Model-Driven Approaches for DevOps: A Systematic Literature Review

No Author Given

No Institute Given

Abstract. DevOps is a combination of tools and methodologies which aims to improve software development, build and deploy processes by shortening this lifecycle and improving software quality. Despite its many benefits it still presents many challenges when it comes to its ease of use and accessibility. One of the reasons is the tremendous proliferation of tools, languages, and syntax which makes the field quite difficult to learn and keep up to date. Since DevOps is now mainstream, it poses issues to many companies who do not manage to accompany this growth.
As it so happens in many other fields, model-driven engineering approaches can be used to cope with such problems. Thus, in this work, we present a systematic literature review to understand how these two realms currently co-exist. We do so by systematically analyzing several digital libraries and identifying publications whose work focuses on the application of various model-driven engineering approaches to the field of DevOps and its different variants (DevSecOps, MLOps, etc.).
We have categorized 34 publications, ranging from 8 different computing domains, from cloud to machine learning, but also several domain-agnostic proposals. We have also found that deployment is by far the most addressed phase of DevOps. Moreover, we also discuss the goals these works try to address, such as how most try to abstract the specific cloud vendor technologies that exist and aim for a more generic and overall solution.

Keywords: DevOps, model-driven, survey

1 Introduction

DevOps is a means for software development and delivery by using various tools and techniques that integrate the worlds of development and operations [16]. Indeed, the word DevOps comes from the combination of Development with Operations. This term is used to describe a culture where these two realms are no longer separated as in the past, but are now just one. The integration of these two fields is achieved through automated development, build, deployment, and monitoring of software. The goals include to achieve reliable, secure, fast and continuous delivery of software, to improve productivity in businesses and workers' well being. Moreover, software engineers that use DevOps are more resilient and equipped for fast changes [29].

However, organizations and developers face many obstacles when adopting DevOps, including changes in the architectural organization, dealing with problems in older infrastructure and the integration of different DevOps tools [20]. Other problems faced by developers are in learning and using the specific DevOps tools which can be overwhelming and can decrease the improvements expected with the adoption of DevOps [23].

Model-driven engineering (MDE) approaches have been used to cope with numerous problems in software development complexity, from aiding in the design process, to the generation of code for different platforms, to improving the communication between different teams and team members [49]. Models, being a partial representation of a system, can ignore details and focus on the relevant parts in each moment and phase of software development and maintenance. Thus, it seems a very reasonable approach to the complex problem of DevOps use. Indeed, over the past decade, several works have been done at the intersection of the fields of MDE and DevOps. However, we have not found any papers summarizing these findings or contributions. These works have applications for numerous DevOps tasks and fields of computing, fields such as cloud-computing , machine learning, data science and cyber-physical-systems and to broad diverse tasks such as testing, deployment and monitoring.

In this paper, we present the first systematic literature review of these works to understand the domains in which model-driven approaches for DevOps is used, which phases of DevOps are covered by MDE approaches, and which are the goals of these works. Our goal with this study is to summarize the work done at the intersection of these fields as well as to identify some possible contributions that can be done in this space. In particular, we seek to answer the following research questions (RQs):

RQ1 Which computing domains are covered by model-driven approaches for DevOps?
RQ2 Which phases of DevOps are covered by model-driven approaches?
RQ3 Which are the goals of the model-driven approaches for DevOps?

With RQ1 we want to better understand the field and for which kinds of domains are already covered by MDE approaches for DevOps. With our work we found 34 published scientific papers proposing model-driven approaches for DevOps. These works cover 8 computing domains, from cloud to blockchain, but we also found some proposals are domain-agnostic.

DevOps can be applied in different phases [56] of software construction and maintenance. Thus, with RQ2 we seek to understand which of these phase are covered by model-driven approaches. We also found that all the 7 phases of DevOps are covered by MDE approaches, although deployment is clearly the most addressed.

We also propose to better understand the goals of these works. In RQ3 we found that several objectives exist, from improving development or deployment phases of DevOps, but that mostly these works try to cope with the different characteristics, technologies and languages available in the cloud vendors solutions.

The remaining of this paper is organized as follows. We start by presenting related work in Section 2. Afterwards, in Section 3, we present the methodology we followed in this work. In Section 4 we present the results of this work, from the papers found to the analysis of the different dimensions of the papers, answering the RQs. Finally, in Section 5 we present some concluding remarks.

2 Related work

Several literature reviews have been made for both the fields of DevOps and model-driven engineering.

Relating to DevOps one survey focuses on finding and understanding different challenges from the perspective of engineers, managers, and researchers [32]. Their process was to search scientific libraries, selecting papers, and then snowballing the papers found. They organized the results into conceptual maps. Another paper showcases a list of security challenges and practices for DevOps software development gathered from doing a systematic literature review [33]. Other researchers have been working in understanding DevOps culture by doing a systematic literature review of various DevOps works [44]. Others have done work to understand the best practices [4], pitfalls [40], and benefits [11] of DevOps. Some aim at studying the impact DevOps has on software quality [12].

In the content of MDE, various surveys have also been made to gather and summarize current contributions to the field. One of the publications reviews 140 research papers to understand the uses of MDE across the development lifecycle of cyber-physical systems and identifies research gaps and areas that need more investigation [39,38]. Another study reviews the use of meta-models of elicitation techniques [24]. Researchers have also been work to understand the benefits of MDE in the software industry and how existing processes can be reused [25]. The use of MDE for reverse engineering was also studied [42]. The application and challenges of MDE in cloud computing have also been studied [41]. A systematic mapping study has been done to understand the role and challenges of MDE for the development of safety and security systems [35]. A systematic review of the use of requirements engineering techniques in model-driven development has been made [34].

However, to the best of our knowledge no survey or other kind of systematic study has been done to know the use of model-driven engineering in the context of DevOps.

3 Research Method

Our research method consisted in realizing a systematic literature review where we queried four digital libraries, namely the *ACM Digital Library*[1], the *IEEE*

[1] `https://dl.acm.org`

Xplore[2], *ScienceDirect*[3] and *SpringerLink*[4], for papers containing both topics of interests, namely model-driven and DevOps. Thus, we started by defining an initial query for searching these libraries for relevant publications:

DevOps AND ("model-driven" OR "model driven")

This query contains only two terms because they represent very well both areas we are studying (although not completely as we will see in a few paragraphs). We did not include terms such as "model based" or others similar as our focus in model-driven approaches, that is, approaches for which models are paramount in contrast with model-based approaches were more are used in a more lightweight way.

We have run this query and the number of papers found for each library can be seen in the second column of Table 1 (First query).

Table 1. Number of papers found in each library.

Venue	**First query**	**Second query**
ACM Digital Library	106	106
IEEE Xplore	33	33
ScienceDirect	101	101
SpringerLink	479	505
Total	718	744

After gathering the papers containing the search terms we manually filtered, by screening the content, the papers whose work involved the application of MDE techniques in various areas of DevOps and using the following inclusion and exclusion criteria:

Inclusion criteria

I_1 The work in the paper applies MDE to the field of DevOps
I_2 The paper analyzes current MDE use in DevOps
I_3 The paper exposes requirements for the use of MDE in DevOps

Exclusion criteria

E_1 Irrelevant to the topics of DevOps and MDE
E_2 Abstracts only
E_3 Proceedings

After filtering the initial 718 papers we ended up with 32 publications. Finally, we read the 32 papers and organized them according to their technical domain and DevOps phase. During the reading we realized several other terms

[2] `http://ieeexplore.ieee.org`
[3] `https://www.sciencedirect.com`
[4] `https://link.springer.com`

related to DevOps were being used, namely ‘MLOps’, ‘DevSecOps’, ‘CloudOps’, ‘AIOps’, and ‘DataOps’, which represent various fields of application of DevOps. Thus, we extended the query to include these terms. The final query is as follows:

((DevOps OR MLOps OR ChainOps OR DevSecOps OR CloudOps OR AIOps OR DataOps)
AND
(“model-driven” OR “model driven”))

We then proceeded to query the same databases and the final number of papers can be seen in Table 1 having this new query produced 26 new papers. This was done on October 2022. As before, we applied inclusion and exclusion criteria to filter irrelevant work using the same process used previously. This new query’s results were merged with the ones from the original one producing a total of 34 papers.

In the next section, we detail our findings.

4 Results

In this Section, we present in detail the results we found starting by answering RQ1.

4.1 RQ1: Which computing domains are covered by model-driven approaches for DevOps?

To answer this RQ we read in detail each of the 34 papers and collected the domains which the papers address. This information was gathered from analyzing the papers collected and not always explicit, when the phases or domain of the paper were not explicit we interpreted the contents of the paper and classified it according to the contributions made on the paper. In the following paragraphs, we describe these works grouped by domain.

Cloud Of the 34 publications under study, 17 (50%) address the cloud domain, being this the most coveted domain. Regarding the modeling language used, most of these projects adapted existing languages already used in the cloud computing domain such as TOSCA [47,55,46,9,7,3,13,48]. We also found that the use of meta-models was prevalent in these publications [47,55,46,7,3,13,18,48].

Internet of Things We found only 4 publications (almost 12% of all papers) in the Internet of Things (IoT) space. One of these publications proposes a meta-model for IoT capable of standardizing current and future IoT architectures [17]. Another project tackles the problem of deploying multiple gateways and connected IoT devices automatically using a model-driven approach [51]. Another group of authors created a modeling infrastructure for the systematic engineering of IoT applications [30].

Safety-Critical Systems For safety-critical systems we found just 1 article (less than 3% of the studied papers) [37]. In this work, the authors propose a modeling framework compatible with the DevOps principles of continuous testing and continuous integration for the design of safety-critical systems while obeying industry standards. In this work, the authors define their modeling language by making use of a meta-model. The authors also expressed that the ability to reason and make proofs over the models was one factor for their use of model-driven engineering.

Cyber-Physical Systems In terms of cyber-physical systems (CPS), we found 3 publications (less than 9%) whose focus was to model the whole development lifecycle of these systems [15,28]. Two publications created their languages through meta-models. As for safety-critical systems, in both papers, the authors express that the ability to reason and make proofs over the models was one factor in their use of model-driven engineering. Another publication uses MDE to provide AI-augmented automation capabilities to the continuous development of cyber-physical systems [8].

Blockchain In the blockchain domain, we found 1 publication whose work consisted in developing a model-driven approach to the creation of smart contracts [27]. The authors designed their language making use of meta-models. Once again, the ability to reason and make proofs over the models was one factor for their use of model-driven engineering.

Machine Learning In this space, we found 2 publications (about 6%) [26,5]. The authors of both articles created their language, but only one publication made use of meta-models [26].

Data Science In this field, 1 project applied MDE techniques to improve the reproducibility and replicability of data science projects. The authors created a model-driven framework that allows users to define data science models in a platform-independent way by separating the definition of the modeling and operation layers [36].

Big Data Regarding big data, we found 1 publication [10]. The language created by the authors is their own and they make use of meta-models.

Domain Agnostic We found 4 publications (almost 12%) whose work is not dedicated to any particular domain. In the approach of one of these works [14] developers can use the proposed framework to create processes and platforms, a platform a sequence of processes, and a process an application or service. Their approach however is dependent on the tools and platform used and does not offer a generic approach to doing each DevOps phase and the entire lifecycle.

This work is based on a previous one [6] and both have their languages and use meta-models [14,6]. We also found another project whose goal was to aid the management side of software projects by applying model-driven techniques. Their work touches on the operations phase of the DevOps lifecycle [52]. Another project uses UML to automate the deployment [43].

Summary In the 34, papers analyzed we found 8 computing domains, plus some works that are domain-agnostic and can potentially be used in different domains. However, 30 papers, about 88%, present domain-specific approaches. This may indicate that creating a model-driven generic approach for DevOps is not the best approach, being a more specific proposal a better path.

It is quite clear that the cloud is the domain to which most researchers dedicate their time, accounting for 50% of all works. In line with the previous reasoning that a more domain-specific approach is probably more fruitful, many domains are still very much uncovered by research. This seems to indicate that there is still space for proposals, especially in the others domains.

Several other domains are not yet covered by any work. An example would be mobile development. Thus, there are still domains where research is necessary.

Several works reused or adapted existing modeling languages, but many others created their own. Moreover, many of the papers presented a meta-model which makes extending the language and the approach potentially easier. One of the main reasons claimed by the authors for the use of a model-driven approach is the fact it is possible to reason using the models, and also to make proofs about the systems.

4.2 RQ2: Which phases of DevOps are covered by model-driven approaches?

In this Section, we analyze the various articles and organized them according to the phases of DevOps addressed by each one.

Development The phase of continuous development involves the tasks of planning and developing software. This process is an application of the agile methodology to the world of software development, where the whole software gets broken down into smaller pieces, and software is updated in smaller batches after being tested [56].

We found several works relating to this phase, these works span various fields of computing including cloud computing [47], IoT [17,30,51], cyber-physical systems [15,28], blockchain [27], machine learning [26,5], data science [36], big data [10] as well as domain agnostic approaches [6,14].

Integration This phase is at the core of the DevOps lifecycle and affects the process of committing new changes in the source code. When doing continuous integration developers are forced to do more frequent software releases and these more frequent releases, in conjunction with the use of various testing techniques such as unit testing and integration testing and other quality assurance practices such as code reviews, make the software more secure and stable [56].

We found several works relating to this phase, spanning various fields of computing including cloud computing [47], IoT [17,30,51], cyber-physical systems [15,28], blockchain [27], data science [36], big data [10] as well as domain agnostic approaches [6,14].

Testing Continuous testing involves the use of automatic testing tools such as Selenium, JUnit, Jenkins, and Docker [56]. This process saves organizations time and increases productivity [56].

Several of the works studied to address this phase. These works span various fields of computing including cloud computing [31], IoT [30], safety-critical systems [37], cyber-physical systems [15,28], blockchain [27], data science [36], as well as domain agnostic approaches [6,14,43].

Monitoring This phase entails performance monitoring and recording of the application, including potential errors such as high memory usage and networking problems. This monitoring allows development teams to find and correct errors earlier and substantially reduce costs [56].

We found several works relating to this phase in different domains such a IoT [30], cyber-physical systems [15,28,8], safety-critical systems [37], blockchain [27], data science [36], big data [10] as well as domain agnostic approaches [6,14].

Feedback This phase comprises the gathering, analysis, and use of feedback coming from the client's end. This information includes performance and errors as well as customer feedback [56].

Some of the works analyzed cope with this phase, spanning various fields of computing including cloud computing [22], IoT [30], safety-critical systems [37], cyber-physical systems [15,28], blockchain [27], data science [36], big data [10] as well as domain agnostic approaches [6,14].

Deployment Continuous deployment affects the process of deploying new code to the production environment. This process is made automatically and reduces the need for having manual and planned releases since new changes to the source code are sent into production after the process of integration and testing. This phase makes extensive use of container technology such as Docker to make use of standardized environments [56].

Most of the works addressing this phase are in the cloud computing domain [47,21,53,48,54,18,2,13,3,7,9,55,46,50]. However, we also found works in other areas including IoT [30], safety-critical systems [37], cyber-physical systems [15,28], blockchain [27], data science [36], big data [10] as well as domain agnostic approaches [6,14].

Operations This phase includes the automation of all operation processes that help developers automate releases, detect issues faster and improve software products. This practice allows for faster delivery times and higher quality software [56].

We found several works relating to this phase. These works span various fields of computing including cyber-physical systems [15,28], blockchain [27], data science [36] as well as domain agnostic approaches [6,14].

Summary We found considerably more papers for the deployment phase, which was due to the considerable amount of work done in improving compatibility between different cloud providers.

We also found several projects that touched the whole DevOps lifecycle and found that the monitoring, integration, and operations phases were the ones with the least amount of work and thus with more space for new contribution.

4.3 RQ3: Which are the goals of the model-driven approaches for DevOps?

In this Section, we discuss the goals addressed by the works under analysis.

One of the main objectives of these works is to *abstract cloud vendors* particularities so the management of could(-deployed) applications can be done in an agnostic way of the particular cloud supplier. These particularities include lock-in issues with auto-scaling configurations [1,2], infrastructure provisioning using a domain-specific language that converts to code for different providers [48], but most of the works focus on management of cloud resources or infrastructure [21,7,48,54,13,46,53,18,46]. In general, most of these articles aim at creating a cloud platform-agnostic way of managing code and doing different operations in different cloud services, improving reusability between those services as well as hiding technical details from the users.

Several works set as goal to improve the *deployment* phase [1,3,50,30,19,18,43]. These span different domains, but most of them focus on the cloud computing domain, these works often have as their goal to abstract from several technical tasks relating to deploying software to the cloud, this abstraction is useful to improve compatibility between different cloud providers. Other works focus not mainly on the deployment phase but mostly on providing an integrated approach for doing DevOps either generic or for a particular field.

Quality is explicitly[5] addressed by four works. One of the goals is to use model-driven engineering to create a framework that allows the creation of data-intensive software that supports reliability, efficiency, and safety requirements [10]. The other work uses continuous deployment to increase the quality

[5] One can admit that the use of DevOps or model-driven approaches can implicitly help to achieve several goals, from complexity management to fast deployment, to better security. However, we focus on the goals researchers explicitly expressed in the papers.

of the software [3]. Two different works explicitly address reliability, which can be seen as a characteristic of quality [10,30].

A couple of works try to cope with *serverless* technologies, either by aiding in creating and managing applications based on such paradigm [9], or by aiding in the decomposition and orchestration in such an environment [55].

The management of *multi-cloud applications* is also addressed only in two works [18,15], the first being from the cloud domain and the second from the cyber-physical systems. In the work in the cyber-physical domain, the author's motivation is the fact that the various components of cyber-physical systems often run in several heterogeneous environments during their lifespan and the author's aim is to provide a layer of abstraction for those different environments to facilitate the development process. The paper in the cloud computing domain tries to solve the problem of dealing with various cloud computing providers when developing large-scale distributed applications due to the lack of compatibility between said providers, their solution was to create a framework and environment capable to facilitate the deployment of multi-cloud applications.

Another project that reached the same areas consists of a conceptual security model to facilitate the adoption of DevSecOps[6].) for the business processes over the cloud [22]. Another work focuses on improving data security and privacy assurance in teams using agile methodologies [31].

Although model-driven is widely used for designing software, we could also find two papers explicitly setting design as a goal, one for designing safety-critical system [37] and another allows us to graphically design cloud applications and deploy and manage them at runtime [13].

The development process has also been addressed in two papers. In the data science, domain [36] we found the authors' goal was to improve the *development* process by providing a pipeline definition of data science projects agnostic of different technologies.

We found only one paper which focuses on *aiding* non-experts to develop smart contract-based distributed applications [27].

Standardization was also only found in one paper. In particular, it proposes an approach to standardize current and future IoT architectures [17].

In Table 2 we present the specific goals authors express in each paper, the computing domain of each paper, as well as the different DevOps phases addressed in the paper. In this table, each publication is represented by its first author's initials and the year it was published.

[6] DevSecOps is a variant of DevOps, but including security controls to provide continuous security assurance [45]

Table 2: Goals found in the different papers under analysis. Dev abbreviates the Development phase, Int the Integration, Test the Testing, Mon the Monitoring, Feeds the Feedback, Dep the Deployment, and Op the Operations.

Goals	**Domain**	**Phases**	**Paper**
Facilitate fast service deployment	Cloud	Dev	H.A, 2016 [1]
Abstract and automate a continuous delivery process of cloud resources	Cloud	Dep, Int	J.S, 2018 [47]
Optimization, and runtime capacity allocation of cloud applications	Cloud	Feed, Dep	M.G, 2015 [21]
Continuous deployment for quality DevOps	Cloud	Dep	M.A, 2016 [3]
Improve compatibility between different cloud resources	Cloud	Dep	D.W, 2016 [53]
Orchestration for cloud resources	Cloud	Dep	H.B, 2019 [7]
Manage infrastructure as code	Cloud	Dep	J.S, 2019 [48]
Low priority of data security and privacy assurance in agile environments	Cloud	Test	R.K,2020 [31]
DevSecOps using open-source software over cloud	Cloud	Feed, Dep	M.G, 2015 [22]
Managing cloud resources	Cloud	Dep	D.W, 2016 [54]
Manage multi-cloud applications	Cloud	Dep	N.F, 2018 [18]
Auto-scaling services on multiple clouds	Cloud	Dep	H.A, 2018 [2]
Cloud resource management	Cloud	Dep	S.C, 2020 [13]
Automatically assigning multiple software deployments	Cloud	Dep	H.S, 2020 [50]
Streamlining DevOps automation for Cloud applications	Cloud	Dep	J.W, 2016 [55]
Infrastructure provisioning for multiple clouds	Cloud	Dev	J.S, 2019 [46]
Rational decomposition and orchestration for serverless computing	Cloud	Dep	G.C, 2020 [9]
Fleet deployment in the IoT	IoT	Dep	S.H,2020 [51]
Continuous deployment of IoT systems	IoT	Dep	N.F, 2019 [19]
Standardize existing IoT architectures	IoT	Dev, Int	B.E, 2020 [17]
Development and deployment of reliable IoT applications	IoT	Dev, Dep	J.K,2022 [30]
Design of safety-critical systems	Safety-critical systems	Test, Mon, Feed	B.M, 2019 [37]

Continued on next page

Table 2 – continued from previous page

Goals	Domain	Phases	Paper
Unified Model-based Engineering, Digital Twins, and DevOps practice in a uniform workflow	Cyber-physical systems	Dev, Int, Test, Mon, Feed, Dep, Op	J.H, 2020 [28]
DevOps environment for multi-cloud applications	Cyber-physical systems	Dev, Int, Test, Mon, Feed, Dep, Op	B.C, 2020 [15]
Digital twins for the engineering of CPS	Cyber-physical systems	Mon	H.B, 2022 [8]
Allow non-experts to develop smart contract-based distributed applications	Blockchain	Dev, Int, Test, Mon, Feed, Dep, Op	W.J, 2021 [27]
MLOps for intelligent enterprise applications	Machine learning	Dev	W.J, 2020 [26]
Modeling autonomic systems for ML and microservices	Machine learning	Int; Dep	N.B, 2021 [5]
Reproducibility and replicability of data science projects	Data Science	Dev, Int, Test, Mon, Feed, Dep, Op	F.M,2022 [36]
Quality-driven development of data-intensive cloud applications	Big Data	Dev, Int, Mon, Feed, Dep	G.S, 2015 [10]
Gathering requirement for a modeling framework for DevOps that can be reusable across different domains	Generic	Dev, Int, Test, Mon, Feed, Dep, Op	F.B, 2020 [6]
Modeling framework for DevOps that can be reusable across different domains	Generic	Dev, Int, Test, Mon, Feed, Dep, Op	A.C, 2020 [14]
Acceptance tests to support software measurement	Generic	Test, Op	L.F, 2018 [43]

Summary Answering RQ3, several goals can be found in the analyzed papers, namely the creation of could vendor agnostic framework to allow abstracting each vendor's specificities, improving DevOps phases as development and deployment, increasing the quality of software, coping with multi-cloud applications and the serverless paradigm, improving the security of software, aiding non-experts and work through standardization.

Some of these objectives are domain-specific, for instance, the ones more related to cloud vendors, but others can be found in different domains (e.g. deployment, development).

Security, especially important in online and distributed applications, is not directly addressed, except for 1 case. This seems to indicate that model-driven approaches for DevOps to improve security are still very much unexplored.

With the current trend of low/no-code platforms, which very much rely on modeling, we found only one paper about helping non-experts. This also seems to indicate that DevOps is still seen as a technology only used by experts which creates opportunities to research how can they be made available to people with less expertise in DevOps.

Overall since some of the main benefits of DevOps include improving developer productivity, reducing labor cost, a better and more organized approach with various benefits including automation, reusability and replicability, and better cohesion between team members [29] we found that most of the work goals aimed at further improving aspects that motivated and are main benefits of the DevOps adoption.

4.4 Intersection between phases and domains

In this section we perform an analysis orthogonal to the research questions. The distribution of the publications according to the DevOps phases and domains is presented in Table 3.

For the papers relating to the domain of cloud computing, we can observe from the table that most contributions focus on improving and managing existing cloud computing processes. Those improvements often involve using MDE to improve or provide reusability of those processes across various platforms and also the creation of platform-agnostic models to do so. The contributions made in this domain are primarily focused on the deployment phase. These contributions are also, for the vast majority, intended for technical users with previous experience doing tasks with the intent of improving developers' productivity.

Other projects offer contributions outside of the deployment phase. These include proposing a model-driven approach to abstract and automate a continuous delivery process of cloud resources in development, testing, and production environments, which touches the development, integration, testing, feedback, deployment, and operations phases [47]. Another project that reached the same areas consists of a conceptual security model to facilitate the adoption of DevSecOps for the business processes over the cloud [31].

In the IoT domain most of the contributions focus on improving the development process. Those contributions focus mainly on the deployment and development phases of the DevOps process.

When it comes to safety-critical systems the work done in this area focuses on the various phases of the DevOps process and aims at developing a model-driven

Table 3. Publications organized according to their domains and DevOps lifecycle phases.

Domain/ Phase	Develop.	Integration	Testing	Monitoring	Feedback	Deployment	Operations
Cloud	J.S, 2018 [47]	J.S, 2018 [47]	R.K, 2020 [31]		M.G, 2015 [22]	H.A, 2016 [1] M.G, 2015 [21] D.W, 2016 [53] J.S, 2019 [48] D.W, 2016 [54] N.F, 2018 [18] H.A, 2018 [2] S.C, 2020 [13] M.A, 2016 [3] H.B, 2019 [7] G.C, 2020 [9] J.W, 2016 [55] J.S, 2019 [46] H.S, 2020 [50]	
IoT	B.E, 2020 [17] J.C,2021 [30] H.S,2022 [51]	B.E, 2020 [17] J.C,2021 [30]	J.C,2021 [30]	J.C,2021 [30]	J.C,2021 [30]	J.C,2021 [30] SN.F, 2019 [19]	
Safety-Critical Systems			B.M, 2019 [37]	B.M, 2019 [37]	B.M, 2019 [37]		
Cyber-Physical Pystems	B.C, 2020 [15] J.H, 2020 [28]	B.C, 2020 [15] J.H, 2020 [28]	B.C, 2020 [15] J.H, 2020 [28]	B.C, 2020 [15] J.H, 2020 [28] H.B, 2022 [8]	B.C, 2020 [15] J.H, 2020 [28]	B.C, 2020 [15] J.H, 2020 [28]	B.C, 2020 [15] J.H, 2020 [28]
Blockchain	W.J, 2021 [27]	W.J, 2021 [27]	W.J, 2021 [27]	W.J, 2021 [27]	W.J, 2021 [27]	W.J, 2021 [27]	W.J, 2021 [27]
Machine Learning	W.J, 2020 [26] N.B, 2021 [5]						
Data Science	F.M,2022 [36]	F.M,2022 [36]	F.M,2022 [36]	F.M,2022 [36]	F.M,2022 [36]	F.M,2022 [36]	F.M,2022 [36]
Big Data	G.S, 2015 [10]	G.S, 2015 [10]		G.S, 2015 [10]	G.S, 2015 [10]	G.S, 2015 [10]	
Domain Agnostic	F.B, 2020 [6] A.C, 2020 [14]	F.B, 2020 [6] A.C, 2020 [14]	F.B, 2020 [6] A.C, 2020 [14] L.F, 2018 [43]	F.B, 2020 [6] A.C, 2020 [14]	F.B, 2020 [6] A.C, 2020 [14]	F.B, 2020 [6] A.C, 2020 [14]	F.B, 2020 [6] A.C, 2020 [14] K.T, 2022 [52]

approach that would allow its users to perform those tasks while abstracting from concrete technologies.

In terms of cyber-physical systems, we found two publications whose focus was to model the whole development lifecycle of these systems [15,28]. This is also the case of the work in blockchain [27] and the one in data science [36]. These three domains are the only ones where all the DevOps phases are covered by the same work (except for the domain agnostic approaches).

We found three publications whose work is not dedicated to any particular domain. Two of these publications span the whole DevOps lifecycle [6,14], while the other is an industrial case study with the goal of deriving requirements for a model-based approach for DevOps by studying the DevOps process used by this company [6]. The goal of one of the other articles [14] is to build upon those requirements to create a conceptual framework for modeling and combining DevOps processes and platforms. In their approach developers can use their framework to create processes and platforms, processes being an application or service and platform a sequence of processes. Their approach however is dependent on the tools and platform used and does not offer a generic approach to doing each DevOps phase and the lifecycle. We also found another project whose goal was to aid the management side of software projects by applying model-driven techniques. Their work touches on the operations phase of the DevOps lifecycle [52]. Another project uses UML to automate the deployment [43].

Summary Only 6 papers (about 17%) cross all DevOps phases, being two in the CPS domain, one in blockchain, another in data science, and two domain agnostic. This may seem to indicate that to create an approach that copes with all the phase is too difficult or not interesting as a solution.

We can also conclude that several intersections are not covered. For instance, the operations phase for cloud, IoT, safety-critical systems, machine leaning, and big data are not addressed by any work. In fact, operations is the most neglected phase.

The machine learning field has only two approaches that only cover the development phase. There is also quite some space for new works to tackle this intersection.

Finally, we can see that safety-critical systems' domain is only addressed by one work and only three phases. Once more, researchers can focus their work in this domain.

5 Concluding Remarks and Future Work

Based on our research we can see that the majority of papers actuate in the domain of cloud computing and more specifically in the deployment phase. Nevertheless, we also found that some papers tackle the other phases of the same domain, but in less quantity. We also found that of the papers relating to the deployment phase of cloud computing most of them focus on interoperability between cloud services.

We also learned that the use of meta-models was prevalent in these projects and found that the ability to reason and create proofs over models was a factor in using model-driven approaches in the spaces of safety-critical systems, cyber-physical systems, and blockchain.

In terms of generic approaches, we found that some authors acknowledge the need for a model-driven approach to the DevOps process and that existing frameworks are still reliant on specific technologies.

Overall, we found a trend of growing interest over the past two years in publications in domains outside of cloud computing, since we can see that, while the vast majority of publications in this domain happened before 2020, the opposite is true for the other domains.

References

1. Alipour, H., Liu, Y.: A model driven method to deploy auto-scaling configuration for cloud services. In: Proceedings of the 4th International Workshop on Release

Engineering. p. 23. RELENG 2016, Association for Computing Machinery, New York, NY, USA (2016). https://doi.org/10.1145/2993274.3011285, https://doi.org/10.1145/2993274.3011285
2. Alipour, H., Liu, Y.: Model driven deployment of auto-scaling services on multiple clouds. In: 2018 IEEE International Conference on Software Architecture Companion (ICSA-C). pp. 93–96 (2018). https://doi.org/10.1109/ICSA-C.2018.00033
3. Artač, M., Borovšak, T., Di Nitto, E., Guerriero, M., Tamburri, D.A.: Model-driven continuous deployment for quality devops. In: Proceedings of the 2nd International Workshop on Quality-Aware DevOps. p. 40–41. QUDOS 2016, Association for Computing Machinery, New York, NY, USA (2016). https://doi.org/10.1145/2945408.2945417, https://doi.org/10.1145/2945408.2945417
4. Badshah, S., Khan, A.A., Khan, B.: Towards process improvement in devops: A systematic literature review. In: Proceedings of the Evaluation and Assessment in Software Engineering. p. 427–433. EASE '20, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3383219.3383280, https://doi.org/10.1145/3383219.3383280
5. Bencomo, N.: Modeling autonomic systems in the time of ml, devops and microservices. In: 2021 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C). pp. 731–731 (2021). https://doi.org/10.1109/MODELS-C53483.2021.00118
6. Bordeleau, F., Cabot, J., Dingel, J., Rabil, B.S., Renaud, P.: Towards modeling framework for devops: Requirements derived from industry use case. In: Bruel, J.M., Mazzara, M., Meyer, B. (eds.) Software Engineering Aspects of Continuous Development and New Paradigms of Software Production and Deployment. pp. 139–151. Springer International Publishing, Cham (2020)
7. Brabra, H., Mtibaa, A., Gaaloul, W., Benatallah, B., Gargouri, F.: Model-driven orchestration for cloud resources. In: 2019 IEEE 12th International Conference on Cloud Computing (CLOUD). pp. 422–429 (2019). https://doi.org/10.1109/CLOUD.2019.00074
8. Bruneliere, H., Muttillo, V., Eramo, R., Berardinelli, L., Gómez, A., Bagnato, A., Sadovykh, A., Cicchetti, A.: Aidoart: Ai-augmented automation for devops, a model-based framework for continuous development in cyber–physical systems. Microprocessors and Microsystems **94**, 104672 (2022). https://doi.org/https://doi.org/10.1016/j.micpro.2022.104672, https://www.sciencedirect.com/science/article/pii/S0141933122002022
9. Casale, G., Artač, M., van den Heuvel, W.J., van Hoorn, A., Jakovits, P., Leymann, F., Long, M., Papanikolaou, V., Presenza, D., Russo, A., Srirama, S.N., Tamburri, D.A., Wurster, M., Zhu, L.: Radon: rational decomposition and orchestration for serverless computing. SICS Software-Intensive Cyber-Physical Systems **35**(1), 77–87 (Aug 2020). https://doi.org/10.1007/s00450-019-00413-w, https://doi.org/10.1007/s00450-019-00413-w
10. Casale, G., Ardagna, D., Artac, M., Barbier, F., Di Nitto, E., Henry, A., Iuhasz, G., Joubert, C., Merseguer, J., Munteanu, V.I., Perez, J.F., Petcu, D., Rossi, M., Sheridan, C., Spais, I., Vladuic, D.: Dice: Quality-driven development of data-intensive cloud applications. In: 2015 IEEE/ACM 7th International Workshop on Modeling in Software Engineering. pp. 78–83 (2015). https://doi.org/10.1109/MiSE.2015.21
11. Céspedes, D., Angeleri, P., Melendez, K., Dávila, A.: Software product quality in devops contexts: A systematic literature review. In: Mejia, J., Muñoz, M., Rocha,

Á., A. Calvo-Manzano, J. (eds.) Trends and Applications in Software Engineering. pp. 51–64. Springer International Publishing, Cham (2020)

12. Céspedes, D., Angeleri, P., Melendez, K., Dávila, A.: Software product quality in devops contexts: A systematic literature review. In: Mejia, J., Muñoz, M., Rocha, Á., A. Calvo-Manzano, J. (eds.) Trends and Applications in Software Engineering. pp. 51–64. Springer International Publishing, Cham (2020)
13. Challita, S., Korte, F., Erbel, J., Zalila, F., Grabowski, J., Merle, P.: Model-based cloud resource management with tosca and occi (2020). https://doi.org/10.48550/ARXIV.2001.07900, `https://arxiv.org/abs/2001.07900`
14. Colantoni, A., Berardinelli, L., Wimmer, M.: Devopsml: Towards modeling devops processes and platforms. In: Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings. MODELS '20, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3417990.3420203, `https://doi.org/10.1145/3417990.3420203`
15. Combemale, B., Wimmer, M.: Towards a model-based devops for cyber-physical systems. In: Bruel, J.M., Mazzara, M., Meyer, B. (eds.) Software Engineering Aspects of Continuous Development and New Paradigms of Software Production and Deployment. pp. 84–94. Springer International Publishing, Cham (2020)
16. Ebert, C., Gallardo, G., Hernantes, J., Serrano, N.: Devops. IEEE Software **33**(3), 94–100 (2016). https://doi.org/10.1109/MS.2016.68
17. El Khalyly, B., Belangour, A., Banane, M., Erraissi, A.: A new metamodel approach of ci/cd applied to internet of things ecosystem. In: 2020 IEEE 2nd International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS). pp. 1–6 (2020). https://doi.org/10.1109/ICECOCS50124.2020.9314485
18. Ferry, N., Chauvel, F., Song, H., Rossini, A., Lushpenko, M., Solberg, A.: Cloudmf: Model-driven management of multi-cloud applications. ACM Trans. Internet Technol. **18**(2) (jan 2018). https://doi.org/10.1145/3125621, `https://doi.org/10.1145/3125621`
19. Ferry, N., Nguyen, P.H.: Towards model-based continuous deployment of secure iot systems. In: 2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C). pp. 613–618 (2019). https://doi.org/10.1109/MODELS-C.2019.00093
20. Ghantous, G.B., Gill, A.Q.: Devops: Concepts, practices, tools, benefits and challenges. In: PACIS (2017)
21. Guerriero, M., Ciavotta, M., Gibilisco, G.P., Ardagna, D.: A model-driven devops framework for qos-aware cloud applications. In: 2015 17th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC). pp. 345–351 (2015). https://doi.org/10.1109/SYNASC.2015.60
22. Guerriero, M., Ciavotta, M., Gibilisco, G.P., Ardagna, D.: Space4cloud: A devops environment for multi-cloud applications. In: Proceedings of the 1st International Workshop on Quality-Aware DevOps. p. 29–30. QUDOS 2015, Association for Computing Machinery, New York, NY, USA (2015). https://doi.org/10.1145/2804371.2804378, `https://doi.org/10.1145/2804371.2804378`
23. Haque, M.U., Iwaya, L.H., Babar, M.A.: Challenges in docker development: A large-scale study using stack overflow. In: Proceedings of the 14th ACM / IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM). ESEM '20, Association for Computing Machinery, New York,

NY, USA (2020). https://doi.org/10.1145/3382494.3410693, https://doi.org/10.1145/3382494.3410693
24. He, C., Mussbacher, G.: Model-driven engineering and elicitation techniques: A systematic literature review. In: 2016 IEEE 24th International Requirements Engineering Conference Workshops (REW). pp. 180–189 (2016). https://doi.org/10.1109/REW.2016.041
25. Hebig, R., Bendraou, R.: On the need to study the impact of model driven engineering on software processes. In: Proceedings of the 2014 International Conference on Software and System Process. p. 164–168. ICSSP 2014, Association for Computing Machinery, New York, NY, USA (2014). https://doi.org/10.1145/2600821.2600846, https://doi.org/10.1145/2600821.2600846
26. van den Heuvel, W.J., Tamburri, D.A.: Model-driven ml-ops for intelligent enterprise applications: Vision, approaches and challenges. In: Shishkov, B. (ed.) Business Modeling and Software Design. pp. 169–181. Springer International Publishing, Cham (2020)
27. van den Heuvel, W.J., Tamburri, D.A., D'Amici, D., Izzo, F., Potten, S.: Chainops for smart contract-based distributed applications. In: Shishkov, B. (ed.) Business Modeling and Software Design. pp. 374–383. Springer International Publishing, Cham (2021)
28. Hugues, J., Hristosov, A., Hudak, J.J., Yankel, J.: Twinops - devops meets model-based engineering and digital twins for the engineering of cps. In: Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings. MODELS '20, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3417990.3421446, https://doi.org/10.1145/3417990.3421446
29. Kim, G., Humble, J., Debois, P., Willis, J.: The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations. ITpro collection, IT Revolution Press (2016), https://books.google.pt/books?id=ui8hDgAAQBAJ
30. Kirchhof, J.C., Rumpe, B., Schmalzing, D., Wortmann, A.: Montithings: Model-driven development and deployment of reliable iot applications. Journal of Systems and Software **183**, 111087 (2022). https://doi.org/https://doi.org/10.1016/j.jss.2021.111087, https://www.sciencedirect.com/science/article/pii/S0164121221001849
31. Kumar, R., Goyal, R.: Modeling continuous security: A conceptual model for automated devsecops using open-source software over cloud (adoc). Computers & Security **97**, 101967 (2020). https://doi.org/https://doi.org/10.1016/j.cose.2020.101967, https://www.sciencedirect.com/science/article/pii/S0167404820302406
32. Leite, L., Rocha, C., Kon, F., Milojicic, D., Meirelles, P.: A survey of devops concepts and challenges. ACM Comput. Surv. **52**(6) (nov 2019). https://doi.org/10.1145/3359981, https://doi.org/10.1145/3359981
33. Leppänen, T., Honkaranta, A., Costin, A.: Trends for the devops security. a systematic literature review. In: Shishkov, B. (ed.) Business Modeling and Software Design. pp. 200–217. Springer International Publishing, Cham (2022)
34. Loniewski, G., Insfran, E., Abrahão, S.: A systematic review of the use of requirements engineering techniques in model-driven development. In: Petriu, D.C., Rouquette, N., Haugen, Ø. (eds.) Model Driven Engineering Languages and Systems. pp. 213–227. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)

35. Mashkoor, A., Egyed, A., Wille, R.: Model-driven engineering of safety and security systems: A systematic mapping study (2020). https://doi.org/10.48550/ARXIV.2004.08471, https://arxiv.org/abs/2004.08471
36. Melchor, F., Rodriguez-Echeverria, R., Conejero, J.M., Prieto, Á.E., Gutiérrez, J.D.: A model-driven approach for systematic reproducibility and replicability of data science projects. In: Franch, X., Poels, G., Gailly, F., Snoeck, M. (eds.) Advanced Information Systems Engineering. pp. 147–163. Springer International Publishing, Cham (2022)
37. Meyers, B., Gadeyne, K., Oakes, B., Bernaerts, M., Vangheluwe, H., Denil, J.: A model-driven engineering framework to support the functional safety process. In: 2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C). pp. 619–623 (2019). https://doi.org/10.1109/MODELS-C.2019.00094
38. Mohamed, M.A., Challenger, M., Kardas, G.: Applications of model-driven engineering in cyber-physical systems: A systematic mapping study. Journal of Computer Languages **59**, 100972 (2020). https://doi.org/https://doi.org/10.1016/j.cola.2020.100972, https://www.sciencedirect.com/science/article/pii/S2590118420300320
39. Mohamed, M.A., Kardas, G., Challenger, M.: A systematic literature review on model-driven engineering for cyber-physical systems (2021). https://doi.org/10.48550/ARXIV.2103.08644, https://arxiv.org/abs/2103.08644
40. Muñoz, M., Negrete, M., Mejía, J.: Proposal to avoid issues in the devops implementation: A systematic literature review. In: Rocha, Á., Adeli, H., Reis, L.P., Costanzo, S. (eds.) New Knowledge in Information Systems and Technologies. pp. 666–677. Springer International Publishing, Cham (2019)
41. Muzaffar, A.W., Mir, S.R., Anwar, M.W., Ashraf, A.: Application of model driven engineering in cloud computing: A systematic literature review. In: Proceedings of the Second International Conference on Internet of Things, Data and Cloud Computing. ICC '17, Association for Computing Machinery, New York, NY, USA (2017). https://doi.org/10.1145/3018896.3036380, https://doi.org/10.1145/3018896.3036380
42. Raibulet, C., Arcelli Fontana, F., Zanoni, M.: Model-driven reverse engineering approaches: A systematic literature review. IEEE Access **5**, 14516–14542 (2017). https://doi.org/10.1109/ACCESS.2017.2733518
43. Rivera, L.F., Villegas, N.M., Tamura, G., Jiménez, M., Müller, H.A.: Uml-driven automated software deployment. In: Proceedings of the 28th Annual International Conference on Computer Science and Software Engineering. p. 257–268. CASCON '18, IBM Corp., USA (2018)
44. Sánchez-Gordón, M., Colomo-Palacios, R.: Characterizing devops culture: A systematic literature review. In: Stamelos, I., O'Connor, R.V., Rout, T., Dorling, A. (eds.) Software Process Improvement and Capability Determination. pp. 3–15. Springer International Publishing, Cham (2018)
45. Sánchez-Gordón, M., Colomo-Palacios, R.: Security as culture: A systematic literature review of devsecops. In: Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops. p. 266–269. Association for Computing Machinery, New York, NY, USA (2020), https://doi.org/10.1145/3387940.3392233

46. Sandobalin, J., Insfran, E., Abrahao, S.: Towards model-driven infrastructure provisioning for multiple clouds. In: Andersson, B., Johansson, B., Barry, C., Lang, M., Linger, H., Schneider, C. (eds.) Advances in Information Systems Development. pp. 207–225. Springer International Publishing, Cham (2019)
47. Sandobalin, J.: A model-driven approach to continuous delivery of cloud resources. In: Braubach, L., Murillo, J.M., Kaviani, N., Lama, M., Burgueño, L., Moha, N., Oriol, M. (eds.) Service-Oriented Computing – ICSOC 2017 Workshops. pp. 346–351. Springer International Publishing, Cham (2018)
48. Sandobalin, J., Insfran, E., Abrahão, S.: Argon: A model-driven infrastructure provisioning tool. In: 2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C). pp. 738–742 (2019). https://doi.org/10.1109/MODELS-C.2019.00114
49. Schmidt, D.: Guest editor's introduction: Model-driven engineering. Computer **39**(2), 25–31 (2006). https://doi.org/10.1109/MC.2006.58
50. Song, H., Dautov, R., Ferry, N., Solberg, A., Fleurey, F.: Model-based fleet deployment of edge computing applications. In: Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems. p. 132–142. MODELS '20, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3365438.3410951, `https://doi.org/10.1145/3365438.3410951`
51. Song, H., Dautov, R., Ferry, N., Solberg, A., Fleurey, F.: Model-based fleet deployment in the iot–edge–cloud continuum. Software and Systems Modeling **21**(5), 1931–1956 (Oct 2022). https://doi.org/10.1007/s10270-022-01006-z, `https://doi.org/10.1007/s10270-022-01006-z`
52. Tsilionis, K., Wautelet, Y.: A model-driven framework to support strategic agility: Value-added perspective. Information and Software Technology **141**, 106734 (2022). https://doi.org/https://doi.org/10.1016/j.infsof.2021.106734, `https://www.sciencedirect.com/science/article/pii/S0950584921001841`
53. Weerasiri, D., Barukh, M.C., Benatallah, B., Cao, J.: A model-driven framework for interoperable cloud resources management. In: Sheng, Q.Z., Stroulia, E., Tata, S., Bhiri, S. (eds.) Service-Oriented Computing. pp. 186–201. Springer International Publishing, Cham (2016)
54. Weerasiri, D., Barukh, M.C., Benatallah, B., Jian, C.: Cloudmap: A visual notation for representing and managing cloud resources. In: Nurcan, S., Soffer, P., Bajec, M., Eder, J. (eds.) Advanced Information Systems Engineering. pp. 427–443. Springer International Publishing, Cham (2016)
55. Wettinger, J., Breitenbücher, U., Kopp, O., Leymann, F.: Streamlining devops automation for cloud applications using tosca as standardized metamodel. Future Generation Computer Systems **56**, 317–332 (2016). https://doi.org/https://doi.org/10.1016/j.future.2015.07.017, `https://www.sciencedirect.com/science/article/pii/S0167739X15002496`
56. Zhu, L., Bass, L., Champlin-Scharff, G.: Devops and its practices. IEEE Software **33**(03), 32–34 (may 2016). https://doi.org/10.1109/MS.2016.81