# A Guided Modeling Approach for Secure System Design

Alex R. Sabau
Research Group Software Construction
RWTH Aachen University
Ahornstraße 55
Aachen, Germany
sabau@swc.rwth-aachen.de

***Abstract*—Designing security architectures (SA) that are maintained independently from the system architectures is a well-researched and established approach for modeling the security perspective of software systems. However, this approach involves several drawbacks in the documentation of SAs. These include an increase in redundancies in the design documents, synchronization errors due to concurrent modification of separate models, and expert knowledge required to design SAs, which is a crucial constraint due to the lack of experts in the security domain. To overcome these drawbacks, this paper presents the foundations, vision, research plan, and preliminary results of a novel architecture modeling approach that aims to eliminate the necessity of designing separate SAs and support architects in designing and documenting secure software systems. The approach establishes a guided process for tracing security requirements to the modeling elements in the AD. Additionally, it utilizes security properties to make design recommendations in the modeling process and separate the AD's security-related parts from its other parts.**

***Index Terms*—security architecture, security modeling, secure system design, security-by-design, architecture modeling**

## I. Introduction

Nowadays, security is considered one of the most important quality attributes of software systems [1]. Be it the high degree of connectivity of cyber-physical systems [2], the broad attack surface posed by microservice applications [3], or the great value of data for data theft [4]; the reasons for this are manifold. Consequently, the worldwide costs related to security incidents in 2023 are estimated to exceed eight trillion US dollars. They are expected to grow linearly further next years to reach around 14 trillion US dollars in 2028 [5].

This development has ultimately increased awareness of the importance of security in today's software systems [6]—a factor that has long been a problem in software engineering and the fight against cybercrime [7]. As a result, the "security by design" paradigm emerged in recent years as a means of enhancing security and is increasingly being propagated by researchers and practitioners [8]. According to this paradigm, security should be considered from the beginning of the SDLC, such as in the analysis and design of the software system, rather than being added to the software product later on [9]. This includes systematically eliciting and managing security requirements and designing the software system's architecture securely early on.

Due to the sheer size of industrial software systems, managing the complexity of their architectures is a difficult task and has therefore been a subject of research for a long time. Moreover, recent studies in the field of cybersecurity show that managing the increased complexity created by developing secure systems is a major challenge for industrial software engineering [10]. These findings are consistent with the results of our recent study, in which we assessed the trends in the current state of security by conducting interviews with practitioners from the German software industry. In these, too, the security-related complexity was most frequently mentioned as the biggest challenge to cope with in upcoming years [11].

This paper presents the foundation, vision, research plan and preliminary results of the author's doctoral thesis. First, the foundations of this paper are explained in Section II. In Section III, the problem is illustrated, and the proposed solution is motivated. Section IV formulates the research questions. The current state and preliminary results will be presented in Section V. In Section VI, the envisioned industrial impact is discussed. Section VII discusses the related work to establish the state of the art and practice. Finally, the paper is concluded in Section VIII.

## II. Foundations

Our work builds upon the architectural concepts of Rozanski and Woods [12]. We use the term *architectural description* (AD) for the set of design documents that describe and document any state of the architecture of a software system. For the according design documents, we use the term *architectural model* (AM). AMs, in turn, are composed of one or many *architectural modeling elements* (AME). If AMs or parts of its AMEs are intended to model security-related aspects of the system, they model aspects of the *security perspective* of the AD.

To design ADs securely, it is nowadays common practice to develop *security architectures* (SA) alongside the system architecture, which are considered independent, yet not isolated architectures [13]. Consequently, a SA itself is a part of the system architecture. It is noticeable that unlike the distinction we make between architecture and its AD, literature hardly ever distinguishes between an SA and its description.

Published in:
2024 IEEE 21st International Conference on Software Architecture Companion (ICSA-C)
979-8-3503-6625-9/24/$31.00 ©2024 IEEE DOI: 10.1109/ICSA-C63560.2024.00026

SAs usually refer to the representation of security-related parts of the system architecture [14]. Therefore, to better distinguish between the SA as the security part of the system architecture and the design documents that describe it, we follow the terminology of Rozanski and Woods and define the term *security architecture description* (SAD). The SAD refers to the set of AMs that describe security-related information in whole or in part by their AMEs. Hence, the SAD both represents a description of an SA's design and reflects the security perspective of an AD and is, thus, a constituent part of it.

Lastly, we must point out that the term "security architecture" is often associated with enterprise architectures [13]. This association is not entirely accurate and is based on the fact that the term is often used interchangeably with ”enterprise security architecture" [15]. This is even reflected in the NIST glossary, which defines SAs as both "a set of physical and logical security-relevant representations of system architecture" and as "an embedded, integral part of the enterprise architecture" [14]. We stay in line with the first, more general meaning of SAs. Thus, the content addressed and presented in this paper does not limit itself to a certain kind of architecture but instead refers to the secure design and its documentation of any kind of architecture in the software domain.

## III. Problem Statement and Motivation

The common practice of creating SADs independently from the AD is a conceivable approach to cope with the high degree of complexity in designing and documenting software systems. It realizes the concepts of quality perspectives and, thus, avoids the ”big ball of mud" problem in the documentation of a system. However, this method has other problems, especially in modeling and maintaining SADs.

**P1 - Feigned separation of inseparable concepts:** Security is a cross-cutting concern. Thus, information that is modeled in an SAD can be modeled or required, at least to certain parts, in other parts of the AD. There is no clear distinction as to whether an AME solely belongs to the SAD or not [16]. Security requirements are often considered functional in nature [17]. Because of this, security solutions are related to and interact with many other functional and non-functional parts of an architecture [12]. Thus, a clear separation between ADs and SADs gives the impression that security can exist in isolation. However, as explained above, this is only partially the case. This separation adds redundancies in the SAD and other parts of the AD. Moreover, reading the SAD can only satisfy stakeholder concerns to a limited extent, as, in some cases, parts of the AD are required in addition to the SAD to satisfy certain concerns. Therefore, separating the SAD from the AD impedes the maintenance of both due to injected redundancies and hampers their understandability and stakeholder communication.

**P2 - Separation creates coupling:** The separation of these inseparable concepts leads to another well-known problem in software engineering: coupling. While the reason does not lie in the concurrent modification of the same file, such as is the case for version control and collaborative editing tools, it lies in the concurrent modification of information in two or more models representing the same entity: the SAD and the AD [18]. This results in inconsistencies and synchronization errors due to concurrent modifications [19]. Adding, removing or changing AMEs in the SAD requires the modifications to be incorporated in the other parts of the AD, too, and vice versa. This hampers the maintenance of both AD and SAD and increases the communication effort between the responsible stakeholders for reasons of conflict detection and resolution.

**P3 - Security needs experts:** Another problem is the lack of human resources. Designing SAs and documenting SADs requires organizations to employ experienced security experts and security architects. Skills of the latter usually contain solid foundations in security architecture frameworks such as TOGAF, SABSA, or OSA, and expertise in designing and documenting SAs [20]. This is a crucial constraint, as experts in the security domain are rare and it is unlikely that this will change soon [21], [22]. If a project team does not have this expertise, the attempt to create SADs quickly leads to an incomprehensible documentation of security in which the security-related information is distributed across different models in the SAD and AD in an unstructured manner. This, again, impedes both the maintenance and understandability of SADs and ADs.

We conclude that maintaining SADs independently from ADs can have major drawbacks. Especially when the required experts are not available, and the architecture and its description grow due to its evolution, these drawbacks can easily lead to a situation in which the AD loses its actual purpose as a planning, documentation, and communication tool. Considering that the cognitive complexity of complex information should be reduced as much as possible to make it comprehensible [23], this approach seems to address this challenge only one-sidedly. We argue that it rather shifts the cognitive complexity to the highly divided structure of the resulting ADs. The redundancy and coupling added between the SAD and AD counteracts the reduction of cognitive complexity. However, unsystematically integrating the SAD into the AD cannot be considered a better approach, as it leads to a high degree of cognitive complexity within the models.

Hence, in this research project, we seek to develop solutions for integrating the SAD into the AD while keeping the cognitive complexity low. We envision that the solutions we will create will eliminate or at least reduce the disadvantages of maintaining independent SADs and support non-experts in creating them. We expressly use the term "cognitive complexity" to emphasize that the focus of this research project is not to reduce the actual, i.e., "ontological complexity" [24] of a system but to ease designing and understanding its documentation. In the following, we present how we aim to solve these problems.

## IV. Research Plan & Design

To begin with, we subsume the above-mentioned problems under the following central research question:

Published in:
2024 IEEE 21st International Conference on Software Architecture Companion (ICSA-C)
979-8-3503-6625-9/24/$31.00 ©2024 IEEE DOI: 10.1109/ICSA-C63560.2024.00026

*How can an architecture modeling approach look like that supports architects in designing SADs being integrated into ADs while staying maintainable and keeping cognitive complexity in the models low?*

From now on, we will use the term *integrated SAD* to refer to SADs being created by this approach. To answer this central research question, we have broken down it into more granular ones. First, we formulate and explain each research question. Then, we present our solution ideas.

**RQ1: How can a SAD be meaningfully separated from an AD without maintaining it as an independent artifact?** This research question mainly addresses P1 and P2. To develop integrated SADs, we must be able to meaningfully separate the constituent parts that belong to the SAD from the AD. Only if we succeed in this will we be able to develop methods that allow the integrated design of SADs in ADs. Hence, the solutions developed to answer this RQ build the foundation of this research project.

**Solution:** We plan to create a conceptual framework that structures ADs and the elements that comprise them. Thereby, a separation of these elements into security and non-security elements will be conceptualized. By this, when visualizing the SAD, filtering methods can be developed that filter the elements to be visualized based on their existence and role in the SAD, keeping the cognitive complexity in the models low. As we will see in Section V, this can be accomplished by associating AMs and AMEs with security requirements.

**RQ2: How can architects be supported in creating integrated SADs?** This (fairly high-level) research question mainly addresses P3. We envision our results for RQ1 to provide the conceptual foundation to integrate the design of SADs in the design of ADs. However, as explained in Section III, an unsystematic implementation of this method results in messy ADs with a high degree of cognitive complexity, hampering maintenance and understandability. Hence, we will explore methods to support architects in creating SADs. To achieve this, we formulated the following more granular questions.

**RQ2.1: How can architectural concepts be combined with security-related concepts?** This research question is a rather foundational one. We claim that AMEs that model security-related information have specific properties they all have in common. If we can classify and structure these, we argue that solutions could be developed that support architects in designing SADs by recommending, at least to some extent, suitable design solutions that satisfy a security requirement. We assume that these solutions will not be limited to integrated SADs and should, thus, be of high value for designing independent SADs.

**Solution:** A solution requires more conceptual foundations to be developed. We must extend our conceptual framework with meaningful concepts that allow us to merge the architectural concepts described by our framework with the properties of design solutions for security requirements. Architectural security requirements are the interface between both domains, as we will see in Section V.

**RQ2.2: How can a recommender system support architects in designing SADs?** To build upon our results of RQ2.1, we want to develop a recommender system. Its purpose is to give architects suggestions in their modeling activity on designing suitable design solutions in the SAD to fulfill specific security requirements.

**Solution:** We plan to design the recommender system to implement our conceptual framework, as this should provide us with all the concepts required to bridge the gap between security designs and architecture modeling. Moreover, we plan to extend the recommender system by incorporating best practices and security design patterns into its recommendations.

**RQ2.3: How can a systematic architecture modeling process look like that supports architects in creating integrated SADs?** To tackle RQ2 from another angle, we want to develop methods to support architects in designing integrated SADs using a systematic process. Combined with the recommender system from RQ 2.2, which we plan to integrate into this process, we envision that our contributions can support architects in designing SADs, even if they are less experienced in this practice.

**Solution:** Our proposed solution foresees a guided architecture modeling process. In this process, the architect first selects a security requirement for which he or she wants to design a solution in the AD. Then, the recommender system uses the data specified in the security requirement and guides the architect to an appropriate design solution. Besides its supporting feature, this approach reveals an additional advantage: the related parts of the SAD could be automatically linked with the security requirements they fulfill, which allows us to use the concepts we aim to develop in RQ2.1. This ensures that the link between architectural elements and their relation to the security requirements is always included in their metadata without the architect having to add it manually, eliminating the human as a possible source of error.

Figure 1 summarizes our research design. We follow the Design Science Research (DSR) according to Peffers et al. [25] as it offers a systematic process and structure for developing research artifacts.

**Design & Development:** We plan to develop RQ1 and RQ2.1 alongside each other, as their concepts, and thus their development, are related. After we have developed the first

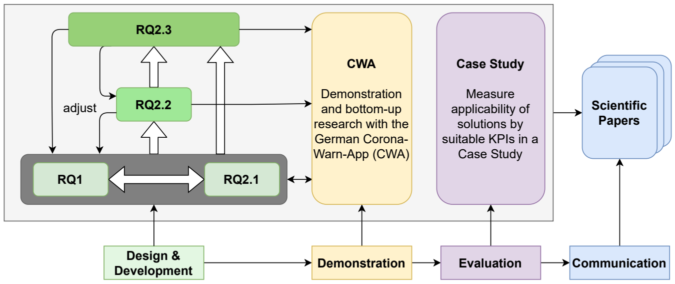
RQ2.3
adjust
RQ2.2
RQ1
RQ2.1
CWA
Demonstration and bottom-up research with the German Corona-Warn-App (CWA)
Case Study
Measure applicability of solutions by suitable KPIs in a Case Study
Scientific Papers
Design & Development
Demonstration
Evaluation
Communication

Fig. 1: Research design following DSR

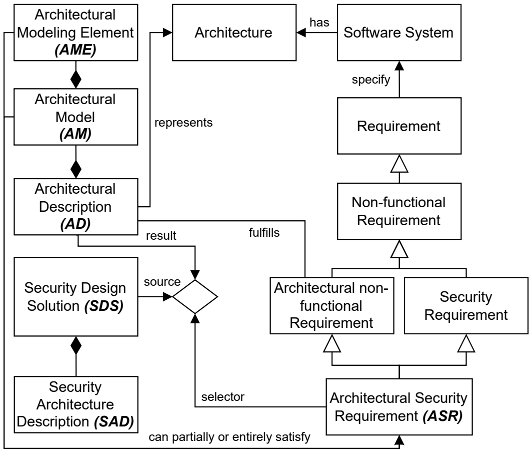
Architectural Modeling Element (AME)
Architecture
has
Software System
specify
Architectural Model (AM)
represents
Requirement
Architectural Description (AD)
Non-functional Requirement
result
fulfills
Security Design Solution (SDS)
source
Architectural non-functional Requirement
Security Requirement
Security Architecture Description (SAD)
selector
Architectural Security Requirement (ASR)
can partially or entirely satisfy

Fig. 2: Architecture concept model (RQ1)

version of the artifacts of RQ1 and RQ2.1, we will begin with developing the research artifact of RQ2.2, the recommender system. In this process, we will adjust the artifacts of RQ1 or RQ2.1 if necessary. Consequently, as we develop the first version of the recommender system, we plan to develop the research artifact of RQ2.3, the guided process model. Again, we will make adjustments to the earlier artifacts if needed.

**Demonstration:** We opted for the open source project of the *Corona-Warn-App (CWA)*[1], the official COVID-19 exposure notification app for Germany, as a means of demonstration. We also use it for the bottom-up development of our research artifacts. We consider it well suited for this research project as it is a sufficiently large, full-fledged, and well-engineered open-source project. Furthermore, the target users raised serious concerns about security and data protection, making the development of a highly secure solution a challenge for its design and development [26].

**Evaluation:** We plan to conduct a case study for evaluation purposes. In this process, we will develop suitable KPIs to make statements about how applicable the process is and how maintainable and understandable the SAD and its models are.

**Communication:** The communication will be done by publishing scientific papers.

## V. Preliminary Results

### A. RQ1: Architecture concept model

We developed the first version of a concept model by adapting the concept models of Rozanski and Woods [12] and Tang et al. [27]. It is depicted in Figure 2. Each software system has an underlying architecture. Its requirements specify it. Some of the requirements are non-functional (NFR), some of which are security requirements and some are architectural NFRs, which require architectural modifications to be satisfied. Security requirements that affect the architecture are *architectural security requirements* (ASR). An AD is composed of one or more AMs, and AMs are composed of one or more AMEs. AMEs and AMs can describe the degree of fulfillment of ASRs.

The model's essence is its ternary relationship between ADs, ASRs, and *security design solutions* (SDS). An SDS is defined as the set of all AMEs and AMs that satisfy an ASR to a certain degree. The idea behind this is: if the set of AMEs and AMs used to model a partial satisfaction of an ASR $R$ fulfills this $R$ as a whole, then this set represents how $R$ is satisfied in the AD. This utility construct allows us to filter the AD for the SDS of $R$, i.e., for all AMEs and AMs that describe how $R$ is satisfied in the AD. Moreover, by filtering an AD for the set of all SDSs of all ASRs, then this resulting set of SDSs describes all design solutions in the AD satisfying ASRs. Consequently, the composition of all SDSs of an AD describes the security-related part of the system. This is consistent to our definition of SADs in Section II.

We were able to demonstrate the feasibility of our concept model in a first application example with the CWA AD. Still, we do not consider the current version to be final, as there are remaining open questions. One central problem is deciding which AMEs are necessary to satisfy an ASR and understand the SDS. Some AMEs might provide semantic context to understand the design even though they are unnecessary to satisfy an ASR. Furthermore, the set of AMEs must be grouped more granularly to preserve semantics.

### B. RQ2.1: Security architecture modeling concept model

In Figure 3, the current version of our concept model for RQ2.1 is depicted. Due to space reasons, we cannot explain it in detail; instead, we present its essence. The ASRs serve as the interface between both concept models. We bridged the gap between the architecture and security domains by adapting the semantics of Soufi [28]. According to the author, (architectural) countermeasures fulfill ASRs and both ASRs and countermeasures mitigate threats. We can use this semantic relation to define security-specific stereotypes for AMEs, such as UML components representing a threat actor and the interface over which the threat actor can intrude the system.

Furthermore, we could define the semantics between security mechanisms and how they affect ASRs. First, security mechanisms often involve data in some way. Secondly, there

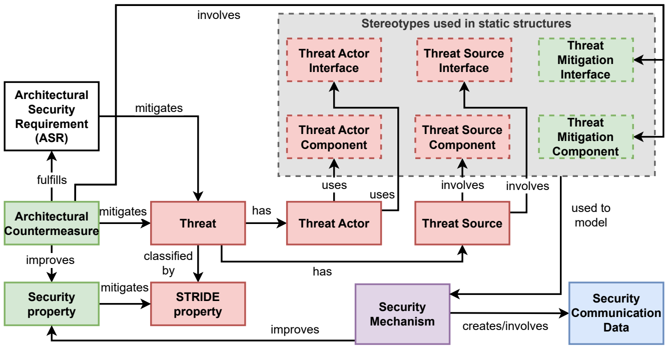

Fig. 3: Security architecture modeling concept model (RQ2.1)

[1] https://github.com/corona-warn-app

Published in:
2024 IEEE 21st International Conference on Software Architecture Companion (ICSA-C)
979-8-3503-6625-9/24/$31.00 ©2024 IEEE DOI: 10.1109/ICSA-C63560.2024.00026

exist various security mechanisms, each of which improves certain security properties. These, in turn, can be defined to mitigate a specific STRIDE property. By this, we could extend our filter concept presented before by a third filter. This filter uses a classification of ASRs, for instance, ASRs representing an authenticity countermeasure, to allow the filtering of the AD by all SDSs of ASRs relating to authenticity. The semantic relation between countermeasures, security properties, and STRIDE properties allows us to explore methods of automated security assessments in the design.

These concepts are in an early stage. However, we were able to replicate the depicted concepts on static models in the CWA AD. Our current aspirations are to refine the security modeling concept model for static models (for instance, Security Communication Data is modeled in static diagrams, too) and extend it by concepts for dynamic models. As these model different semantics, the stereotypes we defined in our model are insufficient to capture all semantics.

## VI. Industrial Impact

Our research has a clear potential industrial impact. Firstly, it supports companies with too little access to security architects to design and systematically document more secure software systems. As pointed out in Section III, the lack of security experts is a recent problem. Thus, many companies and project teams could benefit from this. Additionally, through the guided modeling process combined with the recommender system, architects are expected to make fewer mistakes when designing secure systems. This makes the maintenance of the architecture and its documentation more cost-effective and the system more secure and should also scale with large systems whose maintenance suffers most from their size and complexity. Lastly, tracing design solutions and security requirements—especially through the guided process we propose as it enforces the references to be added—allows architecture audits to become faster and cheaper. This solution allows tools to be developed that register changes to existing design solutions. If AMEs of a design solution, i.e. that already satisfy a security requirement $R$, are changed by new or modified requirements, a tool could register these changes and notify the architect to check again whether the changed AMEs still fulfill $R$.

## VII. Related Work and Tools

We see our research topic to be cross-cutting, i.e., it overlaps with many different areas of secure software design. This makes it hard to present all the related work we identified. To our knowledge, there does not exist a tool or framework that supports and guides architects in the creation of secure software systems by utilizing security-specific properties of architectural designs. Thus, we will briefly present only the most related work of each overlapping field of research that we identified. Furthermore, we do not include research on EA security architectures, as this is a different research area.

Most closely related to our research is work on security architecture and security by design frameworks. Van Opstal [29] proposes a security architecture framework that can be used as a template to create security architectures. However, his solution uses a way broader definition of security architectures. The author defines a security architecture as "the key concept to relate all security activities that need to be performed as part of the development life cycle and as the basis to organize all the associated documentation" [29]. In our work, we focus on the secure architectural design of a system and its systematic and integral documentation. Casola et al. [30] present a novel security-by-design methodology that supports risk management in an almost automated way. Their solution utilizes Service Level Agreements to support the security by design principle and does not offer solutions to create and document more secure system designs. Siavvas et al. [31] and Obaidat et al. [32] present concepts for a security-by-design platform, respectively, a security architecture framework. However, their work addresses specific problems in IoT systems and develops solutions for these. In contrast, our work provides a more generic solution and does not limit itself to software in a specific domain. Furthermore, their work does not focus on designing and documenting secure system design.

Another domain is recommender systems for architectural design. Brandner and Weinreich [33] propose a recommender system for software architecture decision-making. Their approach uses decision models and already captured design decisions to give recommendations for the system design [33]. Some other research is carried out on recommender systems for design patterns (cf. [34]). Several methods are proposed for recommender systems in this domain, such as multi-agent systems [35], Goal-Question-Metrics [36] or ontologies [37]. Our recommender system does not focus on design patterns. In addition, we use a new recommendation approach, as it utilizes security-specific properties to classify security requirements. To our knowledge, this method has not been applied before for architectural design recommender systems.

Lastly, our concepts of requirement traceability are not new. Tools like Enterprise Architect[2] or IBM Doors[3] and Systems Design Rhapsody[4] are well-established tools that allow for tracing requirements to design elements in ADs. However, our solution implements a different modeling approach providing a guided process and a recommender system. The process adds the references automatically at the time of modeling. In both tools, it remains the architect's responsibility to create this reference, which involves the human as a potential source of error. Moreover, our work defines a conceptual framework that other researchers can use and refine. It also seeks to introduce a more lightweight approach to architecture modeling than the tools presented. Lastly, our solution uses security-specific properties to define filtering methods as a flexible approach to extracting information and assessing the SAD from the AD.

## VIII. Conclusion

In this paper, the author's doctoral research project for an integrated modeling approach to security architectures was

[2] https://www.sparxsystems.eu/

[3] https://www.ibm.com/products/requirements-management

[4] https://www.ibm.com/products/systems-design-rhapsody

Published in:
2024 IEEE 21st International Conference on Software Architecture Companion (ICSA-C)
979-8-3503-6625-9/24/$31.00 ©2024 IEEE DOI: 10.1109/ICSA-C63560.2024.00026

proposed. The problem statement of the current practice of developing independent security architectures and the motivation were described. Following, the research questions of this project were posed and the overall research plan following DSR was presented. Our preliminary results show a promising direction that the architecture and security modelling domain can be merged to develop a recommender system and leave much room for future research. Lastly, paper additionally the potential industrial impact of this research and related work was presented. Here it was discussed, that similar solution approaches for modeling secure software systems are missing in the literature.

## References

[1] R. Matulevičius, *Fundamentals of Secure System Modelling*. Springer, 2017.

[2] H. Kayan, M. Nunes, O. Rana, P. Burnap, and C. Perera, “Cybersecurity of industrial cyber-physical systems: A review,” *ACM Computing Surveys*, vol. 54, no. 11s, pp. 1–35, 2022.

[3] P. Billawa, A. Bambhore Tukaram, N. E. Díaz Ferreyra, J.-P. Steghöfer, R. Scandariato, and G. Simhandl, “Sok: Security of microservice applications: A practitioners’ perspective on challenges and best practices,” in *Proceedings of the 17th International Conference on Availability, Reliability and Security*. New York, NY, USA: ACM, 2022, pp. 1–10.

[4] T. Hubbard, G. Weber, and J. Steinhoff, “Protecting data assets in a perilous cyber world,” *The Journal of Government Financial Management*, vol. 66, no. 3, pp. 26–31, 2017.

[5] Statista, “Estimated cost of cybercrime worldwide 2017-2028,” 2023. [Online]. Available: https://www.statista.com/forecasts/1280009/cost-cybercrime-worldwide

[6] Deloitte, “2023 global future of cyber survey: Building long-term value by putting cyber at the heart of the business,” 2023, last accessed: 14.02.2024. [Online]. Available: https://www.deloitte.com/global/en/services/risk-advisory/content/future-of-cyber.html

[7] M. Humayun, M. Niazi, N. Z. Jhanjhi, M. Alshayeb, and S. Mahmood, “Cyber security threats and vulnerabilities: A systematic mapping study,” *Arabian Journal for Science and Engineering*, vol. 45, no. 4, pp. 3171–3189, 2020.

[8] D. B. Johnsson, D. Deogun, and D. Sawano, *Secure by Design*. Manning Publications, 2019.

[9] M. Kreitz, “Security by design in software engineering,” *ACM SIGSOFT Software Engineering Notes*, vol. 44, no. 3, p. 23, 2019.

[10] Deloitte, “2021 future of cyber security,” 2021. [Online]. Available: https://www2.deloitte.com/us/en/pages/advisory/articles/future-of-cyber-survey.html

[11] T. Langstrof and A. R. Sabau, “The current state of security – insights from the german software industry,” 2024, preprint on arXiv.

[12] N. Rozanski and E. Woods, *Software Systems Architecture: Working with Stakeholders using Viewpoints and Perspectives*. Addison-Wesley, 2012.

[13] The Open Group, “Enterprise security architecture,” https://pubs.opengroup.org/togaf-standard/integrating-risk-and-security/integrating-risk-and-security_3.html, The Open Group, Tech. Rep., 2022, last accessed: 15.11.2023.

[14] National Institute of Standards and Technology, “Risk management framework for information systems and organizations,” U.S. Department of Commerce, Washington, D.C., Tech. Rep. Special Publication (SP) 800-37, Revision 2, 2018, 2018.

[15] S. Kim and C. Seong Leem, “Enterprise security architecture in business convergence environments,” *Industrial Management & Data Systems*, vol. 105, no. 7, pp. 919–936, 2005.

[16] D. Mellado, E. Fernández-Medina, and M. Piattini, “Towards security requirements management for software product lines: A security domain requirements engineering process,” *Computer Standards & Interfaces*, vol. 30, no. 6, pp. 361–371, 2008.

[17] M. Riaz, J. King, J. Slankas, and L. Williams, “Hidden in plain sight: Automatically identifying security requirements from natural language artifacts,” in *2014 IEEE 22nd international requirements engineering conference (RE)*. IEEE, 2014, pp. 183–192.

[18] A. R. Sabau, S. Hacks, and A. Steffens, “Implementation of a continuous delivery pipeline for enterprise architecture model evolution,” *Software and Systems Modeling*, vol. 20, pp. 117–145, 2021.

[19] P. Dewan and R. Hegde, “Semi-synchronous conflict detection and resolution in asynchronous software development,” in *ECSCW 2007: Proceedings of the 10th European Conference on Computer-Supported Cooperative Work, Limerick, Ireland, 24-28 September 2007*. Springer, 2007, pp. 159–178.

[20] S. Shrivastava and N. Srivastav, *Solutions Architect’s Handbook: Kick-start your solutions architect career by learning architecture design principles and strategies*. Packt Publishing Ltd, 2020.

[21] S. Furnell and M. Bishop, “Addressing cyber security skills: the spectrum, not the silo,” *Computer fraud & security*, vol. 2020, no. 2, pp. 6–11, 2020.

[22] B. J. Blažič, “Cybersecurity skills in eu: New educational concept for closing the missing workforce gap,” *Cybersecurity threats with new perspectives*, 2021.

[23] J. C. Thomas and J. T. Richards, “Achieving psychological simplicity: Measures and methods to reduce cognitive complexity,” in *Human-Computer Interaction: Design Issues, Solutions, and Applications*. CRC Press, 2009, pp. 179–198.

[24] M. C. Jackson, “How we understand “complexity” makes a difference: Lessons from critical systems thinking and the covid-19 pandemic in the uk,” *Systems*, vol. 8, no. 4, p. 52, 2020.

[25] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, “A design science research methodology for information systems research,” *Journal of Management Information Systems*, vol. 24, no. 3, pp. 45–77, 2007.

[26] W. Lasarov, “Im Spannungsfeld zwischen Sicherheit und Freiheit: Eine Analyse zur Akzeptanz der Corona-Warn-App,” *HMD Praxis der Wirtschaftsinformatik*, vol. 58, no. 2, p. 377, 2021.

[27] A. Tang, P. Liang, V. Clerc, and H. van Vliet, “Traceability in the co-evolution of architectural requirements and design,” 2011.

[28] A. Soufi, “What is a security requirement?” Bachelorthesis, Mälardalen University, Västerås, Sweden, 2021.

[29] T. van Opstal, “A structured approach to software security,” in *ISSE 2009 Securing Electronic Business Processes*, ser. SpringerLink Bücher, N. Pohlmann, H. Reimer, and W. Schneider, Eds. Wiesbaden: Vieweg+Teubner Verlag / GWV Fachverlage GmbH Wiesbaden, 2010, pp. 281–290.

[30] V. Casola, A. de Benedictis, M. Rak, and U. Villano, “A novel security-by-design methodology: Modeling and assessing security by slas with a quantitative approach,” *Journal of Systems and Software*, vol. 163, p. 110537, 2020.

[31] M. Siavvas, E. Gelenbe, D. Tsoukalas, I. Kalouptsoglou, M. Mathioudaki, M. Nakip, D. Kehagias, and D. Tzovaras, “The iotac software security-by-design platform: Concept, challenges, and preliminary overview,” in *2022 18th International Conference on the Design of Reliable Communication Networks (DRCN)*. IEEE, 2022, pp. 1–6.

[32] M. Obaidat, M. Khodiaeva, S. Obeidat, D. Salane, and J. Holst, “Security architecture framework for internet of things (iot),” in *2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, 2019, pp. 0154–0157.

[33] K. Brandner and R. Weinreich, “A recommender system for software architecture decision making,” in *Proceedings of the 13th European Conference on Software Architecture - Volume 2*, L. Duchien, C. Trubiani, R. Scandariato, R. Mirandola, E. M. Navarro Martinez, D. Weyns, A. Koziolek, P. Scandurra, and C. Quinton, Eds. New York, NY, USA: ACM, 2019, pp. 22–25.

[34] M. Z. Asghar, K. A. Alam, and S. Javed, “Software design patterns recommendation : A systematic literature review,” in *2019 International Conference on Frontiers of Information Technology (FIT)*. IEEE, 2019, pp. 167–172.

[35] E. M. Saleh, O. Sallabi, and H. A. Darbi, “A multi-agent system to support design pattern recommendation,” in *Proceedings of the 6th International Conference on Engineering & MIS 2020*, 2020, pp. 1–9.

[36] F. Palma, H. Farzin, Y.-G. Gueheneuc, and N. Moha, “Recommendation system for design patterns in software development: An dpr overview,” in *2012 Third International Workshop on Recommendation Systems for Software Engineering (RSSE)*. IEEE, 2012, pp. 1–5.

[37] M. A. Yasvi and R. Mutharaju, “An ontology design pattern recommendation system,” in *Advances in Pattern-Based Ontology Engineering*. IOS Press, 2021, pp. 258–272.

Published in:
2024 IEEE 21st International Conference on Software Architecture Companion (ICSA-C)
979-8-3503-6625-9/24/$31.00 ©2024 IEEE DOI: 10.1109/ICSA-C63560.2024.00026