Secure Stitch: Unveiling the Fabric of Security Patterns for the Internet of Things *

Emiliia Geloczi†[0000−0001−9847−7656], Felix Klement†[0000−0001−9650−7698], Eva Gründinger, and Stefan Katzenbeisser

University of Passau, Innstraße 43, 94032 Passau, Germany
{emiliia.geloczi, felix.klement, stefan.katzenbeisser}@uni-passau.de
{gruendinger}@fim.uni-passau.de

Abstract. The design of the Internet of Things (IoT) system is a complex process, not only in terms of the balance between resource consumption and extensive functionality but also in the context of security. As various technical devices are now widespread and have access to all kinds of critical information, they become one of the main targets for attackers. Consequently, it is vital to consider the IT security aspect during the development of any system. A practical way to do it is to use security patterns. There are many different patterns that can address particular problems, but not all of them are suitable due to the wide range of requirements in such systems. In this paper, we present a systematic collection and categorisation of IoT-applicable security patterns and analyse gaps in recent research works related to security. We provide a catalogue of 61 patterns organised in a top-down approach that follows the World Forum's IoT Architecture Reference Model, this collection is able to play an important role in the future development of secure IoT solutions.

Keywords: IoT · IoT Security · Design Patterns.

1 Introduction

The proliferation of the Internet of Things (IoT) has ushered in a transformative paradigm wherein countless devices are interconnected, facilitating seamless communication and data exchange. Spanning domains such as smart homes, wearables, industrial systems, and smart cities, the IoT offers unparalleled convenience, efficiency, and connectivity. However, the extensive connectivity inherent in the IoT landscape also introduces significant security challenges [33]. As

* This work has been partially funded by the Bavarian Ministry of Science within the framework of the research cluster "ForDaySec: Security in everyday digitalisation", as well as, by the German Federal Ministry of Education and Research, as part of the Project "6G-RIC: The 6G Research and Innovation Cluster" (project number 825026).

† These authors contributed equally to this work and share first authorship

the IoT ecosystem expands, so does the attack surface, rendering it susceptible to an array of threats encompassing privacy breaches, data manipulation, physical harm, and critical infrastructure disruption. Addressing these security concerns assumes paramount importance in guaranteeing the trustworthiness and dependability of the IoT.

In the field of software development, the utilization of pre-established design patterns is a prevalent practice for addressing recurring issues. These patterns serve the purpose of not only circumventing known problems but also guaranteeing seamless integration and support for systems [35]. Nevertheless, not all security patterns hold the same level of applicability within the realm of the IoT, due to the presence of numerous and ever-evolving requirements [20]. Consequently, the adoption of more specialized patterns significantly diminishes the pool of suitable patterns tailored specifically for the IoT domain.

This paper introduces a comprehensive compilation of systematically organized design patterns that pertain to the mitigation of security challenges in the realm of the IoT. First, we identify existing design patterns according to the several chosen criteria. Then, the patterns are ranked based on seven architecture levels, five fundamental security objectives and ten common vulnerabilities. As a result, to our best knowledge, we provide the most comprehensive catalogue of design patterns suitable for solving security problems in the IoT, which consists of 61 elements and is organised according to a top-down approach. After the analysis of the catalogue, we discover that the included patterns cover all layers of the IoT architecture to varying degrees, address all considered security goals, and can also be used to mitigate the most common vulnerabilities.

The rest of the paper is organised as follows. Section 2 includes background information related to terms used in the catalogue. An overview of related works is presented in Section 3. In Section 4, the overall methodology is described including search and selection procedures. The resulting IoT security pattern catalogue is shown in Section 5. Section 6 contains the evaluation and discussion of the obtained results. The possible application of the presented catalogue is described in Section 7. Finally, Section 8 concludes the paper.

2 Background

To facilitate the reader's initiation into the Security Pattern discourse, we have synthesized the most important aspects. Within this section, we present a concise explanation of the terminology employed in formulating our comprehensive design pattern catalog.

2.1 World Forum Architecture Layers

During the creation of the catalogue, we classify the patterns according to the possible architecture levels at which they can be applied. For this purpose, we use the generally accepted seven architecture layers according to the World Forum Reference Model (WFRM) [4] (see Fig. 1).

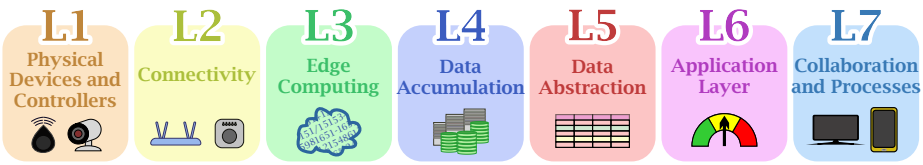

Fig. 1: Seven IoT architecture levels according to WFRM [4].

2.2 Top Ten Common IoT Vulnerabilities

To determine which vulnerabilities can be addressed by the patterns in the catalogue, we focus on the ten most common issues that have been identified by The Open Web Application Security Project (OWASP) community. The OWASP periodically updates and analyses critical problems regarding building and managing IoT systems. According to the last update, the top ten common IoT vulnerabilities are the following [17].

(T1) Weak Passwords
(T2) Insecure Network Services
(T3) Insecure Ecosystem Interfaces
(T4) Lack of Secure Update Mechanisms
(T5) Use of Insecure or Outdated Components
(T6) Insufficient Privacy Protections
(T7) Insecure Data Transfer and Storage
(T8) Lack of Device Management
(T9) Insecure Default Settings
(T10) Lack of Physical Hardening

2.3 Security Objectives

An adversary can pursue different goals, such as violating the confidentiality or integrity of information. In compiling our catalogue, we focus on the following five main possible targets of attackers and analyze the design patterns under consideration to determine whether they can ensure the protection of these targets [26]:

- Confidentiality: Data resources or information should be protected against unauthorized disclosure and improper use.
- Integrity: Data resources or information should be protected against unauthorized changes, destruction, or loss.
- Availability: Data resources or information are accessible to authorized users when they are needed.
- Authentication: Before a user can access information or resources, they must prove their identity and permission.
- Authorization: Verification of user permissions to access or use requested resources.

3 Related Work

There are several works describing the relevance of patterns and architectures that focus specifically on IoT. However, besides common design patterns and frameworks, security patterns in this area are still in their early stages of development and documentation. This section gives an overview of existing IoT pattern catalogues.

Reinfurt *et al.* [25] describe specific patterns for designing IoT systems that can be applied to the domain of smart factory systems. These patterns cover different areas and operation modes like device communication and management as well as energy supply types.

Besides design patterns also different architectural styles can be utilized for creating IoT systems. In [15], Muccini *et al.* provide a number of abstract reference architectures. Through the implementation of a systematic mapping study, a comprehensive selection process was undertaken, resulting in the identification of a set of 63 papers from a pool of over 2,300 potential works. The outcomes of this study play a crucial role in the classification of current and forthcoming approaches pertaining to architectural-level styles and patterns in the domain of IoT.

In order to get a better idea of the landscape of patterns and architectures that have accumulated over the years in research, Washizaki *et al.* [33, 34] analysed the successes and failures of patterns for IoT systems. The authors acknowledge that the development of IoT-specific patterns and architectures has substantial room for improvement due to limitations in documentation and a scarcity of successfully executed implementations.

Fysarakis *et al.* [9] sketch the SEMIoTICS approach to create a pattern-driven framework that is based on already existing IoT platforms. Aiming to guarantee secure actuation and semi-automatic behaviour, the SEMIoTICS project utilizes patterns to encode dependencies between security, privacy, dependability and interoperability properties of smart objects.

Organized in a hierarchical taxonomy, Papoutsakis *et al.* [18] collect and categorize a set of security and privacy patterns. While giving the reader an overview of security- and privacy-related objectives that are relevant in the IoT domain, the goal of this paper is to match these properties to their corresponding patterns. This usable pattern collection should guide developers to create IoT solutions that are secure and privacy-aware by design.

Over the last three years, Rajmohan *et al.* [21–23] published different papers that review the research work regarding patterns and architectures for IoT security and privacy. Despite rising in the number of publications in this area, there is a shortage of pattern IT security solutions at the Network and Device levels. Whereas the Physical Devices and Controller, Connectivity, and Application layers have the largest number of different security solutions.

Through our comprehensive analysis of the existing body of work, we can draw the conclusion that while there exists a multitude of design patterns applicable to the realm of IoT, there remains a notable dearth of design patterns specifically addressing some security concerns.

4 Methodology

This section introduces the methodology used for the creation of our catalogue. To begin with, we describe the chosen search strategy, and then outline the selection procedure including criteria based on which the founded papers have been filtered.

4.1 Search Strategy

As a base for a search of the existing papers relevant to our topic, we used the Systematic Literature Review (SLR) approach introduced by Kitchenham *et al.* [11]. SLR entails a methodical analysis of publications concerning a specific topic, encompassing the meticulous collection and critical evaluation of multiple research studies or papers. The objective of this study is to offer a comprehensive synthesis of the pertinent literature pertaining to a particular research question, ensuring transparency and reproducibility throughout the process. The following points utilize the review protocol that is used to conduct the literature review in a strategic manner and consist of the research questions that should be answered, selection criteria the found papers need to fulfil and a search strategy on how to browse databases in order to find the most relevant publications.

The strategy to find papers that discuss security patterns is divided into two main parts: automatic and manual search. To conduct our primary search, we employ five widely recognized scientific publication databases: IEEE Xplore, ACM Digital Library, ScienceDirect, Web of Science, and Scopus. Given that Scopus and ACM Digital Library already index SpringerLink, we exclude the former from our search process. Additionally, we omit Researchgate and Google Scholar, as they encompass a considerable number of non-peer-reviewed and non-English papers. Following the predefined set of search engines, we employ an automated approach utilizing specific keywords to identify relevant example studies. Subsequently, we proceed with a manual search to address any potential gaps, ensuring the inclusion of any pertinent scientific papers that may have been overlooked during the automated search process.

Furthermore, we meticulously assess all the obtained papers to determine their adherence to the following criteria:

Inclusion Criteria:

IC1 Paper contains (one or more) security pattern that is applicable to an IoT system.

IC2 Paper targets the IoT field, either in a general or specific application domain of IoT.

IC3 Paper discusses security objectives for system design, architecture or infrastructure.

Exclusion Criteria:

EC1 Paper is not written in English language.

EC2 Paper discusses design, privacy or misuse patterns, as well as security architectures not for the IoT domain.

EC3 Paper is not peer-reviewed.

In order for an article to be selected it must meet all inclusion criterion and none of the exclusion criteria:

$$(IC1 \wedge IC2 \wedge IC3) \wedge \overline{(EC1 \vee EC2 \vee EC3)} = 1$$

Lastly, to mitigate the presence of duplicate entries, the final stage of the search process involves the merging of identical papers identified by distinct search engines.

Automatic Search. Following the SLR approach, we identify several keywords in order to create appropriate requests for a successful automatic search. Based on our preceding analysis, we formulate the subsequent search query, which was subsequently employed for our initial database search:

(*"Internet of Things"* || *"IoT"* ||
"Cyber Physical Systems" || *"Web of Things"*)
∧
(*"Security Pattern"* || *"Security Design Pattern"*)

For each search the query string needed to be slightly modified to fit each database's advanced search functionality and guidelines.

Manual Search. By utilizing the snowballing strategy that was introduced by Wholin and Prikladnicki [36], we searched manually for further literature that was missed by the automatic database inquiry. Following references of the already found papers, we looked for relevant publications that include further security design patterns that can be useful for our study.

Following several iterations, we identify a collection of papers that fulfill the inclusion criteria outlined in our SLR. Upon eliminating duplicate entries previously identified during the initial database search, we are able to incorporate an additional eleven articles into our database search results.

4.2 Results of the Search and Selection Procedures

After conducting a search and selecting papers according to the step-by-step strategy described in Section 4.1, we obtain the following results (see Fig. 2): After an automatic search through five scientific databases, we select 160 suitable articles. Next, titles, abstracts and content are checked against the selected inclusion and exclusion criteria. Consequently, a total of nine appropriate articles are identified. Through a manual search, an additional eleven eligible studies are found, thus augmenting the search procedure's outcome to encompass a total of 20 articles. These articles collectively represent a comprehensive compilation of 61 design patterns specifically focused on addressing security concerns in the context of the IoT (see Table 1).

Table 1: Overview of primary IoT security pattern studies.

Year	Author	Title
2021	Fernández *et al.*	A Pattern for a Secure IoT Thing [2]
2021	Papoutsakis *et al.*	Towards a Collection of Security and Privacy Patterns [19]

Year	Author	Title
2020	Fernández *et al.*	Abstract and IoT security segmentation patterns [6]
2020	Fernández *et al.*	Secure Distributed Publish/Subscribe (P/S) pattern for IoT [7]
2020	Fernández *et al.*	A Pattern for a Secure Cloud-Based IoT Architecture [8]
2020	Muñoz *et al.*	TPM, a Pattern for an Architecture for Trusted Computing [14]
2020	Orellana *et al.*	A Pattern for a Secure Sensor Node [16]
2019	Moreno *et al.*	BlockBD: A Security Pattern to Incorporate Blockchain... [13]
2018	Ali *et al.*	Applying security patterns for authorization of users in IoT-based app-s [1]
2018	Schuß *et al.*	IoT Device Security the Hard(Ware) Way [27]
2018	Seitz *et al.*	Fogxy: An Architectural Pattern for Fog Computing [28]
2018	Tkaczyk *et al.*	Cataloging design patterns for internet of things artifact integration [31]
2017	Lee *et al.*	A case study in applying security design patterns for IoT... [12]
2017	Reinfurt *et al.*	Internet of Things Security Patterns [24]
2016	Sinnhofer *et al.*	Patterns to Establish a Secure Communication Channel [29]
2016	Syed *et al.*	A Pattern for Fog Computing [30]
2015	Ur-Rehman *et al.*	Secure Design Patterns for Security in Smart Metering Systems [32]
2014	Ciria *et al.*	The History–Based Authentication pattern [3]
2007	Fernández *et al.*	Security Patterns for Physical Access Control Systems [5]
2005	Kienzle *et al.*	Security patterns repository, version 1.0 [10]

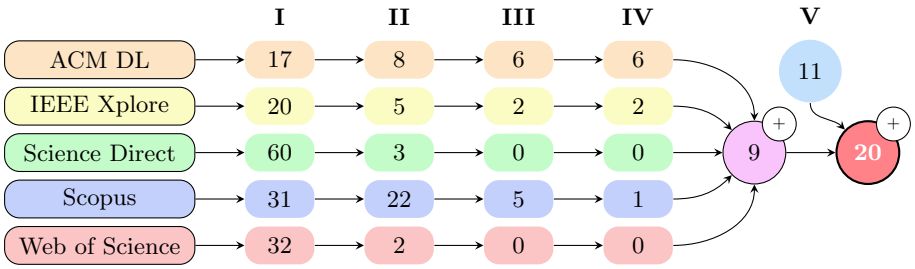

Fig. 2: Overview of the search and selection procedure results consisted of the following states: results after initial search (I), after reviewing the title and abstract (II), after scanning content (III), after cross-check information (IV) and manual search results (V).

5 Catalogue

In this section, all IoT security design patterns collected during the previously explained search process are listed in the form of a catalogue for developers.

Table 2 represents our catalogue and is divided into seven layers according to WFRM, where each pattern corresponds to a specific layer. Additionally, the possibility to solve ten vulnerabilities using each design pattern is reflected. Finally, security objectives are also mentioned that are either addressed (●) or not (○) by this particular pattern, the decisions are made based on the original description of the design pattern.

Table 2: IoT security pattern lookup table.

Layer	Pattern Name	Solution for T										Sec. Objectives				
		1	2	3	4	5	6	7	8	9	10	C	I	A	Ac	Az
L1	Hardware IoT Security [27]		✓		✓	✓						●	○	○	●	○
	Secure IoT Thing [2]		✓	✓				✓				●	●	●	○	○

Layer	Pattern Name	Solution for T										Sec. Objectives				
		1	2	3	4	5	6	7	8	9	10	C	I	A	Ac	Az
	Secure Sensor Node [16]			✓		✓		✓				●	●	●	●	●
	Security Segmentation [6]			✓								○	○	○	●	●
	Trusted Platform Module [14]					✓		✓				●	●	○	●	●
L2	Authenticated Channel [19]		✓					✓				○	●	○	●	○
	Encrypted Channel [19]		✓					✓				●	○	○	○	○
	Middleware Message Broker [31]			✓				✓				●	○	○	○	○
	Middleware Selfcontained Message [31]							✓				●	○	○	○	○
	Orchestration of SDN Network Elements [31]			✓				✓				●	○	○	○	○
	Outbound-Only Connection [24]						✓	✓				●	○	○	○	●
	Password-Based Key Exchange [29]	✓						✓				●	○	○	○	○
	Safe Channel [19]		✓					✓				○	●	○	○	○
	Secure Remote Readout [32]							✓				●	●	○	●	○
	Signed Message [19]							✓				○	●	○	●	○
	Symmetric Key Cryptography [29]							✓				●	○	○	●	○
	Third Party Based Authentication [29]							✓				●	●	○	●	○
	Trusted Communication Partner [24]						✓	✓				●	○	●	●	●
	Web of Trust [29]							✓				○	●	○	●	○
L3	Fog Computing [30]		✓	✓				✓				●	●	●	●	●
	Fogxy [28]		✓	✓				✓				○	○	●	●	●
	Secure Cloud-based IoT Architecture [8]			✓				✓				●	●	●	●	●
L4	Encrypted Storage [10]						✓	✓				●	○	○	○	○
	Redundant Storage [19]							✓				○	○	●	○	○
	Safe Storage [19]						✓	✓				○	●	○	○	○
L5	Alignment-based Translation Pattern [31]			✓								○	○	●	○	○
	BlockBD [13]			✓				✓				●	●	●	○	○
	Discovery of IoT Services [31]			✓								○	○	●	○	○
	Flow-based Service Composition [31]		✓					✓				●	○	○	○	○
	IoT Gateway Event Subscription [31]			✓				✓				●	○	●	○	○
	IoT SSL Cross-Layer Secure Access [31]			✓								●	●	○	●	●
	Middleware Message Translator [31]			✓								○	○	●	○	○
	Middleware Simple Component [31]											●	○	○	○	○
	D2D REST Request/Response [31]			✓								●	○	●	○	●
	Server Sandbox [10]									✓	✓	●	●	●	○	○
	Service Orchestration [31]			✓				✓				●	○	○	○	○
	Translation with Central Ontology [31]			✓								○	○	●	○	○
L6	Access Control to Physical Structures [5]										✓	○	○	○	●	●
	Alarm Monitoring [5]								✓			○	○	○	○	●
	Audit Log [12, 19]								✓			○	●	○	○	○
	Authenticated Session [19]			✓								○	○	○	●	○
	Authorization Enforcer [19]								✓			○	○	○	○	●
	Encrypted Processing [19]							✓				●	○	○	○	○
	Fault Management [19]								✓			○	○	●	○	○
	File Authentication [1]							✓				○	○	○	●	●
	Matrix Authentication [1]							✓				○	○	○	●	●
	Minefield [10]					✓				✓		○	●	○	○	○
	Remote Authenticator/Authorizer [1]							✓				○	○	○	●	●
	Role Based Access Control [1]							✓				○	○	○	○	●
	Safe Processing [19]							✓				○	●	○	○	○
	Secure Distributed Publish/Subscribe [7]							✓				●	●	●	●	●
	Uptime [19]								✓			○	○	●	○	○
L7	Account Lockout [19]	✓										○	○	○	●	○
	Authentication Enforcer [19]			✓								○	○	○	●	○
	Blacklist [19, 24]			✓					✓			○	○	○	●	●
	History-Based Authentication [3]	✓										●	●	○	●	●
	Permission Control [24]						✓		✓	✓		●	○	○	○	●
	Personal Zone Hub [24]						✓					●	○	○	○	●
	Relays [5]											○	○	○	○	●
	Single Access Point [19]			✓								○	○	○	●	●
	Whitelist [24]			✓					✓			○	○	○	●	●

6 Discussion

To ascertain potential avenues for the expansion of our research, we conducted a thorough analysis of the compiled data regarding the patterns' capabilities in operating across various architectural levels, mitigating prevalent vulnerabilities, and addressing paramount security properties. To facilitate a more comprehensive exploration of our findings, we formulate several research questions, which were subsequently addressed and answered through a structured examination of the results.

RQ1: Which layer in the IoT WFRM is covered by the least security patterns? In order to answer this question, we calculated the distribution of patterns for each WFRM layer. The results are illustrated with a corresponding pie chart in Fig. 3.

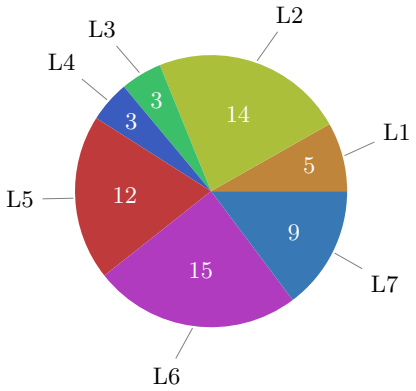

Fig. 3: Distribution of IoT security patterns among the WFRM architecture layers.

According to the calculated distribution of IoT security patterns that are attributed to the different architecture layers, most patterns can be almost equally found in the Connectivity (L2) and Application (L6) layers. On the other hand, the Edge Computing (L3) and Data Accumulation (L4) layers have only three patterns each, hence, they are on the lower end of the pattern coverage. Upon closer examination of the underlying factors contributing to this phenomenon, it can be observed that Edge Computing can be regarded as an autonomous technology infrastructure, which is not universally recognized as an integral component of the IoT across all models and frameworks. If we specifically search for Edge functionality in publications, our success rate in finding such patterns would definitely be significantly higher. But because IoT is the main focus of our research topic, only a few publications that specified IoT as well as Edge technology at the same time could be found. The underrepresentation of data accumulation (L4) in the safety patterns which we observe in our study provides an intriguing foundation for future research endeavors, warranting further in-depth analysis and investigation. The hypothesis is that the limited number of

patterns currently available for addressing security issues in IoT devices can be attributed to the significant role of secure storage in both mobile and stationary computing systems. Thus far, only a few patterns have been developed with a specific focus on resolving security challenges in the realm of IoT devices. In summary, to achieve a more balanced distribution of patterns within the WFRM architecture, it is strongly encouraged to focus on the development of security patterns specifically designed for the Data Accumulation layer in the context of the IoT.

RQ2: Which security goals are covered by the patterns? In order to address the posed inquiry, we conduct an assessment of data encompassing design patterns and security objectives, as presented in Table 1. The tabulated results, available in Table 3, illustrate the cumulative frequency at which each security goal is addressed by the identified patterns.

Table 3: Security objectives addressed by IoT security patterns.

Security Objective	Pattern Count
Confidentiality	30
Integrity	20
Availability	18
Authentication	27
Authorization	24

With a total of 30 design patterns, the confidentiality objective is the goal that is covered the most in our data set. Given its paramount importance, the protection of sensitive data is typically accorded the highest priority among various security requirements. Hence, even if an IoT system is built without any security aspects in mind, the probability that confidentiality is ensured is pretty high. Therefore, there are many patterns that guarantee this objective. Availability, however, is the security goal with the least amount of coverage. The target is to ensure that a system is accessible on user demand can be quite challenging.

Fig. 4 presents the coverage of different security objectives in each layer of the WFRM architecture by design patterns in consideration.

The outlier in the bar plot is definitely the Data Accumulation (L4) layer. With only three security goals being covered and authentication and authorization being absent entirely, hence, we can assume the lack of security solutions for the storage of IoT devices. However, the reason for authentication being neglected lies in these mechanisms usually being implemented in higher layers of the IoT architecture. Interesting to mention is also the lack of availability support in the Collaboration & Processes (L7) layer. But because this layer is focused on user interactions and not the applications of the system themselves, it makes sense that availability is not a priority here.

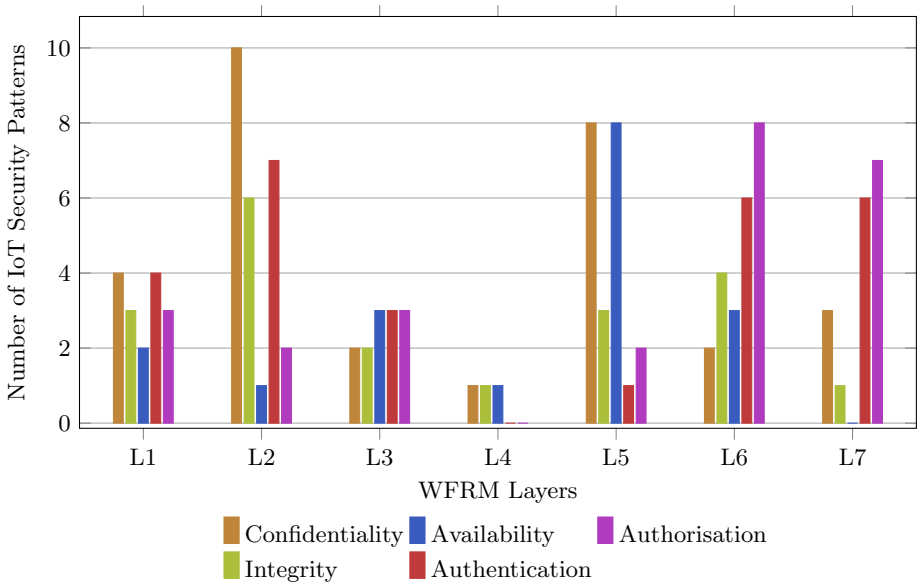

Fig. 4: IoT security patterns with addressed security concerns according to WFRM layers.

RQ3: Which vulnerabilities from the OWASP Top Ten IoT list are possible to solve by security patterns included in our catalogue? Fig. 5 shows which common vulnerabilities can be solved with the found IoT security patterns.

According to the obtained results, we can notice that each common vulnerability defined by OWASP can be solved by at least one IoT security pattern from our catalogue. The most covered is T7 which focuses on insecure data transfer and storage. At least 34 different IoT security patterns that we found have a solution to enhance the security of data handling in IoT devices. On the other hand, T4 is apparently the hardest one to solve with only one pattern addressing this issue. If we look into its description, the problem is the lack of ability to securely update the IoT device. This is a very specific issue that also is highly dependent on the hardware of the device and its user interface. For better update management of IoT devices, further research in terms of security patterns is highly recommended in this area.

Additionally, we examine the WFRM layer distribution of the IoT security patterns for each individual OWASP vulnerability. In Fig. 6, the pie charts display the more detailed results of the previous bar plot. While the pie charts for T2, T3 and T7 show the most diverse range of pattern solutions from four up to six different layers, T4 is covered by only patterns from the Physical Devices & Controllers (L1) layer.

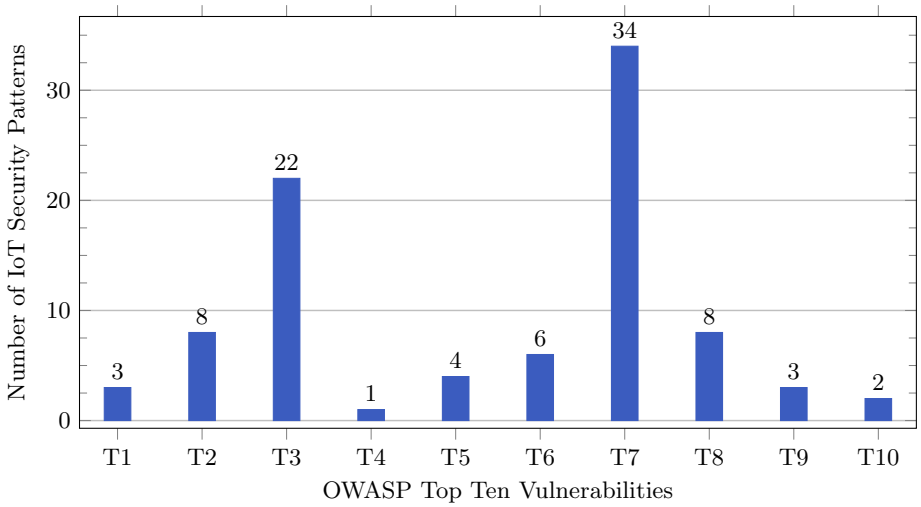

Fig. 5: Number of pattern solutions for the OWASP common vulnerabilities.

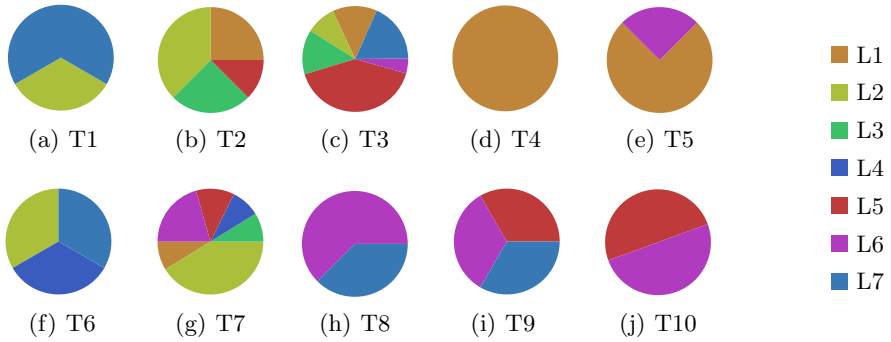

Fig. 6: Layer distribution of pattern solutions for OWASP common vulnerabilities.

Therefore, we see a connection between the found IoT security patterns for a specific OWASP vulnerability and the number of covered layers. This assumption is confirmed by the results displayed in Fig. 7. It showcases the correlation between the pattern quantity and their layer distribution with a value of 0.868. This correlation coefficient always ranges from -1 to 1 and indicates the strength of the relationship between two variables. A value between 0.7 and 1 shows a strong connection and the positive sign indicates, that more patterns as solutions for a specific vulnerability also mean a more diverse distribution of WFRM layers for these IoT patterns.

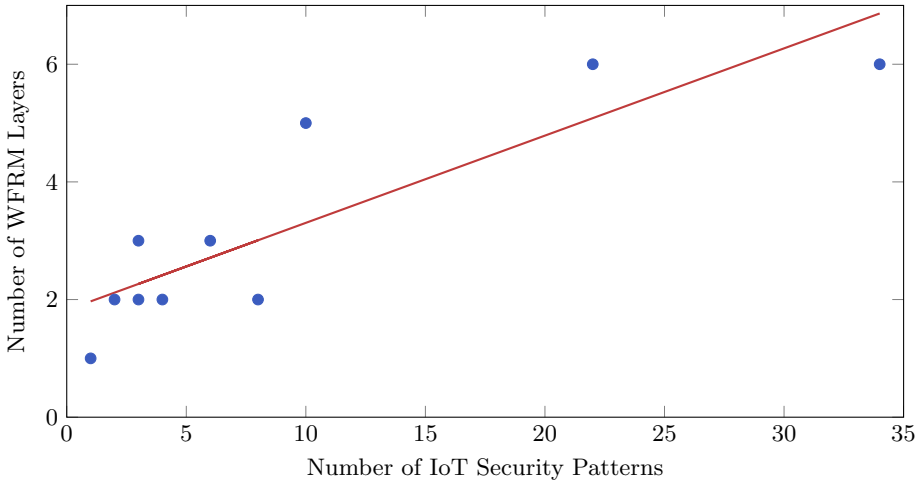

Fig. 7: The correlation between the number of patterns and WFRM layers for each OWASP vulnerability.

7 Use Case Application

In order to demonstrate a possible application of our catalogue, we have chosen the most common and at the same time vulnerable domain where IoT devices are used, specifically Smart Home. The selected scenario is simple to understand and implement, however, it encompasses most of the typical communications and activities on IoT networks that may have various vulnerabilities that need to be addressed.

Before proceeding with the detailed description, it is important to note, that in this example we assume that the system is only used by the owner of the house and no further security measures were taken than the ones that were already integrated into the system.

This smart home contains different connected devices that are distributed in the living room, bedroom, kitchen, bathroom and entrance of the house. All electronic devices run on the custom firmware Tasmota and include four RGB LED bulbs, six light controllers, and five smart plugs. Additionally, a Raspberry PI with HomeBridge allows the integration of HomeKit into the network that controls the following devices: two televisions, four Sonos ZonePlayers, a Ring camera and a doorbell.

After we established the use case example, we can inspect each pattern of our catalogue and check its applicability in the given context. The evaluation results can be found in Table 4 with (-) indicating this pattern is not suitable for this system, (o) being used in cases where the given system has already implemented similar security features this pattern would provide, and (+) marking recommended patterns to implement to further optimize the system design.

Table 4: Use case applicability of design patterns.

Layer	Pattern Name	Rating	Explanation
L1	Hardware IoT Security [27]	+	Exchangeable cryptographic co-processors to secure IoT devices.
	Secure IoT Thing [2]	+	Secure any entity that is connected to sensors/actuators, e.g. Raspberry PI.
	Secure Sensor Node [16]	-	System does not include sensor nodes.
	Security Segmentation [6]	+	IoT devices are divided into subnetworks.
	Trusted Platform Module [14]	+	Attestation of Raspberry PI with integrated cryptographic services.
L2	Authenticated Channel [19]	o	Mutual authentication of communication partners and forward secrecy.
	Encrypted Channel [19]	o	TLS handshake and exchange of cryptographic information.
	Middleware Message Broker [31]	o	HomeBridge controls the flow of messages between IoT devices.
	Middleware Self-contained Message [31]	+	Messages should be "pure and complete" representations of events/commands.
	Orchestration of SDN Network Elements [31]	-	Only required when an IoT SDN is employed.
	Outbound-Only Connection [24]	+	Blocks incoming malicious connection requests.
	Password-Based Key Exchange [29]	+	Common secret is used to generate session key pairs.
	Safe Channel [19]	o	Use certificates to guarantee integrity during message transmission.
	Secure Remote Readout [32]	+	Security Module encrypts measurements before transmitting.
	Signed Message [19]	o	Use digital signatures during the message generation/exchange process.
	Symmetric Key Cryptography [29]	+	Handshake and common secret are exchanged between communication parties.
	Third Party Based Authentication [29]	+	Combination of asymmetric cryptography and session keys.
	Trusted Communication Partner [24]	+	List trusted communication partners and block unknown connection requests.
	Web of Trust [29]	-	Tasmota uses a central self-signed certificate authority.
L3	Fog Computing [30]	-	No cloud-based system.
	Fogxy [28]	-	No cloud-based system.
	Secure Cloud-based IoT Architecture [8]	-	No cloud-based system.
L4	Encrypted Storage [10]	+	Critical data is encrypted before it gets committed to disk.
	Redundant Storage [19]	-	No cloud-based system.
	Safe Storage [19]	+	Guarantee integrity of stored data.
L5	Alignment-based Translation Pattern [31]	o	HomeBridge enables interoperability between different platforms.
	BlockBD [13]	-	No Big Data system.
	Discovery of IoT Services [31]	-	No usage of different IoT services.
	Flow-based Service Composition [31]	-	No usage of different IoT services.
	IoT Gateway Event Subscription [31]	o	HomeBridge sends notifications on updates.
	IoT SSL Cross-Layer Secure Access [31]	o	Only authenticated entities are able to access the external interfaces.
	Middleware Message Translator [31]	o	HomeBridge enables interoperability between different platforms.
	Middleware Simple Component [31]	+	Universally applicable pattern to achieve the best component decomposition.
	D2D REST Request/Response [31]	o	HomeBridge API is used to connect to different IoT devices.

Layer	Pattern Name	Rating	Explanation
	Server Sandbox [10]	+	Isolate server to protect it in case the system gets compromised.
	Service Orchestration [31]	-	No usage of different IoT services.
	Translation with Central Ontology [31]	o	HomeBridge enables interoperability between different platforms.
L6	Access Control to Physical Structures [5]	-	No physical structures need to be accessed.
	Alarm Monitoring [5]	o	Alarm functionality is included in HomeBridge.
	Audit Log [12, 19]	o	HomeBridge has a rolling log screen.
	Authenticated Session [19]	-	System runs on a local server with no internet requirements.
	Authorization Enforcer [19]	-	Only relevant if a system is used by users with different roles.
	Encrypted Processing [19]	+	Integrity of data with e.g. homomorphic functions.
	Fault Management [19]	+	Smart handling of any faulty behaviour of the system.
	File Authentication [1]	-	Only relevant if a system is used by users with different privileges.
	Matrix Authentication [1]	-	Only relevant if a system is used by users with different privileges.
	Minefield [10]	+	Modify Raspberry PI to confuse attackers and simplify threat detection.
	Remote Authenticator/Authorizer [1]	-	System runs on a local server with no internet requirements.
	Role Based Access Control [1]	-	Only relevant if a system is used by users with different roles.
	Safe Processing [19]	+	Guarantee integrity during data processing with e.g. integrity checks.
	Secure Distributed Publish/Subscribe [7]	o	HomeBridge sends notifications on updates.
	Uptime [19]	o	HomeBridge measures and displays the server availability.
L7	Account Lockout [19]	o	Login via password authentication.
	Authentication Enforcer [19]	+	Authentication process that creates proof of identity.
	Blacklist [19, 24]	+	Identification of abusers who are not granted access to the system.
	History-Based Authentication [3]	+	Authentication is based on the user's own history.
	Permission Control [24]	+	User can control which data is shared with the server.
	Personal Zone Hub [24]	-	No cloud-based system.
	Relays [5]	-	No switches in the system.
	Single Access Point [19]	+	Only one entry point into the system with HomeBridge UI.
	Whitelist [24]	+	Identification of trusted partners.

While there are many patterns that are not suitable to be implemented in this kind of smart home scenario, like BlockBD [13] or the Web of Trust [29], just as many are already integrated into the system, e.g. Uptime [19] or Account Lockout [19]. Nevertheless, by going through the catalogue and analysing each pattern individually, we found 25 patterns that can be used to optimize the security measures in this smart home example. Spread across all layers of the WFRM architecture, one can choose from a variety of patterns that include Symmetric Key Cryptography [29], Server Sandbox [10] or even simpler solutions like a combination of a Blacklist [19, 24] and a Whitelist [24].

This use case demonstrates that our template catalogue can serve as a simple guide to improving system security.

8 Conclusion

The aim of this analysis was to create a comprehensive catalogue of IoT security design patterns and to provide a guide for the future development of secure IoT systems.

In the beginning, we defined which IoT devices belong to the IoT spectrum. Further, we selected and described the base elements of our catalogue, such as the list of IoT WFRM architecture layers, the common IoT vulnerabilities and the most important security objectives.

In order to obtain representative results, we answer on three following questions during our research:

RQ1: Which layer in the IoT WFRM is covered by the least security patterns?
RQ2: Which security goals are covered by the patterns?
RQ3: Which vulnerabilities from the OWASP Top Ten IoT list are possible to solve by security patterns included in our catalogue?

Among the 61 design patterns in the catalogue, almost half are applied at two of the seven layers of conventional architecture. We also found a lack of coverage of security goals at the Data Accumulation level. On the other hand, every vulnerability out of 10 on the OWASP list was addressed by at least one pattern, which is a positive discovery.

A collection of security patterns in the IoT field is a good start to get an overview of the current state-of-the-art. But there are many other ways in which researchers and developers can advance secure IoT development and utilize the advantages of standardization. For future work, we identify two possibilities.

The first one is the pattern catalogue expansion. Our IoT security pattern catalogue cannot be called complete in any way. There are surely more security patterns that can be modified into the IoT context as well as other types of patterns that can make the implementation of secure IoT systems easier. A few examples would be privacy patterns, misuse patterns or anti-patterns. Therefore, the expansion of the security pattern catalogue for IoT is definitely a topic for further research.

As a second trajectory for the development of this work, we propose industry practical validation. Technology and science are ever-evolving, therefore the need for different types of patterns for common problems are always exist and require new and modern solutions. The best way to develop new security patterns, that are optimized for applicability and usage in real-world situations, is cooperation with the industry. Only when academia combines its theories and ideas with the practical problems of the corresponding industry, we are able to find the best solutions to solve common issues in the world of IoT.

References

1. Ali, I., Asif, M.: Applying security patterns for authorization of users in iot based applications. In: 2018 International Conference on Engineering and Emerging Technologies (ICEET). pp. 1–5 (Feb 2018)
2. B. Fernández, E., Astudillo, H., Orellana, C.: A pattern for a secure iot thing. In: 26th European Conference on Pattern Languages of Programs. EuroPLoP'21, Association for Computing Machinery, New York, NY, USA (2021), https://doi.org/10.1145/3489449.3489988
3. Ciria, J.C., Domínguez, E., Escario, I., Francés, A., Lapeña, M.J., Zapata, M.A.: The history-based authentication pattern. In: Proceedings of the 19th European Conference on Pattern Languages of Programs. EuroPLoP '14, Association for Computing Machinery, New York, NY, USA (2014), https://doi.org/10.1145/2721956.2721960
4. El Hakim, A.: Internet of things (iot) system architecture and technologies, white paper. (03 2018). https://doi.org/10.13140/RG.2.2.17046.19521
5. Fernandez, E.B., Ballesteros, J., Desouza-Doucet, A.C., Larrondo-Petrie, M.M.: Security patterns for physical access control systems. In: Barker, S., Ahn, G.J. (eds.) Data and Applications Security XXI. vol. 4602, pp. 259–274 (07 2007)
6. Fernández, E., Fernandez, E., Yoshioka, N., Washizaki, H.: Abstract and iot security segmentation patterns (01 2020)
7. Fernández, E., Yoshioka, N., Washizaki, H.: Secure distributed publish/subscribe (p/s) pattern for iot (02 2020)
8. Fernández, E.B.: A pattern for a secure cloud-based iot architecture. In: Proceedings of the 27th Conference on Pattern Languages of Programs. PLoP '20, The Hillside Group, USA (2020)
9. Fysarakis, K., Spanoudakis, G., Petroulakis, N., Soultatos, O., Bröring, A., Marktscheffel, T.: Architectural patterns for secure iot orchestrations. In: 2019 Global IoT Summit (GIoTS). pp. 1–6 (2019). https://doi.org/10.1109/GIOTS.2019.8766425
10. Kienzle, D.M., Elder, M.C., D, P., D, P., Tyree, D., Edwards-hewitt, J.: Security patterns repository, version 1.0 (2006)
11. Kitchenham, B., Charters, S.: Guidelines for performing systematic literature reviews in software engineering **2** (01 2007)
12. Lee, W.T., Law, P.J.: A case study in applying security design patterns for iot software system. In: 2017 International Conference on Applied System Innovation (ICASI). pp. 1162–1165 (May 2017)
13. Moreno, J., Fernandez, E.B., Fernandez-Medina, E., Serrano, M.A.: Blockbd: A security pattern to incorporate blockchain in big data ecosystems. In: Proceedings of the 24th European Conference on Pattern Languages of Programs. EuroPLop '19, Association for Computing Machinery, New York, NY, USA (2019), https://doi.org/10.1145/3361149.3361166
14. Muñoz, A., Fernandez, E.B.: Tpm, a pattern for an architecture for trusted computing. In: Proceedings of the European Conference on Pattern Languages of Programs 2020. EuroPLoP '20, Association for Computing Machinery, New York, NY, USA (2020), https://doi.org/10.1145/3424771.3424781
15. Muccini, H., T. Moghaddam, M.: IoT Architectural Styles, pp. 68–85 (01 2018). https://doi.org/10.1007/978-3-030-00761-4$_5$
16. Orellana, C., Fernandez, E.B., Astudillo, H.: A pattern for a secure sensor node. In: Proceedings of the 27th Conference on Pattern Languages of Programs. PLoP '20, The Hillside Group, USA (2020)

17. OWASP: Iot top 10, https://owasp.org/www-pdf-archive/OWASP-IoT-Top-10-2018-final.pdf
18. Papoutsakis, M., Fysarakis, K., Spanoudakis, G., Ioannidis, S., Koloutsou, K.: Towards a collection of security and privacy patterns. Applied Sciences **11**, 1396 (02 2021). https://doi.org/10.3390/app11041396
19. Papoutsakis, M., Fysarakis, K., Spanoudakis, G., Ioannidis, S., Koloutsou, K.: Towards a collection of security and privacy patterns. Applied Sciences **11**(4) (2021), https://www.mdpi.com/2076-3417/11/4/1396
20. Qanbari, S., Pezeshki, S., Raisi, R., Mahdizadeh, S., Rahimzadeh, R., Behinaein, N., Mahmoudi, F., Ayoubzadeh, S., Fazlali, P., Roshani, K., Yaghini, A., Amiri, M., Farivarmoheb, A., Zamani, A., Dustdar, S.: Iot design patterns: Computational constructs to design, build and engineer edge applications. In: 2016 IEEE First International Conference on Internet-of-Things Design and Implementation (IoTDI). pp. 277–282 (2016). https://doi.org/10.1109/IoTDI.2015.18
21. Rajmohan, T., Nguyen, P., Ferry, N.: A systematic mapping of patterns and architectures for iot security (03 2020)
22. Rajmohan, T., Nguyen, P., Ferry, N.: A decade of research on patterns and architectures for iot security. Cybersecurity **5** (01 2022). https://doi.org/10.1186/s42400-021-00104-7
23. Rajmohan, T., Nguyen, P.H., Ferry, N.: Research landscape of patterns and architectures for iot security: A systematic review. In: 2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA). pp. 463–470 (2020). https://doi.org/10.1109/SEAA51224.2020.00079
24. Reinfurt, L., Breitenbücher, U., Falkenthal, M., Fremantle, P., Leymann, F.: Internet of things security patterns. In: Proceedings of the 24th Conference on Pattern Languages of Programs. PLoP '17, The Hillside Group, USA (2017)
25. Reinfurt, L., Falkenthal, M., Breitenbücher, U., Leymann, F.: Applying IoT Patterns to Smart Factory Systems. In: Proceedings of the 11th Advanced Summer School on Service Oriented Computing. pp. 1–10. IBM Research Division (2017)
26. Samonas, S., Coss, D.: The cia strikes back: Redefining confidentiality, integrity and availability in security. Journal of Information System Security **10**(3) (2014)
27. Schuß, M., Iber, J., Dobaj, J., Kreiner, C., Boano, C.A., Römer, K.: Iot device security the hard(ware) way. In: Proceedings of the 23rd European Conference on Pattern Languages of Programs. EuroPLoP '18, Association for Computing Machinery, New York, NY, USA (2018), https://doi.org/10.1145/3282308.3282329
28. Seitz, A., Thiele, F., Bruegge, B.: Fogxy: An architectural pattern for fog computing. In: Proceedings of the 23rd European Conference on Pattern Languages of Programs. EuroPLoP '18, Association for Computing Machinery, New York, NY, USA (2018), https://doi.org/10.1145/3282308.3282342
29. Sinnhofer, A.D., Oppermann, F.J., Potzmader, K., Orthacker, C., Steger, C., Kreiner, C.: Patterns to establish a secure communication channel. In: Proceedings of the 21st European Conference on Pattern Languages of Programs. EuroPlop '16, Association for Computing Machinery, New York, NY, USA (2016). https://doi.org/10.1145/3011784.3011797, https://doi.org/10.1145/3011784.3011797
30. Syed, M.H., Fernandez, E.B., Ilyas, M.: A pattern for fog computing. In: Proceedings of the 10th Travelling Conference on Pattern Languages of Programs. VikingPLoP '16, Association for Computing Machinery, New York, NY, USA (2016), https://doi.org/10.1145/3022636.3022649

31. Tkaczyk, R., Wasielewska, K., Ganzha, M., Paprzycki, M., Pawlowski, W., Szmeja, P., Fortino, G.: Cataloging design patterns for internet of things artifact integration. In: 2018 IEEE International Conference on Communications Workshops (ICC Workshops). pp. 1–6 (2018)
32. Ur-Rehman, O., Zivic, N.: Secure design patterns for security in smart metering systems. In: 2015 IEEE European Modelling Symposium (EMS). pp. 278–283 (2015)
33. Washizaki, H., Ogata, S., Hazeyama, A., Okubo, T., Fernandez, E.B., Yoshioka, N.: Landscape of architecture and design patterns for iot systems. IEEE Internet of Things Journal **7**(10), 10091–10101 (2020). https://doi.org/10.1109/JIOT.2020.3003528
34. Washizaki, H., Yoshioka, N., Hazeyama, A., Kato, T., Kaiya, H., Ogata, S., Okubo, T., Fernandez, E.B.: Landscape of iot patterns. In: 2019 IEEE/ACM 1st International Workshop on Software Engineering Research Practices for the Internet of Things (SERP4IoT). pp. 57–60 (2019). https://doi.org/10.1109/SERP4IoT.2019.00017
35. Wedyan, F., Abufakher, S.: Impact of design patterns on software quality: a systematic literature review. IET Software **14**(1), 1–17 (2020). https://doi.org/https://doi.org/10.1049/iet-sen.2018.5446, https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-sen.2018.5446
36. Wohlin, C., Prikladniki, R.: Systematic literature reviews in software engineering. Information and Software Technology **55**, 919–920 (06 2013). https://doi.org/10.1016/j.infsof.2013.02.002