) on architecture-centric solutions to develop and manage IoT software. The objectives of the proposed mapping study are to (i) systematically investigate the architecture specific principle, existing solutions, and best practices for IoT software and (ii) analyze the strengths, limitations, and emerging trends to pinpoint futuristic solutions for architecting IoT software. The mapping study represents concentrated knowledge on the role of software architecture for IoTs with following contributions, as illustrated in Figure 1. Figure 1 highlights that:

- The mapping study classifies existing research themes to streamline the prominent challenges, understand the architectural solution, and identify the patterns as best practices to architect IoT based software.
- The mapping study analyses the progression and maturation of research to highlight existing, emerging, and futuristic solution for architecting next generation of IoT software.

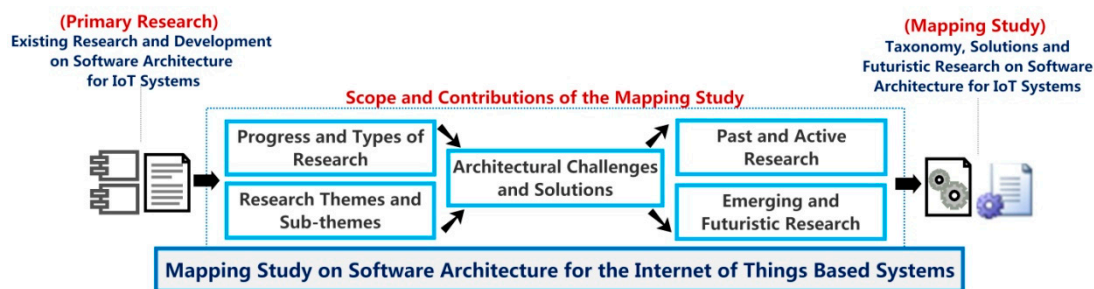


Figure 1. Scope and Contributions of the Mapping Study.

Results and Implications: The results of the mapping study highlight that, in the last five years, there has been significant progress in terms of the published research and development to architect IoT

software. Software architecture-based solution exploit mobile and cloud computing technologies to develop innovative IoT systems that rely on edge computing and software ecosystems to support smart city software. As part of futuristic research, architectural models and languages are required to enable tool support (providing automation), user intervention (supporting human decision) and patterns (exploiting reusable knowledge) that can support industry scale IoTs. The results of the mapping study can benefit:

- Researchers who are interested in understanding a collective impact of the existing research outline a new hypothesis that is based on the state-of-the-art and analyze the role of software architecture in the engineering and development of IoT based systems.
- Practitioners who would like to explore a catalogue of existing solutions understand architectural patterns as the best practices and analyze whether academic research can be leveraged to develop practical and industry scale solutions for IoT software.

Section 2 presents background and related work on software architecture for IoTs. Section 3 presents research methodology to conduct the mapping study. Sections 4–6 present the results of the mapping study as follows. Section 4 highlights the frequency, types, and classification of published research. Section 5 outlines the challenges for IoTs and their architectural solutions. Section 6 presents different phases and trends to highlight the progress of research. Section 7 discusses the implications and conclusions of the mapping study.

2. Background and Related Work

In this section, we first present software architecture and its underlying concepts in the context of IoTs in Section 2.1. Afterwards, we discuss related research on architectures for the IoTs in Section 2.2. The concepts and terminologies that are introduced in this section are used throughout the paper.

2.1. Software Architecture for the Internet of Things

The term ‘Internet of Things’, which is sometimes also referred to as ‘Web of Things’, has diverse applications that range from personalized health monitoring, automated home services, to operationalizing large-scale manufacturing [1,7]. IoT as a concept lacks a unified interpretation and its definition also remains fuzzy due to the diverse applications of the IoT systems. IEEE’s initiative towards a definition of the IoTs aims to consolidate a generic definition, architecture and infrastructure to better conceptualize the term and its applications [7,8] A generic IoT architecture represents a layered structure as per the IEEE standards and available reference architectures in [10]. The layered structure includes at-least a (i) sensing layer, (ii) networking and data communication layer, and (iii) applications layer, as contextualized in Figure 2. Specifically, in Figure 2, we have provided a high-level structural overview of the IoTs (left-hand view), along with its corresponding software architecture (right-hand view). The layered structure for the IoT in Figure 2 highlights that interconnected things, such as a home service robot, a vehicle, and home-based devices are interconnected at the middle layer, which is referred to as interconnection and logic layer. This layer is responsible for managing and operationalizing the interconnection and the desired logic, which helps the things to coordinate and execute over the internet. The layer above the interconnection and logic layer is computation and data storage layer, which manages the computation/analytics and data storage (at a central location) for all of the interconnected things.

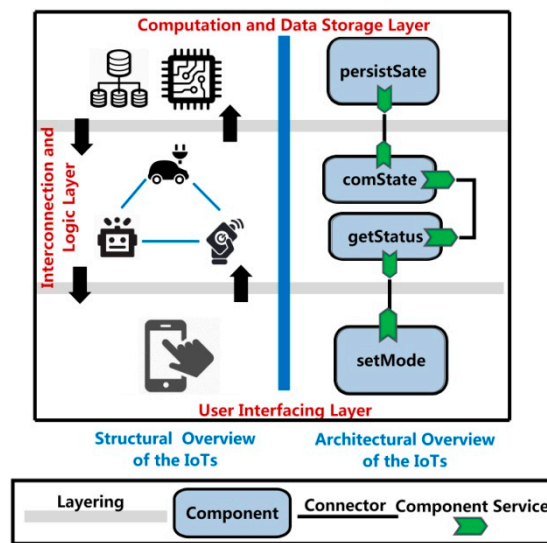


Figure 2. Overview of Software Architecture for Internet of Thing (IoT) based Systems.

The bottom most layer corresponds to the user interfacing layer, which allows the role of user(s) in the system in terms of supervision, and human decision support to monitor or manipulate the interconnected things in the IoTs. In Figure 2, we have presented the software architecture—corresponding to the layered structure for IoTs—that comprises of layers of architectural components and their interconnections [12]. For example, in Figure 2, at the user interfacing layer, the **setMode** architectural component allows for a user to set the mode of a home service robot as idle. The **setMode** component communicates the user’s selected mode to **getStatus** component at the interconnection and logic layer. Finally, the **comState** component communicates the current state of the robot (an individual thing in the IoT) to **persistState** on the computation and logic layer, so that the current state of the home service robot can be persisted and communicated to all other things that are connected to it. The three-layered architecture pattern helps us to layer the system to support separation of concerns and to modularize different architectural components at different layers of system/architectural abstraction [9,12].

In the context of Figure 2, we conclude that, by exploiting the architectural model, complexities of implementation and configuration of the IoT systems can be minimized by means of layering. Moreover, the things and their interconnections in IoTs can be mapped to architectural components and connectors to effectively specify, develop, deploy, and maintain IoT systems [9]. Based on the architectural model, model driven engineering and development can be exploited for the automated generation of the source code (code modules and their interactions) from the corresponding architecture (based on architectural component and their connectors) [10,12]. We aim to investigate the role of software architecture in the engineering, development, operations, and management of the IoT systems in this mapping study.

2.2. Reference Architectures and Survey-Based Studies for IoTs

We now present the most relevant research in terms of the proposed reference architectures and survey-based research (secondary studies) on software architecture for the IoTs. These secondary studies, as summarized in Table 1, help us to understand the gaps in the existing research, which is important in defining the scope and contributions of the proposed mapping study.

Table 1. Summary of existing secondary studies and their research focus.

Study ID	Total Reviewed	Research Focus	Publication Year
[9]	9	Analyze the main characteristics of reference architectures.	2015
[10]	N/A	Impact of Reference Architectures on Industrial IoTs.	2016
[11]	364	Mapping of research on microservices for IoT systems.	2018
[12]	63	Classification of architectural styles for IoT based software.	2018
Proposed Research	88	Mapping of challenges and solution for architecting IoT software.	N/A

2.2.1. Reference Architectures for the IoTs

A number of reference architectures have been put forward in order to conceptualize the building blocks for the IoT systems [8]. Specifically, the reference architectures provide a blue-print of the overall IoT system in terms of the building blocks, hardware and software components, and their interconnections to support the execution and operations of an IoT driven system [9]. We only focus on the most relevant secondary research that explores the available reference architectures from the perspectives of software systems for IoTs.

- Analysis of the Reference Architecture for the IoTs

IoT systems support the utilization of innovative technologies and functionality in an attempt to enhance the productivity and efficiency that can be achieved with a seamless connection among devices. However, the lack of standards for IoT architectures, specifically in the context of industrial IoT systems, is one of the major limitations in the current progress. The research in [10] reviews the recent progress and evolution of the IoT architectures and their impacts on industry scale implementation. The research study in [10] concludes that the time has arrived for software system developers and industry leaders to collaborate regarding engineering, developing, operating, and managing IoT systems that follow standardized approaches for an eco-system of the IoTs. In a similar study presented in [9], the authors analyze a total of nine well known reference architectures (e.g., Reference Architecture Model Industrie 4.0 (RAMI 4.0)) for the IoTs. The results of the research in [9] can help individuals with the selection of an architecture that can be implemented to achieve specific purposes, such as industrial automation, device eco-systems, and other aspects of autonomic systems infrastructures that are based on IoT software.

2.2.2. Styles and Patterns for the IoTs

In the literature, the terms patterns and styles are considered as interchangeable terms and they are often used synonymously, which refer to reusable knowledge and the best practices for architecting software systems [6,12]. Patterns represent recurring solutions from a technical perspective, while architectural styles can be considered as a blue-print or a layout to organize or structure the system as per a given style [13]. We now discuss the existing survey-based studies on software architectural styles for developing the IoT based software systems.

- Microservices for IoTs

Microservices, also known as microservice architecture, is an architectural style that structures a software system in terms of loosely coupled software services for a rapid and continuous delivery and deployment of large and complex systems. In [11], the authors present the results of a mapping study,

which analyzes the state-of-the-art for microservice architectures as the building blocks for IoT systems and cloud computing solutions. The results of the mapping study in [11] highlight the role of reusable frameworks to support microservice architectures for IoT software.

- Architectural Styles for IoTs

In [12], the authors have conducted a mapping study to investigate the role of styles and patterns for the IoT systems. The authors have reviewed a total of 63 research studies to analyze the architectural styles for the IoT based software. The study highlights the research progress and existing solutions, with a discussion regarding future research needs on architectural styles and patterns for the IoT systems. The study in [12] focuses on classifying the existing and highlights the possible futuristic research that can help to develop the patterns and styles for existing and futuristic solutions.

2.3. A Comparative Summary of Existing Research vs. Proposed Mapping Study

Table 1 highlights a summary of the existing secondary studies and their focus in terms of architectural solutions for IoT systems. Based on the review of most related research (i.e., secondary studies on software architecture for the IoTs as in Table 1), we can conclude that in the current decade the research on engineering and architecting the IoTs has significantly grown. A number of secondary studies have been conducted to investigate the state-of-the-art on reference architectures and styles for IoT systems. However, no research exists that investigates the role of software architecture in IoT systems. In comparison to the existing reference architectures and styles, such as [9–12], the proposed mapping study aims to classify the existing research themes to highlight the prominent challenges, understand the architectural solution, and identify the existing patterns as the best practices to architect IoT based software. Moreover, the study focuses on analyzing the progression and maturation of research to highlight the existing, emerging, and futuristic trends of research on architecting IoT software that lacks in existing research.

The details in Table 1 are complementary to the illustration in Figure 3. Specifically, based on the summary of existing secondary studies on architecting IoT software (in Table 1), we position the proposed study and justify its contributions in the context of existing research, as illustrated in Figure 3. Figure 3 illustrates that the proposed mapping study complements existing secondary or survey-based studies that support architectural models and principle to design, develop, deploy and evolve IoT based software systems. From a technical perspective, the proposed mapping study exploits the foundations of IoT architecture in terms of (i) design principle and practices [9,10], along with (ii) design knowledge and reusability from [11,12] to analyze the collective impact and future dimensions of existing research. The existing study regarding the design principle for industrial IoTs [10] also motivates our work to explore IoT architecture from an industrial perspective, while also considering reusable design knowledge [12]. Based on the details in Table 1 and positioning of existing research, as in Figure 3, we conclude that, in contrast to the existing research, the proposed study specifically focuses on software architecture specific challenges, principles, and solutions for developing IoT systems. Further details and technical contributions of the proposed mapping study are presented later in the paper.

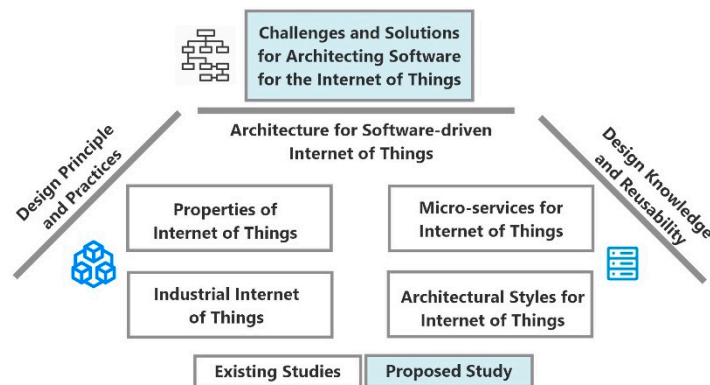


Figure 3. Comparison Overview of the Existing vs. Proposed Study on Architecting IoT Software.

3. Research Method for the Mapping Study

We have used the Evidence-Based Software Engineering (EBSE) method and followed the guidelines to conduct mapping studies and systematic reviews in software engineering to conduct this mapping study [5]. Figure 4 provides an overview of the different steps of the mapping study, namely (i) Specifying the Research Questions, (ii) Collecting Data for the Mapping Analysis, and (iii) Documenting the Mapping Study. In the remainder of this section, we discuss Step I, i.e., Specifying the Research Questions, as illustrated in Figure 4. For space reasons, the details of Step II and Step III of the research method, as in Figure 4, are provided in a dedicated technical report that details the protocol for conducting the mapping study [14].

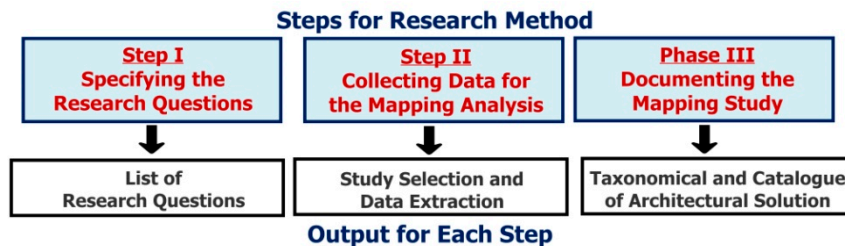


Figure 4. An Overview of the Research Methodology for Conducting the Mapping Study.

Systematic mapping studies focus on a high-level classification and mapping of the existing research to highlight a consolidate impact, research gaps, along with past, present, and futuristic trend of research, in comparison to the systematic literature reviews and literature surveys [5,12]. As per the guidelines for conducting the mapping studies, the role of research questions is fundamental in mapping studies for an objective investigation of the topic under consideration.

3.1. Specify the Research Questions for the Mapping Study

Research questions (RQs) are fundamental to a systematic and objective investigation of the topic under consideration in mapping studies and literature reviews. We have outlined a number of research questions and their objectives, which are classified and presented below.

3.1.1. Type A—Demography and Classification of Research

Research questions (RQs) are fundamental to a systematic and objective investigation of the topic under consideration in mapping studies and literature reviews. We have outlined a number of research questions and their objectives that are classified and presented below.

RQ-1. *What is the frequency and types of publications in the area of software architecture solutions for Internet of Things (IoTs) based Systems?*

Objective(s): To highlight the progress of research in terms of the number of publications over the years. Moreover, the types of publications (e.g., conference proceedings, journal article) reflect the types of research contributions on software architecture for IoT systems.

RQ-2. *What are the existing research themes to support software architecture solutions for IoT systems?*

Objective(s): To identify and classify the recurring solutions as research themes that support architecting of IoT based software. Themes also help us to pinpoint the focus of existing and potential futuristic research.

3.1.2. Type B—Architectural Challenges for Existing and Futuristic Research

RQ-3. *What are the primary challenges for architecting IoT software, and what solutions are provided to address those challenges?*

Objective(s): To identify the primary challenges for engineering of IoT systems and understanding the role of software architecture to address those challenges. Recurring solutions can be systematically identified and documented for their potential reuse as the best practices for architecting IoT software.

RQ-4 *What are existing trends of research for software architecture of IoT systems?*

Objective(s): To highlight the active trends of research that represents the state-of-the-art for architectural solutions pertaining to IoT systems.

RQ-5. *What are the emerging and futuristic trends of research for software architecture of IoT systems?*

Objective(s): To identify the emerging research that also represents the dimensions of futuristic research on architecting the next generation of IoT software and systems.

After specifying the RQs, we now present the results of the mapping study that aim to individually answer each of the RQ (Sections 4–6). Extended details of the research methodology are provided in the protocol for mapping study [14]. Appendix A provides the list of selected studies to conduct the mapping study.

3.2. Strategy to Search and Select the Studies for Mapping Study

After specifying the research questions, the next phase relates to searching the relevant studies and selecting them (based on qualitative analysis) to conduct the mapping study and document the results. This phase involves two steps, which include (i) searching of the primary studies and (ii) qualitative assessment of the primary studies to select the list of studies that are included for review in the mapping study. Both the steps are detailed, as below.

3.2.1. Step I: Searching Primary Studies for Mapping Study

We customized and executed the search strings to ensure a rigorous search and selection of the relevant primary studies, as illustrated in Figure 5, to different digital libraries, including IEEE Xplore, ACM Digital Library, Springer, Science Direct, Scopus, and Google Scholar. Further details regarding the composition of search strings, search strategy, and customized search strings are provided in [14] (Table 2). As in Figure 5, based on the composed search strings, the total results returned referred to as Hits were 28,954 studies with total Retrieved studies amounting to 21,345 studies. We screened a total of 3675 studies to finally include 88 studies for review in the mapping study. In the context of Figure 5, the terms Hits, Retrieved, Screened, and Included refer to the identification, analysis, and selection of the most relevant studies. The extended details of the research method including a qualitative assessment of the identified primary studies and threats to the validity of mapping study are presented in our technical report [14]. The results of the mapping study and presented in the following sections (Sections 4–6).

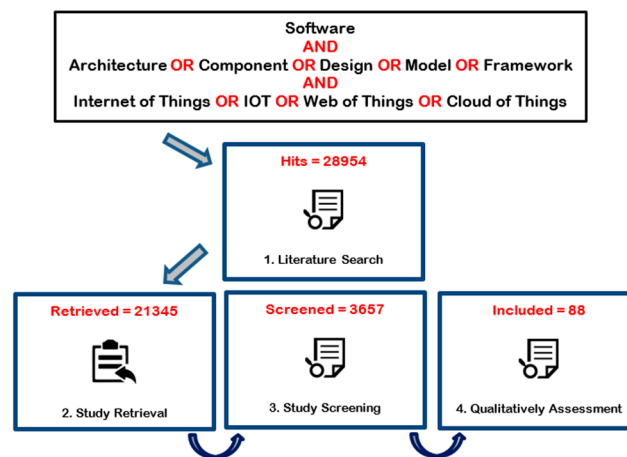


Figure 5. Identification and Selection of Primary Studies—Overview of Literature Search Process.

Table 2. A summary of the strategic steps in the literature search process.

Search Step	Description
Step 1: Deriving Search Terms	We derived search terms from RQs in Section 3.1
Step 2: Considering Synonyms and Alternatives for Search Terms	We considered the alternatives keywords for deriving the literature search strings: - Software Architecture as [Software Design, Software Component, Software Framework] The other relevant terms for architecture like ‘Software Structure’ or ‘Software Styles’ were excluded as they resulted in a large number of irrelevant literature hits. - Internet of Things as [Internet of Things, IoT, Web of Things, Cloud of Things].
Step 3: Combining Search Terms to Compose Search Strings	We combined the search terms to compose the search strings: - Boolean OR operators were used to incorporate alternative spellings and synonyms - Boolean AND operators is used to link the search terms. Number of unique search string depends.
Step 4: Dividing and Customizing Search Strings	We divided and customized the search strings so that they could be applied to different databases. We assigned the unique IDs to every (sub-) search string.
Step 5: Managing Study References	We managed the references to identified studies with Refworks.

3.2.2. Step II: Qualitative Assessment for Inclusion/Exclusion of Studies for Review

Qualitative assessment of the identified studies includes screening and criteria-based assessment, as per the guidelines in [5] and detailed in Table 3. The qualitative assessment in Table 3 helps us to include/exclude studies in an objective way. The equation for quality ranking provides a numerical quantification for individual studies, based on the quality assessment checklist in Table 3, is given below. Quality Assessment Criteria (QAC) is based on five factors that can provide a maximum score of 1.

Table 3. Summary of the study selection process (screening and qualitative assessment).

Step I—Screening				
<i>Is the study in English language?</i>		YES	NO	
<i>Is the study a scientific peer-reviewed published research (no white papers or technical reports)?</i>		YES	NO	
<i>Is the study not a secondary study?</i>		YES	NO	
<i>Is the study not a book or a book chapter?</i>		YES	NO	
<i>Is the study focused on software architecture-based method for robotic systems?</i>		YES	NO	
If [YES] to all four criteria then go to Step II, otherwise exclude study				
Step II—Qualitative Assessment				
Quality Assessment Criteria (QAC)		YES = 1.0	PARTIALLY = 0.5	NO = 0
QAC 1: Contribution	Problem definition and proposed solution(s) clearly presented?			
QAC 2: Context	Research environment in which the study was carried out properly explained?			
QAC 3: Methodology	Research methodology and its organization clearly stated?			
QAC 4: Evaluation	Contributions of the study properly evaluated?			
QAC 5: Synthesis	Lessons learnt, limitations and future research explicitly mentioned?			

3.3. Threats to the Validity of the Mapping Study

We have followed the procedure and guidelines of empirical software engineering and systematic mapping study from [5] to conduct and document the mapping study. However, we had a few deviations from and customizations of the guidelines for conducting the mapping studies, as per the context and the needs of the mapping study. Any customization of the methodological steps for conducting the mapping study may lead to some potential threats to the validity of this mapping study. In the following, we highlight some potential threats and how their impact(s) could be minimized.

3.3.1. Threat I—Identification and Selection of Primary Studies

The first threat relates to a systematic selection of the primary studies that are part of the mapping study. One of the primary challenges prior to the literature search is about formulating the search string(s) with the most appropriate keywords to retrieve all possible relevant literature. Specifically, the key terms ‘Internet of Things’ and ‘Software Architecture’ requires all of their synonyms (e.g., IoT, Web of Things, Framework) to be part of the search strings to ensure that all the relevant data/publications have been retrieved. We have explicitly devised an inclusion and exclusion criteria to qualitatively and objectively identify, analyze, and select each study for the review to support this. Composition of search strings, their execution, and explicit inclusion/exclusion criteria results in increased time and efforts; however, it has ensured that all of the relevant studies have been selected and analyzed for the mapping study.

3.3.2. Threat II—Evaluating Quality of the Selected Studies

After study selection, the next threat related to the qualitative evaluation of the studies to assess the quality of the research that was reported in each individual study. The quality of the mapping study is proportional to the quality of the research reported in individual primary studies. In case the quality

of the primary studies is sub-standard, it impacts the quality of the mapping study and its impact on the findings. To minimize this threat, i.e., to ensure that only quality-oriented primary studies have been selected for the review, we have performed two-steps, which include (i) criteria-based assessment of the quality of the selected primary studies and (ii) structured format to extract and document the required data from the selected studies.

3.3.3. Threat III—Synthesizing Data and Reporting the Results

The final threat is regarding the bias and possible errors in synthesizing the extracted data from primary studies and reporting the results as a mapping study. We have tried to minimize this threat by (i) defining different data points and (ii) capturing data in a structured format that corresponds to each defined data point. For example, the mapping and correlation between the challenges for IoT systems and their architectural solutions (see, RQ-3) only becomes possible once such information has been explicitly extracted from each primary study and documented in the template [14]. Moreover, we have used the ACM classification scheme for taxonomical classification and categorization of the existing solutions to classify and compare the results.

We have tried our best to identify any potential threats to the validity and derive a strategy to minimize the impacts of threat(s) on the mapping study. Further details regarding the research method and threats to the validity of the research method are in [14].

4. Frequency, Types, and Classification of Research on Architecting IoT-based Software Systems

First, we answer **RQ-1** to highlight the frequency and types of published research to indicate the progression and growth of research in Section 4.1. We answer **RQ-2** by deriving a taxonomical classification to investigate the existing research themes in Section 4.2.

4.1. Frequency and Types of Research Publications

To answer RQ-1, we discuss the frequency of research publications—as a reflection of the research progression—over the years (ranging from year 2011 to 2018), as illustrated as a bar graph in Figure 6a. In Figure 6a, the horizontal axis represents the number of years, while the vertical axis represents the total number of publication(s) corresponding to each year. For example, in the year 2017, there were a total of 23 published studies, such as [S6, S78, S79] which are relevant to architecting the IoT based software. Please note that the notation [Sn] (*n* is a number) represents a reference to studies included in the review, as listed in the Appendix A. The notation also maintains a distinction between the ‘Bibliography’ and ‘Selected Studies’ for this mapping study. In general, the graph in Figure 6a reflects a continuous growth of research starting from year 2011 (with 1 relevant study) to year 2017 (having 23 relevant studies). Specifically, the last five years following 2014 can be viewed as more progressive in terms of increased number and diversified types of research publications, such as workshop papers, conference proceedings, and journal articles. The initial published research (from years 2011 to 2013) primarily focused on the unification of cloud computing model to develop the IoT systems [S14, S42, S35]. Later on (during the years 2014 to 2016), the research focused on a diverse set of topics that mainly include, but not limited to, big data systems [S43, S51], security and privacy [S2, S14], web of things [S8, S18], and agent-based systems [S10] that the IoTs drive. Most recently (years 2017 and 2018), the research is focused on fog and edge computing [S64, S40] and self-adaptive software systems [S68, S79] based on the IoTs. Further analysis of the research contributions are discussed later in the subsequent sections of the paper.

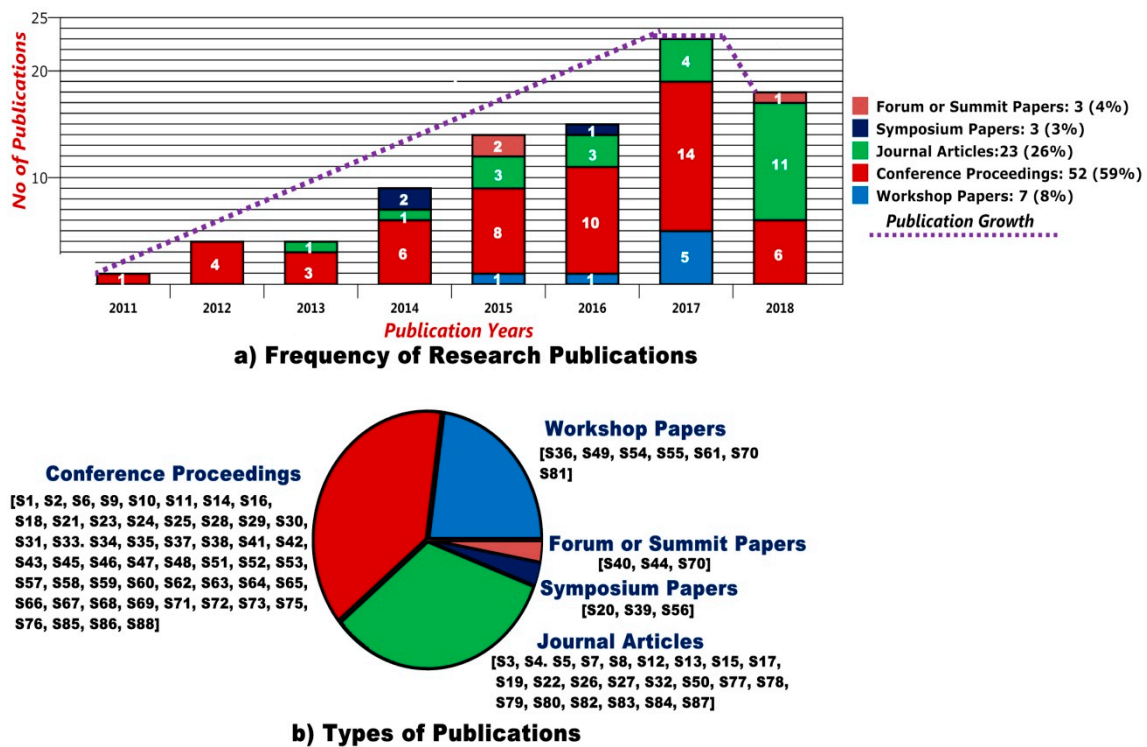


Figure 6. An Overview of the frequency and types of publications.

We conclude that the last four years, collectively, can be seen as most progressive in terms of the total number of publications (70/88, i.e., approximately 80% of the total reviewed studies). These years represent the emerging and futuristic research on the role of software architecture to model, develop, evolve, and operate IoT based software systems.

After discussing the frequency, we must also highlight the types of publications that reflect the diversity of the published research in terms of five distinct categories, namely Workshop Papers, Forum or Summit Papers, Symposium Papers, Conference Proceedings, and Journal Articles. Figure 6b presents the relative distribution of the types of publications. Figure 6b highlights that an overwhelming majority of the research publications (75/88, i.e., approx. 85%) are either conference proceedings or journal articles, which represent not only the proposed solutions, but also their validation to demonstrate the strength and applicability of the research. The research disseminated from conference proceedings and journal articles has mainly focused on the application of software architecture to address the current challenges of big data [S77, S80], system security [S59, S78], and ecosystems [S64, S75] of IoT software.

4.2. A Taxonomical Classification of the Research on Architecting IoT Software

We now answer RQ-2, which aims to investigate the existing research themes and their sub-themes that support architectural solution for the IoT based software systems. The answer to RQ-2 is based on our derived taxonomy of research themes, as presented in Figure 7 and Table 4. The research taxonomy refers to a systematic discovery, naming, organization and presentation of various research themes and their subthemes that provides a blue-print of topic under investigation to highlight the state of research and its contributions. We have derived the taxonomy by reviewing each study and applying (i) the thematic analysis and (ii) the ACM computing classification processes [15]. Figure 7 and Table 4 complement each other while presenting the taxonomy of research themes. Figure 7 mainly focuses on a visual illustration of the hierarchical classification of the identified research themes and their sub-themes, whereas Table 4 provides the corresponding evidence (i.e., reference to the published studies) and total number of studies that support a particular theme or sub-theme. We have

organized the focus and contributions of the research at two different levels for fine-grained analysis of existing research.

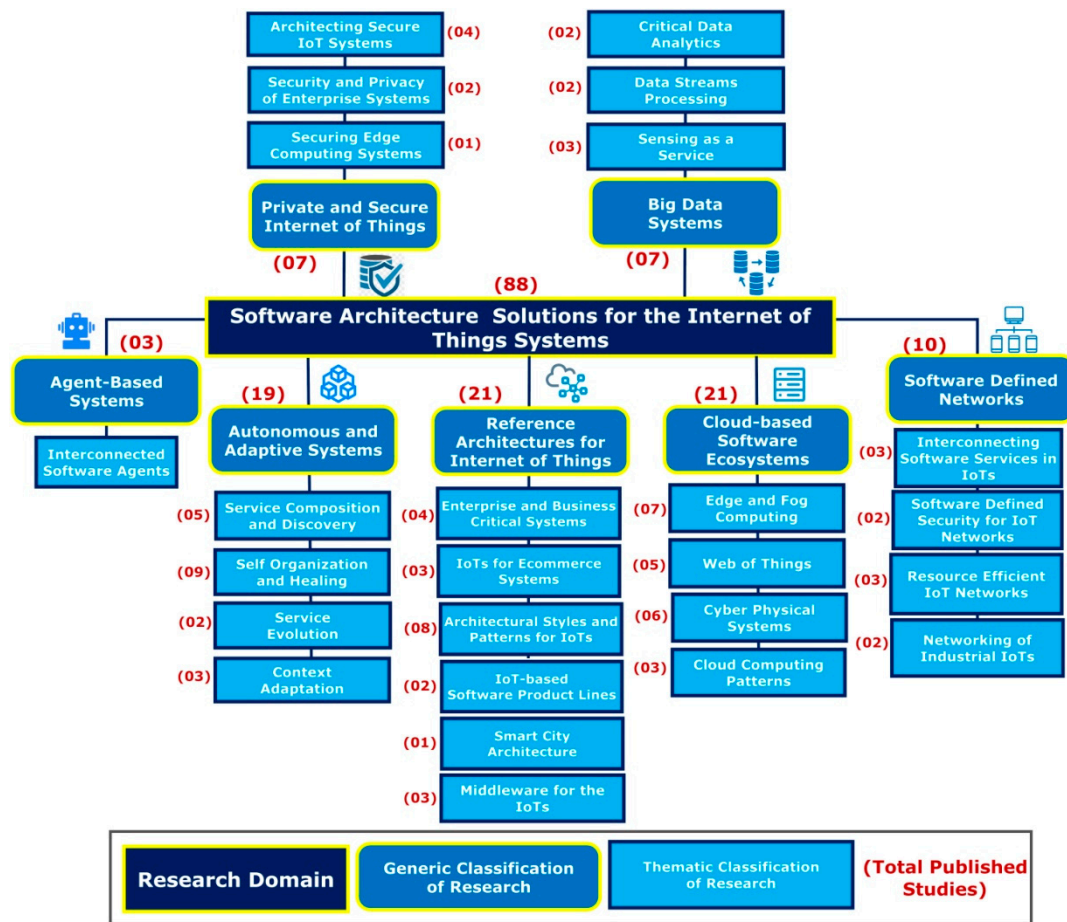


Figure 7. Taxonomical classification of research on architecting software for the internet of things.

Table 4. Research themes, sub-themes, and corresponding evidences (published research studies).

No.	Research Area		Research Themes with Corresponding Evidences	Total Studies
1	Cloud-based Software Ecosystem (21/88 studies, i.e., 24% approx. of total studies)	a	Edge and Fog Computing [S1, S64, S5, S9, S40, S42, S44]	7
		b	Web of Things [S8, S18, S32, S46, S54]	5
		c	Cyber Physical Systems [S15, S17, S19, S34, S70, S75]	6
		d	Cloud Computing Patterns [S29, S47, S53]	3
2	Private and Secure Internet of Things (07/88 studies, i.e., 08% approx.)	a	Securing Edge Computing Systems [S61]	1
		b	Security and Privacy of Enterprise Systems [S2, S78]	2
		c	Architecting Secure IoT Systems [S14, S24, S37, S59]	4
3	Software Defined Networks (10/88 studies, i.e., 11% approx.)	a	Interconnecting Software Services in IoTs [S4, S23, S45]	3
		b	Software Defined Security for IoT Networks [S81, S25]	2
		c	Resource Efficient IoT Networks [S33, S84, S88]	3
		d	Networking of Industrial IoTs [S12, S86]	2

Table 4. Cont.

No.	Research Area		Research Themes with Corresponding Evidences	Total Studies
4	Reference Architectures for the Internet of Things (21/88 studies, i.e., 24% approx.)	a	Enterprise and Business Critical Systems [S31, S60, S63, S72]	4
		b	IoT for Ecommerce Systems [S11, S39, S73]	3
		c	Architectural Styles and Patterns [S7, S26, S27, S28, S30, S41, S76, S87]	8
		d	IoT based Software Product Lines [S16, S74]	2
		e	Smart City Architecture [S82]	1
		f	IoT Middleware [S22, S48, S62]	3
5	Autonomous and Adaptive Systems (19/88 studies, i.e., 22% approx.)	a	Service Composition and Discovery [S3, S13, S49, S50, S52]	5
		b	Self Organization and Healing [S6, S21, S35, S36, S55, S57, S67, S68, S83]	9
		c	Service Evolution [S38, S56]	2
		d	Context Adaptation [S58, S69, S79]	3
6	Big Data Systems (7/88 studies, i.e., 08% approx.)	a	Sensing as a Service [S43, S51, S85]	3
		b	Data Stream Processing [S66, S71]	2
		c	Critical Data Analytics [S77, S80]	2
7	Agent-based Systems (3/88 studies, i.e., 03% approx.)	a	Interconnected Software Agents [S10, S20, S65]	3

These two levels are named as Generic Classification (i.e., Research Themes) and Thematic Classification (i.e., Research Sub-themes), as in Figure 7 and Table 4. For example, the study [S5], in general, supports Cloud-based Software Ecosystems (i.e., generic classification), whereas, [S5] specifically focuses on supporting Edge and Fog Computing (i.e., thematic classification). In the following, we discuss both types of classification, which help us to present the challenges and architectural solutions with the emerging and futuristic trends later.

4.2.1. Generic Classification

It also referred to as **research themes** represent a high-level organization of the existing research into various themes based on the contributions and focus of the research. As in Figure 7, we have identified and presented a total of 07 different research themes, which focus on architecting IoT software, with their evidences in Table 4.

4.2.2. Thematic Classification

It also referred to as research sub-themes provides a specialized and focused extension of the generic classification by highlighting the core contributions of research that fall under the generic classification. For example, studies [S6, S21, S35] generically classified under the Autonomous and Adaptive Systems research support the specific research theme of Self-organizing and Self-healing Systems. The thematic classification represents a fine-grained analysis and presentation of the research themes and their focus. We have identified a total of 25 sub-themes under the 07 research themes that support architectural solutions for IoT systems. For example, in Table 4, the generic classification, named Cloud-based Software Ecosystem, supports four research themes, namely (i) Edge and Fog Computing (7/88 studies, representing 08% approx. of total studies), (ii) Web of Things, (iii) Cyber Physical Systems, and (iv) Cloud Computing Patterns.

It is important to mention here that we have classified each study under one research theme, as presented in Table 4. Some of the studies were possible candidates for their classification under more than one theme. For example, the study [S85] that is classified under the theme Sensing as a Service (Big Data Systems) supports a network of services for crowd-sensed data. [S85] was a potential candidate for its additional possible classification under the theme Secure Software Services (Software Defined Networks), as it relies on service security for sensing data in ad-hoc networked environments. For

each such study, like [S85], we focused on the primary contribution(s), while ignoring any secondary aspects of the research. Therefore, [S85] was only classified under the theme Sensing as a Service, as the primary contribution of this study is service-driven sensing of big data, whereas security of service is a secondary concern. We conclude that each study was classified under one theme that was based on the primary contribution, while discarding the secondary contributions to avoid any overlaps.

5. Architectural Solutions and Patterns for the IoT Software

We now answer RQ-3 that presents challenges and their architectural solutions to engineer and develop IoT software. The challenges and architectural solutions are derived from the taxonomical classification of the existing research (from Figure 7) and are structured as in Tables 5–11, as detailed in Section 5.1. After presenting architectural solutions, we also compare them for an objective evaluation of their strengths and limitations in Section 5.2.

Table 5. Challenge I—Overview of the challenges and architectural solutions for cloud-based software ecosystem.

Challenges for IoT Software	Architectural Solutions
Challenge 1: Architecting Cloud-based Software Ecosystem	
<p><i>Primary Challenge:</i> How to support a (heterogeneous) eco-system of cloud-based software that supports (infrastructure/platform/software) as a service model offered by various cloud providers to third party cloud service subscribers?</p>	<p><i>Proposed Solution:</i> The IoT based solution supports the communication among multi-clouds - the interconnected things that produce and consume data—using a central Cloud Hub. The Cloud Hub facilitates the interaction among the different clouds to collaborate and connect as part of the ecosystem.</p>
Application Areas/Research Focus and Available Evidences	Architectural View
<p>1-a. <i>Edge and Fog Computing</i> exploits architectural configurations to push the computation and storage of devices to the edge of the network to improve performance of the IoTs. Available Evidence(s): [S1, S64, S5, S9, S40, S42, S44]</p>	<p>The architectural view presents a generic view in terms of highlighting the Cloud Hub as a mediator that helps various cloud providers to bind with each other using the mediator as illustrated in Figure 8a.</p>
<p>1-b. <i>Web of Things (WoT)</i> support potentially connected web of devices and things to support universally accessible IoTs Available Evidence(s): [S8, S18, S32, S46, S54]</p>	
<p>1-c. <i>Cyber Physical Systems</i> represent heterogeneous and massively distributed systems to support IoTs for cyber physical systems. Available Evidence(s): [S15, S17, S19, S34, S70, S75]</p>	
<p>1-d. <i>Cloud Computing Patterns</i> represent generic and reusable knowledge and practices that supports reuse-driven development, maintenance and evolution of software architectures for IoT-based cloud-ecosystems. Available Evidence(s): [S29, S47, S53]</p>	

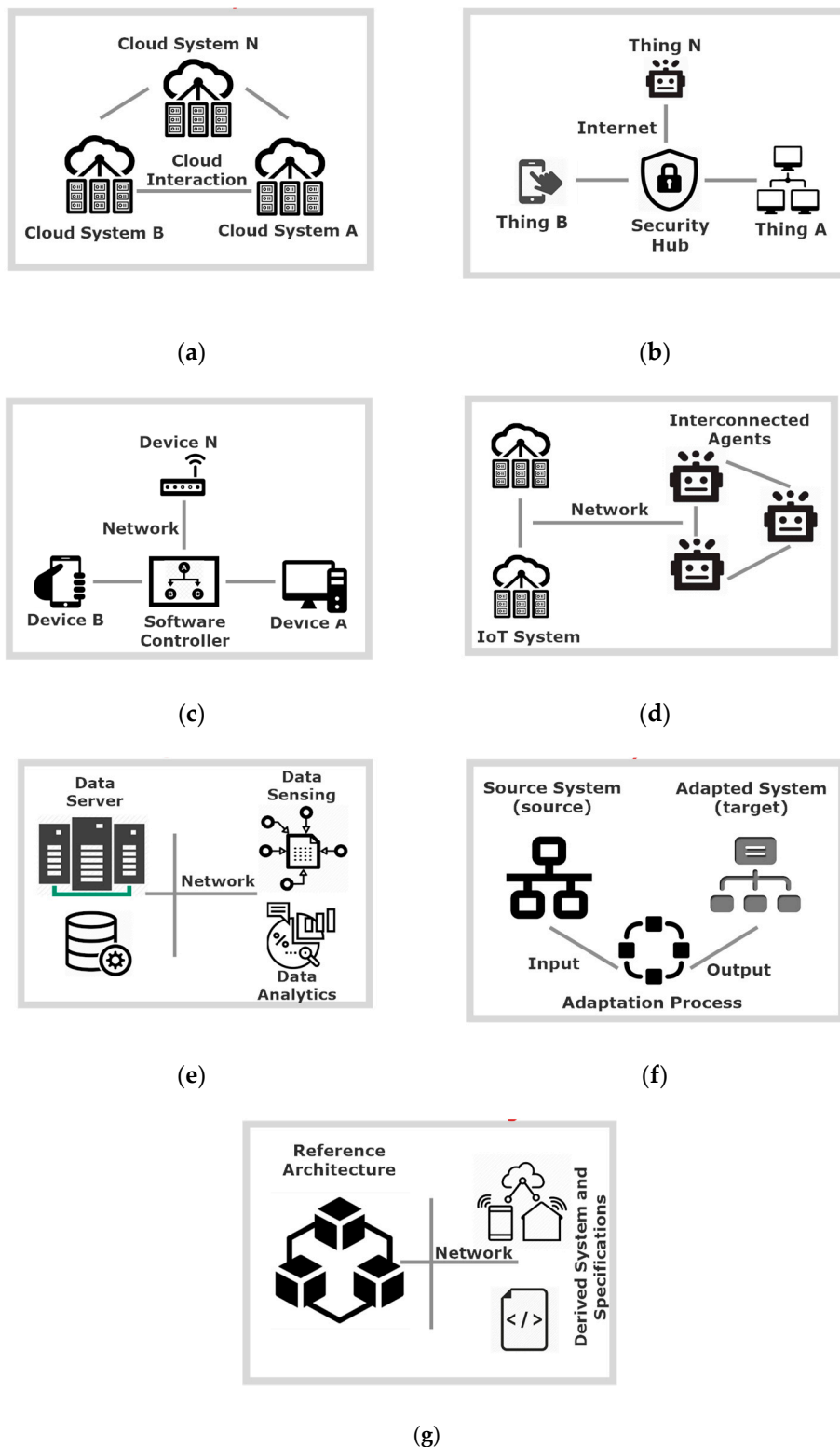


Figure 8. (a) Architectural View of Cloud Software Ecosystems. (b) Architectural View of Private and Secure IoTs. (c) Architectural View of SDN IoTs. (d) Architectural View of Agents Systems for IoTs. (e) Architectural View of IoT based Big Data Systems. (f) Architectural View of Autonomous and Adaptive IoTs. (g) Architectural View of IoT Reference Architectures.

Table 6. Challenge II—Overview of the challenges and architectural solutions for secured and privacy preserving IoTs.

Challenge 2: Secured and Privacy Preserving Internet of Things	
<p><i>Primary Challenge:</i> How to enable the security and privacy of data that is produced consumed or shared by the interconnected things (i.e., heterogeneous devices) in the IoT systems?</p>	<p><i>Proposed Solution:</i> The solution introduces a Security Hub that exploits the mediator architectural pattern to mediate the communication between the connected devices/things and ensure the security of the IoT system.</p>
Application Areas/Research Focus and Available Evidences	Architectural View
<p>2-a. <i>Securing Edge Computing Systems</i> supports secure communication, computation and storage of the nodes deployed as edges in IoT based architectural configurations. Available Evidence(s): [S61]</p>	<p>The architectural view presents the mediator, i.e., Security Hub that monitors and manages the data communication between the devices that may introduce the communication latency as illustrated in Figure 8b.</p>
<p>2-b. <i>Security and Privacy of Enterprise Systems</i> supports the security of computation infrastructures and privacy of data in industry scale enterprise software based on IoTs. Available Evidence(s): [S2, S78]</p>	
<p>2-c. <i>Architecting Secure IoT Systems</i> provides reference architecture and a framework as guidelines and enabling methods to design and develop secured IoTs. Available Evidence(s): [S14, S24, S37, S59]</p>	

Table 7. Challenge III—Overview of the challenges and architectural solutions for software defined networking for IoTs.

Challenge 3: Enabling Software Defined Networking (SDN) for the Internet of Things	
<p><i>Primary Challenge:</i> How to enable software-driven (logic-based) dynamic interconnection and coordination of the things in IoT based systems?</p>	<p><i>Proposed Solution:</i> The proposed solutions integrate a software controller that monitors and manages the necessary logic for runtime manipulation of the interconnections in IoT systems.</p>
Application Areas/Research Focus and Available Evidences	Architectural View
<p>3-a. <i>Interconnecting Software Services in IoTs</i> supports the runtime coordination and security of software as a service in IoT based systems. Available Evidence(s): [S4, S23, S45]</p>	<p>The architectural view presents a Software Controller that acts as an orchestrator for runtime management and configuration of the devices to support software and logic-driven networking of the IoTs as illustrated in Figure 8c.</p>
<p>3-b. <i>Software Defined Security for IoT Networks</i> enables software-driven monitoring and management of security for the IoT systems. Available Evidence(s): [S81, S25]</p>	
<p>3-c. <i>Resource Efficient IoT Networks</i> enables software systems to support the computation, storage, energy, and communication efficiency for IoT networks. Available Evidence(s): [S33, S84, S88]</p>	
<p>3-d. <i>Networking of Industrial IoTs</i> is supported with dynamically configured software services that can manipulate and automate the devices in industrial IoTs. Available Evidence(s): [S12, S86]</p>	

Table 8. Challenge IV—Overview of the challenges and architectural solutions for engineering agent-based systems for IoTs.

Challenge 4: Engineering Agent-based Systems for the IoTs	
<i>Primary Challenge:</i> How to exploit the devices and things in the IoT systems to engineer and develop autonomous and adaptive agents that are interconnected and web accessible?	<i>Proposed Solution:</i> The proposed solutions consist of two parts, i.e., IoT System and the Interconnected Software Agents. The agents (i.e., autonomous, adaptive and distributed components) can be connected to the IoT systems to perform their tasks.
Application Areas/Research Focus and Available Evidences	Architectural View
<p>4-a. <i>Interconnected Software Agents</i> can support a collaborative network of agents that actively coordinate to complete the assigned tasks in an IoT system.</p> <p>Available Evidence(s): [S10, S20, S65]</p>	<p>The architectural view presents network-based coordination between the IoT system and software agents as illustrated in Figure 8d.</p>

Table 9. Challenge V—Overview of the challenges and architectural solutions for supporting IoT based big data systems.

Challenge 5: Supporting IoT based Big Data Systems	
<i>Primary Challenge:</i> How to exploit the interconnected and distributed devices in an IoT system to sense (collect real-world) data and analyze it (process in real-time) for decision support systems?	<i>Proposed Solution:</i> The proposed solution exploits ‘on the edge’ deployed devices that acts as context-aware data sensors and communicate it with the backend server. The backend cloud-based servers can store and process the data with back and forth communication between front-end devices and back-end servers.
Application Areas/Research Focus and Available Evidences	Architectural View
<p>5-a. <i>Sensing as a Service:</i> support loosely coupled and autonomous (web-) services that exploit the devices and sensors to collect the data.</p> <p>Available Evidence(s): [S43, S51, S85]</p>	<p>The architectural view presents the front-end (on the edge) IoT devices that sense data and communicates it with the backend server that processes the data as illustrated in Figure 8e.</p>
<p>5-b. <i>Data Stream Processing:</i> is managed by IoT sensors that can be deployed at different network locations to collect and process live streams of data as part of real-time data analytics.</p> <p>Available Evidence(s): [S66, S71]</p>	
<p>5-c. <i>Critical Data Analytics:</i> is managed by IoT devices that are deployed ‘on the edge’ to sense and process critical data that includes health, context and urban analytics.</p> <p>Available Evidence(s): [S77, S80]</p>	

Table 10. Challenge VI—Overview of the challenges and architectural solutions for autonomous and adaptive IoTs.

Challenge 6: Autonomous and Adaptive IoTs	
<i>Primary Challenges:</i> How to support architectural models that enable autonomous and adaptive IoTs that dynamically adapts the structure and behavior of the IoT systems at runtime?	<i>Proposed Solution:</i> The solution supports an adaptation process based on IBM’S framework for autonomic computing that takes a (source/existing) IoT systems and then adjusts its structure and behavior to enable an adapted (target/new) IoT system.

Table 10. Cont.

Application Areas/Research Focus and Available Evidences	Architectural View
<p>6-a. <i>Service Composition and Discovery</i>: supports dynamic discovery and composition of the IoT software services to enable dynamic systems as per the contextual requirements. Available Evidence(s): [S3, S13, S49, S50, S52]</p>	<p>The architecture presents an adaptation process that contains the adaptation context and logic to dynamically reconfigure/adapts a source system to a target system at runtime in Figure 8f.</p>
<p>6-b. <i>Self-Organization and Healing</i>: enables IoT software that is fault-tolerant and it can self-organize to continue its operations under continuously varying requirements and frequent maintenance and evolution. Available Evidence(s): [S6, S21, S35, S36, S55, S57, S67, S68, S83]</p>	
<p>6-c. <i>Service Evolution</i>: is supported with dynamically reconfigured services in the IoTs systems that evolve over-time as per changes in the business and technical requirements and operational environments. Available Evidence(s): [S38, S56]</p>	
<p>6-d. <i>Context Adaptation</i>: is enabled with context-aware services in the IoTs that adapt their behavior as per the contextual requirements. Available Evidence(s): [S58, S69, S79]</p>	

Table 11. Challenge VII—Overview of the challenges and solutions for reference architecture and models for IoTs.

Challenge 7: Reference Architectures and Models for IoTs	
<p><i>Primary Challenge</i>: How to establish guidelines and frameworks that provides the templates and blue-print to design and develop IoT systems for real world applications?</p>	<p><i>Proposed Solution</i>: The solutions provide reference architectures as generic and high-level (abstract) solutions that can be instantiated with specialized architectures as (concrete) solutions for IoTs.</p>
Application Areas/Research Focus and Available Evidences	Architectural View
<p>7-a. <i>Enterprise and Business Critical Systems</i>: are developed based on reference architectures and process automation that support enterprise systems and production/assembly lines to enable industrial IoTs. Available Evidence(s): [S31, S60, S63, S72]</p>	<p>The architectural view presents reference architectures as a reference framework that serves as a template and a collection of documented guidelines to architect and design IoT systems as in Figure 8g.</p>
<p>7-b. <i>IoT for Ecommerce Solutions</i>: are supported by business process automation and reference architectures that automate those business processes to support the development and operations of IoT based e-commerce systems. Available Evidence(s): [S11, S39, S73]</p>	
<p>7-c. <i>Architectural Styles and Patterns for IoTs</i>: represent generic knowledge and best practices to model, develop, and evolve IoT architectures with enhanced reusability and efficiency. Available Evidence(s): [S7, S26, S27, S28, S30, S41, S76, S87]</p>	
<p>7-d. <i>IoT-based Software Product Lines</i>: are supported by reference architectures that represent a generic solution to derive specialized product lines that enable Industrial IoT systems. Available Evidence(s): [S16, S74]</p>	
<p>7-e. <i>Smart City Architecture</i>: are enabled by interconnected IoT devices that act as backbones for smart and autonomous infrastructures for digitized urban services. Available Evidence(s): [S82]</p>	
<p>7-f. <i>IoT Middleware</i>: provides a technological layer that abstracts the hardware and software complexities to support IoT devices (hardware) and their corresponding (software) applications to support IoT systems. Available Evidence(s): [S22, S48, S62]</p>	

5.1. Architectural Challenges and Solutions for IoT Software

Tables 5–11 presents structured data that acts as a catalogue to highlight the problem solution view based on four types of information, which include (1) Challenges for IoT Software that pinpoints the engineering, development, deployment, and maintenance specific issues, (2) Architectural Solutions that provide an architecture-centric solution to address specific challenges, (3) Application Areas/Research

Focus and Available Evidences in terms of published studies corresponding to the presented challenges and available solutions, and (4) Architectural View that provides a visual representation (generic and abstract illustration) of the proposed solution. During the data collection and synthesis process (see Figure 4 and details in [14]), we documented the information regarding the recurring challenges and their architectural solutions that help to address those challenges, as detailed in Tables 5–11 and exemplified below.

- 1 **Challenge(s) for IoT Software:** How to secure the IoT software and ensure the privacy of critical data contained by the software? See Table 6 Challenge II.
- 2 **Architectural Solutions:** To integrate a Security Hub that exploits the mediator architectural pattern to mediate the communication between the connected devices/things and ensure the security of the IoT system. Specifically, the Security Hub that mediates between the connected things monitors the communication between the things and ensures the detection of any security threats and necessary actions to alleviate the threat to preserve the security and privacy of the IoTs, as in Figure 8b. However, each thing in the IoT can only communicate through the central Security Hub, which may lead to increased data and communication latency issues while the things communicate to each other via the mediator.
- 3 **Application Areas/Research Focus and Available Evidences:** include (a) Securing Edge Computing Systems [S61], (b) Securing Enterprise Systems [S2, S78], and (c) Secure System Architecture [S14, S24, S37, S59], as in Table 6.
- 4 **Architectural View:** is presented as ‘Private and Secure Internet of Things’ in Figure 8b.

5.2. Comparative Analysis of the Architectural Solutions for IoT Software

In Table 12, we also highlight the prominent characteristics of the existing software architectures for IoTs based on following criteria, which have been adopted from [16,17]. Previously, we classified and compared some of the well-known existing software architectures for IoTs [9–12] in terms of architectural principle and usefulness (see Table 1). In this section, we specifically discuss the identified architectures from existing research and development, which are presented in Figure 8a–g and Tables 5–11. In Table 12, a criteria based summarization of the identified software architectures helps us to discuss the following based on guidelines for architectural comparison and evaluation [16,17]. The discussion below and presentation in Table 12 can facilitate the objective/criteria-based selection of a specific architecture design of an IoT based software system.

- Design Principle of the Architecture highlights the underlying principle and practices that enable architectural design for the problem/system under consideration. For example, as in Table 12, the IoT architecture for cloud-based software ecosystem exploits the concept of architectural components (computing servers) along with architectural connectors (server interconnections) to develop an ecosystem of the cloud computing.
- Motivations for Architectural Adoption reflect the rational as perceived benefits for adopting the architecture. For example, IoT architecture for cloud-based ecosystem support the communication and coordination of heterogeneous cloud-based servers that are supported by architectural configurations.
- Limitations of the Architectural Adoption presents some of the shortcomings or limitations that may be inherent in a specific architecture. For example, two of the primary limitations for cloud-based software ecosystem architecture are performance and communication latency, due to heterogeneous cloud and their interconnections.
- Potential Applications of the Architecture represents the areas or domain where architectural solutions can be applied. For example, cloud-based software ecosystem architecture can be used to design and develop cyber physical systems and edge/fog computing systems.
- Attributes for Architectural Evaluation represents the quality attributes (also known as non-functional properties) that must be evaluated for an objective assessment of effectiveness and

the usability of the desired architecture. For example, cloud-based software ecosystem architecture must be evaluated in terms of system scalability and integration to multi-cloud systems.

Table 12. Criteria-based assessment of the existing software architectures for IoT systems.

<i>Design Principle of the Architecture</i>	<i>Motivations for Architectural Adoption</i>	<i>Limitations of the Architectural Adoption</i>	<i>Potential Applications of the Architecture</i>	<i>Attributes for Architectural Evaluation</i>
IoT Architecture for Cloud-based Software Ecosystem				
Interconnected cloud computing servers	Heterogeneous cloud interconnection and ecosystem	- Performance - Communication latency	- Cyber physical systems - Edge/Fog computing	- Scalability - Multi-cloud integration
Architecture for Secure and Private IoT Software				
Security and privacy preserving computation and communication of IoTs	Assurance for data integrity and security at design/architecture level	- Extensive computation and storage resources - Performance issues for resource-constrained mobile devices	Enterprise and business critical systems	- Data integrity - Security vs. Performance tradeoffs
IoT Architecture for Software Defined Networking				
Adaptive software for networking of IoTs	Network and interconnections can be managed and executed dynamically	Extensive data and logic to be programmed	- Web of Things - Vehicular Networks	- Adaptation and Maintainability - Runtime Performance
IoT Architecture Agent-based Software Systems				
Collaborative things and agents	Collaborative and mission critical computations and communications	- Communication latency - Security and trust of collaborative agents	- Crowd sourced computations - IoT based robotics	- Scalability of multi-agent system - Computation and communication performance
IoT Architecture for Big Data Software Systems				
Architectural nodes as computational entities and data stores	- Distributed and heterogeneous data processing - Data replication and backup	- Complex implementation and deployment - Frequent maintenance and evolution	- Data analytics - Data-driven smart systems	- Data integrity - Efficient data processing and storage
IoT Architecture for Autonomous and Adaptive Software Systems				
Autonomic computing for adaptive software system and services	- Dynamic adaptive IoTs - Autonomous computing	Complex implementation and deployment	- Smart city systems - Adaptive Industrial IoTs	- Fault tolerance - Maintenance and evolution
Reference Architectures and Models for IoT based Software Systems				
Model-based software design and development	- Modelling for implementation - Automated engineering	- Levels of Abstraction	Applicable to a wide variety of systems for model-based design and development of IoTs	- Design to implementation traceability - Customization and extensibility

6. Phases and Trends of Research for Architecting IoT Software

We now answer RQ-4 and RQ-5, which aim to investigate the past, present and futuristic research on architecting software systems for the IoTs. First, we answer RQ-4, which analyses the temporal progression and evolution of research, as different phases in Section 6.1. Later, we answer RQ-5 to highlight the active and emerging trends of research that indicate the possible dimensions of futuristic research on architecting IoT software in Section 6.2. We have used the taxonomy of research classification and available evidences (from Figure 7 and Table 4) to categorize the published research into three distinct phases in order to systematically investigate the historical progression of research and to highlight the potential areas for futuristic research, as detailed below and illustrated in Figure 9.

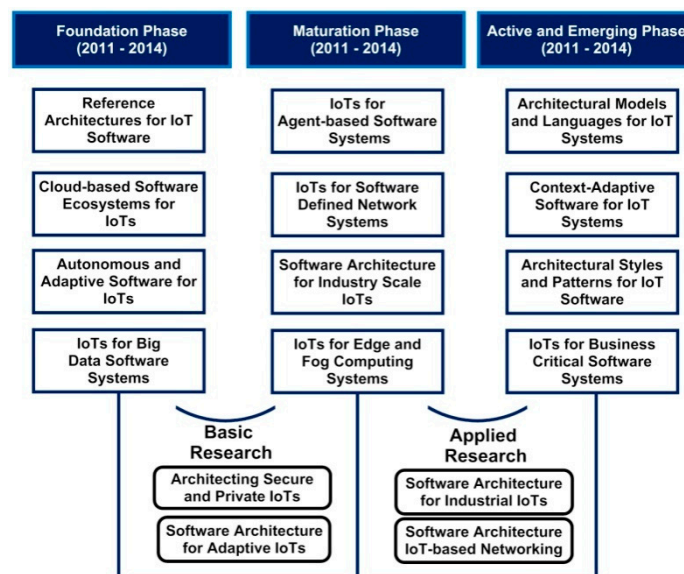


Figure 9. Overview of the Three Phases of Research on Architecting IoT Software.

6.1. Phases and Temporal Progression of Research on Architecting IoTs

The different phase of research represents a historical progression and evolution (detailed in Section 6.1.1) that has laid the foundations for existing and emerging trends of research and the potential dimensions of the futuristic research (Section 6.1.2). The discussion regarding different phases of research is guided by the illustrations in Figure 9.

6.1.1. Historical Progression and Evolution of Research

- **Foundation Phase (2011 to 2013)**

It represents the earliest phase and emergence of research on architecting the IoT software. This phase only consists of 09 studies (i.e., 10% approx. of total reviewed literature). In Figure 9, this phase highlights the inception of research on the role of software architecture on engineering and development of IoT systems with main focus on reference architectures [S35, S87, S88] and cloud-based software eco-systems [S42, S53] to support IoTs. This phase also introduced the concept of Web of Things, which exploits the web-based technologies to connect various devices and things as part of the IoTs [S46].

- **Maturation Phase (2014 to 2016)**

It represents the years during which the foundation level research started to gain momentum and maturity based on the quality as well as the quantity of the published research. This phase consists of 38 studies (i.e., 43% approx.) that extended the earlier research theories and concepts to develop industry scale IoT software [S12, S86]. As illustrated in Figure 9, some innovative concepts

were introduced during this phase, which include software-driven networking for IoTs [S23, S85] and IoT-driven agents [S10, S20]. For example, the studies, like [S9, S44], extended the theory behind cloud eco-systems to introduce edge computing for IoT systems, reference architectures, product lines [S23, S74, S86], for IoTs and autonomous context-adaptive IoTs [S52]. The research published during this phase can be considered as fundamental for the active and futuristic research on architectural solutions for IoT software.

6.1.2. Existing and Emerging Trends of Research

- **Active and Emerging Phase (2017 and 2018)**

It represents the most recent phase with ongoing and potentially futuristic research on architecting IoT software. This phase can be considered as the most progressive in terms of the number of published studies (41/88, i.e., 47% approx.). In the current phase, the research on architecting the IoTs is primarily focused on extending the basic research with new concepts and their applications to industrial IoTs [12] and business critical software systems [S31, S60], as in Figure 9. Specifically, the mapping study suggests that three research themes are the most active with innovative solutions that extend the basic research on autonomous and adaptive systems, cloud ecosystems, and reference architectures. The active and futuristic research on architecting autonomous and adaptive IoT software is focused on establishing theoretical foundations and implementing practical solutions that are related to architectural modeling notations and description languages [S21, S68, S83] and self-organizing and self-healing IoTs [S40, S55] that can adapt themselves as per the evolution of their context [S58, S79]. Architectural patterns and styles as reusable knowledge and best practices are considered as emerging and futuristic trends to architect cloud-based IoT ecosystems that include, but not limited to, patterns for edge and fog computing systems [S7, S19, S26, S27]. The reference architectures that were introduced earlier and had been used as blue-prints and architectural templates to develop IoT software are now being used to develop industrial scale IoTs and business critical software that support e-commerce and enterprise activities [S31, S60, S63, S72, S73], as in Figure 9.

6.2. Active and Futuristic Research on Architectural Solutions for IoT Software

Based on the years of publication as in Figure 9, we have considered the years from 2011 to 2016 as past research, whereas the research being published during the years 2017 and 2018 represents active and emerging trends as an indication of possible futuristic research. The research and development of architecting the IoT software has enabled (a) the establishment of the theoretical foundations and (b) fundamental research on software architecture for the IoTs during the past phase (2011 to 2016). In Figure 9, we highlight some of the prominent trends as the dimensions of possible futuristic research on architecting IoTs, summarized as below.

6.2.1. Models, Languages, and Patterns for Architecting IoT Software

Architectural languages and patterns have proven to be useful for architecture-centric specification and the development of software-intensive systems [6,12,13]. In the context of the IoTs, the research in [12,13] reflects the recent trends that investigate the role of patterns and styles as the best practices and knowledge that can be reused to effectively and efficiently develop IoT systems. However, architectural patterns and styles represent empirically discovered knowledge, which needs systematic identification, documentation, and application that can be integrated with architectural languages to enable pattern and style-based architecting of the IoT software [16]. Specifically, the exiting research and development on software languages and patterns can be fostered to establish the foundations for innovative solutions, such as pattern language, pattern catalogues, and architectural modeling languages to architect the next generation of solutions for software-driven IoT systems.

6.2.2. Context-Sensitive and Dynamic Adaptive IoTs

Autonomous and adaptive systems can dynamically adapt the behavior of the system to cope with the changing requirements and conditions of operational environments that also involve self-organization and self-healing [S6, S21]. Context-aware computing has been exploited to develop systems that can exploit contextual information to adapt systems on the go with the recent emergence and fast adoption of the mobile computing technologies, as per the needs of the context [S58, S59]. The emerging and futuristic trends suggest that possible unification of the IoTs and context-aware computing can lead to context-sensitive IoTs that can rely on contextual (i.e., temporal, environmental, geographical, etc.) information to adapt their behavior and operations. The usage and application of context-sensitive IoTs can range from Big Data to Crowd Sensing and Agent Systems. In the context of big data systems that process and transmit large amount of data, the IoT gateway with limited resources of data (pre-) processing may face performance or bottleneck issues [18]. In such scenarios, IoT gateway consumes more time than expected, in pre-processing the collected data and/or transmitting the data to the connected servers. Therefore, quality attributes or non-functional properties, such as performance, scalability, and availability, must be explicitly addressed while architecting IoT-driven big data systems. IoTs that support crowd sensed data, such as real-time traffic flows and environmental circumstances, can exploit contextual information to support advanced recommender systems.

For context-sensitive and dynamic adaptive IoTs, one of the major challenges is mobility management, which relates to connectivity and performance issues for IoT devices and applications that move from one geographic location/network to another one. The solution in [19] provides a distributed computing architecture (based on Edge/Fog/Cloud monitoring system and a Capillary Container Orchestrator) to handle highly dynamic IoT environments to address the challenges of mobility management. In this architecture, container-based microservices running on the Edge node will be offloaded to the upper layer, called the Fog node, to manage situations where IoT devices are moving from one geographical point to another one.

6.2.3. IoTs for Critical Software Systems

Critical systems refer to software-intensive systems that are central to the operations of a particular domain, such as health, safety, and business critical systems [S11, S39]. Industrial IoTs have been increasingly adopted to support automation and enhance the efficiency of industrial systems, such as production and manufacturing [7,9]. The reference architectures for IoT systems establish foundations and standards, and provide frameworks to develop business and enterprise critical systems [S31, S60]. The futuristic research and development on IoT also demand systems that can be applied to other domains, such as health care, process automation, and self-operating systems (e.g., robotics, automobiles, agents) to realize the potential of the IoTs in routine activities and increased adoption in real life. This means that research on reference architectures must go beyond the fundamental theories to focus on concrete solutions, scenarios, and case studies to engineer, develop, and operate critical systems.

6.2.4. Reference Architectures and Models for Industry Scale IoTs

Reference architecture and models is one of the most researched areas in IoT software. Reference architectures provide a blue-print and a set of guidelines for the architecture-centric development of the IoT systems [9,10]. One specific example of industrial IoTs is RAMI 4.0 that can be extended with concrete solutions to support industrial automation, device eco-systems, and other aspects of autonomic systems infrastructures that are based on IoT software. As part of future research, industrial IoTs need to rely on standardized practices and processes to develop solutions and systems that can be used across the industries [20]. In such circumstances, the role of reference architecture is fundamental in providing a blue-print of the overall IoT system in terms of the building blocks, hardware, and

software components and their interconnections to support the execution and operations of an IoT driven system [21].

6.2.5. IoT Architectures for Smart Cities, Systems, and Infrastructures

In recent years, smart city systems and infrastructures are being increasingly adopted by individuals and organizations to automate urban services, improved lifestyles, and accelerated socio-economic development [2]. From a technical perspective, the devices or things that are connected to and managed or operated by IoT Architecture produce and consume a large amount of data among the interconnected devices. In such an architectural context for IoTs, architecturally significant requirements, such as performance, availability, and security of the system, becomes important. Among others, smart homes and smart vehicular networks are two of the widely used examples of IoT in smart city systems [582]. Future research on IoTs for smart systems requires autonomous and adaptive systems that can dynamically configure their execution, security, and fault-tolerance based on contextual information. Software architecture and its underlying principle can be exploited to model, develop, operationalize, and evolve smart city systems, along with their functionality and desired quality.

Fog of Everything (FoE) is one of the recently introduced concepts that unifies IoTs and fog computing [22]. Traditionally, fog computing and Internet of Thing/Everything (IoE) are considered are two distinct computing paradigms with complementary views and contributions. The study in [21] promotes the integration of fog computing and FoEs to foster a number of computing and network-intensive pervasive applications to support the operational aspects of the future internet. In the context of Internet of Everything (IoE) for smart city systems, the research in [23] presents a fog computing based smart city architecture that is named Fog Computing Architecture Network (FOCAN). FACON is a multi-layered architecture that supports the execution of smart city applications to compute, route, and communicate with one another to support a smart computation and communication environment. One of the usage and main benefit of the FOCAN architecture is that IoT devices (as portable and context-aware computers) can provide required services in a cost and energy efficient manner.

Architectural management is an important activity that focuses on managing different phases in the process for architecting IoT software, such as architecture-centric design, development, validation, deployment, and maintenance of smart systems [24]. As part of future research, Web of Things based software demand architectural configurations that support elasticity (dynamic addition or removal of devices and servers) to support industry-driven and scalable IoTs [25]. One of the most critical challenge is to design industrial IoTs that are based on architectures that support the scalability of the systems when the number of connected devices increases.

7. Conclusions of the Mapping Study

Internet of Things based systems have fast emerged as technological solutions and enabling platforms to interconnect the devices, humans, systems, and services as connected things to ensure autonomous systems and enhanced digital services. The research on software architecture solutions for IoT systems have progressed for almost a decade to establish theories, propose frameworks, and develop tools to address various challenges that relate to architecting and implementing IoT systems. We have used the evidence-based software engineering approach to conduct a systematic mapping study as an empirical investigation regarding the role of software architecture and its implications on IoT based software systems. The results of the mapping study highlight that prominent architectural solutions for IoTs include, but not limited to, software ecosystems, autonomous and adaptive software, security and privacy, software defined networking, reference architectures, agent systems, and big data analytics. A taxonomical classification and analysis of the existing research suggest that active and emerging solutions are primarily focused on models, languages, and patterns for architecting IoT software, engineering IoTs for critical software, and developing industry scale and product line based IoTs. The active and emerging research indicates the next generations of solution for IoT systems. The

proposed mapping study aims to consolidate a collective knowledge in terms of progress, impacts, and limitations of the research to benefit:

- Researchers who may be interested in conceptualizing, understanding, and analyzing the body of knowledge that is based on the research and development of architecture-centric solution for IoT software. The systemized and structured knowledge, in terms of research taxonomy and catalogue of architectural solutions, can assist the researchers to gain insights into the progress of research to formulate innovative theories and establish new hypotheses to be tested. The mapping study can help the researchers to identify: (i) the frequency and evidence of research progression, (ii) research challenges with active and emerging themes, and (iii) architectural challenges for developing a wide range of IoT solutions.
- Practitioners who are interested to know about existing research and its implications to engineer and develop practical solutions (i.e., cloud-based systems and industrial IoTs). The mapping of the challenges and their architectural solutions can indicate the possible areas and approaches where academic research and development can be exploited for industrial solutions. The active and emerging trends of research highlight the needs for solutions that can exploit patterns and languages to develop industry scale IoTs.

Author Contributions: Conceptualization, A. Alreshidi and A. Ahmad; methodology, A. Alreshidi; validation, A. Ahmad; investigation, A. Alreshidi; data curation, A. Alreshidi and A. Ahmad; writing—original draft preparation, A. Alreshidi and A. Ahmad; writing—review and editing, A. Alreshidi and A. Ahmad; visualization, A. Ahmad; supervision, A. Alreshidi; project administration, A. Alreshidi; funding acquisition, A. Alreshidi.

Funding: This research was funded by Research Deanship at the University of Ha'il under the Grant Number 'BA-1514'.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

List of Qualitatively Selected Studies (Published Research) for the Systematic Mapping Study			
Study ID	Title of the Study	Publication Type	Publication Year
[S1]	Cloud Ecosystems Support for Internet of Things and DevOps Using Patterns	Conference	2016
[S2]	Internet of Things (IoT): Security challenges, business opportunities & reference architecture for E-commerce	Conference	2015
[S3]	A Taxonomy of IoT Client Architectures	Journal	2018
[S4]	An Open Internet of Things System Architecture based on Software-defined Device	Journal	2018
[S5]	Responsive Data Architecture for the Internet of Things	Journal	2016
[S6]	Architecting Emergent Configurations in the Internet of Things	Conference	2017
[S7]	Software architecture pattern selection model for Internet of Things based systems	Journal	2018
[S8]	A Software Architecture Enabling the Web of Things	Journal	2015
[S9]	Application of cloud computing in the emergency scheduling architecture of the Internet of Things	Conference	2015
[S10]	Modeling landing control system of carrier-based aircraft on Internet of Things	Conference	2014

List of Qualitatively Selected Studies (Published Research) for the Systematic Mapping Study

Study ID	Title of the Study	Publication Type	Publication Year
[S11]	Digital Enterprise Architecture—Transformation for the Internet of Things	Conference	2015
[S12]	Adaptive Transmission Optimization in SDN-Based Industrial Internet of Things with Edge Computing	Journal	2018
[S13]	A Scalable and Self-Configuring Architecture for Service Discovery in the Internet of Things	Journal	2014
[S14]	A Novel Secure Architecture for the Internet of Things	Conference	2011
[S15]	Designing a Cyber-Physical Cloud Computing Architecture	Journal	2015
[S16]	Building a Framework for Internet of Things and Cloud Computing	Conference	2014
[S17]	Software Abstractions for Component Interaction in the Internet of Things	Journal	2016
[S18]	From the Internet of Things to the web of things — enabling by sensing as-a service	Conference	2016
[S19]	A Component Architecture for the Internet of Things	Journal	2018
[S20]	Infrastructure Management Support in a Multi-agent Architecture for Internet of Things	Symposium	2014
[S21]	Software architecture of self-organizing systems-of-systems for the Internet-of-Things with SosADL	Conference	2017
[S22]	A Reference Model for Internet of Things Middleware	Journal	2018
[S23]	A SDN-based architecture for horizontal Internet of Things services	Conference	2016
[S24]	A Multiple Layer Security Architecture for Internet of Things into MVC Design	Conference	2018
[S25]	An hybrid and proactive architecture based on SDN for Internet of Things	Conference	2017
[S26]	Design and Implementation of a Hardware Versatile Publish-Subscribe Architecture for the Internet of Things	Journal	2018
[S27]	T-REST: An Open-Enabled Architectural Style for the Internet of Things	Journal	2018
[S28]	A Layered Protocol Architecture for Scalable Innovation and Identification of Network Economic Synergies in the Internet of Things	Conference	2016
[S29]	IoT gateway for smart metering in electrical power systems—software architecture	Conference	2017
[S30]	The Internet of Things (IoT): A Study of Architectural Elements	Conference	2017
[S31]	Research and Application of Manufacturing Recourse Management in Manufacturing Enterprise Based on Internet of Things	Conference	2017
[S32]	An Avatar Architecture for the Web of Things	Journal	2015
[S33]	Graph Based Clustering for Two-Tier Architecture in Internet of Things	Conference	2016
[S34]	A novel clock synchronization architecture for IoT access system	Conference	2016
[S35]	Self-Organised Middleware Architecture for the Internet-of-Things	Conference	2013
[S36]	A Commitment-Based Approach to Realize Emergent Configurations in the Internet of Things	Workshop	2017
[S37]	A Reference Architecture for Improving Security and Privacy in Internet of Things Applications	Conference	2014
[S38]	Complex Event Recognition Notification Methodology for Uncertain IoT Systems Based on Micro-Service Architecture	Conference	2018

List of Qualitatively Selected Studies (Published Research) for the Systematic Mapping Study			
Study ID	Title of the Study	Publication Type	Publication Year
[S39]	Research on intelligent supermarket architecture based on the Internet of Things technology	Symposium	2014
[S40]	CEFIoT: A fault-tolerant IoT architecture for edge and cloud	Forum	2018
[S41]	Microservices approach for the internet of things	Conference	2016
[S42]	Moving Application Logic from the Firmware to the Cloud: Towards the Thin Server Architecture for the Internet of Things	Conference	2012
[S43]	An Architecture to Support the Collection of Big Data in the Internet of Things	Conference	2014
[S44]	Edge computing enabling the Internet of Things	Forum	2015
[S45]	A distributed software-defined multi-agent architecture for unifying IoT applications	Conference	2017
[S46]	A resource oriented architecture for the Web of Things	Conference	2012
[S47]	IoT Mashup as a Service: Cloud-Based Mashup Service for the Internet of Things	Conference	2013
[S48]	A Reference Architecture for federating IoT infrastructures supporting semantic interoperability	Conference	2017
[S49]	Automatic Generation of Distributed Run-Time Infrastructure for Internet of Things	Workshop	2017
[S50]	Event-Aware Framework for Dynamic Services Discovery and Selection in the Context of Ambient Intelligence and Internet of Things	Journal	2016
[S51]	A Holistic Architecture for the Internet of Things, Sensing Services and Big Data	Conference	2013
[S52]	A Decentralized Locality-Preserving Context-Aware Service Discovery Framework for Internet of Things	Conference	2015
[S53]	Architecture and measured characteristics of a cloud based internet of things	Conference	2012
[S54]	CoTWARE: A Cloud of Things Middleware	Workshop	2017
[S55]	Self-Healing for Distributed Workflows in the Internet of Things	Workshop	2017
[S56]	A Notification Management Architecture for Service Co-evolution in the Internet of Things	Symposium	2016
[S57]	A management architectural pattern for adaptation system in Internet of Things	Conference	2016
[S58]	A message broker based architecture for context aware IoT application development	Conference	2017
[S59]	Thingtegrity: A Scalable Trusted Computing Architecture for the Internet of Things	Conference	2016
[S60]	A Reference Separation Architecture for Mixed-Criticality Medical and IoT Devices	Conference	2017
[S61]	An Architectural Mechanism for Resilient IoT Services	Workshop	2017
[S62]	A policy-based coordination architecture for distributed complex event processing in the internet of things: doctoral symposium	Conference	2016
[S63]	The design and development of an intelligent tutoring system as a part of the architecture of internet of things (IoT)	Conference	2017
[S64]	A new security middleware architecture based on fog computing and cloud to support IoT constrained devices	Conference	2017
[S65]	Towards a Reference Architecture for Swarm Intelligence-Based Internet of Things	Conference	2017
[S66]	A Graph-Based Cloud Architecture for Big Stream Real-Time Applications in the Internet of Things	Conference	2014
[S67]	Applying Architecture-Based Adaptation to Automate the Management of Internet-of-Things	Conference	2018

List of Qualitatively Selected Studies (Published Research) for the Systematic Mapping Study			
Study ID	Title of the Study	Publication Type	Publication Year
[S68]	Formally Describing Self-organizing Architectures for Systems-of-Systems on the Internet-of-Things	Conference	2018
[S69]	A Framework of Adaptive Interaction Support in Cloud-Based Internet of Things (IoT) Environment	Conference	2014
[S70]	An Architecture for Interoperable IoT Ecosystems	Workshop	2016
[S71]	An Open-Source Cloud Architecture for Big Stream IoT Applications	Conference	2015
[S72]	An Integrated Architecture for IoT-Aware Business Process Execution	Conference	2018
[S73]	Integrating Traditional Stores and e-Commerce into a Multi-tiered Recommender System Architecture Supported by IoT	Conference	2017
[S74]	Towards Defining Families of Systems in IoT: Logical Architectures with Variation Points	Summit	2015
[S75]	Architecture and Scheduling Method of Cloud Video Surveillance System Based on IoT	Conference	2015
[S76]	The Internet of Things: Insights into the building blocks, component interactions, and architecture layers	Conference	2018
[S77]	An energy-efficient internet of things (IoT) architecture for preventive conservation of cultural heritage	Journal	2018
[S78]	SecureSense: End-to-end secure communication architecture for the cloud-connected Internet of Things	Journal	2017
[S79]	COLLECT: COLLaborativE ConText-aware service oriented architecture for intelligent decision-making in the Internet of Things	Journal	2017
[S80]	A new architecture of Internet of Things and big data ecosystem for secured smart healthcare monitoring and alerting system	Journal	2018
[S81]	New Security Architecture for IoT Network	Workshop	2015
[S82]	Smart City Architecture and its Applications Based on IoT	Journal	2018
[S83]	Software architecture of self-organizing systems-of-systems for the Internet-of-Things with SosADL	Journal	2017
[S84]	Evaluating energy efficiency of Internet of Things software architecture based on reusable software components	Journal	2017
[S85]	Software-defined wireless network architectures for the Internet-of-Things	Conference	2015
[S86]	A System Architecture for Software-Defined Industrial Internet of Things	Conference	2015
[S87]	PMDA: A physical model driven software architecture for Internet of Things	Journal	2013
[S88]	EfiIoT: An efficient software architecture for internet of things	Conference	2012

References

1. Al-Fuqaha, A.; Guizani, M.; Mohammadi, M.; Aledhari, M.; Ayyash, M. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Commun. Surv. Tutor.* **2015**, *17*, 2347–2376. [[CrossRef](#)]
2. Zanella, A.; Bui, N.; Castellani, A.; Vangelista, L.; Zorzi, M. Internet of Things for Smart Cities. *IEEE Internet Things J.* **2014**, *1*, 22–32. [[CrossRef](#)]
3. Hamblen, J.O.; van Bekkum, G.M.E. An Embedded Systems Laboratory to Support Rapid Prototyping of Robotics and the Internet of Things. *IEEE Trans. Educ.* **2013**, *56*, 121–128. [[CrossRef](#)]
4. Evans, D. *The Internet of Things How the Next Evolution of the Internet Is Changing Everything*; White Paper; CISCO: San Jose, CA, USA, 2011.

5. Petersen, K.; Feldt, R.; Mujtaba, S.; Mattsson, M. Systematic Mapping Studies in Software Engineering. In Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering, Bari, Italy, 26–27 June 2008; Volume 8, pp. 68–77.
6. Capilla, R.; Jansen, A.; Tang, A.; Avgeriou, P.; Babar, M.A. 10 Years of Software Architecture Knowledge Management: Practice and Future. *J. Syst. Softw.* **2016**, *116*, 191–205. [CrossRef]
7. Wan, J.; Tang, S.; Shu, Z.; Li, D.; Wang, S.; Imran, M.; Vasilakos, A.V. Software-defined Industrial Internet of Things in the Context of Industry 4.0. *IEEE Sens. J.* **2016**, *16*, 7373–7380. [CrossRef]
8. Fremantle, P. *A Reference Architecture for the Internet of Things*; White Paper; WSO2: Mountain View, CA, USA, 2014. Available online: <http://wso2.com> (accessed on 19 June 2019).
9. Cavalcante, E.; Alves, M.P.; Batista, T.; Delicato, F.C.; Pires, P.F. An Analysis of Reference Architectures for the Internet of Things. In Proceedings of the 1st International Workshop on Exploring Component-based Techniques for Constructing Reference Architectures, Montréal, QC, Canada, 6 May 2015.
10. Weyrich, M.; Ebert, C. Reference Architectures for the Internet of Things. *IEEE Softw.* **2016**, *33*, 112–116. [CrossRef]
11. Campeanu, G. A Mapping Study on Microservice Architectures of Internet of Things and Cloud Computing Solutions. In Proceedings of the 2018 7th Mediterranean Conference on Embedded Computing (MECO), Budva, Montenegro, 10–14 June 2018.
12. Henry, M.; Moghaddam, M.T. IoT Architectural Styles. In *European Conference on Software Architecture*; Springer: Berlin, Germany, 2018.
13. Croes, E. Software Architectural Styles in the Internet of Things. Master's Thesis, Radboud University Nijmegen, Nijmegen, The Netherlands, August 2015.
14. Alreshidi, A.; Ahmad, A. *Protocol for Systematic Mapping Study on Software Architecture for the Internet of Things*; Technical Report; College of Computer Science and Engineering, University of Hail: Hail, Saudi Arabia, 2019; pp. 1–13. Available online: https://docs.wixstatic.com/ugd/396772_7ef3c1c76d3d47cbbbe21bd34d5a0da8.pdf (accessed on 19 June 2019).
15. Boyatzis, R.E. *Transforming Qualitative Information: Thematic Analysis and Code Development*; Sage: Newcastle Upon Tyne, UK, 1998.
16. Babar, M.A.; Ali, M.; Zhu, L.; Jeffery, R. A Framework for Classifying and Comparing Software Architecture Evaluation Methods. In Proceedings of the 2004 Australian Software Engineering Conference, Melbourne, Victoria, Australia, 13–16 April 2004; pp. 309–318.
17. Ahmad, A.; Jamshidi, P.; Pahl, C. Classification and Comparison of Architecture Evolution Reuse Knowledge—A Systematic Review. *J. Softw. Evol. Process* **2014**, *26*, 654–691. [CrossRef]
18. Jain, S.M. Monitoring IoT Gateways. U.S. Patent Application 15/706, 791, 21 March 2019.
19. Taherizadeh, S.; Stankovski, V.; Grobelnik, M. A Capillary Computing Architecture for Dynamic Internet of Things: Orchestration of Microservices from Edge Devices to Fog and Cloud Providers. *Sensors* **2018**, *18*, 2938. [CrossRef] [PubMed]
20. Ciczozzi, F.; Spalazzese, R. MDE4IoT: Supporting the Internet of Things with Model-Driven Engineering. In *International Symposium on Intelligent and Distributed Computing*; Springer: Berlin, Germany, 2016.
21. Internet of Things Related Standards. IEEE Standards Association. Available online: <https://standards.ieee.org/initiatives/iot/stds.html> (accessed on 2 February 2019).
22. Baccarelli, E.; Naranjo, P.G.; Scarpiniti, M.; Shojafar, M.; Abawajy, J.H. Fog of Everything: Energy-efficient Networked Computing Architectures, Research Challenges, and a Case Study. *IEEE Access* **2017**, *5*, 9882–9910. [CrossRef]
23. Naranjo, P.G.; Pooranian, Z.; Shojafar, M.; Conti, M.; Buyya, R. FOCAN: A Fog-supported Smart City Network Architecture for Management of Applications in the Internet of Everything Environments. *J. Parallel Distrib. Comput.* **2018**. [CrossRef]
24. Babar, M.A.; Dingsøyr, T.; Lago, P.; Vliet, H.V. *Software Architecture Knowledge Management*; Springer Science & Business Media: Berlin, Germany, 2009.
25. Mainetti, L.; Mighali, V.; Patrono, L. A Software Architecture Enabling the Web of Things. *IEEE Internet Things J.* **2015**, *2*, 445–454. [CrossRef]

