A Systematic Mapping Study of Security Concepts for Configurable Data Storages

Richard May
Harz University Wernigerode, Germany
rmay@hs-harz.de

Christian Biermann
Harz University Wernigerode, Germany
cbiermann@hs-harz.de

Jacob Krüger
Ruhr-University Bochum, Germany
jacob.krueger@rub.de

Gunter Saake
Otto-von-Guericke University Magdeburg, Germany
saake@ovgu.de

Thomas Leich
Harz University Wernigerode, Germany
tleich@hs-harz.de

ABSTRACT

Most modern software systems can be configured to fulfill specific customer requirements, adapting their behavior as required. However, such adaptations also increase the need to consider security concerns, for instance, to avoid that unintended feature interactions cause a vulnerability that an attacker can exploit. A particularly interesting aspect in this context are data storages (e.g., databases) used within the system, since the adapted behavior may change how (critical) data is collected, stored, processed, and accessed. Unfortunately, there is no comprehensive overview of the state-of-the-art on security concerns of configurable data storages. To address this gap, we conducted a systematic mapping study in which we analyzed 50 publications from the last decade (2013–2022). We compare these publications based on the configurable systems, data storages, and security concerns involved; using established classification criteria of the respective research fields. Overall, we identified 14 research opportunities, which we discuss in detail. Our key insight is that the security of configurable data storages seems to be under-explored and is rarely considered in a practice-oriented way, for instance, regarding relevant security standards. Furthermore, data storages and their security concerns are usually only mentioned briefly, even though they are either highly configurable or store critical data. Our mapping study aims to help practitioners and researchers to understand the current state-of-the-art research, identify open issues, and guide future research.

CCS CONCEPTS

• **Security and privacy → Vulnerability management**; • **Software and its engineering → Software product lines**.

KEYWORDS

Security, Data Storage, Configurable Systems, Software Product Line Engineering, Mapping Study

SPLC '22, September 12–16, 2022, Graz, Austria

ACM ISBN 978-1-4503-9443-7/22/09…$15.00
https://doi.org/10.1145/3546932.3546994

ACM Reference Format:
Richard May, Christian Biermann, Jacob Krüger, Gunter Saake, and Thomas Leich. 2022. A Systematic Mapping Study of Security Concepts for Configurable Data Storages. In *26th ACM International Systems and Software Product Line Conference - Volume A (SPLC '22), September 12–16, 2022, Graz, Austria.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3546932.3546994

1 INTRODUCTION

In recent years, the amount of data (e.g., recordings, datasets) has massively increased, due to the growing complexity and interconnection of software systems [93, 106]. To handle this data in an appropriate way, it is stored, managed, and processed using a variety of storage systems, ranging from centralized databases over on-demand cloud storages to decentralized blockchains [27, 37, 112]. A growing number of such storages and the software systems they are part of is configurable [76], meaning that such systems comprise variability so that they can be configured to specific requirements, such as customer demands, industry standards, hardware limitations, or legal regulations. So, configurability allows developers to derive a set of similar, but adapted variants of a system by enabling or disabling certain features. While each feature ideally provides functionalities that increase the value of a system, it is also more complex to manage such configurable systems [6, 99].

Typically, the data stored in a software system involves private or security sensitive entries (e.g., personal data, company secrets). To protect this data, it is essential to process and store it in a secure way by relying on security patterns, models, and standards [100, 108]. Due to the uniqueness and complexity of configurable systems, there is a particularly high risk of becoming the subject of cyber attacks or simply revealing sensitive data [1, 64]. Attacks are often based on the exploitation of certain system vulnerabilities or malicious administrators and users, leading to unauthorized data access, data loss, or even data manipulation [20, 45, 63].

Not surprisingly, immense research in the intersection of configurable systems, particularly work based on software product lines (SPLs), and security concerns has been conducted in recent years [46, 64, 78]. There are several works that also investigate configurable data storages, for instance, in cloud robotics applications [43] or secure dynamic SPLs in cloud environments [66]. In this paper, we are particularly interested in such papers, since the

data storage of a configurable system is a major target for cyber attacks and feature interactions may lead to data breaches [11, 40].

Despite extensive research on security and configurable systems, **we are not aware of a systematic overview that focuses on the security of configurable data storages.** Consequently, there is no comprehensive systematization of such storages, their security and variability concerns, or of future research directions. We conducted a systematic mapping study [98], aiming to address this research gap. Precisely, we analyzed 50 papers that are concerned with security and configurable data storages by searching through Scopus,[1] IEEE Xplore,[2] and the ACM Guide to Computing Literature[3] for the last decade (2013–2022). Based on this dataset, we are able to summarize the state-of-art research and emphasize research gaps regarding the intersection of security and configurable storage systems. In detail, we contribute the following:

- A systematic and comprehensive overview of recent research on security in the context of configurable data storages.
- A discussion of what properties have been researched and 14 opportunities for future research.
- An open-access repository including our detailed study results to increase the overall comprehensibility and ensure replicability.[4]

We argue that our contributions help researchers and practitioners in identifying and understanding security concerns of configurable data storages more easily.

2 BACKGROUND

In the following section, we provide relevant background information on *configurable systems*, *data storages*, and *security*.

2.1 Configurable Systems

A configurable system is characterized by a number of features (i.e., user-visible functionalities [6]) that can be configured to meet customer needs, such as user requirements, hardware constraints, or legal regulations. We classify configurable systems into two categories: configurable software systems (e.g., plugin-based systems) and configurable storage systems (e.g., cloud-based systems). Usually, such systems rely on concepts, methods, and techniques related to SPLs [6, 71, 99, 105]. Configurable features in an SPL are managed through variability mechanisms (e.g., variability models, such as feature models) to organize, implement, and document features with their dependencies [6, 28, 62, 91, 109]. Such mechanisms are typically supported by tools that check whether a configuration is valid, propagate configuration decisions [6, 67], and derive valid variants automatically (e.g., FeatureIDE [83], pure::variants [15]). Consequently, configurable systems can be described using the definitions of problem space (i.e., the domain abstraction), solution space (i.e., the implementation), and a mapping between both spaces (i.e., connection between both spaces allowing to derive variants automatically) [6]. Moreover, every configurable system can be verified based on certain attributes. For this purpose, three established strategies exist: feature-based (i.e., analyzing each feature in isolation without considering configurations or dependencies), product-based (i.e., analyzing a system configuration based on its code or an abstraction), and family-based (i.e., analyzing the whole configurable system including valid configurations) [118].

2.2 Data Storages

A data storage, a medium that is able to store specific data in a certain way, is usually classified according to one of three structures it employs, namely centralized, decentralized, or distributed [36, 128]. Centralized storages (e.g., centralized relational SQL databases) are built around one single unit handling all major processing or storage tasks (e.g., one server). Consequently, all machines are connected to the central unit where the data is stored [104]. In contrast, decentralized storages (e.g., decentralized NOSQL databases in a blockchain) rely on the distribution of processing or storage steps among several units with no or limited coordination [9, 128]. So, the dependency on an individual processing unit is much weaker than in centralized solutions [9, 12, 82]. If a decentralized storage provides (close) coordination between independent units, it is called a distributed storage [122, 128], for instance, inter-cloud environments relying on a variety of storages [22, 74].

Data storages usually operate in certain environments that are oriented towards self-hosting (i.e., an environment located on a local machine) or outsourcing. In the context of outsourcing, which has become increasingly widespread for enterprises in recent years, there are several common technologies, such as cloud, edge, and fog computing [23, 29, 37]. Such environments use a complex combination of software and hardware components, and are able to provide server-based storage space at a high level of efficiency, flexibility, scalability, and on-demand availability [80, 115]. Outsourced solutions often serve as underlying technologies based on which data storing and processing functionalities can be implemented, leading to service-based systems or infrastructures; usually called Software as a Service (SaaS) or Infrastructure as a Service (IaaS) [72, 73, 110].

2.3 Security

According to norms and standards established in practice (e.g., ISO/IEC 27000 series [53], ISO/IEC 25010 [52], ISO/IEC 29100 [57], NIST Guide for Conducting Risk Assessments [59]), security is a property of a software system aimed at protecting stored or processed data against a wide range of threats caused by attacks, vulnerabilities, errors (i.e., bugs), or nature (e.g., in the context of hardware). A threat is defined as an unwanted, but possible, event that results in a harm to a system. If there is a concrete possibility of exploiting such a threat, for instance, in terms of vulnerabilities, the threat poses a security risk [53, 59]. To minimize such risks, a regular risk assessment using defined monitoring, measurement, and analysis processes according to the data lifecycle of the individual system is essential [54, 56]. Furthermore, risks are also reduced by fulfilling defined security goals implemented via mitigation techniques, such as cryptographic control mechanisms [53, 54, 59]. According to the ISO/IEC 27000 series [53] and ISO/IEC 25010 [52], three security goals are particularly important, which are also known as the CIA triad [77, 107]:

- *Confidentiality:* The data is only available to authorized entities.
- *Integrity:* The data can only be modified by an authorized entity.

[1] www.scopus.com
[2] ieeexplore.ieee.org
[3] dl.acm.org/search/advanced
[4] https://doi.org/10.5281/zenodo.6802492

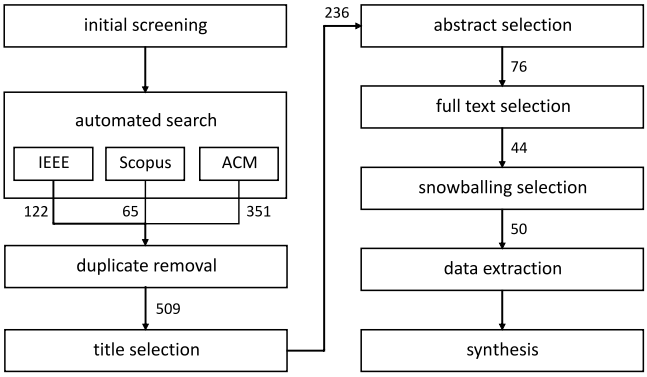

Figure 1: Methodological overview of our systematic mapping study. The numbers indicate the exact amount of papers that we considered relevant after the previous step.

- *Availability:* The system ensures timely and reliable access to its data for authorized entities.

In the context of information security, these three goals are usually extended by [52, 53, 59]:

- *Accountability:* Any action can be traced to a unique entity.
- *Authenticity:* The identity of an entity can be clearly proven to be the one claimed.
- *Non-repudiation:* It is possible to prove the occurrence of every action and what entities were involved.

Information security is often associated with further security goals, such as reliability or privacy [53, 57]. However, such goals are usually subordinated to the six we described, have particular legal dependencies (e.g., legal regulations such as GDPR or HIPAA [120]), or their fulfillment is significantly related to or supported by the fulfillment of the six goals.

3 METHODOLOGY

Our objective for this study was to identify, classify, and discuss research in the intersection of configurable data storages and security. To achieve this objective, we conducted a systematic mapping study based on the guidelines of Petersen et al. [98]. In the following, we describe the individual steps of our study, for which we display an overview in Figure 1.

3.1 Initial Screening

At first, the first author employed an initial screening to ensure the need for our study. Precisely, we aimed to ensure that there are no recent studies addressing our objective, and that a sufficient and relevant number of papers exists. Therefore, we searched the literature databases Scopus, IEEE Xplore, and the ACM Guide to Computing Literature using the following search string without any other constraints (e.g., a certain time frame):

```
"software product line" and "security"
```

Based on this general search string, we obtained around 1,000 papers emphasizing the relevance and research interest for security in the context of configurable systems. Thus, we considered our research objective valuable for conducting a systematic mapping study. Note that we [64] previously conducted a mapping study on safety and security for configurable systems. However, our goals differ from this and other studies referenced therein, since we cover another body of research (cf. Section 6). We focus on the intersection of security and data storages, which has not been the subject of previous mapping studies.

3.2 Study Design

Building on the identified research interest, we decided to conduct a systematic mapping study. Precisely, we focused on security and configurable data storages, also considering potentially underlying configurable software systems that may be the actual focus of a paper. We considered keywords that cover a wide range of storage systems, without any restrictions regarding their architecture (e.g., centralized or decentralized), including solutions ranging from databases to service-based cloud storages. Moreover, we built on the experiences we obtained during the initial screening and considered the previous studies we identified to derive synonyms. So, we defined the following search string:

```
("software as a service" OR "SaaS" OR "infrastructure
as a service" OR "IaaS" OR "service-based" OR
"service-oriented" OR "on-demand" OR "stor*" OR
"cloud" OR "database" OR "blockchain" OR "edge" OR
"fog") AND
("software product line" OR "product famil*"
OR "system famil*" OR "software famil*" OR
"variant*rich" OR "config* system") AND
("security" OR "secure")
```

This string comprises relevant terms in the context of data storages (e.g., "database," "on-demand"), configurable systems in the context of SPLs (e.g., "software product line," "system family"), and security in general (e.g., "secure")—since we assumed that more specific security concerns are rarely reported in the primary search fields.

Using this search string, the first author employed an automated search on Scopus, IEEE Xplore, and the ACM Guide to Computing Literature; searching within the typical fields title, abstract, and keywords. These literature databases ensure a certain quality by indexing only peer-reviewed publications. Moreover, Scopus and the ACM Guide to Computing Literature provide papers from a variety of publishers. This way, we reduced the threat of missing highly relevant publications from other publishers. To further improve the completeness of our dataset, we additionally applied a backwards snowballing (i.e., analyzing the references of the already selected papers) [127]. Note that we performed only one iteration of snowballing to limit the effort of our study.

3.3 Selection Criteria

To select relevant papers, we defined the following criteria:

IC_1 The paper is written in English.
IC_2 The paper has been published between 2013 and 2022.
IC_3 The paper is a peer-reviewed conference paper or journal article (e.g., excluding keynote summaries or posters).
IC_4 The paper has more than three pages.
IC_5 The paper deals with security and configurable data storages in the context of SPLs.

We intentionally focused on the last decade (IC_2) of research to limit the number of papers and cover the most recent advancements in

both research and practice. Note that we did not perform a detailed quality assessment of all selected papers. In fact, since our study was likely to cover a variety of research methods that cannot be properly compared against each other, common quality criteria are not applicable. However, by defining a minimum number of pages (IC_4) and considering only peer-reviewed papers (IC_3), we assume that each paper meets a certain quality that allows us to understand the addressed problem. We intentionally excluded papers that do not address security and data storages in the context of SPLs (IC_5), for instance, papers proposing a configurable SaaS system including a data storage or discussing security concerns, but without building on concepts from product-line engineering.

3.4 Data Extraction

To extract data in a systematic way, the first and second authors defined categories covering the two relevant areas security and configurable data storages. More specifically, we based the extraction categories on related papers, particularly our previous mapping study [64] and the papers referenced therein. Moreover, we defined the categories based on concepts established in research on security as well as product-line engineering. Particularly, the categories regarding security rely on established classifications of security goals (cf. Section 2.3). However, we extended the criteria to gain more detailed insights, especially with respect to the intersection of security and configurable storages. Overall, we defined the following extraction categories classified into four thematic topics:

(1) **Publication**
- *Publication year* of a paper.
- *Domain* of the research (e.g., production, automotive).
- *Perspective* [64] of a paper (indicated by an arrow (→), which reads as "employed to"):
  - *Security → SPL*: the paper focuses on the security of a configurable system.
  - *SPL → security*: the paper focuses on a security concern based on product-line concepts.
  - *Storage → SPL*: the paper focuses on a data storage used to support certain SPL tasks (e.g., storing variants).
  - *SPL → Storage*: the paper focuses on a storage system based on product-line concepts.

(2) **Storage**
- *Type of storage medium* a paper is concerned with (e.g., a relational database in a SaaS cloud environment [26]).
- *Structural organization* of the proposed storage system (i.e., centralized, decentralized, distributed [48, 128]).
- *Stored data* indicating what data is (securely) stored (e.g., feature models [111], source code [75]).

(3) **Security**
- *Standard* a publication refers to (e.g., standards of the ISO/IEC 27000 series, such as ISO/IEC 27001 [54], ISO/IEC 27002 [55]).
- *Security goals* a publications aims to achieve [4, 59, 107]:
  - *CIA triad: confidentiality*, *integrity*, and *availability*.
  - *Information security: authorization*, *accountability*, and *non-repudiation*.
- *Specification*, indicating how certain security goals are considered, documented, or managed (e.g., as quality attribute [97] or non-functional requirements [103]).
- *Security threats* a paper is concerned with (e.g., credential reuse or SQL injection attacks [49]).
- *Mitigation techniques* proposed or implemented to mitigate the impact of cyber attacks and to achieve certain security goals (e.g., encryption techniques, such as AES [125], or technologies, such as Intel SGX [24]).

(4) **Configurable system**
- *Variability focus*, indicating whether the storage itself is configurable or only the surrounding software system.
- *Projection* of an SPL considered in the paper, namely problem space, solution space, or the mapping between both [6].
- *Verification*, indicating whether the described method follows a product-, feature-, or family-based analysis strategy [118].
- *Evolution*, indicating whether storage evolution is considered in the paper [16].
- *Tool support* reported in the paper (e.g., FeatureIDE [83]).

Based on this data, we aim to synthesize a detailed overview on configurable data storages. We remark that when we organized and synthesized the data (cf. Section 3.5), we found that many of the data fields were highly diverse and would obfuscate the presentation in Table 1. For this reason, we provide a more concise overview in the table to focus on our core analysis, and publish the detailed overview of the individual entries for each paper in our dataset.[4]

3.5 Conduct

The first author conducted the automated search on March 15th, 2022, resulting in a total of 538 papers (65 from Scopus, 122 from IEEE Xplore, 351 from the ACM Guide to Computing Literature). Subsequently, the whole selection and extraction process was also performed by the first author. After integrating the papers into the literature review tool Rayyan QCRI,[5] we labeled each paper as include, exclude, or duplicate (i.e., removing 29 duplicates). Then, we employed our selection criteria on titles and abstracts, resulting in 76 papers for the full-text analysis. After reading the full texts, we removed 32 more papers, since they only superficially dealt with configurable data storages or mentioned these aspects only in the context of related work. We employed one iteration of backwards snowballing on the remaining 44 papers. Finally, we identified 50 papers as being relevant.

For the data extraction, we used an Excel spreadsheet and performed an open-coding-like process to identify concrete instances of data fitting to our extraction categories. Afterwards, we employed an open-card-sorting-like methodology. This way, we classified recurring information in the extracted data and synthesized common themes for each category. For this purpose, we derived and adjusted categories, extraction results, and subsequent interpretations of the results (i.e., research opportunities) based on discussions among the authors.

4 RESULTS

In this section, we describe the data we extracted from the 50 selected papers. Note that this section is generally structured based on the four thematic categories (i.e., publication, storage, security, and configurable system) as well as the individual extraction criteria. We provide an overview of our data in Table 1.

[5] www.rayyan.ai

4.1 Publications

First, we describe the results connected to the papers themselves, providing a general overview of our data. Namely, we discuss what domains and aspects of configurable data storages the papers cover.

Publication Years. Most papers in our dataset have been published between 2014 and 2017. In total, 36 out of 50 papers stem from this period, with an average of nine papers each year. Before and after this period, we identified considerably fewer publications (e.g., none in 2022). Between 2018 and 2021, the number of published papers in our dataset is considerably lower compared to the previous period (i.e., 11 papers, around three per year).

Domains. The papers we selected cover a variety of domains. In 28 papers, the domain was explicitly mentioned whereas 22 papers only describe SPL-based research for universal or unspecified domains. Most papers (8) are concerned with the production domain, they usually describe configurable systems or configurations to support certain production systems, machines, or processes, such as cyber-physical systems [7, 124]. Moreover, we identified publications focusing on retail (8 papers, e.g., an ERP system [3]), administration and management services (5 papers, e.g., an eGovernment tool [39]), general web applications (4 papers, e.g., an online survey tool [61]), and payment (4 papers, e.g., an ePayment system [101]). Other domains we found occurred only once, for instance, mobile services [79], medical systems [87], geographic systems [17], tourism systems [14], and general Linux-based systems [95].

Perspectives. Regarding the first perspective category (security and SPL), we identified that most papers (37) are concerned with security for configurable systems (security → SPL), such as configurable systems extended with adaptive Intel Software Guards [68]. Ten publications focus on the application of SPL concepts to implement certain security concerns (SPL → security), such as security configurations of cyber-physical systems [124]. We found three papers for which a distinct classification is not possible, since they consider both perspectives, for instance, a security-policy-driven system for SaaS configurations [5]. Regarding our second perspective category (storage and SPL), 28 papers deal with storages for configurable systems (Storage → SPL), for instance, a configurable eCommerce system with a database [10]. In addition, in 18 papers, the perspective regarding the application of SPL concepts to assure a configurable storage system is described (SPL → Storage), for example, variability models used to model cloud applications [14]. Similar to the first perspective, we found papers considering both perspectives (4), for example, for using SPL concepts to configure robotics applications running on a configurable cloud [43].

4.2 Storages

Second, we describe the results that are concerned with the data storages of a configurable system in more detail, including the type of storage, the structural organization, and the stored data.

Type of Storage. Regarding storage types, we found that authors usually mix storage mediums (e.g., a database) with underlying infrastructure (e.g., an outsourced cloud system). Interestingly, in 17 papers, a database is implemented based on a cloud system. In 13 of these cases, it is explicitly stated that these are considered as SaaS or IaaS environments. A combination of a cloud and a fog system with a database was described once [117]. Overall, in 31 papers a database was implemented with 14 cases providing no details on the underlying infrastructure. In this context, five relational databases are described, and in seven other papers SQL (usually MySQL) is stated to be the database language (i.e., also relational). Moreover, we identified eight papers in which only a cloud system was generally defined as storage system.

Structural Organization. We found all types of structural organization of data storages within our dataset, namely centralized (44), distributed (14), and decentralized (1). For one paper, we could not extract a concrete structure [114]. We remark that it is not always clear whether the described or assigned organization is related to the storage medium (e.g., a database), the storage environment (e.g., a cloud), or the overarching software system. Unfortunately, there is usually insufficient information about the exact focus of the organizational structure in the papers. Cloud systems or service-oriented platforms (i.e., SaaS or IaaS) are usually referred to as large-scale distributed (software) systems by definition [22, 35]. However, this does not necessarily apply to the entire data storage, since such storage environments often consist of databases that are centralized. We argue that only a few papers are actually likely to refer to distributed in terms of the storage (i.e., regardless of the software infrastructure). For instance, one paper explicitly mentions an inter-cloud system based on resources and storage mediums distributed across multiple clouds [74].

Stored Data. The papers in our dataset mention a variety of data to be stored. However, three types of data seem most dominant: 17 papers mention data (strongly) related to the variability of a system, such as concrete features (5), variability models (10), or variants (2). 14 papers refer to general application data (e.g., data of sensors [124]), and 14 other papers to the source code of the (configurable) software (e.g., parts of source code of a robotics system [43]). Other stored data include user data, such as documents (5), application meta data (3), plugins (1), or software patches (1). In five papers, the type of data was unspecified (4) or related to a specific data model (1). Additionally, we assessed whether only the system, or both system and user, can access the stored data. Overall, in 41 papers the system and user can access the data. Accordingly, in nine papers only the system can access the data.

4.3 Security

Third, we describe our results regarding the security concerns of configurable data storages, including standards, specification, security goals, security threats, and proposed mitigation techniques.

Standards. We found five papers that mention or consider security standards or legal regulations. The ISO/IEC 27000 series, precisely ISO/IEC 27000 [53] and ISO/IEC 27001 [54], are mentioned in two papers [34, 41]. The NIST Cloud Computing Security Reference Architecture [44] is also referenced by two papers [34, 65]. Other standards are mentioned only once and are quite diverse. Preuveneers et al. [102] refer to the secure payment standards PCI DSS and PA DSS [19]. Fernandez et al. [33] describe SAML [47] to securely exchange authentication and authorization data. Moreover, Shaaban et al. [111] reference the industry communication-related standard IEC 62443 [50] and IEEE 1686 [51] in the context of their

Table 1: Overview of the data we extracted for each paper based on the categories we defined in Section 3.4.

			Perspectives				Storage				Security goals: CIA triad			Security goals: Information security							Projection			Verification				
Reference	Publication year	Domain	Security → SPL	SPL → Security	Storage → SPL	SPL → Storage	Type of storage	Structural organization	Stored data	Standard	Confidentiality	Integrity	Availability	Authorization	Accountability	Non-repudiation	Specification	Security threats	Mitigation techniques	Variability focus	Problem space	Mapping	Solution space	Feature-based	Product-based	Family-based	Evolution	Tool support
Benlachgar and Belouadha [14]	2013	O	○	●	●	○	C,D	C	○	○	○	○	○	○	○	○	●	○	●	S_1,S_2	○	○	●	○	○	○	○	○
Galster et al. [39]	2013	A	○	●	○	●	D	D_1	●	○	◑	○	○	○	◑	○	●	●	○	S_1	●	○	○	○	○	○	◑	○
Marinho et al. [79]	2013	O	○	●	○	●	C,D	D_1	●	○	○	○	●	●	○	○	◑	○	○	S_1	○	○	●	○	○	○	●	◑
Gherardi et al. [43]	2014	P_1	○	●	◑	◑	C,D	C	●	○	○	○	◑	◑	○	○	◑	●	●	S_1	○	●	●	○	○	○	○	●
Matar et al. [81]	2014	R	○	●	●	○	C,D	C	●	○	○	○	◑	○	○	○	◑	●	○	S_2	○	●	●	○	●	○	●	●
Moens and De Turck [85]	2014	U	○	●	●	○	C	C	●	○	○	○	◑	○	○	○	◑	●	●	S_1	●	●	●	○	◑	○	○	○
Moens et al. [87]	2014	O	○	●	●	○	C	C	●	○	○	○	●	◑	◑	○	●	●	●	S_1	●	●	●	○	◑	○	○	○
Nguyen et al. [92]	2014	W	○	●	○	●	D	C	●	○	○	○	○	●	○	○	◑	●	●	S_1	○	●	●	○	○	○	○	○
Parra et al. [94]	2014	U	○	●	○	●	D	D_1	●	○	○	○	◑	○	○	○	●	○	○	S_1	○	●	●	○	◑	○	○	●
Walraven et al. [126]	2014	U	○	●	○	●	C	C	●	○	○	○	●	◑	○	○	●	●	●	S_1,S_2	●	●	●	○	●	○	●	●
Azzolini et al. [10]	2015	R	○	●	○	●	D	C	●	○	◑	○	○	◑	○	○	◑	●	○	S_1	○	○	●	○	◑	○	●	◑
Baresi and Quinton [13]	2015	P_1	○	●	○	●	C	D_1	●	○	○	○	◑	○	○	○	◑	○	○	S_1	○	○	●	○	○	○	●	○
Brisaboa et al. [17]	2015	O	○	●	○	●	D	C	●	○	◑	○	○	◑	○	◑	◑	●	●	S_1,S_2	○	○	●	○	○	○	◑	○
Fernandez et al. [33]	2015	U	●	○	●	○	C	C	○	●	○	○	◑	●	●	○	●	●	●	S_2	○	●	●	○	○	○	○	○
Fernandez et al. [34]	2015	U	●	○	●	○	C	C	○	●	◑	○	◑	●	◑	○	●	●	●	S_2	○	●	●	○	○	○	○	○
Fernandez and Hamid [32]	2015	U	●	○	●	○	C	C	○	○	○	○	●	●	○	○	●	●	●	S_2	○	●	●	○	○	○	○	○
Galindo et al. [38]	2015	U	◑	◑	○	●	D	D_1	●	○	◑	○	●	◑	○	◑	●	●	●	S_1	○	○	●	◑	○	○	◑	○
Garcia et al. [41]	2015	A	○	●	○	●	D	D_1	●	●	◑	◑	○	○	○	○	◑	○	○	S_1	○	○	●	○	○	◑	●	○
Moens et al. [86]	2015	U	○	●	●	○	C	D_1	●	○	○	○	◑	◑	◑	○	◑	●	●	S_1	●	●	●	○	◑	○	○	●
Tizzei et al. [119]	2015	U	○	●	●	○	C,D	C	●	○	○	○	●	◑	○	○	●	○	●	S_2	○	○	●	○	○	○	○	○
Van Landuyt et al. [121]	2015	U	○	●	●	○	C	D_1	●	○	○	○	◑	○	○	○	◑	●	○	S_1,S_2	●	●	○	○	○	○	●	○
Cao et al. [21]	2015	U	○	●	●	○	D	C	●	○	○	○	◑	◑	○	○	●	●	○	S_1,S_2	○	○	●	○	○	○	○	●
Ali et al. [3]	2016	R	○	●	○	●	C	C	●	○	○	○	○	○	○	○	●	○	○	S_1	●	●	●	○	○	○	○	●
Dig et al. [30]	2016	U	◑	◑	○	●	C	C	●	○	○	○	○	○	○	○	◑	●	○	S_1	●	○	○	○	○	○	●	○
Jumagaliyev et al. [61]	2016	W	○	●	●	○	C,D	C	●	○	○	○	○	○	●	○	●	●	○	S_2	○	○	●	○	○	○	●	●
Leite et al. [74]	2016	U	○	●	●	○	C,D	D_1	●	○	○	○	●	●	○	○	◑	●	●	S_1,S_2	○	○	●	○	○	○	○	○
Metzger et al. [84]	2016	P_1	○	●	●	○	C,D	D_1	●	○	○	○	◑	○	○	○	◑	○	○	S_1	○	●	●	○	○	○	◑	●
Passos et al. [95]	2016	O	○	●	○	●	D	C	●	○	○	●	●	◑	◑	○	●	●	●	S_1	●	●	○	○	◑	○	●	○
Perrouin et al. [96]	2016	W	○	●	○	●	D	C	●	○	○	○	○	●	◑	○	◑	●	○	S_1	●	●	○	○	○	○	◑	○
Preuveneers et al. [101]	2016	P_2	●	○	◑	◑	C,D	D_1	●	○	●	●	●	●	●	●	●	●	●	S_1	○	○	●	○	○	○	○	●
Preuveneers et al. [102]	2016	P_2	●	○	◑	◑	C,D	D_1	●	●	●	●	●	●	◑	●	●	●	●	S_1	○	●	○	○	○	○	○	●
Syed and Fernandez [117]	2016	U	●	○	●	○	F,D	D_1	●	○	○	○	◑	◑	◑	○	●	●	●	S_2	○	○	●	○	○	○	◑	○
Arrieta et al. [7]	2016	P_1	○	●	○	●	C	C	●	○	○	○	○	◑	○	○	●	○	○	S_1	●	●	○	○	○	○	○	●
Alférez and Pelechano [2]	2017	U	○	●	○	●	D	C	●	○	○	○	◑	○	○	○	●	○	○	S_1,S_2	●	●	●	○	●	○	●	●
Jalil and Bakar [58]	2017	P_1	○	●	○	●	C	C	●	○	○	○	◑	◑	○	○	●	●	●	S_1	●	●	●	○	●	○	○	○
Leite et al. [75]	2017	A	○	●	●	○	C,D	C	●	○	○	○	●	●	○	○	◑	●	●	S_1,S_2	○	○	●	○	○	○	◑	○
Mohamed et al. [88]	2017	W	○	●	◑	◑	C,D	C	●	○	○	●	●	◑	○	○	◑	●	●	S_1,S_2	○	●	●	○	○	○	◑	○
Munoz et al. [89]	2017	A	○	●	○	●	C,D	C	●	○	○	○	◑	◑	●	○	●	○	●	S_1	○	○	●	○	○	◑	●	●
Khan et al. [65]	2017	U	○	●	●	○	C,D	C	●	●	○	○	◑	◑	○	○	●	●	●	S_1,S_2	○	●	●	○	○	○	○	◑
Butting et al. [18]	2018	U	○	●	○	●	C	C	●	○	○	◑	○	○	○	○	●	●	●	S_1	●	●	●	○	○	○	◑	○
Krieter et al. [66]	2018	U	○	●	○	●	C	D_1	●	○	●	●	●	◑	○	◑	●	●	●	S_1	●	○	○	○	○	○	○	○
Aouzal et al. [5]	2019	U	◑	◑	●	○	C	C	●	○	◑	●	●	●	◑	○	●	●	●	S_2	●	●	●	○	○	○	●	○
Krieter et al. [68]	2019	U	○	●	○	●	C	C	●	○	◑	●	●	◑	○	○	●	●	●	S_1	○	●	●	○	●	○	○	◑
Shaaban et al. [111]	2019	P_1	●	○	○	●	C	C	●	●	◑	●	○	◑	○	◑	●	●	○	S_1	○	○	●	○	◑	○	○	○
Varela-Vaca et al. [123]	2019	A	●	○	○	●	D	C	●	○	○	◑	●	●	○	○	●	●	●	S_1	●	●	○	○	●	○	●	●
Assunção et al. [8]	2020	R	○	●	○	●	C	D_2	●	○	○	○	◑	◑	○	○	◑	●	○	S_1	●	○	○	○	○	○	◑	○
Siegmund et al. [114]	2020	U	○	●	○	●	C,D	–	○	○	○	○	◑	◑	◑	○	●	○	●	S_1	●	○	○	○	○	○	◑	○
Varela-Vaca et al. [124]	2020	P_1	●	○	○	●	D	C	●	○	●	●	◑	◑	◑	◑	●	●	●	S_1	●	●	○	○	●	○	○	●
Varela-Vaca et al. [125]	2021	P_1	●	○	○	●	D	C	●	●	●	●	●	●	◑	●	◑	●	●	S_1,S_2	○	●	○	○	◑	○	○	●
Zhang et al. [129]	2021	U	○	●	○	●	C,D	C	●	○	○	○	○	◑	●	○	◑	○	○	S_1	●	○	○	○	○	○	●	○

General: ● Completely fulfilled; ◑ Partly fulfilled or subsequently assigned by the authors; ○ Not fulfilled
Domain: A: Administration and Management; P_1: Production; P_2: Payment; R: Retail; W: Web Applications; U: Unspecified SPL; O: Other
Type of storage: C: Cloud; F: Fog; D: Database – **Structural organization:** C: Centralized; D_1: Distributed; D_2: Decentralized – **Variability focus:** S_1: Software; S_2: Storage

configurable production system. Only one publication [34] focuses on the compliance with privacy regulations, precisely the HIPAA.

Specification. In most cases, security is considered as a general, but important, system goal (25). However, this is usually not described in detail, but assigned by us based on the overall context. Seven papers mention security as a non-functional requirement, quality attribute, or a system requirement. Four papers refer to security as a system feature.

Goals. Regarding the security goals described in the papers, most mention availability (38) or authorization (36) to a certain degree, leading to a mix of a CIA triad and information security goal. In 17 papers, accountability, often mentioned in the context of reliability, is addressed. Surprisingly, confidentiality and integrity, which are essential parts of the CIA triad, are only referred to 15 and 13 times, respectively. Non-repudiation is addressed the least (9). Overall, we could not identify any papers that did not describe security goals. However, these goals are usually only mentioned and not described further, except for integrity. Summing up all mentions and concrete descriptions, security goals are named 105 times in a certain context and described or explained in more detail in 57 instances, which is just above half of all their mentions.

Security Threats. We extracted a variety of concerns that could threaten the security of the studied systems. Most papers (23) mention unauthorized data access and general privacy concerns (17). In

addition, the overall variability of configurable systems and their storages (9) as well as the not sufficiently secured communication between clients, clients and software or storage, and software and storage (6) could lead to security issues. Four papers mention trust in the overall configurable system as a security threat, for instance, malicious administrators. Other threats include software bugs and untrusted maintenance (3), data manipulation (2), data theft (2), general vulnerabilities (2), insecure hardware (1), and malware (1).

Mitigation Techniques. Mitigation techniques are usually described in a quite general way without details. Most publications refer to encryption mechanisms or security protocols (26), such as symmetric encryption algorithms like AES [125], or network security protocols like SSH [74], SSL [102], or TLS as successor technology to SSL [124]. In 18 papers, access-control mechanisms are proposed, for instance, through account management [88]. Other mitigation techniques include data isolation (8), firewalls (4), signatures (2), software-misuse patterns (2), security-measure models (1), and parallel variant execution (1). Interestingly, in two papers by Krieter et al. [66, 68], the use of SPL concepts based on a security technology described in detail and called Intel SGX [24] is given, fulfilling all security goals except accountability.

4.4 Configurable Systems

Fourth, we describe our data concerned with the variability of the configurable data storage and its surrounding system, including the variability focus, projection, verification, evolution, and tool support of the studied system.

Variability Focus. Most papers focus only on the variability of the software system itself (31), and refer to (apparently) non-configurable data storages. This observation is further supported by the fact that the data storage only—and not the software system—is configurable in only eight papers. In 11 papers, both software and storage are configurable, showing the close connection of the software system and its data storage.

Projection. We found that the solution space as well as a mapping between problem and solution space are covered by 20 papers, for instance, a configurable robotics system based on a component model [43]. Furthermore, 15 papers address the problem space and the mapping, for example, the formal definition and verification of security configurations for cyber-physical systems [124]. In nine papers, all three types of projection are considered, for instance, modeling policy-driven middleware for SPL-based SaaS application configurations and referring to concrete challenges [5]. We also identified solutions that focus on only one projection, either solution space (16), problem space (6), or the mapping (2). Summing up all papers covering a certain projection, most papers address the solution space (36), while the mapping is covered in 28 papers and the problem space in 21.

Verification. Within our dataset, verification is described or mentioned in 18 papers. Most papers (17) are concerned with the verification of a whole system, for example, a cloud ERP production model [58]. Within one paper, a family-based verification is proposed [41]. In contrast, we could not identify any technique verifying individual features.

Evolution. Regarding the evolution of the configurable data storages and their security concerns, we found 11 papers that mention evolution or consider it partly. For instance, Alférez and Pelechano [2] describe the dynamic evolution of service compositions with the aim to deal with unexpected events of the open world. In another 15 papers, we found more detailed descriptions on the evolution.

Tool Support. Regarding tool support, we extracted data from 22 papers related to five tools. Five papers mention FeatureIDE [83] as a modeling tool. In five more papers, own prototypes or frameworks are described, for instance, CyberSPL for developing an SPL for the verification of security policies according to system configurations [123]. Moreover, other tools and languages are mentioned, including UML for feature modeling (3) and eight tools used only one time, such as pure::variants [86], configuration generator [2], Xfeature [21], or HyperFlex Toolchain [43].

5 DISCUSSION

After describing the data we extracted, we now discuss the core insights we obtained from our analysis. For this purpose, we provide an overview of our synthesis and describe open **research opportunities (ROs)** that we derived from the results through collaborative analyses and discussions among this paper's authors. We argue that these ROs require more intensive research, since they are not sufficiently described in our sample of papers—which we consider representative of our research community, seeing the diverse set of researchers involved as well as domains and research reported.

5.1 Configurable Storages

Interestingly, we found that the concrete understanding of what a data storage actually is varies heavily across the selected papers. Precisely, data storage is either referred to as a medium (e.g., a database), the environment of the medium (e.g., a configurable cloud system), or the cooperation of both; usually with a focus on the environment. Our understanding refers to data storage as a medium embedded into an environment, which has actually no concrete storing abilities without implementing the medium. However, we assume that data storage is usually used as a generic term for systems that are concerned with certain storage goals or functionalities, leading to our argument **(RO_1)**: *A uniform definition and understanding of data storages, especially in the area of configurable systems, is needed* to ensure comparable analyses and research on data storages and the data stored (e.g., in the context of security).

Surprisingly, a uniform assignment of each storage's structural organization (i.e., centralized, decentralized, distributed) was hardly possible, since the concrete perspective of a paper was often unclear. We assume that there exist several possible layers (i.e., perspectives) regarding the structure of a system with a data storage, such as:
(1) the storage medium (e.g., a database),
(2) the storage environment (e.g., a cloud system),
(3) the software system (e.g. a medical system), and
(4) the storage hardware (e.g., the server system infrastructure).
Each layer may contain systems that are either centralized, decentralized, or distributed. Since **(RO_2)** *uniform definitions regarding the perspectives of the structural organization are not defined* in any of the papers, comparable assignments are not or hardly possible at the moment. However, we assume that the structural organization

described in the papers usually refers to the storage environment in combination with the software system (e.g., distributed, multi-tenant cloud systems), where the actual storage medium is usually centralized. We strongly recommend to initiate further research in the context of the structural organization of configurable data storages, for instance, defining and formalizing the different perspectives in terms of a framework.

Our results further highlight that usually only the software system, or both the software system and data storage combined, are actually variable. This insight indicates the dependency and interaction of storages and software systems in the context of variability, namely that both should be considered together if possible. Interestingly, we could not identify research concerned with how variability of the system impacts the data in the storage (e.g., configuring the data itself). In addition, the actual configurability of the storage medium, at least the storage environment that is configurable and scalable by definition (e.g., clouds), is rarely addressed. This may be due to an insufficient relevance of these systems in the field of SPLs or due to a lack of concrete solutions and research in this area. Nevertheless, we argue that **(RO$_3$)** *more research is needed regarding the variability of storage mediums and environments, taking into account their interactions with configurable software systems.*

5.2 Security of Data

To achieve the security goals we defined, security-related concerns must be studied more extensively in the context of configurable storages, but this research is lagging behind the technological advances. This is a well-known problem in the context of data storages [116]. Particularly, this need arises with respect to norms, standards, and legal regulations. Unsurprisingly, there is usually no security standard addressed in the papers, although there exists a variety of international standards published by NIST, ISO, or IEC. Only in two papers, the most relevant standards of the ISO/IEC 27000 series are described to provide essential definitions (ISO/IEC 27000 [53]), requirements (ISO/IEC 27001 [54]), or ways of monitoring, measuring, analyzing, and evaluating security concerns (ISO/IEC 27004 [56]). Moreover, relevant legal regulations (usually in the context of privacy for personal and medical data) are only described in one paper. So, we argue that **(RO$_4$)** *security for configurable data storages is lacking practical orientation based on established norms, standards, and legal regulations*, leading to a major barrier regarding the transfer of theoretical knowledge into practice. Possible reasons for not referring to established standards could be the missing need for a practical standard in the context of purely theoretical considerations or the assumption that the configurable systems automatically rely on the ISO/IEC 27000 series; since it is one of the most established security standards. Also, the limited number of relevant systems and variability mechanisms in practice (e.g., configurable robotics cloud applications in Industry 4.0 [43]) may result in an insufficient interest in researching how to protect these systems against potential cyber attacks based on established standards.

Our findings further emphasize that **(RO$_5$)** *there is a lack of concrete specifications of mitigation techniques* (e.g., concrete encryption algorithms, such as AES-256 or RSA-2048). For instance, in the context of encryption, it is not clear what is encrypted (e.g., data, storage medium, or communication) and how it is encrypted (e.g., using 1024 or 2048 bit in the context of asymmetric RSA encryption). We argue that we need definitions of different protection levels and assigned security measures, depending on which data, medium, environment, or system (including variants) need to be protected. For example, there is a different need for protecting personal data, such as the birth date or medical data of a person. However, in the analyzed papers, security measures are often described in a general way (e.g., firewalls [117] or access control [86]) resulting in no or only limited practical relevance in the context of actual cyber attacks (e.g., man-in-the-middle or brute force attacks). Especially in the case of configurable systems, their data storages, and the communication between both, a specification that is as precise as possible is essential, due to the increased attack surface caused by the variability [64]. We suggest to not only specify mitigation techniques, but to also assign them to concrete threats or risks (i.e., cyber attacks) to increase the practical relevance of research.

Interestingly, similar to the data storages, the perspectives on the security measures differ in the analyzed papers. In detail, some measures refer to the overall software system (e.g., firewall [114]), while others focus on the storage environment (e.g., access control of cloud users [86]) or the storage medium (e.g., encryption of a database and its communication [102]). However, it is not always clear what exactly a security measure refers to, although there exist categorizations of security controls, for instance, by NIST [60]. Thus, we argue that **(RO$_6$)** *a concrete assignment of mitigation techniques to their target area (i.e., their perspective) is usually missing*, leading to a decreasing comparability of existing research.

In contrast to other research on security and configurable systems [64], the security goals of the CIA triad are not the ones addressed most often (cf. Figure 2) within the papers we selected. Instead, availability and authorization are typically mentioned. We argue that this is the result of these two security goals being more closely related to data storages. Furthermore, we assume that some security goals that are relevant, but have not been mentioned, are considered as given (e.g., integrity and non-repudiation). In the context of databases, both can be assumed to be automatically fulfilled by the general properties of a database. However, we emphasize that **(RO$_7$)** *concrete security goals (e.g., according to the definitions of ISO/IEC 27000 [53]) are usually missing in the context of configurable storages*, implying a lack of practical orientation.

5.3 Configurable Systems and Storages

We identified a variety of domains, implying that most of the selected papers are not concerned with domain specifics. Thus, most of the analyzed research should be transferable and applicable to a large number of use cases. However, we note that this may be a hurdle for actually implementing the corresponding techniques in practice, since these solutions are not yet specified for a concrete domain or use case. We identified a trend towards the production domain, more specifically cyber-physical systems and robotics applications. So, we see a high application potential for security research on configurable data storages in cyber-physical, cloud, robotics, and Industry 4.0 systems—which share similar properties, involve large amounts of data, and have high security concerns [31]. Similarly, considering the projections used in the papers, most focus on the mapping between problem space and solution space or

the problem space only, meaning that actual (implemented) solutions for secure configurable storages are missing. Consequently, we argue that there is a need to **(RO$_8$)** *provide sufficiently concrete solutions that can serve as working examples for practice in concrete domains and improve collaborations with practitioners.* Furthermore, we could hardly find any papers published with industrial partners (e.g., in the production domain). However, these would significantly increase the practical relevance of the configurable storages, especially regarding domain-specific requirements.

The configurable systems we examined are mostly not verified or only verified in the context of an individual product, which is why we argue that **(RO$_9$)** *we are lacking feature-based and family-based security verifications.* We argue that it would be valuable to verify configurable storages also in these ways, especially to fulfill domain-specific requirements and security demands in a verified manner. Addressing this opportunity could increase the confidence in configurable systems and their data storages, also improving the trust in the behavior of feature interactions. Moreover, costs could be reduced by avoiding potential system adaptions [64, 69], for instance, in terms of updates that could also lead to new security risks.

Surprisingly, about half of the publications are concerned with the evolution of the systems, which is more than in previous studies [64]. This may be due to the papers' focus on cloud systems, which are possibly subject to more evolution-related modifications, due to their high scalability according to the systems' requirements. Furthermore, we found that **(RO$_{10}$)** *tools to support the secure engineering and evolution of data storages are usually missing.* The combination of these two issues results in the problem that papers serving an evolving solution space (and are thus most predestined for practice) lack a clear understanding of how existing tools could support practitioners. Thus, we would recommend to at least name the tools used (e.g., FeatureIDE [83]) and report on the experiences of evolving configurable systems and their data storages (e.g., how the security of data is ensured despite evolutionary changes).

5.4 Configurable Data Storages and Security

Not surprisingly, most of the papers we analyzed consider security as a non-functional requirement, a quality attribute, or an overall system goal. This leads us to the issue that **(RO$_{11}$)** *security is usually only addressed from a high-level perspective*, resulting in a lack of concrete security measures. However, in four papers, security is addressed as one relevant feature of the configurable system and/or its data storage. We argue that considering security as a feature could help to create more awareness and add more attention to security, namely relevant threats, risks, and mitigation techniques. Consequently, security would no longer be one of many goals, but part of the overall system, which is configurable. Security measures resulting in concrete features (e.g., access-control mechanisms based on strong encryption algorithms to avoid brute force attacks) should be modeled nearly equivalent to other features.

We found that the data storages described in the papers are usually not explained in detail, for instance, whether they are SQL or NoSQL databases or how configurable they actually are. So, we argue that every type or even variant of a data storage can possibly lead to new requirements that are relevant for systems, models, or users interacting with them. For example, securing the

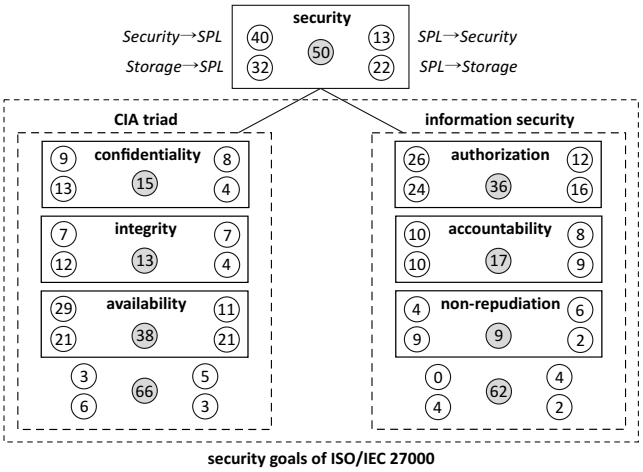

Figure 2: Overview of the security goals and perspectives of the selected papers. The numbers indicate the corresponding papers, with those in the middle representing the overall amount (which can be more than 100 %, since papers can involve multiple perspectives and goals).

underlying software system against SQL injection attacks may depend on the configurability of the data storage [49]. Moreover, it is not clear what impact the type of data storage (e.g., relational database) together with certain variability aspects (e.g., version updates) has regarding potential security risks. Specifically, we found only papers reporting on or recommending general data storages without considering variability, for instance, overviews of concrete security challenges for a data storage [25]. This is why we argue that **(RO$_{12}$)** *more research is needed in the context of variable data storages (i.e., databases) to understand relevant dependencies and minimize security risks caused by a storage's variability.*

Interestingly, some papers stated variability as one issue that could compromise the security of a configurable system or its storage. However, although this threat was recognized, only in one paper a concrete mitigation technique was proposed, precisely the parallel execution of variants [92]. Nevertheless, this issue shows that **(RO$_{13}$)** *the treatment of security threats or risks caused by variability and their relationships requires more research.*

In Figure 2, we can see that most of the papers we analyzed do not focus on configurable data storages themselves. Instead, the papers are more concerned with configurable systems that also include a data storage (e.g., to store user data or variability-related data). Surprisingly, the number of papers in our sample that considers security or privacy goals is decreasing since 2018. We argue that **(RO$_{14}$)** *the research area of configurable systems, especially configurable data storages, and security is under-explored*, since data storages are usually interpreted as a part of a system rather than an individual system with own requirements. Consequently, security is often related to the overall system, but researchers should also focus on the storages that store and provide critical data.

5.5 Threats to Validity

We are aware of some threats that could impair the internal and external validity of our mapping study. First, the papers' authors

usually do not share the same understanding of certain terms and definitions. This threat is particularly relevant in the context of security goals (e.g., distinguishing between authorization and authentication [53]) and what is actually understood as a data storage (e.g., a database as a medium or a cloud system as an environment). Second, we are missing in-depth details regarding the content and consistency of the analyzed papers. In detail, some authors describe their work in great detail, while others only mention some essential properties (e.g., concrete encryption algorithms) or describe them briefly. However, this is probably due to the fact that we considered both conference papers and journal articles in our study. Although we aimed to mitigate such issues in our data analysis, we cannot ensure that this did not impair the comparability or led to misinterpretations on our side (e.g., in the context of assigning categories). Third, we found a few issues regarding the fulfillment of certain criteria, such as several papers not providing any descriptions or names of tools used—leading to a decreased comprehensibility and replicability of the paper. Fourth, besides these threats to the internal validity, the external validity of our study could be impaired by the number of publications we considered (50). Although we searched in three databases, we are aware that the smaller the number of included papers, the higher the potential impact of erroneous classifications. We assume that we missed papers that are relevant for our study, due to our overall search strategy, which does not cover all potential subtopics (e.g., variability modeling in the automated search).

Although such issues threaten our findings, we aimed to mitigate them by relying on established literature databases covering relevant and peer-reviewed papers, as well as snowballing to avoid technical problems [70, 113]. We considered a large number of papers during our systematic analysis (i.e., 538 initial papers), leading to a decreased threat of missing papers that would change our analysis. Additionally, the insights we identified are similar across all papers we studied, increasing our confidence that we did not miss important papers. Thus, we argue that our study provides detailed, reliable, and highly valuable insights regarding security in the context of configurable data storages.

6 RELATED WORK

We found six related papers contributing literature reviews or mapping studies in the field of configurable systems also involving security or data storage. However, security is usually only extracted or mentioned as one of several quality attributes or non-functional requirements from a high-level perspective, without a detailed analysis of security concerns (e.g., threats, attack mitigation techniques). Moreover, data storages are typically not addressed in the context of configurable systems and security. In contrast to the related work we found, our study provides a comprehensive and systematic overview of security in the context of configurable data storages, providing a detailed analysis of these aspects.

Myllärniemi et al. [90] conducted a literature review of 29 SPL-related papers (2000–2010) focusing on variability as a quality attribute, where security is considered from a high-level perspective. Benlachgar and Belouadha [14] reviewed four SPL models for SaaS applications (without any time restriction), including an assessment of their relevance. Security aspects are not considered. Mahdavi-Hezavehi et al. [78] report a literature review of 46 papers (2000–2011) focusing on the variability of service-based software, including general security goals. The authors state that only a few papers actually consider security in a general way as a quality attribute. Hammani [46] surveyed non-functional requirements from nine papers in the context of modeling and verifying SPLs (without any time restriction). However, they only provide a high-level overview regarding security without focusing on details of the security concerns or data storages. Geraldi et al. [42] reviewed 56 SPL-related papers (2006–2018) that describe concepts or applications related to the Internet of Things. As cloud systems are closely associated with the Internet of Things, they are considered as part of the study, but not examined in detail. Security is only generally considered from a high level in the context of non-functional requirements. We [64] presented a systematic mapping study of 65 papers (2011–2020) with a focus on safety and security for configurable software systems, which is closely related to this paper. We complement the previous mapping study by providing a detailed analysis of security research on configurable data storages, which we did not investigate before.

7 CONCLUSION

In this paper, we reported a systematic mapping study of security in the context of configurable data storages. Precisely, we reviewed 50 papers (2013–2022) from a variety of domains. We provided key insights and 14 opportunities for future research. Particularly, we emphasize that, despite the high relevance of security for configurable systems, little research has been concerned with configurable data storages—which store, manage, and provide access to potentially critical data of customers, the organization, or the system itself. Generally, we are missing a detailed understanding of how the current research is related to established security mechanisms and established standards in practice. As a result, the transfer of (theoretical) concepts into practice is impaired. Furthermore, there is no uniform understanding of data storages, which challenges comparisons between papers. This issue can only be solved in the future through a uniform understanding of relevant terms and technological layers, as well as by consistently addressing security goals within configurable systems as features of both the system and the data storage.

In future work, we aim to expand on this study to improve our understanding of the concepts, processes, and relationships of configurable systems and data storages. One objective is to assess the impact of the binding time, for instance, analyzing security-related differences of configurability at design time and runtime. Moreover, we plan to analyze and compare the technological structures' and the different layers' impact on "regular" and configurable storages to develop techniques and security patterns supporting the engineering and evolution of configurable storages.

REFERENCES

[1] M. Acher, G. Bécan, B. Combemale, B. Baudry, and J.-M. Jézéquel. 2015. Product Lines Can Jeopardize Their Trade Secrets. In *ESEC/FSE*. ACM.
[2] G. H. Alférez and V. Pelechano. 2017. Achieving Autonomic Web Service Compositions with Models at Runtime. *Computers & Electrical Engineering* 63, 1 (2017).
[3] M. Ali, E. S. Nasr, and M. H. Gheith. 2016. A Requirements Elicitation Approach for Cloud Based Software Product Line ERPs. *AMECSE*.
[4] J. M. Anderson. 2003. Why We Need a New Definition of Information Security. *Computers & Security* 22, 4 (2003).

[5] K. Aouzal, H. Hafiddi, and M. Dahchour. 2019. Policy-Driven Middleware for Multi-Tenant SaaS Services Configuration. *International Journal of Cloud Applications and Computing* 9, 4 (2019).

[6] S. Apel, D. Batory, C. Kästner, and G. Saake. 2013. *Feature-Oriented Software Product Lines*. Springer.

[7] A. Arrieta, G. Sagardui, and L. Etxeberria. 2015. Cyber-Physical Systems Product Lines: Variability Analysis and Challenges. *Jornadas de Computación Empotrada* (2015).

[8] W. K. G. Assunção, J. Krüger, and W. D. F. Mendonça. 2020. Variability Management Meets Microservices: Six Challenges of Re-Engineering Microservice-Based Webshops. In *SPLC*. Springer.

[9] G. Ayoade, V. Karande, L. Khan, and K. Hamlen. 2018. Decentralized IoT Data Management Using BlockChain and Trusted Execution Environment. In *IRI*. IEEE.

[10] R. P. Azzolini, C. M. F. Rubira, L. P. Tizzei, F. N. Gaia, and L. Montecchi. 2015. Evolving a Software Products Line for E-Commerce Systems: A Case Study. In *ECSA*. ACM.

[11] A. Bamrara. 2015. Evaluating Database Security and Cyber Attacks: A Relational Approach. *The Journal of Internet Banking and Commerce* 20, 2 (2015).

[12] Z. Bao, Q. Wang, W. Shi, L. Wang, H. Lei, and B. Chen. 2020. When Blockchain Meets SGX: An Overview, Challenges, and Open Issues. *IEEE Access* 8, 1 (2020).

[13] L. Baresi and C. Quinton. 2015. Dynamically Evolving the Structural Variability of Dynamic Software Product Lines. In *SEAMS*. IEEE.

[14] A. Benlachgar and F.-Z. Belouadha. 2013. Review of Software Product Line Models Used to Model Cloud Applications. In *AICCSA*. IEEE.

[15] D. Beuche. 2012. Modeling and Building Software Product Lines with Pure::Variants. In *SPLC*. ACM.

[16] J. Bosch. 2002. Maturity and Evolution in Software Product Lines: Approaches, Artefacts and Organization. In *SPLC*. Springer.

[17] N. R. Brisaboa, A. Cortiñas, M. R. Luaces, and M. Pol'la. 2015. A Reusable Software Architecture for Geographic Information Systems Based on Software Product Line Engineering. In *Model and Data Engineering*. Springer.

[18] A. Butting, R. Eikermann, O. Kautz, B. Rumpe, and A. Wortmann. 2018. Controlled and Extensible Variability of Concrete and Abstract Syntax with Independent Language Features. In *VaMoS*. ACM.

[19] A. Calder and N. Carter. 2011. *PCI DSS: A Pocket Guide*. IT Governance Publishing.

[20] F. Campanile, L. Coppolino, S. D'Antonio, L. Lev, G. Mazzeo, L. Romano, L. Sgaglione, and F. Tessitore. 2017. Cloudifying Critical Applications: A Use Case from the Power Grid Domain. In *PDP*. IEEE.

[21] Y. Cao, C.-H. Lung, and S. A. Ajila. 2015. Constraint-Based Multi-Tenant SaaS Deployment using Feature Modeling and XML Filtering Techniques. In *COMPSAC*, Vol. 3. IEEE.

[22] A. Celesti, F. Tusa, M. Villari, and A. Puliafito. 2010. Security and Cloud Computing: Intercloud Identity Management Infrastructure. In *WETICE*. IEEE.

[23] C. Correia. 2020. Safeguarding Data Consistency at the Edge. In *DSN-S*. IEEE.

[24] V. Costan and S. Devadas. 2016. Intel SGX Explained. *IACR Cryptology ePrint Archive* (2016).

[25] CSA. 2012. *Top Ten Big Data Security and Privacy Challenges*. Technical Report.

[26] C. Curino, E. P. Jones, R. A. Popa, N. Malviya, E. Wu, S. Madden, H. Balakrishnan, and N. Zeldovich. 2011. Relational Cloud: A Database-as-a-Service for the Cloud. In *CIDR*. Pacific Grove.

[27] M. Cusumano. 2010. Cloud Computing and SaaS as New Computing Platforms. *Communications of the ACM* 53, 4 (2010).

[28] K. Czarnecki, P. Grünbacher, R. Rabiser, K. Schmid, and A. Wąsowski. 2012. Cool Features and Tough Decisions: A Comparison of Variability Modeling Approaches. In *VaMoS*. ACM.

[29] M. De Donno, K. Tange, and N. Dragoni. 2019. Foundations and Evolution of Modern Computing Paradigms: Cloud, IoT, Edge, and Fog. *IEEE Access* 7 (2019).

[30] D. Dig, R. Johnson, D. Marinov, B. Bailey, and D. Batory. 2016. COPE: Vision for a Change-Oriented Programming Environment. In *ICSE*. ACM.

[31] B. C. Ervural and B. Ervural. 2018. Overview of Cyber Security in the Industry 4.0 Era. In *Industry 4.0: Managing the Digital Transformation*. Springer.

[32] E. B. Fernandez and B. Hamid. 2015. A Pattern for Network Functions Virtualization. In *EuroPLoP*. ACM.

[33] E. B. Fernandez, N. Yoshioka, and H. Washizaki. 2015. Cloud Access Security Broker (CASB): A Pattern for Secure Access to Cloud Services. In *AsianPLoP*, Vol. 15. ACM.

[34] E. B. Fernandez, N. Yoshioka, and H. Washizaki. 2015. Patterns for Security and Privacy in Cloud Ecosystems. In *ESPRE*. IEEE.

[35] I. Foster, Y. Zhao, I. Raicu, and S. Lu. 2008. Cloud Computing and Grid Computing 360-Degree Compared. In *GCE*. IEEE.

[36] B. Furht and A. Escalante. 2010. Cloud Computation Fundamentals. In *Handbook of Cloud Computing*. Springer.

[37] M. Gabel and J. Mechler. 2017. Secure Database Outsourcing to the Cloud: Side-Channels, Counter-Measures and Trusted Execution. In *CBMS*. IEEE.

[38] J. Á. Galindo, D. Dhungana, R. Rabiser, D. F. Benavides Cuevas, G. Botterweck, and P. Grünbacher. 2015. Supporting Distributed Product Configuration by Integrating Heterogeneous Variability Modeling Approaches. *Information and Software Technology* 62, 6 (2015).

[39] M. Galster, P. Avgeriou, and D. Tofan. 2013. Constraints for the Design of Variability-Intensive Service-Oriented Reference Architectures – An Industrial Case Study. *Information and Software Technology* 55, 2 (2013).

[40] A. M. Gamundani and L. M. Nekare. 2018. A Review of New Trends in Cyber Attacks: A Zoom into Distributed Database Systems. In *IST-Africa*. IEEE, 1–9.

[41] C. Garcia, M. Paludo, A. Malucelli, and S. Reinehr. 2015. A Software Process Line for Service-Oriented Applications. In *SAC*. ACM.

[42] R. T. Geraldi, S. Reinehr, and A. Malucelli. 2020. Software Product Line Applied to the Internet of Things: A Systematic Literature Review. *Information and Software Technology* 124 (2020).

[43] L. Gherardi, D. Hunziker, and G. Mohanarajah. 2014. A Software Product Line Approach for Configuring Cloud Robotics Applications. In *CLOUD*. IEEE.

[44] NIST Cloud Computing Security Working Group. 2013. *NIST Cloud Computing Security Reference Architecture*. Standard. NIST.

[45] H. S. Gunawi, V. Martin, A. D. Satria, M. Hao, T. Leesatapornwongsa, T. Patana-anake, T. Do, J. Adityatama, K. J. Eliazar, A. Laksono, and J. F. Lukman. 2014. What Bugs Live in the Cloud?. In *SOCC*. ACM.

[46] F. Z. Hammani. 2014. Survey of Non-Functional Requirements Modeling and Verification of Software Product Lines. In *RCIS*. IEEE.

[47] J. Hughes and E. Maler. 2005. Security Assertion Markup Language (SAML) v2.0 Technical Overview. *OASIS SSTC Working Draft* (2005).

[48] M. Hugoson. 2007. Centralized Versus Decentralized Information Systems. In *HiNC*. Springer.

[49] M. Humayun, M. Niazi, N. Z. Jhanjhi, M. Alshayeb, and S. Mahmood. 2020. Cyber Security Threats and Vulnerabilities: A Systematic Mapping Study. *Arabian Journal for Science and Engineering* 45, 4 (2020).

[50] IEC 62443 2020. *Security for Industrial Automation and Control Systems*. Standard. IEC.

[51] IEEE 1686 2013. *Standard for Intelligent Electronic Devices Cyber Security Capabilities*. Standard. IEEE.

[52] ISO/IEC 25010:2011-03 2011. *Systems and Software Engineering – SQuaRE - System and Software Quality*. Standard. ISO.

[53] ISO/IEC 27000:2018 2018. *Information Technology – Security Techniques – Information Security Management Systems*. Standard. ISO.

[54] ISO/IEC 27001:2013 2013. *Information Security Management Systems – Requirements*. Standard. ISO.

[55] ISO/IEC 27002:2013 2013. *Information Technology – Security Techniques – Information Security Management Systems – Code of Practice for Information Security Management*. Standard. ISO.

[56] ISO/IEC 27004:2016 2016. *Information Technology – Security Techniques – Information Security Management – Monitoring, Measurement, Analysis and Evaluation*. Standard. ISO.

[57] ISO/IEC 29100:2011 2011. *Information Technology – Security Techniques – Privacy Framework*. Standard. ISO.

[58] D. Jalil and M. S. A. Bakar. 2017. Adapting Software Factory Approach into Cloud ERP Production Model. *International Journal of Computer Science and Information Security* 15, 1 (2017).

[59] Joint Task Force Transformation Initiative. 2012. *Guide for Conducting Risk Assessments*. Technical Report NIST SP 800-30r1. National Institute of Standards and Technology.

[60] Joint Task Force Transformation Initiative. 2020. *Security and Privacy Controls for Information Systems and Organizations*. Technical Report NIST SP 800-53r5. National Institute of Standards and Technology.

[61] A. Jumagaliyev, J. Whittle, and Y. Elkhatib. 2016. Evolving Multi-Tenant SaaS Cloud Applications using Model-Driven Engineering. *MODELS* (2016).

[62] K. C. Kang, S. G. Cohen, J. A. Hess, W. E. Novak, and A. S. Peterson. 1990. *FODA Feasibility Study*. Technical Report CMU/SEI-90-TR-21. Carnegie Mellon University.

[63] A. Kenner, S. Dassow, C. Lausberger, J. Krüger, and T. Leich. 2020. Using Variability Modeling to Support Security Evaluations: Virtualizing the Right Attack Scenarios. In *VaMoS*. ACM.

[64] A. Kenner, R. May, J. Krüger, G. Saake, and T. Leich. 2021. Safety, Security, and Configurable Software Systems: A Systematic Mapping Study. In *SPLC*. ACM.

[65] A. Khan, J. Hintsch, G. Saake, and K. Turowski. 2017. Variability Management in Infrastructure as a Service: Scenarios in Cloud Deployment Models. In *ICNC*. IEEE.

[66] S. Krieter, J. Krüger, N. Weichbrodt, V. Sartakov, R. Kapitza, and T. Leich. 2018. Towards Secure Dynamic Product Lines in the Cloud. In *ICSE*. ACM.

[67] S. Krieter, R. Schröter, T. Thüm, W. Fenske, and G. Saake. 2016. Comparing Algorithms for Efficient Feature-Model Slicing. In *SPLC*. ACM.

[68] S. Krieter, T. Thiem, and T. Leich. 2019. Using Dynamic Software Product Lines to Implement Adaptive SGX-Enabled Systems. In *VaMoS*. ACM.

[69] J. Krüger and T. Berger. 2020. An Empirical Analysis of the Costs of Clone- and Platform-Oriented Software Reuse. In *ESEC/FSE*. ACM.

[70] J. Krüger, C. Lausberger, I. von Nostitz-Wallwitz, G. Saake, and T. Leich. 2020. Search. Review. Repeat? An Empirical Study of Threats to Replicating SLR

Searches. *Empirical Software Engineering* 25, 1 (2020).
[71] J. Krüger, M. Pinnecke, A. Kenner, C. Kruczek, F. Benduhn, T. Leich, and G. Saake. 2018. Composing Annotations Without Regret? Practical Experiences Using FeatureC. *Software: Practice and Experience* 48, 3 (2018).
[72] G. Kulkarni. 2012. Cloud Computing – Software as Service. *International Journal of Cloud Computing and Services Science* 1, 1 (2012).
[73] J. Y. Lee, J. W. Lee, S. D. Kim, et al. 2009. A Quality Model for Evaluating Software-as-a-Service in Cloud Computing. In *SERA*. IEEE.
[74] A. F. Leite, V. Alves, G. N. Rodrigues, C. Tadonki, C. Eisenbeis, and A. C. M. A. De Melo. 2016. Autonomic Provisioning, Configuration, and Management of Inter-Cloud Environments Based on a Software Product Line Engineering Method. In *ICCAC*. IEEE.
[75] A. F. Leite, V. Alves, G. N. Rodrigues, C. Tadonki, C. Eisenbeis, and A. C. M. A. Melo. 2017. Dohko: An Autonomic System for Provision, Configuration, and Management of Inter-Cloud Environments Based on a Software Product Line Engineering Method. *Cluster Computing* 20, 3 (2017).
[76] L. Lesoil, M. Acher, A. Blouin, and J.-M. Jézéquel. 2021. Deep Software Variability: Towards Handling Cross-Layer Configuration. In *VaMoS*. ACM, 1–8.
[77] B. Lundgren and N. Möller. 2019. Defining Information Security. *Science and Engineering Ethics* 25, 2 (2019).
[78] S. Mahdavi-Hezavehi, M. Galster, and P. Avgeriou. 2013. Variability in Quality Attributes of Service-Based Software Systems: A Systematic Literature Review. *Information and Software Technology* 55, 2 (2013).
[79] F. G. Marinho, R. M. C. Andrade, C. Werner, W. Viana, M. E. F. Maia, L. S. Rocha, E. Teixeira, J. B. Ferreira Filho, V. L. L. Dantas, F. Lima, et al. 2013. MobiLine: A Nested Software Product Line for the Domain of Mobile and Context-Aware Applications. *Science of Computer Programming* 78, 12 (2013).
[80] D. S. Markovic, D. Zivkovic, I. Branovic, R. Popovic, and D. Cvetkovic. 2013. Smart Power Grid and Cloud Computing. *Renewable and Sustainable Energy Reviews* 24 (2013).
[81] M. A. Matar, R. Mizouni, and S. Alzahmi. 2014. Towards Software Product Lines Based Cloud Architectures. In *IC2E*. IEEE.
[82] T. McConaghy, R. Marques, A. Müller, D. De Jonghe, T. McConaghy, G. McMullen, R. Henderson, S. Bellemare, and A. Granzotto. 2016. BigchainDB: A Scalable Blockchain Database.
[83] J. Meinicke, T. Thüm, R. Schröter, F. Benduhn, T. Leich, and G. Saake. 2017. *Mastering Software Variability with FeatureIDE*. Springer.
[84] A. Metzger, A. Bayer, D. Doyle, A. M. Sharifloo, K. Pohl, and F. Wessling. 2016. Coordinated Run-time Adaptation of Variability-intensive Systems: An Application in Cloud Computing. In *VACE*. IEEE.
[85] H. Moens and F. De Turck. 2014. Feature-Based Application Development and Management of Multi-Tenant Applications in Clouds. In *SPLC*. ACM.
[86] H. Moens, B. Dhoedt, and F. De Turck. 2015. Allocating Resources for Customizable Multi-Tenant Applications in Clouds using Dynamic Feature Placement. *Future Generation Computer Systems* 53 (2015).
[87] H. Moens, E. Truyen, S. Walraven, W. Joosen, B. Dhoedt, and F. De Turck. 2014. Cost-effective Feature Placement of Customizable Multi-Tenant Applications in the Cloud. *Journal of Network and Systems Management* 22, 4 (2014).
[88] F. Mohamed, R. Mizouni, M. Abu-Matar, M. Al-Qutayri, and J. Whittle. 2017. An Integrated Platform for Dynamic Adaptation of Multi-Tenant Single Instance SaaS Applications. In *FiCloud*. IEEE.
[89] D.-J. Munoz, M. Pinto, and L. Fuentes. 2017. Green Software Development and Research with the HADAS Toolkit. In *ECSA*. ACM.
[90] V. Myllärniemi, M. Raatikainen, and T. Männistö. 2012. A Systematically conducted Literature Review: Quality Attribute Variability in Software Product Lines. In *SPLC*. ACM.
[91] D. Nešić, J. Krüger, Ş. Stănciulescu, and T. Berger. 2019. Principles of Feature Modeling. In *ESEC/FSE*. ACM.
[92] H. V. Nguyen, C. Kästner, and T. N. Nguyen. 2014. Exploring Variability-Aware Execution for Testing Plugin-Based Web Applications. In *ICSE*. ACM.
[93] A. Oussous, F. Benjelloun, A. A. Lahcen, and S. Belfkih. 2018. Big Data Technologies: A Survey. *Journal of King Saud University – Computer and Information Sciences* 30, 4 (2018).
[94] C. Parra, D. Joya, L. Giral, and A. Infante. 2014. An SOA Approach for Automating Software Product Line Adoption. In *SAC*. ACM.
[95] L. Passos, L. Teixeira, N. Dintzner, S. Apel, A. Wąsowski, K. Czarnecki, P. Borba, and J. Guo. 2016. Coevolution of Variability Models and Related Software Artifacts. *Empirical Software Engineering* 21, 4 (2016).
[96] G. Perrouin, M. Acher, J.-M. Davril, A. Legay, and P. Heymans. 2016. A Complexity Tale: Web Configurators. In *VACE*. IEEE.
[97] A. Peruma and D. Krutz. 2018. Security: A Critical Quality Attribute in Self-adaptive Systems. In *SEAMS*. IEEE.
[98] K. Petersen, S. Vakkalanka, and L. Kuzniarz. 2015. Guidelines for Conducting Systematic Mapping Studies in Software Engineering: An Update. *Information and Software Technology* 64 (2015), 1–18.
[99] K. Pohl, G. Böckle, and F. Van Der Linden. 2005. *Software Product Line Engineering: Foundations, Principles, and Techniques*. Springer.
[100] R. A. Popa, C. M. S. Redfield, N. Zeldovich, and H. Balakrishnan. 2011. CryptDB: Protecting Confidentiality with Encrypted Query Processing. *SOSP* (2011).
[101] D. Preuveneers, T. Heyman, Y. Berbers, and W. Joosen. 2016. Feature-Based Variability Management for Scalable Enterprise Applications: Experiences with an E-Payment Case. In *HICSS*. IEEE.
[102] D. Preuveneers, T. Heyman, Y. Berbers, and W. Joosen. 2016. Systematic Scalability Assessment for Feature-Oriented Multi-Tenant Services. *Journal of Systems and Software* 116 (2016).
[103] N. Ragab, A. Ahmed, and S. AlHashmi. 2015. Software Engineering for Security as a Non-Functional Requirement. In *Intelligent Data Analysis and Applications*. Springer.
[104] K. Ramamritham. 1996. Real-Time Databases. *International Journal of Distributed and Parallel Databases* (1996).
[105] M. Rosenmüller, S. Apel, T. Leich, and G. Saake. 2009. Tailor-Made Data Management for Embedded Systems: A Case Study on Berkeley DB. *Data & Knowledge Engineering* 68, 12 (2009).
[106] S. Sagiroglu and D. Sinanc. 2013. Big Data: A Review. In *CTS*. IEEE.
[107] S. Samonas and D. Coss. 2014. The CIA Strikes Back: Redefining Confidentiality, Integrity and Availability in Security. *Journal of Information System Security* 10, 3 (2014).
[108] V. Sartakov, N. Weichbrodt, S. Krieter, T. Leich, and R. Kapitza. 2018. STANlite – A Database Engine for Secure Data Processing at Rack-Scale Level. In *IC2E*. IEEE.
[109] I. Schaefer, R. Rabiser, D. Clarke, L. Bettini, D. Benavides, G. Botterweck, A. Pathak, S. Trujillo, and K. Villela. 2012. Software Diversity: State of the Art and Perspectives. *STTT* 14, 5 (2012).
[110] N. Serrano, G. Gallardo, and J. Hernantes. 2015. Infrastructure as a Service and Cloud Technologies. *IEEE Software* 32, 2 (2015).
[111] A. M. Shaaban, T. Gruber, and C. Schmittner. 2019. Ontology-Based Security Tool for Critical Cyber-Physical Systems. In *SPLC*. ACM.
[112] H. Shafagh, L. Burkhalter, A. Hithnawi, and S. Duquennoy. 2017. Towards Blockchain-Based Auditable Storage and Sharing of IoT Data. In *CCSW*. ACM.
[113] Y. Shakeel, J. Krüger, I. von Nostitz-Wallwitz, C. Lausberger, G. C. Durand, G. Saake, and T. Leich. 2018. (Automated) Literature Analysis - Threats and Experiences. In *SE4Science*. ACM.
[114] N. Siegmund, N. Ruckel, and J. Siegmund. 2020. Dimensions of Software Configuration: On the Configuration Context in Modern Software Development. In *FSE*. ACM.
[115] L. V. Silva, P. Barbosa, R. Marinho, and A. Brito. 2018. Security and Privacy Aware Data Aggregation on Cloud Computing. *Journal of Internet Services and Applications* 9 (2018).
[116] M. Strohbach, J. Daubert, H. Ravkin, and M. Lischka. 2016. Big Data Storage. In *New Horizons for a Data-Driven Economy*. Springer.
[117] M. H Syed and E. B. Fernandez. 2016. Cloud Ecosystems Support for Internet of Things and DevOps using Patterns. In *IoTDI*. IEEE.
[118] T. Thüm, S. Apel, C. Kästner, I. Schaefer, and G. Saake. 2014. A Classification and Survey of Analysis Strategies for Software Product Lines. *ACM Computing Surveys* 47, 1 (2014).
[119] L. P. Tizzei, L. G. Azevedo, M. de Bayser, and R. F. G. Cerqueira. 2015. Architecting Cloud Tools using Software Product Line Techniques: An Exploratory Study. In *SAC*. ACM.
[120] S. A. Tovino. 2017. The HIPAA Privacy Rule and the EU GDPR: Illustrative Comparisons. *Seton Hall Law Review* 47, 4 (2017).
[121] D. Van Landuyt, S. Walraven, and W. Joosen. 2015. Variability Middleware for Multi-Tenant SaaS Applications: A Research Roadmap for Service Lines. In *SPLC*. ACM.
[122] M. Van Steen. 2002. Distributed Systems – Principles and Paradigms. *Network* 2 (2002).
[123] Á. J. Varela-Vaca, R. M. Gasca, R. Ceballos, M. T. Gómez-López, and P. B. Torres. 2019. CyberSPL: A Framework for the Verification of Cybersecurity Policy Compliance of System Configurations using Software Product Lines. *Applied Sciences* 9, 24 (2019).
[124] Á. J. Varela-Vaca, D. G. Rosado, L. E. Sánchez, M. T. Gómez-López, R. M. Gasca, and E. Fernández-Medina. 2020. Definition and Verification of Security Configurations of Cyber-Physical Systems. In *Computer Security*. Springer.
[125] Á. J. Varela-Vaca, D. G. Rosado, L. E. Sánchez, M. T. Gómez-López, R. M. Gasca, and E. Fernandez-Medina. 2021. CARMEN: A Framework for the Verification and Diagnosis of the Specification of Security Requirements in Cyber-Physical Systems. *Computers in Industry* 132 (2021).
[126] S. Walraven, D. Van Landuyt, E. Truyen, K. Handekyn, and W. Joosen. 2014. Efficient Customization of Multi-Tenant Software-as-a-Service Applications with Service Lines. *Journal of Systems and Software* 91 (2014).
[127] C. Wohlin. 2014. Guidelines for Snowballing in Systematic Literature Studies and a Replication in Software Engineering. In *EASE*. ACM.
[128] J. Wu. 2017. *Distributed System Design*. CRC Press.
[129] Y. Zhang, H. He, O. Legunsen, S. Li, W. Dong, and T. Xu. 2021. An Evolutionary Study of Configuration Design and Implementation in Cloud Systems. In *ICSE*. IEEE.