



DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Information Systems

**Automated documentation of Business
Domain assignments and cloud application
information from an application
development pipeline**

Nicolás Corpancho Villasana





DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Information Systems

**Automatisierte Dokumentation von Business
Domain Zuordnungen und Cloud
Anwendungsinformationen aus einer
Anwendungsentwicklungspipeline**

**Automated documentation of Business Domain
assignments and cloud application information
from an application development pipeline**

Author:	Nicolás Corpancho Villasana
Supervisor:	Prof. Dr. Florian Matthes
Advisor:	Martin Kleehaus
Submission Date:	15.02.2019



I confirm that this master's thesis in information systems is my own work and I have documented all sources and material used.

Munich, 15.02.2019

Nicolás Corpancho Villasana

Acknowledgments

First of all I want to thank my advisor Martin Kleehaus for putting a lot of time and effort supporting me during the last months. His constructive feedback was of enormous value for the fulfillment of this thesis.

Furthermore, I want to thank Dr. Matheus Hauder and Martin Leichter for advising me on the industrial part of this work. Your support and feedback were also of immense value.

I would like to thank Prof. Dr. Florian Matthes for giving me the opportunity to write my research topic at his chair and for his feedback.

In addition, I want to thank my team at the industry partner for making a collaboration possible, for participating in interviews and for the help during the last months.

Last but not least I would like to thank my family and friends for supporting me during the time of this work.

Abstract

Enterprise Architecture Management (EAM) has emerged to be a settled instrument to increase data quality, reduce IT costs and to reduce the error-prone effects of the process of the Enterprise Architecture information collection.

Although EAM has gained importance over more than a decade to improve the alignment of business with IT and transmit a holistic view of the entire organization along with its application landscape, this discipline has been shown to be more complex than expected.

One of the major challenges for an organization is the documentation of Enterprise Architecture (EA) information. Most of the EA documentation is still collected manually. The high number of applications within the application landscape coupled with data redundancy and the inconsistent data leads to a high complexity of EA documentation. This produces a time consuming and highly error-rated documentation and collection process of EA information. Maintaining and gathering information for the EA is (as well) a very expensive task. The lack of governance is also a major challenge for EA itself. The absence of standardized tools integration, continuous delivery and build pipelines, etc., adds to the complexity of a clear and transparent EA documentation.

Since one of the goals of EAM is to create a completely holistic view of the EA, it should integrate internal sources, such as PPM tool and external sources, such as cloud service providers.

This thesis will focus on the automated documentation of cloud applications information from an application development pipeline, and the business domain assignments based on the hypothesis that the automated documentation of EA cloud applications leads to a reduction of IT costs and effort, while increasing the data quality of EA information and data.

In order to enhance a holistic view, the EA Tool can be enriched with dynamic data to enable a continuous process of monitoring application performance and infrastructure since the already available documented EA information is mostly static data coming from EA data sources.

Keywords: Enterprise Architecture Management, automated Enterprise Architecture Documentation, Business Domains, Application Development Pipeline, Cloud Application Information, Application Performance Monitoring

Contents

Acknowledgments	iii
Abstract	iv
List of Figures	1
List of Tables	3
List of Abbreviations	4
1. Introduction	6
1.1. Motivation	6
1.2. Research Questions	7
1.3. Approach	8
2. Foundations	12
2.1. Enterprise Architecture Management	12
2.1.1. EAM Definition	12
2.1.2. EAM Stakeholders	13
2.1.3. EAM Use Cases	15
2.2. Enterprise Architecture Documentation	16
2.2.1. Definition	17
2.2.2. Information Sources	17
2.2.3. EA Documentation Challenges	19
2.3. Technology trends influencing EA	21
2.3.1. Agile development and Continuous Practices	21
2.3.2. Cloud Computing	22
2.3.3. Monitoring	23
3. Related work	25
3.1. Literature approaches	26
3.2. Derivation of requirements	28
3.3. Summary	30

4. Solution approach of an automated EAD process	32
4.1. Solution architecture	32
4.2. Approach for an automated documentation	34
4.2.1. Requirements	34
4.2.2. Process description	37
4.3. Approach in usage scenario	39
4.3.1. Tool selection	39
4.3.2. Fulfillment of requirements	40
4.3.3. Scenario process description	40
5. Prototype implementation	46
5.1. Motivation	46
5.2. Technical specification	47
5.3. Main views	47
5.3.1. Overview	47
5.3.2. Detailed View	48
5.3.3. Visualizations View	55
5.4. Component diagrams	56
5.4.1. Components communication	59
5.4.2. Web component	59
5.4.3. Server component	61
5.5. Class diagram	64
6. Evaluation	66
6.1. Case Study Design	66
6.1.1. Case study objectives	66
6.1.2. Case study definition	66
6.1.3. Case study methodology	67
6.2. Challenges influencing the IT landscape	68
6.2.1. External challenges	68
6.2.2. Internal challenges	70
6.3. AS-IS IT landscape	73
6.3.1. Current EA documentation	74
6.4. Target IT landscape	77
6.5. Derived requirements	79
6.6. Case study evaluation	80
6.6.1. Evaluation goal and methodology	80
6.6.2. Approach evaluation	81
6.6.3. Tool evaluation	90

Contents

6.6.4. Evaluation summary	98
7. Limitations	99
7.1. Technical challenges	99
7.2. Governance challenges	99
8. Conclusion	100
8.1. Summary	100
8.2. Future work	101
A. Appendix	102
A.1. Evaluation questionnaire	102
Bibliography	109

List of Figures

1.1. Research approach	11
2.1. Different stakeholder groups with interest in EAM [22]	13
4.1. Solution architecture	33
4.2. Application development process	35
4.3. Mapping of the the project structure	36
4.4. Alignment of tools and Archimate	40
4.5. Automated EAD process	41
5.1. High level architecture of the prototype	47
5.2. Overview of the artifacts	49
5.3. Filtertree in overview	50
5.4. General section in Detailed View of artifact	51
5.5. Jira-Monitoring section in Detailed View of artifact	53
5.6. Github-Monitoring section in Detailed View of artifact	54
5.7. Jenkins-Monitoring section in Detailed View of artifact	54
5.8. Actions section in Detailed View of artifact	55
5.9. Communications and business domain assignment diagram	57
5.10. Adjacency matrix	58
5.11. Adjacency matrix cell information box	59
5.12. Components communication	60
5.13. UML component diagram of the web component	62
5.14. UML component diagram of the server component	63
5.15. Class diagram of server response	65
6.1. AS-IS IT landscape	73
6.2. Build process	76
6.3. Deployment process	76
6.4. Target IT landscape	77
6.5. Question 2.3: Do you use any of the mentioned information sources to retrieve EA relevant information?	83

List of Figures

6.6. Question 2.4: Do you think this information sources could contain relevant EA information?	84
6.7. Comparison of approaches	86
6.8. Question 2.8: Imposing the team to incorporate a pipeline-script in the repository is easy to establish	87
6.9. Question 2.9: Imposing the team to use a predefined toolchain for the application development is easy to establish	88
6.10. Solution architecture of the prototype in the current IT landscape of the enterprise	90
6.11. Question 4.4: The following information displayed in the general section of the detailed view of an application/service is NOT useful	91
6.12. Question 4.14: The following information in the Jira monitoring section is useful	94
6.13. Additional implemented section: Governance monitoring section	98

List of Tables

3.1. Automated EAD requirements derived from literature approaches . . .	28
3.2. Data challenges in the integration of different information sources . . .	29
3.3. Topics covered by the literature approaches	31
6.1. Automated EAD requirements derived from the case study	79
6.2. Interviewed industry partners evaluation	81
6.3. Question: Who collects the EA information of a new developed applica- tion/service for the EA Tool?	82
6.4. Automated EAD requirements derived from the case study	95

List of Abbreviations

AM	Application Monitoring
API	Application Programming Interface
APM	Application Performance Monitoring
BCM	Business Continuity Management
BIA	Business Impact Analysis
BRM	Binary Repository Manager
CC	Cloud Computing
CD	Continuous Delivery
CI	Continuous Integration
CMDB	Configuration Management Database
EA	Enterprise Architecture
EAD	Enterprise Architecture Documentation
EAM	Enterprise Architecture Management
EAMPC	Enterprise Architecture Management Pattern Catalogue
ESB	Enterprise Service Bus
IT	Information Technology
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
PaaS	Platform as a Service

List of Tables

PPM	Project Portfolio Management
REST	Representational State Transfer
SLA	Service Level Agreement
SaaS	Software as a Service
VCS	Version Control Service

1. Introduction

1.1. Motivation

Companies operate in a dynamic marketplace defined by fast-changing technologies, shortened product life cycles and increasing specialization and competition in global value chains. Due to these changes organizations experience a culture shift that encourages collaboration for improving the quality of software applications while being able to develop them more quickly and reliably. The ability to adapt to the changes has become fundamental for companies to have an advantage over the competitors. However, when organizations try to adapt to create an advantage it results in a more complex landscape within the enterprise. [1] The landscape becomes more heterogeneous when trying to incorporate new methodologies, technologies and tools. This leads to a landscape of incompatible and costly information systems, business processes and organizational structures. [1] These changes also result in new software development methodologies such as agile practices, DevOps and continuous deployment of containerized applications, which have significant influences on the further development of Enterprise Architectures (EA). These includes a re-prioritization of conflicting goals, such as product-oriented vs. process-oriented IT organizations, continuous evolution of the application landscape vs. long-lived stable products, small microservices vs. large monolithic applications and cloud-based environments vs. on-premise IT infrastructures.

Over the last decade EAM is seen as a strategic advantage. [42] Due to the rapidly changing environments enterprises need to adapt as a competitive factor. [1] EAM has been established as an important instrument for managing the complexity of the IT landscape. EAM improves the alignment of business and IT, creates transparency of the application landscape within the enterprise and reduces the landscape complexity and its costs. [1][42]

In order to achieve reduce the complexity of the landscape, a highly accurate, consistent and uniform EA documentation is needed. However, EA often ends up with a scarce documentation.[42] EA documentation is one of the main problems when it comes to the collection of EA information, since most of the information is collected manually. Today's EA documentation is a very complex process due to the immense application landscape consisting of redundant and inconsistent data. [1] The

collection process is contemplated as very time consuming process and the data quality is insufficient. [24][15][42][26] Most of the organizations have no dedicated process for the collection of EA information which confirms that the lack of governance in EA projects is one of the major challenges since it is difficult to document information for a "plethora of stakeholders". [25] [33] Some organizations have tried to automate the process of EA documentation but the automated process is mostly limited to import manually a file which contains manually collected data/information. Organizations should aim for a direct integration of information sources into EA Tools and target an automated EA documentation of external information sources such as cloud providers.

Some organizations show the attempts of an automated EA documentation using Network Scanners and ESBs but the information concludes to be incomplete and not up-to-date. The use of external information sources such as cloud providers will increase the actuality and data quality of the information delivering structured static and dynamic data enriching the EA Tool. In order to obtain an increased data actuality and data quality changes in cloud-based environments should trigger an update of the EA documentation. This update can be triggered by integrating monitoring tools in the cloud infrastructure or by monitoring changes through the cloud providers API. In 2004, ter Doerst stated: "in 7 years from now, enterprise architecture will be a real-time tool for management and redesign of the enterprise for better performance, flexibility and agility". [46] As shown in the survey of Farwick et al. [15] organizations are far away from using an EA Tool as a real-time tool. This thesis will propose a solution for improving the EA documentation regarding external sources, specially applications running in a cloud-based environment. A common problem in EA is the lack of governance. Therefore this thesis will propose a solution approach and a prototype implementation for an automated EA documentation. The approach will integrate the implemented prototype into the development pipeline to automate the documentation of cloud applications, enriching the documentation by assigning it to business domains and enhance the documentation with runtime data to enable a continuous process of monitoring application performance and infrastructure.

1.2. Research Questions

To support this, the following research questions (RQ) will be answered during this thesis.

RQ1. How to assign the application landscape to business domains? The first research question will demonstrate a solution approach to assign the business domain model to the application landscape.

RQ2. How to obtain EA relevant information from the runtime behaviour of

cloud-based environments? The goal of this research question is to identify what the possibilities are to obtain runtime information of applications running in a cloud-based environments and what information is relevant for EAM.

RQ3. How to automate the assignment process with an integrated toolchain? The third research question will describe a process integrating the most common tools used within a company to automate the business assignments and the enterprise architecture documentation process.

RQ4. How does a prototype implementation of the automated EA documentation process of cloud applications look? To demonstrate that the automated documentation of applications running in cloud-based environments is possible regarding the toolchain described in the question above (RQ3) this research question will describe a prototype implementation of the automated documentation process of cloud applications. The prototype implementation based on an open source project is an application and service inventory containing the EA relevant information and metadata of these. One of the technology trends mentioned before is Cloud migration. This reduces infrastructure and maintenance costs, reduces transparency but increases complexity of EA documentation due to the higher number of applications. The growing number of applications is therefore a challenge. To increase the independence of the teams and increase the reusability of these, the open source project was expanded to answer questions like: Which service runs where? Which domain does it belong to? What does it do? Who is responsible for that?

1.3. Approach

This thesis is composed by a literature review and a case study of the solution proposal. The literature review is divided into 4 parts, the same as the case study phase. Figure 1.1 gives a clear overview of the research approach.

First, the scope of research of this work will be described resulting in a set of research questions. The second part of the literature review is the topic conceptualization. The aim is to get an overview of EA and how data is collected in organizations and what technology trends affect the current documentation process. For the literature review (scope of research, topic conceptualization, literature search and literature analysis) an investigation according to the guidelines and the process model of Webster and Watson (2002) was carried out. [52] The research relates to different terms or areas found in the second part: topic conceptualization. First, different terms have been identified to cover different aspects of each chapter. Among other things, a search with a number of combinations of the relevant terms was completed for the individual chapters. The databases used for the search provide a variety of publications, which is why a broad

coverage of the terms is possible. The databases used for this work are as follows:

- ACM Digital Library
- Google Scholar
- IEEE Xplore / Electronic Library Online (IEL)
- University Library of the Technical University of Munich

For the search results the following search criteria were used in meaningful combinations:

- Enterprise Architecture
- Enterprise Architecture Management
- Automated EA documentation
- Cloud Computing
- Continuous Delivery/ Integration
- Microservices
- Application Performance Monitoring

In addition, a reverse search and a forward search were performed. This results in a larger total of results. Since the Technical University of Munich provides its students with a number of licenses, it is possible to use scientific research databases. The quality of the search results is ensured by the licenses for the databases and proves to be very helpful.

As noted previously, organizations struggle with the documentation of their current situation due to the immense application landscape consisting of redundant and inconsistent data. Organizations seek to automate that EA documentation process; that is why the focus of the literature will depict the approaches to automate the EA documentation process with the relevant data sources. These documentation processes are named and compared in the third and forth part of the literature review.

The second part of this work will suggest a new approach based on the approaches found in the literature. A new solution approach is introduced due to current research endeavours that lack in integrating cloud-based environments such as Platform as a Service (PaaS) and Software as a Service (SaaS) for automated EA documentation.

The solution approach will include the technology trends that presently influence the EAM such as agile development (continuous delivery and integration), cloud aspects and decomposition of legacy systems into application components or microservices. Regarding the Enterprise Architecture Documentation (EAD), the advantages and the challenges are depicted. Once the solution approach has been developed it has to be analyzed and evaluated in a productive environment. For the development and evaluation phase of the solution approach an insurance company was examined regarding its process of EA documentation. The implementation phase of this work will include Application Performance Monitoring (APM) to increase the data actuality of the documentation. Performance data is gathered during the EA documentation process and a set of metrics is generated based on the performance data. The performance of the application is measured using the metrics to conclude operation and strategic decisions for the EAM. It follows that extending the EAD with application monitoring metrics allows EAM to derive new information. The analysis of this information is applicable to more use cases such as storage and power monitoring and even running costs calculation of cloud applications based on resources consumption. [41] Therefore the automated EA documentation process will combine static EA information with dynamic data from APM.

The proposed solution approach for automating Enterprise Architecture Documentation by investigating major technology trends that influence EAM, how they affect EAM and how an automated EAD process can lead to a more consistent and up-to-date documentation of the cloud infrastructure.

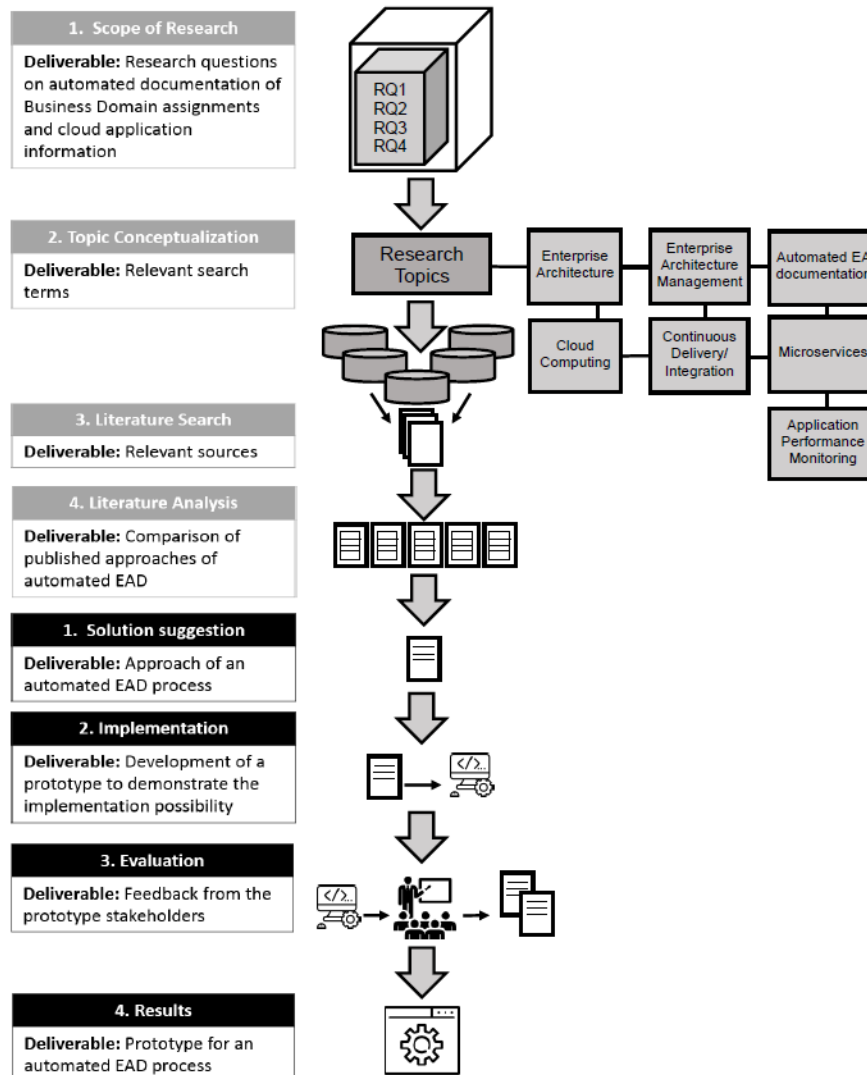


Figure 1.1.: Research approach

2. Foundations

2.1. Enterprise Architecture Management

This chapter provides a theoretical overview of the discipline Enterprise Architecture Management. The most important concepts, related fields and their challenges, related to EA are described in this chapter.

The aim is to describe the EAD process of organizations. This section will present the specific EA information sources and the key problems regarding the EAD. First EAM is defined and the stakeholders are mentioned. Then the different use cases of EAM will be presented. Since documenting the current state of the enterprise IT landscape is one of the uses cases, the different approaches found in the literature for automating that documentation process are presented.

2.1.1. EAM Definition

The term Enterprise Architecture Management is composed of three words: Enterprise, Architecture and Management. According to ANSI/IEEE Std 1471-2000 architecture is defined as ‘the fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution.’ Applying the previous mentioned definition to the context of enterprises, the EA refers to the fundamental organization of an enterprise, embodied in its components (e.g. organizational units, stakeholders, locations, business processes), their relationships to each other, the principles, methods and models that are used in the design and realization of the enterprise’s organizational structure. [7] Additionally, the term management according to Mary Parker Follet refers to ‘the art of getting things done through people’. [6][7][29][49] Thus, the three terms result in the following definition: EAM is promoted as an instrument to improve the alignment of business and IT, ideally suggesting a common language and a framework across the company to determine which business and technical domains, business processes, information systems and technical building blocks are used conveying a holistic view of the entire organization to realize cost savings potentials, and increase availability and fault tolerance. [24]

2.1.2. EAM Stakeholders

For a successful operation of EAM all relevant stakeholder groups need to be identified and involved. When analyzing which stakeholder groups play a role in the organization's EAM initiative central functions, departments and project organizations, as well as IT and external organizations have to be included. Figure 2.1 shows an overview of the stakeholder groups that typically influence, have an interest or can benefit from EAM. [22]

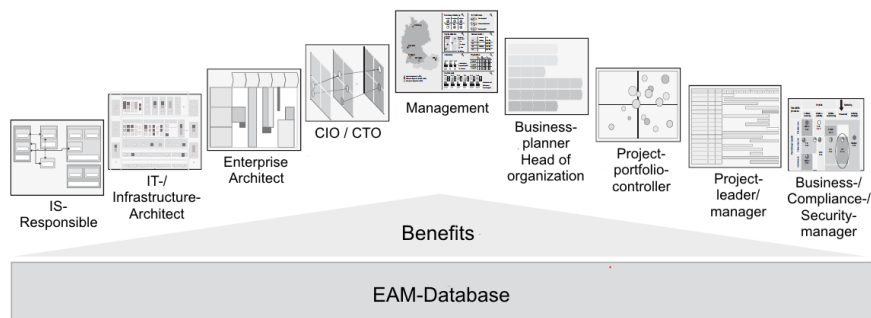


Figure 2.1.: Different stakeholder groups with interest in EAM [22]

The following list shows stakeholder groups that are often involved in EAM initiatives in practice. The structuring is based on the TOGAF categories: [47]

- Corporate Functions
- End-User Organization
- Project Organization
- System Operations
- External

Not all categories are explained in detail. Only some stakeholder groups, relevant for further sections of this work will be described briefly. These are named below and their tasks and EAM concerns are also explained. Some category mappings may vary by company. For example, enterprise architects are assigned to the stakeholder category IT. [22][47]

Corporate Functions

CIO (CTO): How are we performing on delivering our strategic goals? The CIO (CTO) covers the strategic corporate planning and definition of long-term target and

framework requirements as well as planning and control systems and the corporate organization. The benefits of an EAM initiative is to ensure an optimization of the day-to-day business for the CIO (CTO) gathering a cross-company information report from the EAM database to provide the implementation of the corporate/enterprise goals. [22]

End-User Organization

Business manager(including head of business units or areas and department heads): responsible for increasing the business IT alignment and further development of business and business architecture. Business managers endeavour to discover and remove technical redundancies and divisional differences in business processes and business capabilities and their IT support. [22]

Project Organization

Project leader or managers: Responsible for the operative planning and control of a project. The expected benefits are reduced project preparation and input for project execution as well as input for operational planning and control of the project. In addition to project managers, other stakeholder groups of a project organization may also benefit from EAM results. Examples include business analysts, software architects or solution architects. Solution architects often ensure the proper implementation of the development planning in projects through collaboration or review. [22]

System Operations

Enterprise Architects: Mainly responsible for implementing and designing and EAM initiative. Also, they are in charge of the implementation of technical standards and principles as well as their use in the information system landscape and in the operational infrastructure and the operational infrastructure development and the provision of SLAs at the operational infrastructure level. [22]

Externals

This category includes partners and suppliers, such as outsourcing service providers. The application of EAM to this category is the improvement of technical standards and specifications for the target development as input and framework conditions for services. The analysis of dependencies and effects of changes and the fulfillment of SLAs are also covered. [22]

2.1.3. EAM Use Cases

In regard to the discipline EAM, the use case types can be divided into two categories: operational and strategic use cases. This section will only list some key uses cases.

Operational EAM use cases

Operational use cases aim to support the current business cost-effectively and reliably with the help of IT, while continuously improving the IT support. The main challenges are: cost reduction (to reduce complexity of the IT landscape), the optimization of day-to-day business, and identify risks. [22]

As mentioned above only key uses cases are named in this section. Here is a list of some operational use cases:

Meta model as a common language: Before gathering the EA data about the current state, a meta model of the architecture has to be defined; this describes the elements and relationships needed to constitute the EA meta model in order to enable a common language across the organization. Collecting information from different layers requires the involvement of a multitude of stakeholders, e.g. business process owners, project managers, business architects, etc. Although working for the same organization, the definitions used by these stakeholders differ widely. This communication problem is often referred to as the communication gap between business and IT. [44] This gap restricts adequate communication and collaboration in the EA management process. [7] There are often more interactions between IT and business, as expected. To overcome this gap an EAM common language is needed. The resulting metamodel is used as the reference in the dialogue with all parties. [22]

EA Documentation: In order to enable EAM and its life cycle, the current (as-is) state of the EA has to be documented. The different elements and layers covering business, organizational and infrastructure aspects have to be enclosed to provide a holistic view on the enterprise. The information gathered about the current situation, enables the planning for future states. When complementing the current and planned states of the EA an ideal target (to-be) state should be envisioned, which can be derived from the long-term vision of the enterprise. [7] This use case will be described in detail in the section 2.2.

Standardization and homogenization: This is the definition of technical standards with the monitoring of compliance and promotion of implementation. The goal of this use case is to reduce the IT complexity supporting the re-usability of proven technical building blocks. This increases the technical quality. The standardization and homogenization of technical standards results in a continuous cost reduction through the use of economies of scale and the bundling and reduction of the different know-how

expertise. [22]

Project portfolio management: This involves the creation of transparency about applications needed to optimize the support of business through IT, and the planning of the roadmap for target architecture. This ensures the implementation of planned future development, IT consolidation and compliance in planning, prioritization, and overall governance as well as monitoring of the project portfolio. Through targeted management of project resources costs and also the IT complexity can be reduced. [22]

Demand Management: This is the planning and controlling the flow of strategic and operational business requirements for implementation through demand management. Demand Management translates the requirements between business and IT. As a bottleneck for the planning and control of the implementations of the business, efficiency ensures that the technical goals and business requirements are adequately implemented.

Strategic EAM use cases

Strategic use cases intend to systematically improve the development and strategic alignment of the different responsible factors. The main challenges are the setting of specifications and guarantees of compliance (EA Governance) and the progression of Business-Innovation and -Transformation.

IS-Portfolio management: Identification of the potentials for the optimization in the IT landscape through the analysis of the architecture. One of the goals is to reduce the IT complexity through continuous IT consolidation and thus create a sustainable reduction of IT costs. Another goal is to increase business IT alignment by evaluating the contribution of the IT to the business.

Business-Transformation: Standardization and homogenization of the future IT landscape concerning the implementation of enterprise and IT strategy. The confidence in decisions can be ensured through the analysis of the dependencies and the effects of the IT landscape in relation to the business landscape.

According to the ANSI/IEEE Std 1471-2000 the main tasks of EAM are documentation (operational), communication (operational), and analysis of architectures (strategic). The following section will give a general outline of the enterprise architecture documentation.

2.2. Enterprise Architecture Documentation

EAM begins with the documentation of the current state according to the defined EA metamodel to derive future plans for an improved EA. Organizations endeavour with the documentation of the current state due to the complexity of enterprise architecture.

This section will provide an overview of the enterprise architecture documentation process.

2.2.1. Definition

One of the core products of EAM is to document the current implementation of business processes, IT systems, and infrastructure. [30] The documentation needs to cover the current and complete view of the as-is IT landscape with the right degree of detail. [30][50]

2.2.2. Information Sources

A survey conducted by Farwick et al. [15] presents the major information sources for an EAD. This survey was done to provide a comprehensive analysis of possible information sources and their appropriateness for EA since research activities still seek to automate the data collection process.

The information sources correspond to the tools used by the participants of the survey. In addition, the participants also expressed their concerns regarding the EA relevant data delivered by the tool in respect of the following data properties in the EA context: actuality, completeness, correctness and granularity. [15]

Network Scanners and Monitors

The most common potential information source for an automated EAD are Network Scanners and Monitors. Operations teams often use these sources to monitor the infrastructure and its performance. As analyzed in the survey more than 60 percent of all organizations use Network Scanners. Relevant data for network scanners are servers, applications and databases. However, the data gathered from this sources is too granular. The data can then not be mapped to the EA meta model. Nonetheless the correctness and the actuality of the data are considered positive. Since network scanners cannot cover all technologies the quality of the collected data is worst of all information sources.

Configuration Management Database

Configuration Management Databases (CMDBs) are databases that store relevant information about hardware instances used in an organization's IT services and the relationships between those instances and the related incidents at an operational level. A CMDB should contain data about server, database and application instances. This data is mostly collected and maintained manually. Same as with the network scanners,

the data is too granular and organizations often use CMDBs for a strategic planning of EA since they provide all instances within the organization. [15]

Project Portfolio Management tools

Projects are often modeled as a part of the EA metamodel. Project Portfolio Management (PPM) tools can trigger the EA maintenance process since a PPM tool often contains information about the start and end date, budget information and artifacts affected by the project. This data should be integrated with the EA tool since the aspect of project changes is a central aspect of EAM.[15]

The actuality and completeness of the data is in most cases maintained regularly but it depends on project management of the organizations. On the other hand it is difficult to decide which project is actually architecture relevant so the granularity is affected by that.

Enterprise Service Bus

There is no common definition for Enterprise Service Buses (ESB) but in most cases ESBs act like a layer on top of the applications allowing the communication between them. The messages sent by the applications can be routed through the ESB which makes the communication possible. The transformation of the messages facilitates the connection of different applications to use their own native message formats. This layer enables the integration of third party and legacy systems. This leads to a total dependency of the IT landscape. [36] The analysis and optimization of dependencies between applications is also a central aspect of the EAM discipline. This is the reason why the actuality and correctness of the data quality attributes received very positive evaluations. However, the integration of an ESB and transformation and mapping of the communication between the applications is very challenging. This affected the answers about the data granularity. [15]

Change Management Tool

Change Management Tools are used to improve the procedure of implementing changes in the IT-landscape. Frequently the tools are maintained manually and thus do not cover all changes. This is reflected in the answers for data actuality, correctness and completeness. The negative outcome for the granularity also implies that the data is difficult to map to EA metamodel. Nevertheless Change Management tools can trigger a manual action to keep the EA tool up-to-date.

License Management Tools

License Management Tools provide an overview of acquired software licenses. The information provided by the tool is the number of installations, number of users, costs, acquisition date and type and duration of licenses. The information was seen as EA relevant. Actuality, completeness, and correctness had a good outcome, whereas granularity has a moderated outcome. A large number of participants of the survey could not value the data quality attribute of a license management tool. This indicates that EA stakeholders do not always access these tools and therefore the integration depends on the usage.

Excel Import/ File Import

Many of the participants of the survey pointed out that the import of file, specially importing excel-sheets is a potential information sources since many of the organizations still rely on the usage of Microsoft Excel sheets for keeping data. [15]

External Sources: Cloud Services

The usage of cloud services was mentioned frequently for collecting infrastructure data. [14] Apart from documenting traditional hardware, cloud infrastructure information needs to be collected in an EA tool. Cloud-based environments are much more volatile than traditional environments. Therefore, it is important to integrate EA tools with cloud infrastructure to enable tracking of changes occurring in the cloud. [14]

2.2.3. EA Documentation Challenges

Nowadays the documentation process of the EA is performed in most of the organizations manually. This leads to many challenges:

Modeling

As mentioned before, a meta model of the architecture has to be defined before documenting the current architecture. The model will be used as a common language between the different stakeholders. However, the model should depict the alignment between the business and the IT in way that it can be understood by the stakeholders. This is not often a very easy task.

Stakeholders

The documentation of the current architecture should cover the different perspectives of the stakeholders. Often the management level has difficulties deriving meaningful metrics as a result of the return on investment. This is why the management level does not often support EA and its documentation since it is misunderstood as a project and it is not seen as a process to improve the IT landscape.

Tooling

An important challenge regarding EAD is the tool selection. Most of the tools do not offer enough support to integrate other information sources within the company. The lack of integration leads to a very expensive process of integrating the existing tools for a broad EAD. Another problem regarding the tool is the maintenance of the EA model. Most of the tools impede an easy adaptation of their metamodels.

Data challenges

As mentioned already several times, most organizations collect EA data manually. This is the biggest challenge regarding EAD. Due to the problems integrating the information sources, the amount of information that needs to be collected is immense. The data sources contain information on different granularity levels and the transformation of this data is a complicated process. Collecting EA data manually is a very time consuming and therefore an expensive task. Since the collection process is performed manually the odds of producing high error rated data is very high. This concludes with inconsistent and redundant data. [8]

EA Governance

The lack of governance is also a very important aspect when documenting the enterprise architecture. In most organizations the absence of governance is a major issue since there are no governance guidelines and principles respecting EAD.

Fast changing environment

The fast changing environment of an organization is an influential factor. The infrastructure and the technologies vary rapidly and affects the organization when it comes to time to market. How can a enterprise make use of the developing market to improve its enterprise architecture and productivity to increase its performance and promote the time to market of services and products. [30][33][42]

The following section will explain in detail which current trends have an impact on EA.

2.3. Technology trends influencing EA

This section will describe the main technology trends that influence EA. The first technology trend presented is agile development and continuous practices. Later the modularization of systems is being explained. Subsequently cloud computing is described and ultimately monitoring of applications is presented.

These technology trends have nowadays an impact on EAM. How to integrate these to an automated EAD is explained in the next section.

2.3.1. Agile development and Continuous Practices

During the last decade some development methodologies have emerged and gained an important role for enterprises. We can sum the two following definitions to the term "continuous practices". The combination of agile development and EAM was research by Canat et al. [9] and is possible. However, the main challenge is the lack of communication between enterprise architects and developers and that should increase the collaboration. [9]

Agile development

"Agile methodologies emphasize rapid and flexible development. The main characteristics of agile development are short releases, flexibility, and minimal documentation". [21]

Continuous Delivery Definition

"Continuous Delivery (CD) is a software engineering discipline in which teams keep producing valuable software incrementally in short cycles and ensure that the software can be reliably released at any time." [11]

Continuous Integration Definition

"Continuous Integration (CI) is a widely established development practice in software development industry, in which members of a team integrate and merge development work (e.g., code) frequently, for example multiple times per day. CI enables software companies to have shorter and frequent release cycle, improve software quality, and

increase their teams' productivity. This practice includes automated software building and testing."[45][11]

Impact of agile development and continuous practices on EA

EAM follows a top-down approach derived from the goals and strategy of the enterprise. [23] However, agile development is used on a project-level. This level is more granular since the project is decomposed into requirements. This is due to the short-term planning of the project increasing the dynamic of the project being able to change requirements faster and enabling a quicker time-to-market. Therefore it leads to shorter life cycles. This does not mean that due to the short amount of time between these cycles agile teams do not document. On the contrary, teams need to document their results in detail since its not always ensured that the same team will still work on the projects.

The documentation from agile teams is too granular for EAM. Integrating these two perspectives would lead to a more dynamic EA since it would focus more on the collaboration with the agile development of a project and would benefit of the resulting documentation of the agile teams.

EAM would not be perceived as an 'ivory tower' while working together with the agile teams and it would take the advantage of cooperating with the teams to how these projects influence the To-Be IT landscape and if these projects ensure follow IT Governance rules, guidelines and compliance. [23]

2.3.2. Cloud Computing

Definition

"Cloud computing is a model for enabling universal, on-demand and convenient network access to a shared pool of configurable computing resources (e.g., servers, applications, storage, networks and services) that can be quickly provisioned and released with little to no management effort or service provider interaction". [27][3]

Impact of cloud computing on EA

Cloud computing is influencing the IT environment. It is easy to deploy applications to cloud-based environments. The requirements and the needed resources can be easily adapted to the user. This adaptation of the infrastructure enables reduced costs and the need for investment in computing infrastructure. This is why organizations are increasing the investments to migrated their infrastructure to cloud service providers (CSP). Moving the infrastructure concludes to a significant cost reduction of computing

resources. Not having to maintain these resources allows the organizations to focus on their core competences. [27]

One of the reasons for moving to the cloud is to reduce the total cost of ownership and replacing legacy systems. [27] In addition to this reason the continuous growth of the agile development has aroused another trend which is coupled to cloud computing. The importance of small teams in the context of agile development and the increasing significance to reduce time-to-market has made emerged the concept of microservices.

"A microservices architecture is a cloud-native architecture that aims to realize software systems as a package of small services. Each service is independently deployable on a potentially different platform and technological stack. It can run in its own process while communicating through lightweight mechanisms such as RESTful or RPC-based APIs." [4]

As mentioned before migrating legacy systems to the cloud is one of the main reasons. However, the migration of monolithic architecture means changing the architecture to a microservice architecture. The main problem of a microservice architecture is the very large number of small structures running on a cloud-based environment which leads to an increased complexity of the IT landscape. The high degree of heterogeneity of these structures also increments the complexity. On the other hand, the migration produces many advantages. As already mentioned the main advantage of having a microservice architecture the ability to reduce the time-to-market and also the adaptability to change the technology stack of small structures to avoid technological lock-ins. [4]

Another advantage of merging agile development with the trend of microservices is the ability to enable on-demand deployment integrating CD and/or CI to the cloud-based environment. This enables the possibility to detect failures and to receive feedback during the deployment process with the practice of continuous monitoring (CM). [4]

Establishing a connection between cloud computing and EAM would solve the time consuming and manual task of capturing EA relevant information and would increase the actuality of the data.[14]

2.3.3. Monitoring

As mentioned above the complexity of the IT landscape increases due to the challenges named. This raises the need for monitoring and analyzing tool. Monitoring the performance of applications affects in many ways. Application Monitoring (AM) tools, such as Dynatrace, brings many benefits to the enterprises. The AM tools not only deliver metrics about the system health, the consumption of the computing resources of the platform, they also deliver many other thousands of metrics such as response time, http calls, failure rates, etc.[41]

Impact on EA

Extending the EA context with application monitoring metrics allows EAM to derive new information. Frank et al. [20] recommended the synchronization of performance indicators at runtime level or indicator systems with the EAM view. [20][14] Retrieving metrics such as open connections, latency and the number of cores in the host CPUs led to congestion and decreased the performance of the system. Applying this knowledge, the system can be configured to perform better. [41] The analysis of the measurements of AM tools is applicable to more use cases such as on-line advertisement marketing, click stream storage, and power monitoring. [41] From a IT-governance perspective it is useful to monitor applications to realize and control changes in the cloud. For this a view of the applications running on cloud-based environments of the enterprise is needed. Thus, the enterprise architects can supervise the laws and regulations that need to be applied to these environments. Regulations regarding the storage of private data in a public cloud is prohibited by the European Union. For this an application inventory of the individual cloud is required because this regulation forbids the export of the private data to non-EU countries. There cloud applications need to be compliant with this regulation called EU Data Protection Directive. The negligence of this regulation can result in a significant financial damage and the growth of uncertainty from the customers.[14]

This work will present an automated process of documenting EA integrating the depicted technology trends influencing EAM. The integration of the technology trends facilitates the EAD and enables the automation of it. The purpose of the automated EAD is to derive improved EAM use cases.

3. Related work

This work will target the automation of the documentation use case trying to improve the strategic analysis with the help of application monitoring as a result of the automated EAD. First this chapter will present different solution approaches related to automated EAD. Then the different proposed solutions will be compared and evaluated.

Hauder et al. [24] describes the EA documentation process as very time consuming, error-prone and a process that requires a lot of manual effort. Inspired by these challenges Hauder examines the further challenges for an automated EAD investigating model transformations from various information sources, conducted a survey among 123 EA practitioners and incorporated a literature review. The discovered problems among EAD were grouped into four categories. The main challenge of the first category data is the appropriate selection of a relevant EA information source and the data quality retrieved. The second category identified is transformation. It reports the problems regarding the alignment and maintenance of data from diverse information sources to a central repository. The third category business and organization deals with the question of the added value of an automated process and what the impact to the organizational structure is. The last category describes the challenges related to the tool.

A survey conducted by Farwick et al. [15] reveals more details about the status quo of EA documentation processes in organizations, which information sources can be used to obtain relevant EA information, what integration problems exist for these information sources and what data quality attributes like data actuality, data completeness, data correctness and the granularity level of the data, can be expected from them. The survey presents a list with the possible information sources and the expected data quality attributes. The survey declares that the gap between the retrieved data and the EA model prevents enterprises for automating the documentation process due to different granularity levels and the difficulties mapping the data to the EA model.

Roth et al. [42] reaffirms the struggle of the organizations regarding the EA documentation process with an empirical evaluation on the application of this process. The survey implies the EAD of 140 organizations to validate the challenges mentioned by Hauder et al. [24] and by Farwick et al. [15] The work derives future research directions from the findings and gives an overview of the currently applied techniques in EAD.

3.1. Literature approaches

Farwick 2010

Farwick et al. [14] proposes in his work that cloud infrastructure should be documented in an EA model to increase the understanding between the cloud infrastructure in relation to other business information systems and finally to the overall business goals. The purpose of the publication is to present an approach how to consolidate runtime information of different information sources such as cloud environments to an EAM view. The work presents a conceptual approach and a prototypical implementation using the open-source cloud infrastructure Eucalyptus as well as the open-source EAM tool Iteraplan. The automated approach of Farwick et al. integrates different information sources in a central model, which updates the model, verifies the information and pushes the new information to the EAM tool. The publication proposes as a future work taking into consideration the integration of other cloud service models like PaaS and SaaS for the usage of generic APIs for standardization purposes. The synchronization problems with other information sources and the development of the central model and the metamodel are presented as future research due to the intensive work of the installation of agents for monitoring the different information sources.

Buschle 2012

The presented approach of Buschle et al. [8] examines a specific Enterprise Service Bus (ESB) in an enterprise for interlinking business applications and processes as information sources. The transformation rules for the data are used to apply an automated EA documentation. The evaluation of the approach was carried out with a productive ESB in an enterprise in the fashion industry.

Holm 2014

Holm et al. [26] recommends using a network scanner for the automated process of data collection for producing EA models based on the IT infrastructure of enterprises. Manual effort is still required to make the models completely valuable. The results were evaluated empirically and demonstrate an accurate outcome with little effort.

Välja 2015

The work of Välja et al. [48] describes how to automate the process of EA modeling. The information is retrieved from different data sources using common data processing methods. The work shows that challenges of manual modeling can be overcome and

data quality issues can be solved. The approach shows that it is possible to create automatically enterprise IT architecture models that are timely and scaleable. [48]

Farwick 2015

The work presents a semi-automated approach for documenting EA with the respective tool. It focuses on the adaptability of the metamodel at runtime, the integration of different EA information sources and the versatility and scalability of the visualizations of the tool. The tool was presented as a prototype in the original article and has progressed to a commercial product called Txture.[16]

Johnson 2016

Johnson et al. [28] introduces an approach to automate the modeling of EA. The modeling process is seen as a probabilistic state estimation problem. Therefore a Dynamic Bayesian Networks is used to solve the estimation problem. The suggested approach proposes extensions to the model of ArchiMate. Using a Dynamic Bayesian Network detecting uncertainties surrounding the IT landscape becomes possible. Filtering the relevant information from the irrelevant remains still a challenge in this approach and it is therefore seen as a topic for further studies.

Landthaler 2018

Landthaler et al. [32] presents a machine-learning based approach for detecting and identifying the ArchiMate metamodel entity "application component" in the IT landscape of the enterprise. The presented approach discovers and classifies binary strings of application executables on target machines. The main challenge is that the binary strings of executables differ depending on the devices. That means that the same binary string is different for the same application version, device type and OS version. Evaluating the data reflected to major problems of this approach: the many-label nature of the classification problem and limited existence of sparse classification of the results. The advantage of this approach is that all executable binaries are discovered independently from their name or installation path. For further evaluation of the approach it is necessary to examine the work on a heterogeneous environment. That means different operations systems and different versions of the same application to improve the classification of the applications based on related functionality.[32]

Bogner 2016

Bogner et al. [5] examines methodologies to integrate the growing amount of small structures like microservices, Internet of Things and mobility services that are emerging in today's IT environments. Micro-granular architectures increase the degree of heterogeneity of enterprise's IT landscapes and thus hinder classical EAM approaches to deal with the diversity and distribution presented in the architecture landscapes. The work enlarge EA methodologies by extending earliest reference metamodels with elements for a more adaptable models and EA-mini-descriptions. The EA-mini-descriptions provide an adaptable metamodel for the microservices and the descriptions can be grouped to form superordinated entities. It also proposes that an EA approach should integrate small structures to enable a holistic view and should be flexible. The main intention of the paper is to identify adaptability issues of microservice architectures and to present a different approach than classical EAM approaches. The development of a prototype and the evaluation and validation of the results in practical use cases is requested as future research.

3.2. Derivation of requirements

In the previous section the different approaches were presented. These cover various aspects of EAD and have different outlooks for future works. From the introduced approaches and their respective problems, requirements can be derived for future solutions regarding automated EAD. The following table presents the derived requirements:

Table 3.1.: Automated EAD requirements derived from literature approaches

Id	Requirement	Source
RL1	Integration of different information sources.	[14][17][5][28][32]
RL2	Dynamic metamodel	[24][15][42][14][8][26][48][5]
RL3	Business added value	[24][42][14]
RL4	Tool support	[24][42][26][17]
RL5	Integration of cloud environments (PaaS and SaaS)	[14][17][5]
RL6	Integration of runtime KPIs	[14][17][5]

RL1: Most of the mentioned works propose the integration of different information sources. The survey conducted by Farwick et al. [15] shows that organizations use different information sources and/or see the information sources as potential EA relevant data sources. The findings are affirmed by the published approaches. Regarding the

data of the various sources, there are several data aspects that need to be covered. The integration presents the following data challenges:

Table 3.2.: Data challenges in the integration of different information sources

Id	Challenge	Source
DC1	Data granularity	[24][15][42][26][48]
DC2	Data actuality	[24][15][42][26][48][28][32][5]
DC3	Data completeness	[24][15][42][26][48]
DC4	Data correctness	[24][15][42][26][48][28][32]

DC1: Data granularity describes detail level of information. This challenge affects the mapping of the different model since the granularity and abstraction level of the models is different. If the data too fine-grained it is not relevant stakeholder of EA.

DC2: Data actuality defines how up to date is the information contained in the information source. To ensure the actuality of the data the EA repository requires an detection of changes. The same applies to the different information sources. The changes in the real world need to trigger a maintenance of the EA relevant data.

DC3 and DC4: Data completeness and Data correctness are further quality attributes of the gathered data. They refers to the amount of relevant data and accuracy covered by the information source.

The above mentioned data challenges lead to an EA model maintenance. As describes by the approaches the underlying model needs to be flexible.

RL2: As mentioned before data is retrieved from numerous information sources. The collected information varies regarding the data content and granularity levels (DC1). Therefore a transformation of the data to the target metamodel is required. The transformation of the data should convert the model of the source to the target model. Duplicate EA elements or attributes should also be identified and removed from the target model (DC4). The various models are also diverse regarding the granularity levels of the models (DC1). This is why the target model for the collected data should be dynamic. This means that it has to be able to allow different granularity levels since the different stakeholder may consider the retrieved information as EA relevant.

RL3: The business added value of an automated EAD is not considered as enough regarding the return of investment. The initial investment required for an automated EAD is to large and requires too much effort. The data owners of the information sources need to be involved in the process of the information collection and need to maintain the imported EA information. This is also seen as a challenge according to the survey conducted by Hauder et al. [24][35][34]

RL4: Tool support is also mentioned as a requirement in the related approaches

for an automated EAD. The main challenge regarding the tooling aspect is that the majority of the tools does not support an integration of different information sources due to the lack of a public API. Therefore, Iteraplan is used in most of the approaches as the EA Tool since it offers a public API to integrate various data sources. [35][34] Another problem regarding the EA Tools is the absence of customizing visualizations for analyzing the collected data. [24][35][34]

RL5: Cloud infrastructures are emerging in many organizations. [5] According to that the integration of cloud environments need to be coupled to EAM. Cloud environments supports an enterprise in many ways. Enterprises can host the applications in a public or a private cloud and these also vary in the service offering. Cloud can be contracted as IaaS, PaaS and/or SaaS. Similar to traditional EA relevant information sources these options of cloud need to be documented. The integration of different cloud environments present the same data challenges as mentioned in table 3.2. Some of the approaches have proposed an installation of agents [32][14] in the underlying infrastructure to retrieve information from installed applications. Considering the usage of PaaS and SaaS as a cloud infrastructure the generic cloud API can make the installation of these agents obsolete. Another advantage of cloud infrastructure is that it also offers the possibility to integrate runtime information of the applications which leads to the next requirement: R6 - Integration of runtime KPIs.

RL6: The integration of runtime KPIs is also proposed in the approaches found during the literature reasearch. [20][14] An enhancement of runtime KPIs improves the actuality of the collected data (DC2). In also increases the data completeness and correctness. [20][14][24]

3.3. Summary

This section will give an overview of the topics covered by the different approaches presented in literature.

Table 3.3 shows that current research endeavours lack in integrating cloud aspects (PaaS and SaaS) with its respecting structures such as microservices for automated EA documentation. [8] Also agile methodologies and continuous delivery and integration are not taken into consideration when it comes to the automation of the EAD. New approaches can be derived from the topics that are not covered and the requirements derived from the already existing approaches.

The approach that will be presented in the next chapter will integrate the EAD process within the application development pipeline to enable a continuous automated EAD. The approach will also include an automated documentation of business domain assignments by extracting the business domains from the PPM tool during the applica-

Table 3.3.: Topics covered by the literature approaches

Year	Author	CD	CC	AM
2012	Hauder et al.			X
2013	Farwick et al.		X	X
2013	Roth et al.			
2010	Farwick et al.		X	
2012	Buschle et al.			
2014	Holm et al.			
2015	Välja et al.			
2016	Farwick et al.	X	X	X
2016	Johnson et al.			
2018	Landthaler et al.			
2016	Bogner et al.	X	X	X

tion development pipeline. The integration of an automated EAD within a continuous delivery pipeline has not been covered in the literature. In addition to that, cloud environments like PaaS and SaaS will be integrated by the approach of this work for an automated EAD of cloud applications.

4. Solution approach of an automated EAD process

This chapter will propose a new automated process for EAD derived from the lack of integrating cloud infrastructure for automated EA documentation in the context of agile methodologies and continuous delivery and integration. The proposed approach was derived from the topics that are not covered and the requirements derived from the already existing approaches in section 3.

The first section 4.1 will give an overview of the solution architecture. The second section 4.2 will describe how to automate the EAD from the application development pipeline and what requirements need to be fulfilled. The last section 4.3 will describe the process in more detail showing the approach in a sample scenario with common tools.

4.1. Solution architecture

Driven by the requirements of the literature and the deficiency of automated EAD approaches for cloud infrastructures, the following solution was developed.

Figure 4.1 shows the solution architecture of the automated EAD solution.

The components that are content of this work are:

- A cloud infrastructure
- A Version control service
- A PPM Tool
- A CD/CI Tool
- A EA Tool

The prototypical implementation for this approach is illustrated in figure 4.1 as "Automated EAD Tool". It is integrated as a middle layer to aggregate the information of the components mentioned above since many of the existing EA Tools require a lot of effort to integrate different information sources. An adaption of the EA metamodel

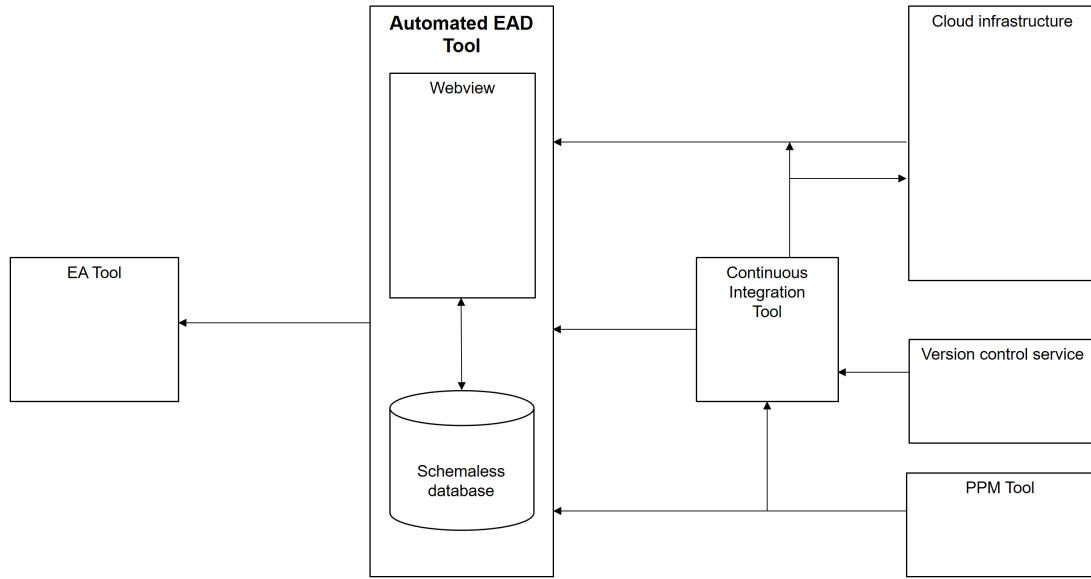


Figure 4.1.: Solution architecture

within the EA tool is still an intensive process. Therefore the automated EAD tool as a middle-layer is useful to aggregate the different collected content.

The main reason for an integration of a PPM Tool is to enable the assignment of the application landscape to the business domains asked in **RQ1**. As a requirement for the assignment process, the product owner and the agile teams need to ensure that the business information like business domains, business subdomains and product assignments are specified in the PPM tool. A possibility to establish these assignments in agile development methodologies is to encourage the agile teams and their product owners to incorporate the business information in their projects. There are several ways to include the business information in a project. However, adding the information to the demands of the project seems effective. The responsible person in agile teams to manage the requirements is the product owner. For that reason the product owner should control that the business domains, business subdomains and product assignments are maintained and updated in the demands of the projects. This information can then be aggregated to the information of the developed application. The aggregation is done during the automated documentation process.

To obtain EA relevant information from the runtime-behavior (**RL6**) of cloud-based environments asked in **RQ2** an integration of the cloud infrastructure is required (**RL5**). The goal of the integration of the cloud infrastructure is to identify how to obtain

runtime information of applications running in a cloud-based environments and what information is relevant for EAM.

CD/CI Tools and VCS can automate build and deployment processes. As described by Chen et al. [11] and by Drews et al. [13] DevOps teams do not document small changes in the application development pipeline. Thus a documentation process can be integrated in the pipeline with no effort to automate the EAD and improve the data quality attributes mentioned in table 3.2 Therefore an integration of the application development pipeline and the according tools are content of this work to automate the documentation and assignment process.

The proposed solution is structured in two main parts:

- A Webview
- A schemaless database with a REST API

The integration of different information sources required in **RL1** and that are of relevance for this approach, are the cloud infrastructure a VCS, a CD/CI tool and the PPM tool. The collection of the relevant information of this sources is achieved due to a REST API layer connected to the database.

Many of the approaches request a dynamic metamodel. A possible solution to a dynamic metamodel are required in **RL2** is the usage of a schemaless database. A schemaless database enables the storage of structured and unstructured data since it does not require a static and predefined schema and allows the integration of different granularity levels (DC1).

An aggregated information consisting of the runtime information and the business domain assignments can be exported from the prototypical implementation to the EA Tool if the tool supports a data integration through an API: **RL4**.

4.2. Approach for an automated documentation

This section introduces an automated EAD process. The focus of this work is the process of an automated EAD from an application development pipeline. The process is described in subsection 4.2.2.

Figure 4.2 shows the application development process with its components and actors. The requirements and the process description are described in the following subsections.

4.2.1. Requirements

To enable an automated documentation the fulfillment of the following requirements are needed:

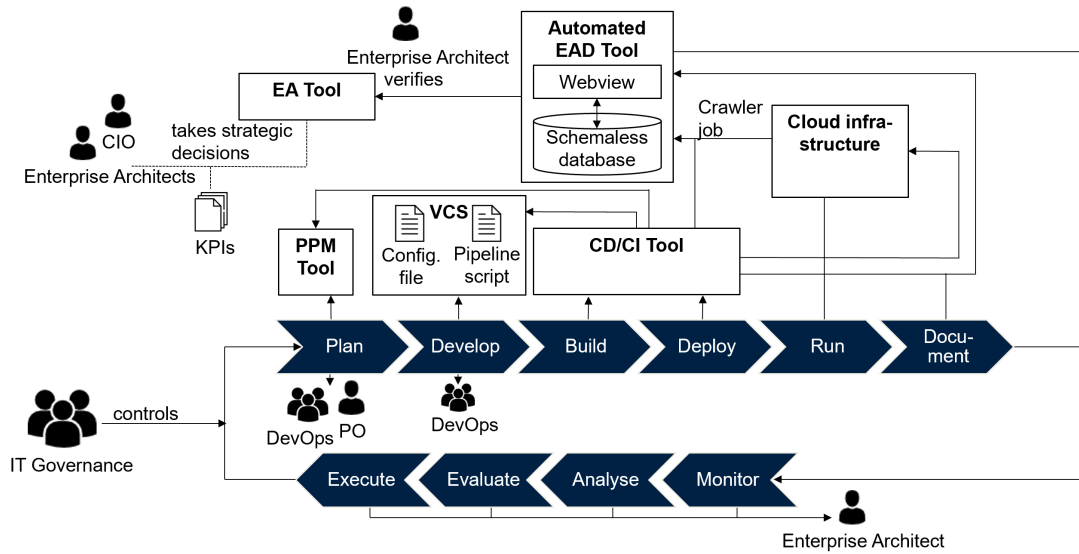


Figure 4.2.: Application development process

- A **predefined project structure** in the PPM tool
- The integration of a **configuration file**
- A **pipeline-script**

Predefined project structure

A predefined project structure in the PPM tool and the fulfillment of the project structure enables the possibility to retrieve automatically the business information of the tool through the API. The proposed structure for a project of this work is that every project should be at the same level as an application. Figure 4.3 shows proposed project structure and the alignment with Archimate definitions.

The definition of an application component according to the Archimate standard is: "An application component is defined as a modular, deployable, and replaceable part of a software system that encapsulates its behavior and data and exposes these through a set of interfaces." [40]

Since the Archimate definition do not cover a definition for a parent application and a child application, the project has to be aligned to a parent application component. This application component as illustrated in figure 4.3 contains different application components. Each project requirement should be assigned to the children application

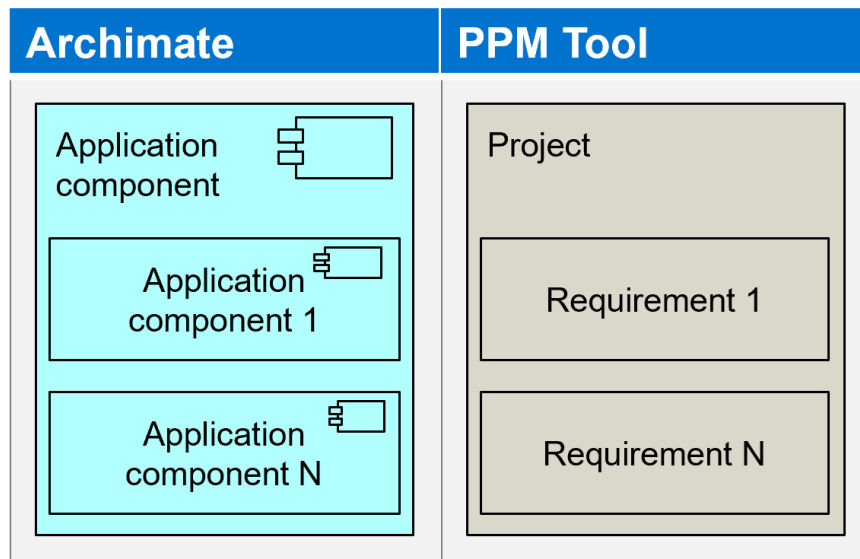


Figure 4.3.: Mapping of the the project structure

components. Depending on the selected PPM tool, the business specific information can be stored at the project level or at the requirements level. In this approach the business information of the PPM tool is added at the requirements level. Each requirement should be assigned to a business domain and if applicable also to a subdomain. The product information also stored at the requirements level corresponds to the parent application component. The application owner would be mapped to the project owner in the PPM tool.

Configuration file

The addition of a configuration file in the repository has different functionalities. The main purpose for including a configuration file is that it contains a link to the PPM tool to retrieve the business information. The other purpose of the configuration file is to enable a federated enterprise architecture. The DevOps team manually maintain the links to other tools in this file. The added links can be stored as attributes of the applications in the metamodel of the tool enabling the linkage of the information enclosed by other tools.

Pipeline-script

A continuous delivery pipeline enables automation of the build and deployment pipeline and thus a documentation process can be integrated in the pipeline with no effort to enable the automation of EAD due to the reason that DevOps teams do not document minor changes. In this manner the data quality attributes mentioned in table 3.2 are automatically improved and documented. A manual separated documentation process is no longer necessary. Therefore the application development pipeline should integrate the documentation process within the continuous delivery pipeline. The idea is to include a similar mandatory script for each repository that enables the standardization of build, deployment and documentation process for every application. The intention of the script is that with small manual effort the DevOps team adapts the script. The adaptation of the script involves mainly the adjustments of authorization credentials. The script describes the flow of an artifact through different stages of the application pipeline. [38]

4.2.2. Process description

The goal of this approach is to document EA relevant information automatically and to aggregate automatically business information to assign the application landscape to business domains (RQ1, RQ3). To ensure this automated aggregation of information the documentation process needs to cover the application development pipeline depicted in figure 4.2.

The application development pipeline starts with the planning of a new software product. Therefore a PPM tool is needed. An automated documentation is only possible if the requirement of the project structure of subsection 4.2.1 is satisfied.

When developing a new product/application the DevOps team must meet the two other requirements. The version control repository has to contain the configuration file with the link to the PPM tool and if possible other tools containing information about the product such as a wiki page for the product. The second requirements is that the pipeline script is included in the repository. This approach purposes a separate repository for each child application component. A deployment of the parent artifact is not needed because the deployment is usually separated into the deployment of small structures that communicate with other small structures. The automated documentation process of this approach is therefore included into the deployment of each application component. [5] Decomposing an application into small structures like microservices and the development process of these lead to a quicker process of delivery to the customer to reduce the cycle time and release risks. [10] Increasing the release frequency also leads to an accelerated time to market. [10][12] Therefore the

CD/CI tool starts the build process from the pipeline script included in the repository of the small structures. Before building the artifact the pipeline script verifies that the configuration file is included into the repository and retrieves the business information for the specified application.

Deployment failures rarely happen therefore the deployment process can be easy automated within the pipeline script. [10]

After the successful deployment the pipeline script makes a call to the cloud infrastructure API to verify that the application is successfully running and to retrieve the runtime-behavior of the application. The API call depends on the cloud infrastructure API. In this case the call is done via the name of the application to retrieve the runtime information.

As acknowledged by the survey conducted by Farwick et al. [15] manual events can trigger an update of the documentation process. The build and deployment process can be used as triggers. Therefore the application development pipeline is used to trigger an automated EAD. [38] The documentation process pushes an aggregation of the collected EA information during the application pipeline to the automated EAD tool. This aggregation contains the business information from the PPM tool and the runtime-behavior of the application. The EAD tool contains a schemaless database with a REST API to facilitate the documentation.

To update the runtime information of the applications running on cloud-based environments another process was defined in the CD/CI tool. The process is responsible for crawling the cloud infrastructure every certain time collecting and updating the runtime information in the automated EAD tool.

An up-to-date EA information of cloud applications can be ensured and the manual effort for documentation processes can be eliminated. The inconsistency and redundancy of EA information retrieved from the cloud infrastructure can be removed through this process. DevOps teams do not document smaller changes of an application in the EA tool. However, this continuously documentation process integrates the data sources used within the DevOps toolchain automatically. Similar documentation approaches are called "self-reporting architecture". [13]

The enterprise architect has the possibility to continuously monitor applications. It enables the detection of operational anomalies and performance related issues. [4] During the analyze and evaluation phase the enterprise architect can identify potential improvements from the monitored data. Afterwards, the enterprise architect executes the planned improvements while the IT Governance controls the fulfillment of policies and methods to ensure the alignment of the IT and the business goals. Based on that the IT Governance is also responsible for the management and monitoring of risks of IT resources. [37]

Depending on the EA tool in place an export from the automated EAD tool to the

EA tool is possible. A mapping between both tools is still needed. Once the mapping is implemented, the enterprise architect can verify the quality and consistency of the exported data in the EA tool. Since the automated EAD tool is only a middle-layer to aggregate information of different information sources the stakeholder will still take the decisions from the reportings and KPIs in the EA tool.

4.3. Approach in usage scenario

To test the conceptual approach in a sample scenario the following tools described in subsection 4.3.1 were used. The following figure 4.5 shows the automated EAD process in the scenario with the selected tools.. The focus of this work is the process of an automated EAD from an application development pipeline. The process is explained in detail in subsection 4.3.3.

4.3.1. Tool selection

The following tools were used to test the conceptual approach of an automated documentation of Business Domain assignments and cloud application information from an application development pipeline.

- The version control service Github is used as a web-based hosting for the code repositories.
- The open source automation server Jenkins is used to automate the application development process as a continuous integration and continuous delivery tool.
- To manage projects the Atlassian tool Jira is used. The tool enables an efficient Requirements Management for the project. [19]
- The cloud infrastructure is enabled by the usage of the open source, multi-cloud application platform CloudFoundry.
- The existing open-source project Pivio is used as an automated EAD tool. The project is explained in detail in chapter 5.
- The EA tool Iteraplan is used to test the approach. A lite version fo Iteraplan is available as a docker image. Iteraplan also allows the integration of other repositories through a public API.

4.3.2. Fulfillment of requirements

The definitions and components of the different tools selected for this approach need to be aligned. Figure 4.4 shows the mapping of definitions between the different tools and the alignment to the Archimate definition.

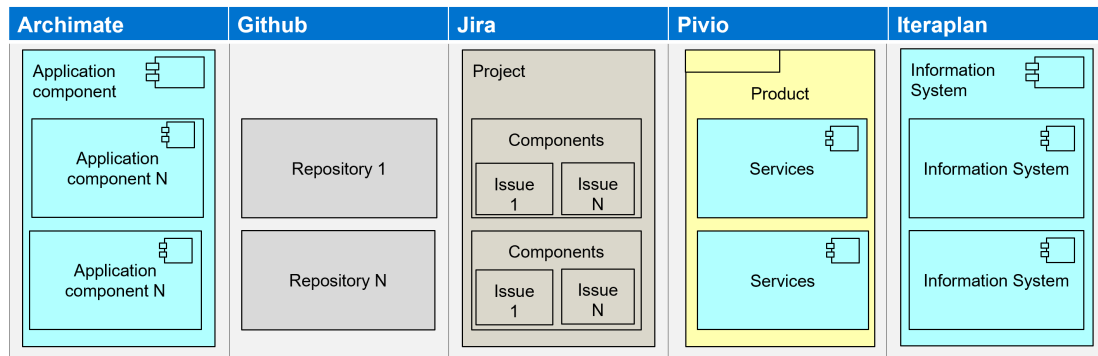


Figure 4.4.: Alignment of tools and Archimate

For the application development process sample Springboot applications were developed in different repositories and the communication between these applications was implemented.

4.3.3. Scenario process description

To retrieve EA relevant information for an application running on a cloud-based environment the information is retrieved already during the application development pipeline and continuous integration tool. The EA relevant information collection process within the continuous integration tool is divided into two jobs. The first job is illustrated in figure 4.5 as Groovy script. The second job is the crawler job which runs every certain time collecting the cloud and runtime information of the service. To enable the first job a groovy script has to be included into the repository of the version control service (requirement: pipeline script). The groovy-script represents the pipeline script. The pipeline shown in the picture above is divided into the following stages:

Get sources

This stage gets the latest code of the version control service. In this case the web-based hosting service for version control Github is used. The CD/CI Tool Jenkins downloads the repository locally.

4. Solution approach of an automated EAD process

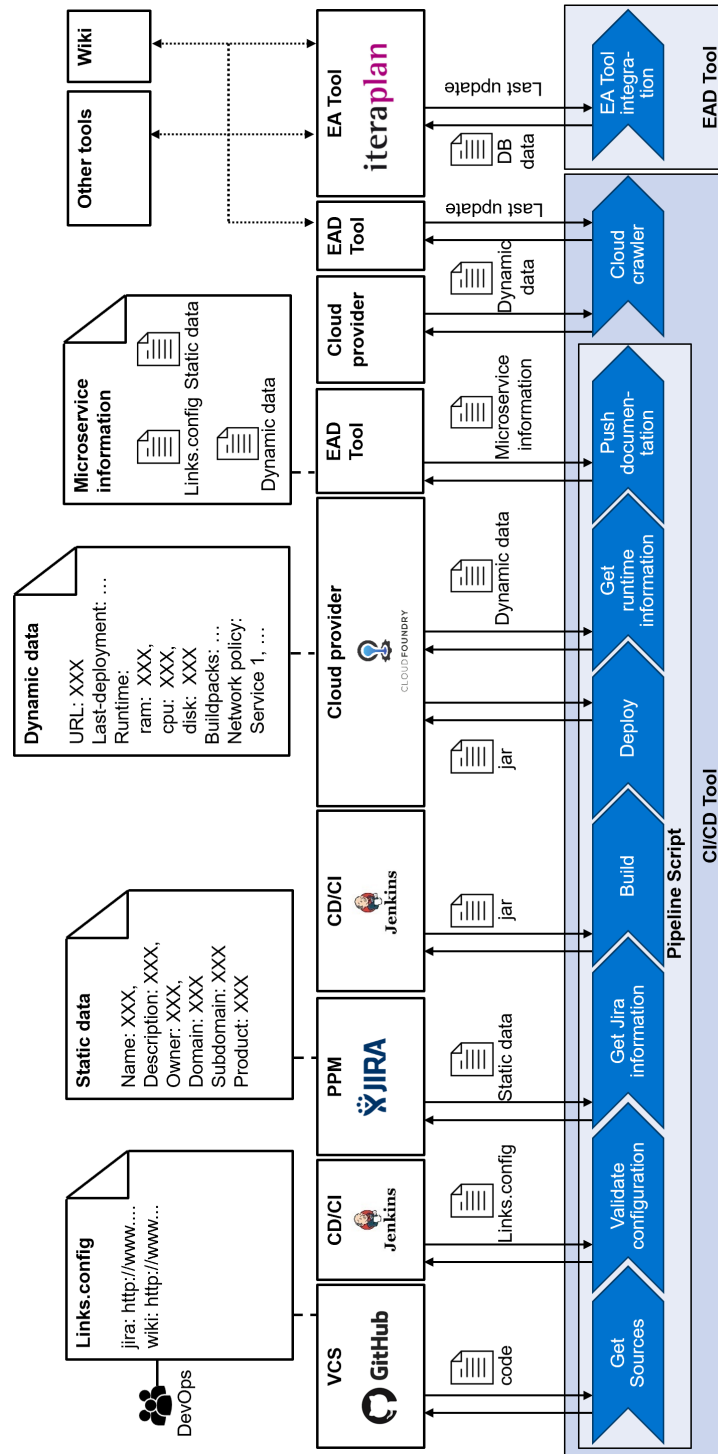


Figure 4.5.: Automated EAD process

Validate configuration

To enable the retrieval of the business specific information and to enable the federated approach of the EA documentation, the configuration file has to be validated. The configuration file contains the links to other tools such a link to the PPM (Jira), a link to the CMDB, a link to the wiki, etc. The developers need to manually maintain and update the configuration file.

If the configuration file exists in the repository and it contains a link to the PPM tool in this case Jira the pipeline does not fail. Otherwise the pipeline fails to disable the inconsistent documentation regarding the business specific information.

Get Jira information

The stage "Get jira information" collects the business specific information. The stage is divided into two small parts to gather the information.

The first part makes a call to retrieve the information stored at the project level in jira. This information gets the project id, the project name, the project owner and the project description.

The second part iterates through the issues assigned to the jira project. Every jira issue contains a standard field "component", which means to which child component of the parent application the issues is assigned to. Since jira does only allow to add customized fields at an issue level, each issue needs to contain the information of the domain, subdomain and product. To ensure the data quality attributes, the fields need to be mandatory when creating a new issue. The aggregated information of this part is stored at a global variable in the groovy script to be pushed at the documentation stage together with the information collected during the other stages.

As shown in the picture above this stage aggregates information from jira to the information being documented. Therefore every deployed artifact can be aligned to a domain, subdomain and a product during the pipeline. This stage is the innovative part of the documentation process. Disabling the deployment of a service to the cloud before having the business-specific information of the service ensures that the EA information is consistent and complete.

Build

Since the build stage together with the deployment stage are one of the most time consuming stages, the build and deployment stages were defined after collecting the business-specific information. The reason for this is that a completeness of the documentation is ensured. The build-stage builds the downloaded code of the version control service with the commands defined in this stage. In this case Springboot

applications were used to test the approach. Therefore Maven and/or Gradle commands were used to build the code. Depending on the size of the repository this stage may take some time.

Deploy

The deployment stage is also one of the most time consuming stages during this proposed pipeline. For the demonstration of this documentation process CloudFoundry is used as a multi-cloud application platform. This stage first connects to the cloud API endpoint with the credentials stored in the continuous delivery tool (jenkins). After authenticating, the organization and space of CloudFoundry are selected. The artifact is then pushed to the platform. If a manifest-yaml-file is defined in the repository, the specified information of that file is used for the cloud configuration. After executing the push command automatically, the platform downloads the buildpacks and software dependencies of the pushed artifact. The platform returns a message with the status of that the artifact. This means if the artifact was successfully or unsuccessfully deployed.

Get Runtime Information

Once the service was successfully deployed to the cloud the runtime information is retrieved. The information collected in this stage contains the following attributes:

- Status of the services: is the service down or is the service running?
- How many instances of this services are running on the cloud?
- How much RAM does the service need of the predefined resources?
- How much CPU does the service consume?
- How much Disk of the predefined resources does the service expend?
- What buildpacks or software dependencies does the service require?
- What services does the deployed service communicate with?

The runtime information that can be gathered from the API may vary depending on the cloud infrastructure.

Push Documentation

During the pipeline execution the collected information of the individual stages are put in the same global variable of the script. The export of this variable is formatted as a json. As shown in the above picture the json contains the information of the configuration file, the business-specific information and the runtime information. This json is pushed to the tool with a simple HTTP-POST-Method.

Cloud Crawler

To updated the EA relevant information retrieved during the process, a crawler was implemented in the continuous delivery tool. The job that triggers an information update is depicted as **Cloud Crawler**. The cloud crawler is a job in the CD tool that retrieves the runtime information of the cloud-based environment and updates the information in the tool. The crawler job is divided in four stages. The following figure shows the cloud crawler job.

Get Automated-EAD-Tool-App

In this stage the job first retrieves a list of all artifacts listed in the Automated-EAD-Tool.

Get Apps-List

During this stage the job gets a list of the artifacts that are hosted in the specified organization and space of the platform. This artifacts can be either running, stopped or crashed. Independently of the status of the service the platform API endpoint will return a complete list of the services.

Get individual Runtime Info

Subsequently getting the list of all artifacts of the organization and space of the platform, this stage collects the individual runtime information. This stage iterates through the list of artifacts to retrieve the individual information of the them on the platform. The runtime information contains the attributes described in "Get Runtime Information". This is the most time consuming stage of the crawler job since the job has to retrieve the individual runtime information for each artifact.

Push Documentation

Ultimately the retrieved information of the individual services is pushed to the Automated-EAD-Tool to update the information of the already documented artifact to

4. Solution approach of an automated EAD process

ensure an uptodate documentation. The job is scheduled time-based. In the jenkins instance the crawler job is performed every 15 minutes.

5. Prototype implementation

This chapter will introduce a prototypical implementation of a tool to cover the automated EAD process presented in chapter 4. An existing open-source project **Pivio**¹ was further developed to cover the presented requirements in subsection 3.2.

5.1. Motivation

Based on the findings during the literature research the implementation of a prototypical solution needs to cover the technology trends that influence nowadays the EA documentation. Furthermore the prototypical solution requires an integration of agile development methodologies, cloud infrastructures and microservices. Concurrently with the literature review the existing open-source project Pivio was found. It mainly focuses on service documentation. This project was adapted and further developed to align it with the presented approach in section 4.

Pivio

Pivio is a service registry. The main purpose of the project is to retrieve metadata of the deployed services. This metadata includes the name, owner and VCS information including runtime environment data and service dependencies.

Figure 5.1 shows the high level architecture of the tool. The three main components used in this approach are:

- The web component which acts as a webview
- The server component: A simple REST API
- A database component connected to the server

The original open-source project was used as a solution for microservice discovery. During this work the project was extended to document an application during the application development pipeline and to maintain and update this information.

¹<http://pivio.io/>

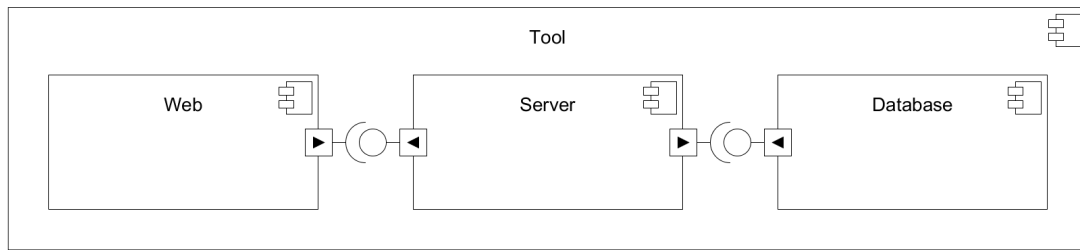


Figure 5.1.: High level architecture of the prototype

5.2. Technical specification

The web component was implemented as a Springboot² application with the integration of the Thymeleaf³ framework as a view layer for MVC-based applications. In addition to that the D3.js⁴ library was included to enable the possibility to produce dynamic, interactive data visualizations in the web component.

The server component was originally implemented as a Springboot application with Elasticsearch⁵ as the database component. During this thesis the server component was changed to a Node.js⁶ application with a MongoDB⁷ database due to maintenance reasons.

5.3. Main views

This section will introduce the main views of the tool. The tool has three core views. The first view will be explained in subsection 5.3.1. It contains an overview of all artifacts that were documented according to the approach in chapter 4. The second view in subsection 5.3.2 illustrates a detailed view of an artifact. The last view in subsection 5.3.3 shows different visualizations of the documented artifacts.

5.3.1. Overview

This subsection introduces the main page of the tool. The overview is divided into two parts. The first part displays a filter functionality described in 5.3.1. The second

²<https://spring.io/>

³<https://www.thymeleaf.org/>

⁴<https://d3js.org/>

⁵<https://www.elastic.co/>

⁶<https://nodejs.org>

⁷<https://www.mongodb.com/>

part as shown in figure 5.2 shows the overview of the artifacts running on the cloud infrastructure.

Every artifact is displayed as a card. The cards contains the artifact name, the owner, the description of the artifact, the last changes of the artifact and the status.

The status is depicted as a red or green circle. Red circles represents that the artifact has crashed or has been stopped. The green circle shows that the artifact is running. The status was added during this thesis. The goal of adding the status of the artifacts to the overview page is that the enterprise architects and other stakeholders can immediately react to a failure.

Filter tree

The filter tree was also implemented during this work. The main reason to add a filter functionality to the overview page was that more and more enterprises are moving their infrastructure to the cloud and therefore the amount of artifacts running on cloud environments has increased significantly. [3]

An average enterprise has more than 900 artifacts hosted in a cloud infrastructure. [27] Therefore the implementation of a filter functionality in the overview page is seen as useful.

The filter contains the metadata of every artifact. The metadata includes several properties of the artifacts such as the name, the short name, the status, the description, the changes (last upload and last update), the domain, the subdomain, the product, the owner and the links to other tools. The filter functionality depicted as "Quick Search..." enables a search through all the metadata properties mentioned before. A hierarchical tree was included as a visualization to enable an overview of artifacts per different areas. The hierarchical tree is structured in the following form: Cloud environment, domains, subdomains, products, artifacts belonging to the product and links to other tools. An example of the hierarchical view can be seen in figure 5.3. Clicking on a node expands or collapses the tree and updates the search box automatically with the metadata on the node and the list of artifacts gets updated as well.

5.3.2. Detailed View

The core view of the tool is the detailed view of an artifact. This view shows several properties and KPIs for one artifact. The main driver to cover a wide range of information regarding one artifact is to involve different stakeholder to use the tool.

The view is divided into nine sections. These sections are explained below.

5. Prototype implementation



Figure 5.2.: Overview of the artifacts

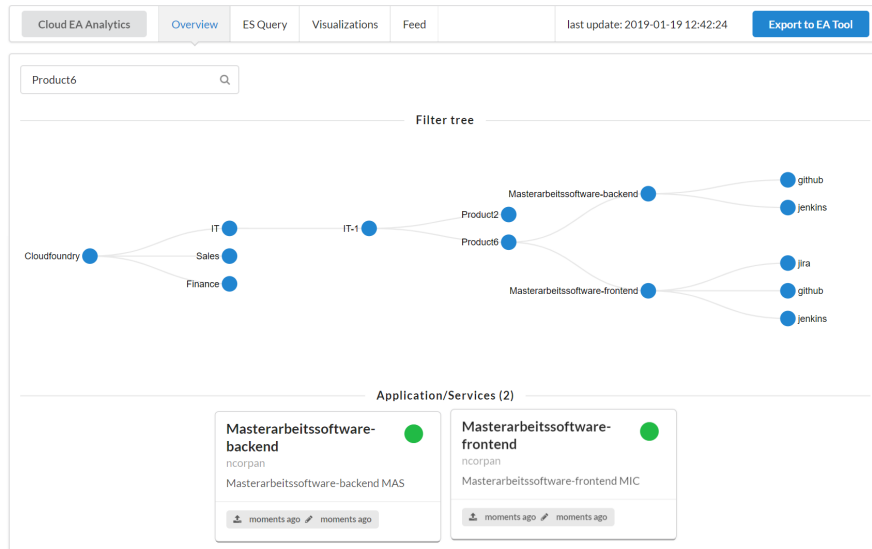


Figure 5.3.: Filtertree in overview

General section

This section shows general information about an artifact. Figure 5.4 shows this section. At the top of the section the status and the name of the artifact are shown. The description, the shortname and the type are retrieved from the PPM tool.

The business specific information is gathered from the issues within the PPM tool. This information includes the business domains, subdomains and the product of the artifact.

This section also contains the last changes of the documentation of the artifact and the date when the artifact was initially documented. The links to enable a federated enterprise architecture are also shown in this section. Further additional information is displayed too.

Runtime section

The runtime information is displayed on a separate section displayed in figure 5.4. This information includes the number of instances of the artifact, the memory used, the CPU consumption and the disk required for the deployment of the artifact. It also includes the host type which means where the artifact is hosted since more than one cloud provider can be used.

In addition to the runtime behavior of the artifact, the running costs are also displayed. These are calculated from the resource utilization. In this example the costs

The screenshot displays the 'General' section of an artifact's detailed view. The artifact is named 'Masterarbeitssoftware-frontend' and is described as 'Masterarbeitssoftware-frontend MIC'. It is a 'software' type with a shortname 'MIC'. The business specific information includes 'Domain IT', 'Subdomain IT-1', and 'Product Product6', with the owner 'ncorpan'. The 'Changes' section shows a change from 'vor 2 Wochen' to 'vor 2 Wochen'. The 'Links' section lists links to Jira, Github, Jenkins, CF-link, and Iterplan. The 'Additional Information' section shows a key 'ED'. The 'Runtime' section shows 1/1 instances, 142.7M of 1G RAM, 0.4% CPU, and 142.2M of 1G Disk, hosted on 'cloudfoundry' with a running cost of \$ 0.03 per hour. The 'Metrics monitoring (Agent)' section shows the URL 'masterarbeitssoftware-frontend.cfapps.io', Prometheus metrics 'masterarbeitssoftware-frontend.cfapps.io/prometheus', a response time of 1.15676799E8, and 13586 HTTP calls. The 'Services' section shows the service name 'Masterarbeitssoftware-backend'. The 'Software Dependencies' section lists buildpacks: client-certificate-mapper=1.8.0_RELEASE, container-security-provider=1.16.0_RELEASE, java-main, java-opts, java-security, jvmskill-agent=1.16.0_RELEASE, and open-jd...

General

Masterarbeitssoftware-frontend

Description

Masterarbeitssoftware-frontend MIC

Shortname (MIC) Type software

Business specific

Domain IT Subdomain IT-1 Product Product6

Owner ncorpan

Changes

vor 2 Wochen vor 2 Wochen

Links

Jira <http://vmmatthes32.informatik.tu-muenchen.de:6000/rest/api/2/project/ED>

Github <https://github.com/Nicocovi/Microservice2>

Jenkins <http://131.159.30.173:8081/job/Masterarbeitssoftware-frontend>

CF-link masterarbeitssoftware-frontend.cfapps.io Iterplan null

Additional Information

key ED

Runtime

Instances 1/1 RAM 142.7M of 1G CPU 0.4% Disk 142.2M of 1G

Host cloudfoundry

Running costs \$ 0.03 per hour

Metrics monitoring (Agent)

URL masterarbeitssoftware-frontend.cfapps.io

Prometheus metrics masterarbeitssoftware-frontend.cfapps.io/prometheus

Response time 1.15676799E8

HTTP Calls 13586

Services

Service name Masterarbeitssoftware-backend

Software Dependencies

Buildpacks

- client-certificate-mapper=1.8.0_RELEASE
- container-security-provider=1.16.0_RELEASE
- java-main
- java-opts
- java-security
- jvmskill-agent=1.16.0_RELEASE
- open-jd...

Figure 5.4.: General section in Detailed View of artifact

are calculated based on the memory per artifact instance. The cloud platform Cloud-foundry uses the unit "Gigabyte per hour" (GB-Hr) to calculate application costs. The rate used in this example is 0.03 dollar per GB-Hr. This means running an artifact that consumes 1 GB for 1 hour, the cost will be 0.03 dollar. The memory and duration are two parameters that can vary. These parameters determine the costs of the artifact. More instances increase the costs proportionally.

In summary, cloud infrastructure resources can be allocated better based on the displayed resource consumption and the running costs of the application. An efficient allocation of IT resources can lead to a decrease of infrastructure costs.

Metrics monitoring section

More use cases can be derived from monitoring and analyzing applications and their KPIs. [41] This is the reason why a section containing KPIs was added to the detailed view of an artifact. The metrics monitoring section is shown in figure 5.4.

The two selected KPIs for this section are the response-time and the number of http-calls of an artifact. The response-time KPI is displayed because it has different meanings. One of them is that a higher response time means an increased waiting time for the user. For that reason it is displayed to show a possible decreased performance of the system. The total number of http-calls can be used for decommission purposes. If the artifact has high costs and total the number of calls is too low the enterprise architect should take measures.

This section also includes the link to the artifact and links to monitoring agents of the artifact. These agents are used to retrieve extra KPIs which are not covered by the monitoring tools offered by the cloud providers. These agents show more technical KPIs such as heap memory consumption and java virtual machine information. The links can be used by other stakeholder such as DevOps teams and CloudOps teams. This increases the involvement of stakeholders in regard to the tool.

Services section

The services section contains an overview of the services connected to the artifact shown in figure 5.4. This section is used for dependency management of the artifact.

Software dependencies section

The software dependency section illustrated in figure 5.4 shows the reliance of the artifact on other software packages. The presence of software dependencies are an important indicator of software complexity. A high degree of dependencies reflects the

cohesion and coupling of the shown artifact, which can be seen as a software quality index. [51]



Figure 5.5.: Jira-Monitoring section in Detailed View of artifact

Jira monitoring section

This section displayed in figure 5.5 shows the aggregated information of the PPM tool to the artifact. As mentioned in 4.3.1 Jira is used to manage projects.

This section shows the development of the project in regard to the total amount of issues and the amount open issues. A pie chart was added in addition to visualize the progress. As named in 4.3.3 every Jira issue contains a standard field "component". The related components are displayed in this section to display other artifacts that are related to the parent application/product. The difference to the services section 5.3.2 is that the shown artifact is not necessary connected to all components of the project. Other stakeholder like the product owner might have an interest to see all components/services of the project.

Github monitoring section

Figure 5.6 shows the section of the VCS' repository. For demonstration purposes of the approach Github is used as the VCS.

This section is connected to the API of Github to show metrics of the repository. The metrics implemented during this work are the number of contributors of the repository and the lines of code. This KPIs serve as an indicator for the size of the service.

In combination with the Jira section the stakeholder can see the ongoing procedure and development of the project in one single page.

5. Prototype implementation

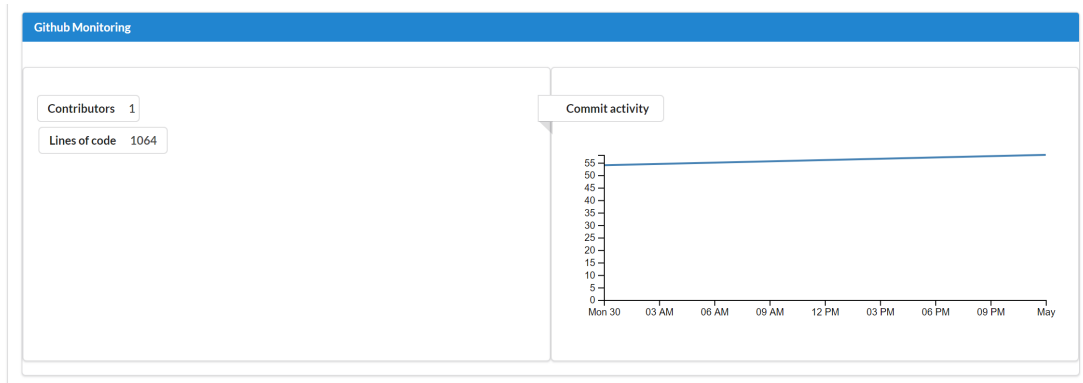


Figure 5.6.: Github-Monitoring section in Detailed View of artifact

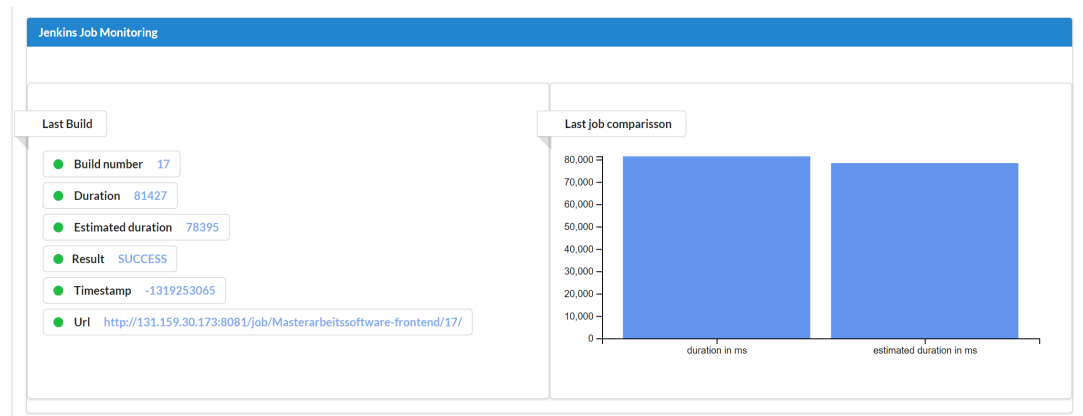


Figure 5.7.: Jenkins-Monitoring section in Detailed View of artifact

Jenkins section

Figure 5.7 shows the section containing information of the CD/CI tool. Jenkins is used as the CI tool during this approach. The last build API of Jenkins is used to display information about the build deployment. This section contains on the left side the following metrics: the last build number, the duration of the last build, the estimated duration of the build process, the result of the last build, the timestamp and the url to the last build in the tool. On the right side it shows a bar chart comparing the duration of the last build and the estimated duration. The comparison is used to see whether the last build was affected by external factors such as latency or connection issues.

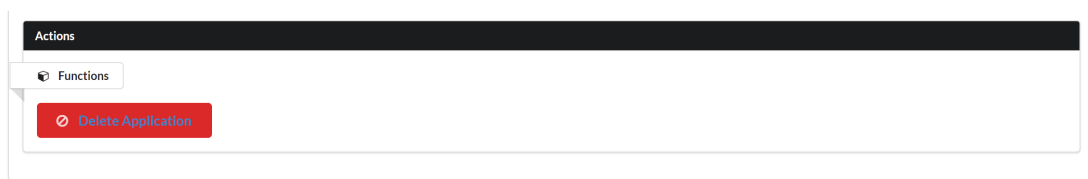


Figure 5.8.: Actions section in Detailed View of artifact

Actions section

The actions section displayed in figure 5.8 includes the manual delete functionality of the artifact. Further functionality can be added in this section such as an update function for a manual information update to include additional data.

5.3.3. Visualizations View

Another essential view of this tool is the visualizations view. The view was implemented during this thesis using the D3.js⁸ library for producing dynamic, interactive data visualizations in the web component. The visualizations view is divided into two diagrams. The first diagram is a combination of a hierarchical edge bundling diagram and a sunburst diagram. The second visualizations is an adjacency matrix.

Communications and business domain assignment diagram

As already mentioned the "Communications and business domain assignment diagram" is a combination of hierarchical edge bundling diagram and a sunburst diagram. As shown in figure 5.9 both diagrams have been used to achieve a unique visualization.

⁸<https://d3js.org/>

Every artifact listed in the tool is displayed around the business information. The blue lines represent the communication between the different services running on the cloud. The purpose of the diagram is to visualize the dependency between the services. By clicking on a service, the detail view of the service is displayed. Hovering a service highlights the connections (blue lines) and the dependent services of the hovered service.

In addition to the communications, a sunburst layout was developed to aggregate the business information. The domains and products of the displayed services are shown underneath the communications diagram allowing to visualize the assignment of the services to the products and the business domains. Due to this visualization it is easier to detect which other domains and products are affected if a service fails.

Adjacency Matrix

An adjacency matrix was developed to enable a dynamic diagram. The purpose of this matrix is to allow the possibility to rearrange the rows and columns regarding on different filter criteria. Figure 5.10 shows the matrix ordered by the service with most dependencies (connections). The possible orders are:

- by name
- by domain
- by frequency

Enabling the possibility to order by these criteria it is easier to visualize and analyze the dependencies of the services. The services with the most dependencies are always shown at the beginning of the matrix (left side).

By clicking on a cell it is possible to see the domains of the source service and the destination service. This communication information box shown in figure 5.11 can be extended to include detailed information on the specific connection.

5.4. Component diagrams

This section will first introduce a component diagram to see the communication between the different tools. Then this section shows two main parts of the tool: the web component and the server component.

5. Prototype implementation

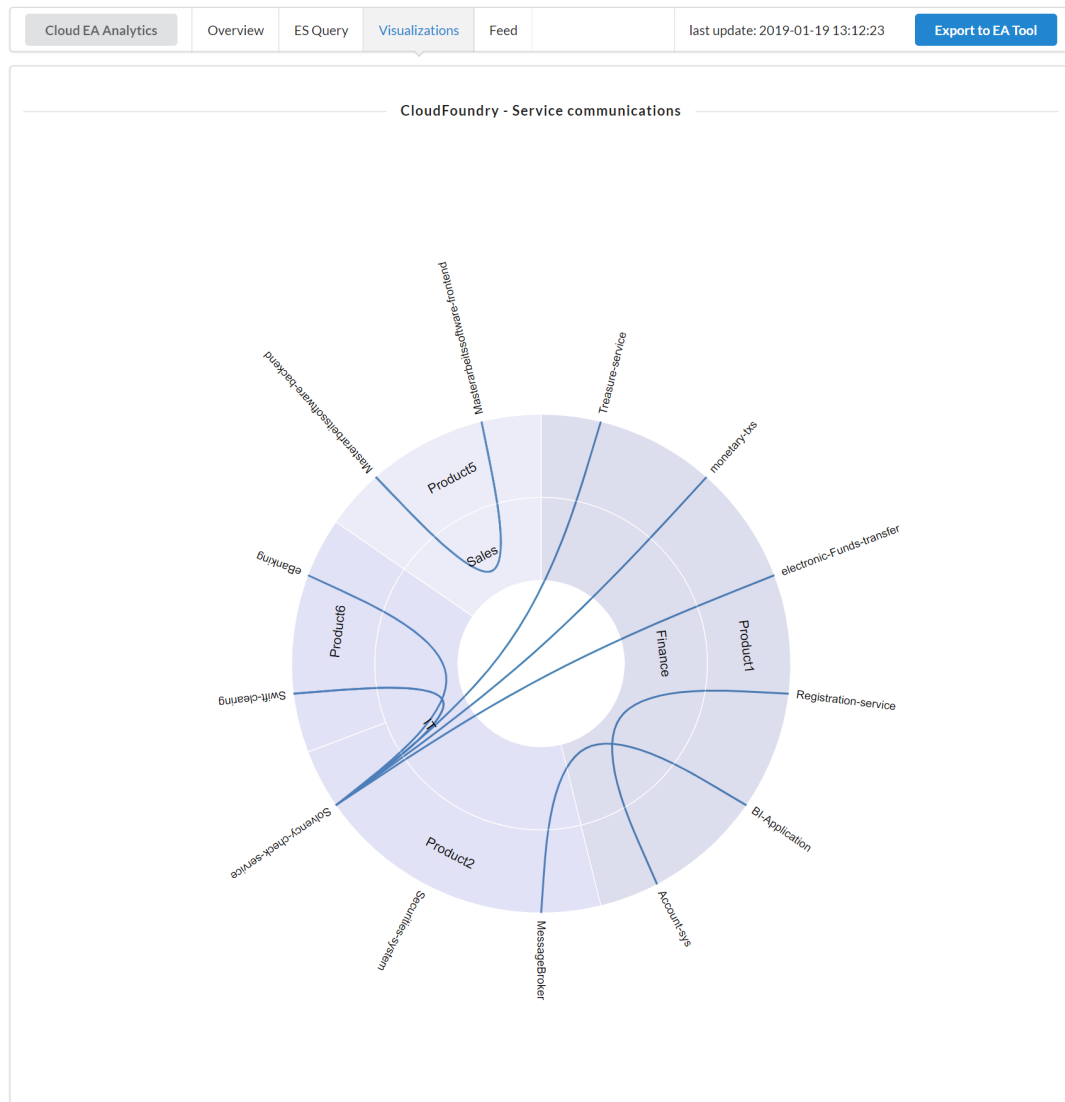


Figure 5.9.: Communications and business domain assignment diagram

5. Prototype implementation

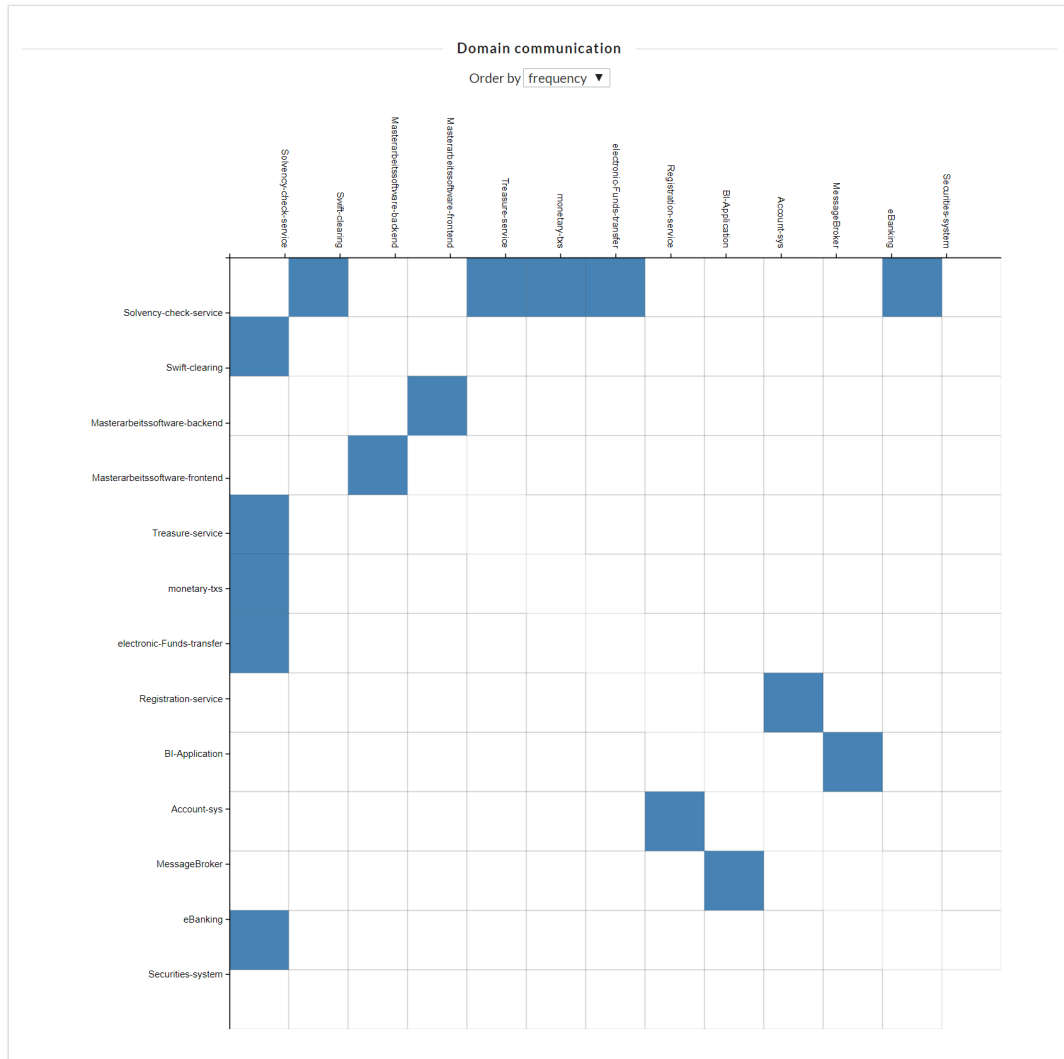


Figure 5.10.: Adjacency matrix

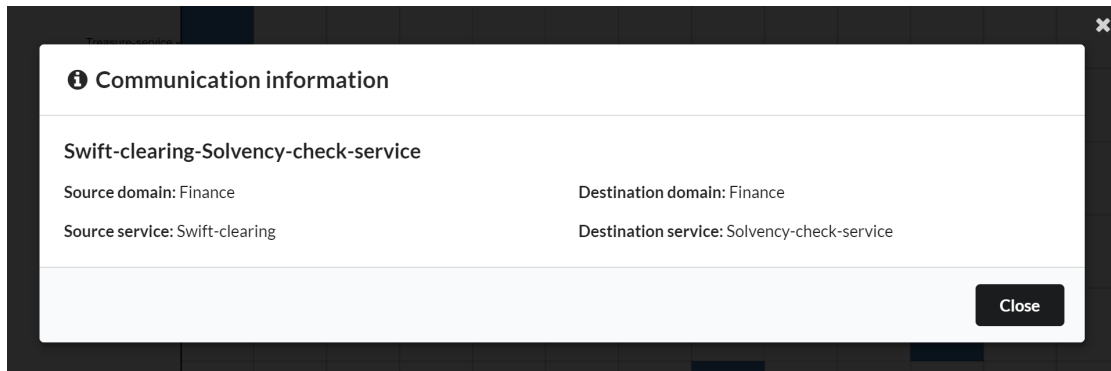


Figure 5.11.: Adjacency matrix cell information box

5.4.1. Components communication

Figure 5.12 shows the communication between the major components of this work.

As mentioned in 5.1 the Tool consists of three components. The The server component is a simple REST API layer for the database component. The Webview component visualizes the information stored in the database. It is also connected to an Adapter component which collects on runtime the information of the specific artifact being displayed. The Adapter retrieves the information from the PPM Tool, the VCS and the CD/CI Tool. The CD/CI Tool runs the crawler job explained in 4.3.3 to update the runtime information of the artifact. The artifact is hosted in a cloud infrastructure and can is typically a microservice or a container.

5.4.2. Web component

Figure 5.13 shows the UML component diagram of the web component.

The main part the of web component is the ServerConnector. This component contains the individual calls to the server component. It uses the component ServerConfiguration to get the current server address. The ServerConfiguration is linked to the application properties file where the links are hold.

To retrieve the data for the overview page the ListController forwards a request for getting the data to the ListService component. The ListService gets the data from the ServerConnector and creates an Overview. The Overview is mapped via the ModelMapperComponent to the OverviewModel which is used by the OverviewTemplate for the data visualization.

The same principle is used for the detailed view of an artifact. The DetailController forwards the calls to the DetailService. The DetailService creates a Document with the data gathered from the ServerConnector. The DetailController fills then

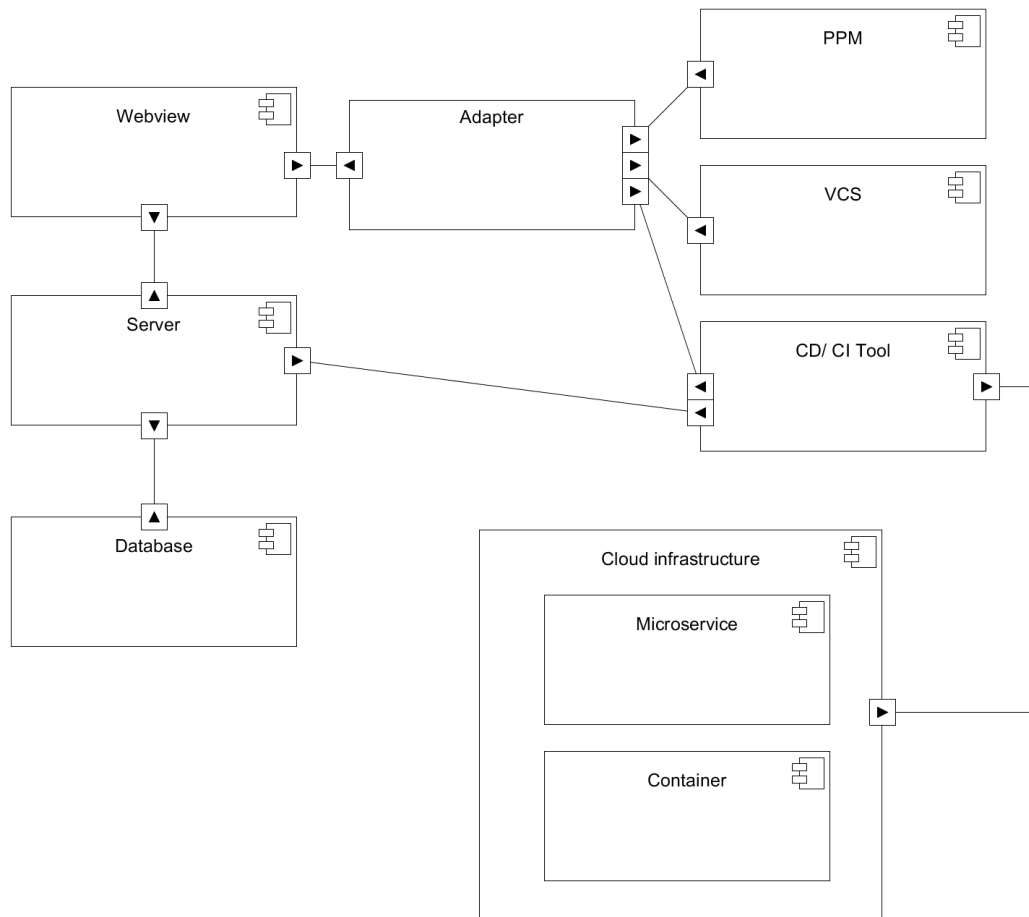


Figure 5.12.: Components communication

the DocumentViewModel with the received Document from the DetailService. The DetailviewTemplate uses the data of the DocumentViewModel to visualize an artifact.

A different principle was used for the visualizations. The VisualizationsController delegates the call to the VisualizationsService which gets the data from the ServerConnector. The Controller passes the data directly to the template. No model is used for this template because the data is parsed several times for the individual visualizations within the template.

Similar to the ListController, the ExporterController delegates the calls to the Service which gets the data from the ServerConnector. The Controller exports then the data to the EA Tool in place.

The PPM Monitoring component is an adapter component for the connected PPM tool. It retrieved the business specific information of the tool. In this case Jira was used as a PPM tool. This adapter collect the business domain, the business subdomain, the product information and the application owner via the API of PPM tool.

The VSC Monitoring component is also an adapter for the code repository. It gathers the development information of the repository. This information includes metrics for determining the size and complexity of the code. Metrics like number of contributors, lines of code and a commit activity diagram are displayed in the GUI.

The CD Monitoring component is collect the information of the last build of the application. Statistics of the last job like build result, duration and estimated duration are used to determine whether the last build was affected by external factors such as latency or connection issues. This adapter can be connected to any CD/CI tool to gather more information.

The governance monitoring component was implemented to monitor other EA relevant criteria for the enterprise. Possible criteria are mentioned in 6.6.3.

5.4.3. Server component

During this work the server component of open-source project was changed to a node.js application with a MongoDB database. Figure 5.14 shows the UML server component diagram.

The server component includes the main application component, and a controller, a route and a schema for storing applications into the database. The application component illustrated in figure 5.14 as "App" defines the API paths and requires the Route component for forwarding the incoming calls. The Route component maps the implemented CRUD-Methods in the Controller component to the respective paths. The Schema component defines the model for the database and the controller requires it to validate and parse the incoming data. This data is stored in the database component.

5. Prototype implementation

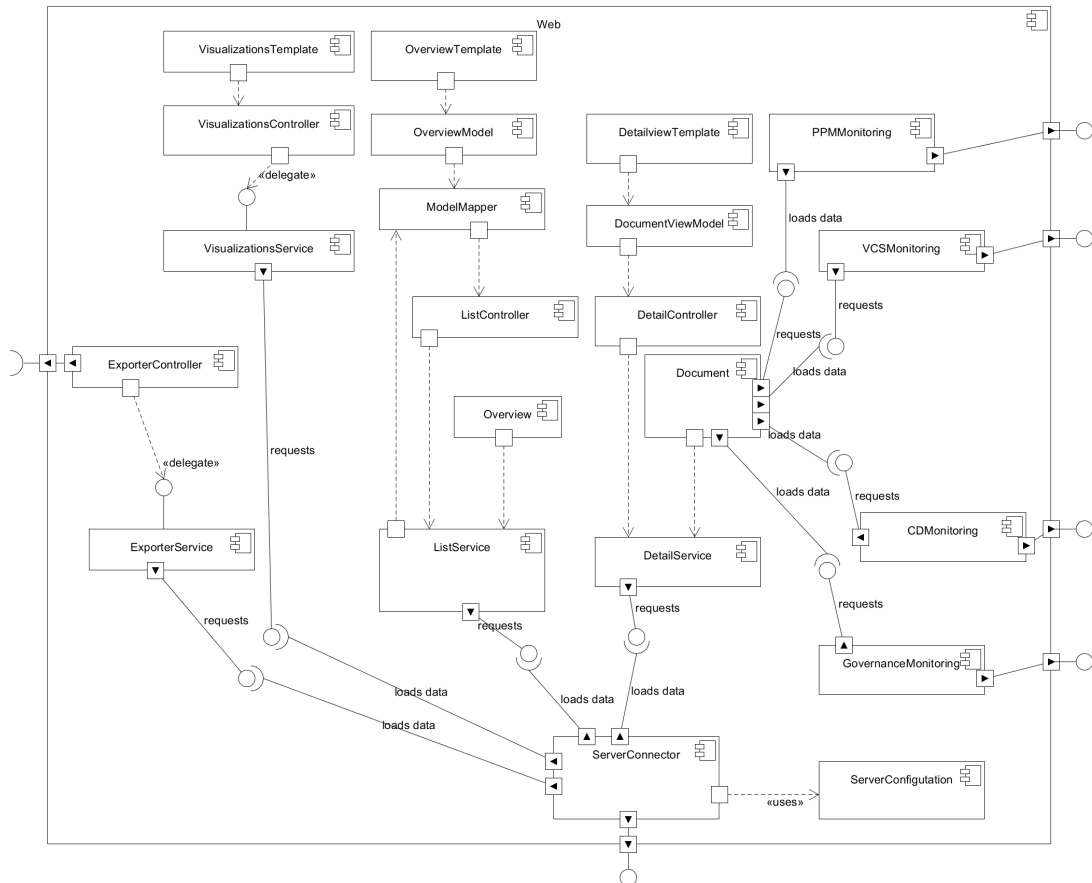


Figure 5.13.: UML component diagram of the web component

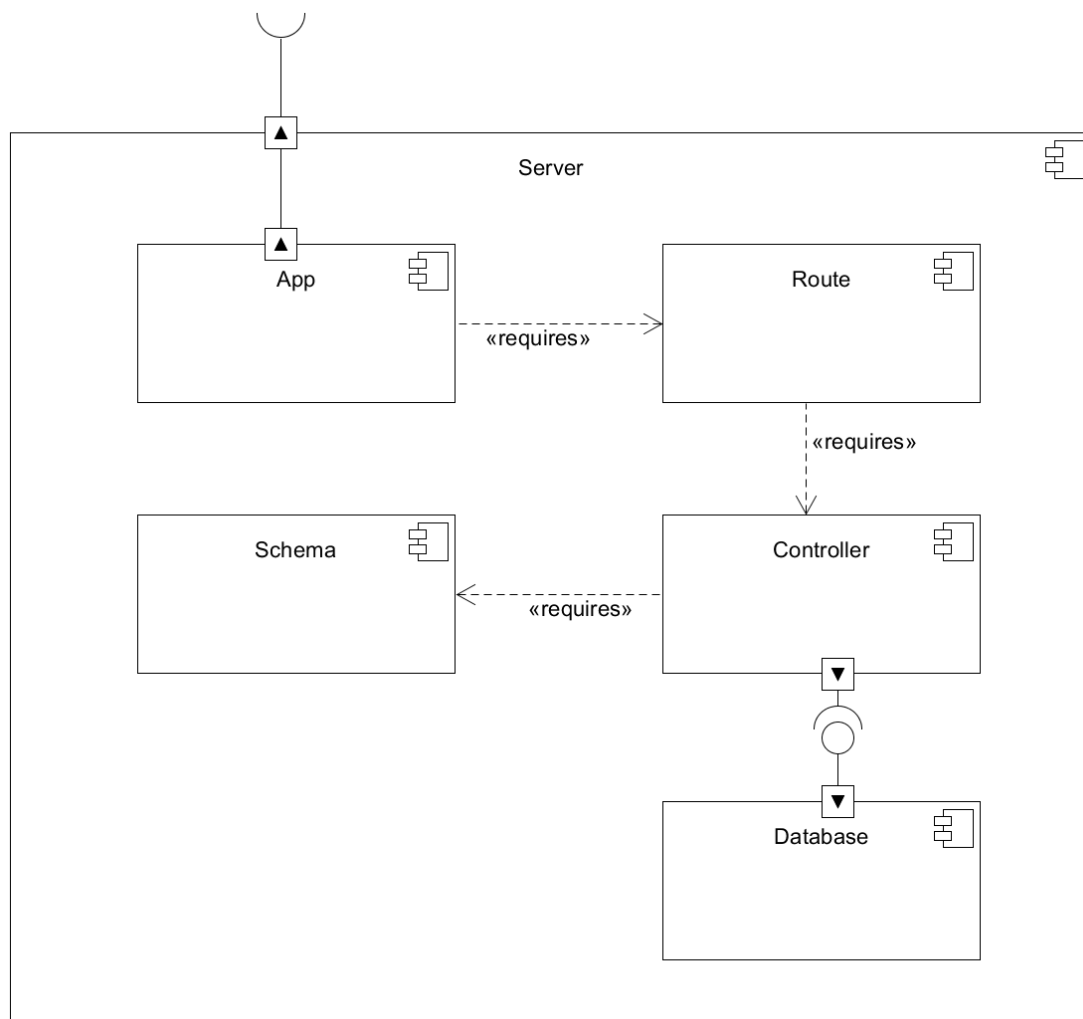


Figure 5.14.: UML component diagram of the server component

5.5. Class diagram

The web component retrieves the data from the server component via a REST API. The return of the server is a JSON containing the properties of one service. This JSON response is mapped to the classes shown in figure 5.15.

Every response is mapped to a document. A document contains several properties. One of the properties is the information about the runtime. The runtime properties are modeled in a separate class. A document can contain one service which is composed by a list of buildpacks and one-to-many "Provides" objects. The list of buildpacks represents the list of software-dependencies mentioned in 5.3.2. "Provides" objects are also modelled in a separate class. One object represents a single connection to another service running on the cloud. The "Provides" class includes the service name, the protocol and the port of the communication.

The tools used during this approach are modeled as properties of a document to facilitate the usage of the links.

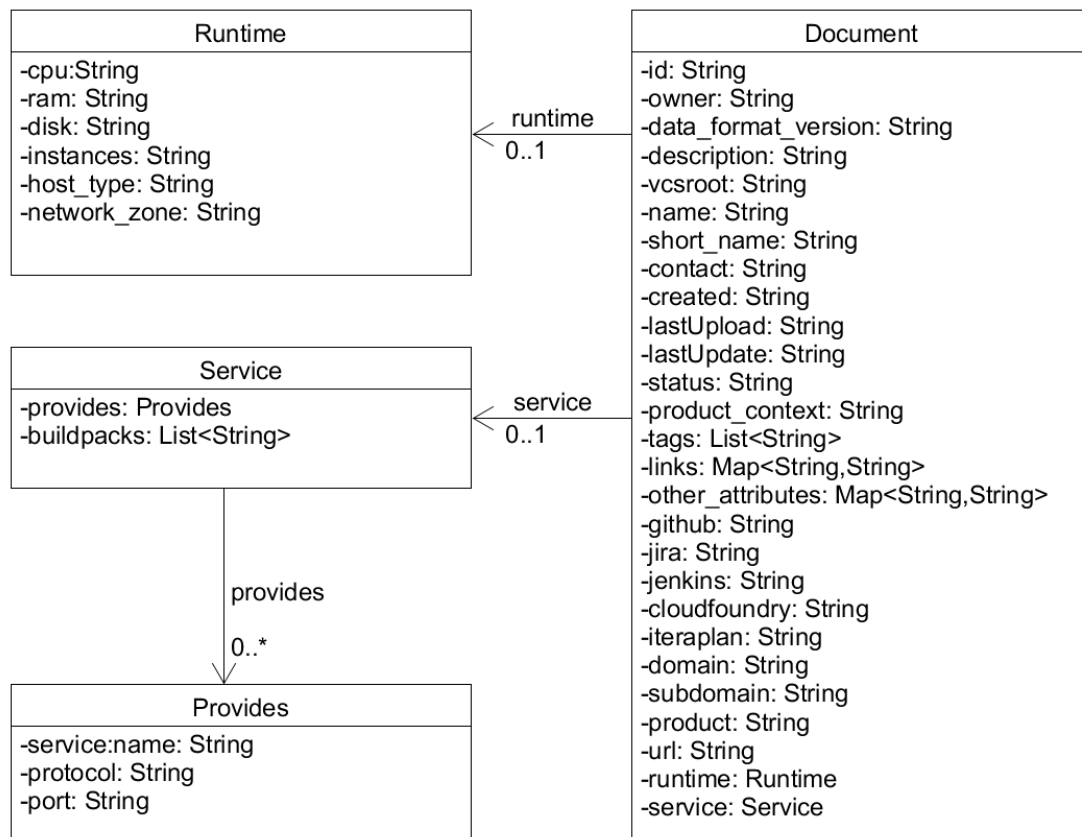


Figure 5.15.: Class diagram of server response

6. Evaluation

The first section of this chapter introduces the methodology of the case study. The second section will explain the challenges that influence the IT landscape of the insurance company. These are divided into internal and external challenges. In the next section 6.3 the current IT landscape of the enterprise is described. In addition to the as-is IT landscape, the current process of the EA documentation in the insurance company was analyzed during this master-thesis. In the following section 6.4 the target IT landscape will be described. In section 6.5, the derived requirements to automate the EAD process will be presented. At the end of this chapter the prototype is evaluated.

6.1. Case Study Design

The case study was conducted according to the research methodology for software engineering from Per Runeson. [43]

6.1.1. Case study objectives

This case will identify the current practice and the challenges regarding the EAD process in a german insurance company. The case study evaluated the suggested solution of an automated EAD process and its prototype derived from the requirements found out during the literature research. To improve the solution the prototype was evaluated in the company to derive further requirements for an automated EAD.

6.1.2. Case study definition

The subject of this case study is an international insurance company located in Germany. The investigation in this case will mainly focus on the Enterprise Architecture Documentation process within the company. To understand the process, the study will take a look at the IT landscape to retrieve information how the EA relevant components interact with each other and what challenges influence the IT landscape. This will give an overview to find out why the enterprise is documenting EA relevant information in that way. The following numbers give a brief overview of the complexity of the

enterprise. The IT landscape of the enterprise consists approximately of 2000 applications, 400 running on cloud based environments and the enterprise has nearly 1000 developers.

6.1.3. Case study methodology

Research questions - what to know?

The research questions that are going to be answered are mentioned in 1.2. RQ1, RQ2 and RQ3 are relevant for the case study. The aim is to see if the questions can be answered during the case study and the prototype evaluation to improve the automated EAD.

Methods — how to collect data?

As mentioned by Runeson et al. [43] the following information sources can be used to collect data:

- Different data sources
- Archival data
- Interviews

During the case study a literature research was conducted to derive challenges and requirements from literature to be able to compare the information to the current situation of the company.

An analysis of the archived data of the company was executed during the thesis in order to gain more knowledge about the IT landscape, the reasons of the current documentation process, challenges changing the landscape and what the enterprise expects to be improved or even automated.

To retrieve more information informal and semi-structured interviews were conducted. The interviews were divided into 2 parts. The first part explained the motivation and the objectives of the thesis. The motivation introduced a brief overview of the scope of the thesis and the objectives includes the research questions to be answered. The second part of the interviews consisted of questions regarding the findings during the research of the current situation.

6.2. Challenges influencing the IT landscape

This section will describe the influencing challenges that have an impact on the IT landscape of the German insurance company. To understand the IT landscape an overview of the main influencing challenges is presented. First, the external ones such as regulations are explained. Secondly, the internal challenges are mentioned. Finally, the EAD process will be described.

6.2.1. External challenges

The section will describe the most important external challenges that change the IT landscape of the enterprise and thus have an impact on the EAD. There are three regulations that mainly influence the EAM. The first regulation is the General Data Protection Regulation (GDPR). The second factor affecting the IT landscape is the VAIT Regulation. The last regulation influencing the IT landscape is the international standard ISO 22301.

General Data Protection Regulation

The General Data Protection Regulation¹ (GDPR) is a European Data Protection Regulation that enforces all member states of the European Union to harmonize data privacy laws. The GDPR is related to the processing of personal data. It assures fundamental rights of persons, especially the right to the protection of personal data. GDPR enforces the enterprises to know exactly which information systems store personal data and how this information systems can be secured to protect the personal data from external hazards. An inventory of information systems related to the processing and storing of personal data is therefore required. This regulation demands a transparent and complete documentation of the IT landscape.

VAIT Regulation

The regulation "Versicherungsaufsichtliche Anforderungen an die IT" (engl. "Insurance supervisory requirements for IT") (VAIT) affects the IT of insurance companies with headquarters in Germany. The use of IT in companies, including IT services offered by IT service providers, is of central importance to insurance companies. The circular letter published in the context of the regulation mentioned contains information on the interpretation of the regulations on business organization in the Insurance Supervision Act (in German "Versicherungsaufsichtsgesetz") (VAG) insofar as they relate to the

¹<https://eugdpr.org/>

technical organizational equipment of the companies. It makes these regulations binding for Federal Financial Supervisory Authority (in german "Bundesanstalt für Finanzdienstleistungsaufsicht") (BaFin), thereby ensuring consistent application to all companies and groups. The circular letter provides a flexible and practical framework, in particular for the management of IT resources, information risk management and information security management. [2]

The main requirement affecting the EAD is the IT operations requirement. The IT operation must implement the fulfillment of the requirements resulting from the implementation of the business strategy as well as the IT-supported business processes. The components of the IT systems and their relationships to each other should be managed appropriately and the inventory information collected should be updated regularly and on an ad hoc basis. The stock information includes in particular: Inventory and purpose of the components of the IT systems with the relevant configuration information, location of the components of the IT systems, compilation of the relevant information on warranties and other support contracts (possibly linking), information on the expiration date of the support period of the components of the IT systems, accepted Period of unavailability of the IT systems and the maximum tolerable data loss. [2]

ISO 22301

The ISO 22301² standard represents the latest international policy for Business Continuity Management (BCM) and was released in 2012. Its objective is to assist in the reduction of business interruptions due to unforeseen emergencies. This norm is an update of the standards ISO 31000 and ISO 27001. It is considered a universal standard in the sense that they apply to companies of all sizes and regardless of the used technologies.

To ensure the reduction of unforeseen emergencies an IT emergency system should be introduced to act as a holistic management system. That includes monitoring and regular review of the IT landscape. These two aspects enable one of the main components of the standard execution: the Business Impact Analysis.

The Business Impact Analysis (BIA) is a complex task that includes important corporate resources as a precautionary measure: specialists, executives and corporate management. The analysis includes the collection and identification of processes and functions, the required resources such as staff, but also hardware resources like IT facilities, buildings, warehouses with their technical equipment. The analysis also include dependencies on IT processes, the definition of the core processes and impacts and recovery times. [39]

²<https://www.iso.org/>

6.2.2. Internal challenges

The internal challenges presented in the following paragraphs are mainly the drivers that have an impact on the IT landscape of the enterprise.

Main business system

The main business system of the insurance company has developed over time into a monolithic system. System components were built to connect new systems or applications to enable a communication between those. The added layers, components and adapters led to an not transparent and not maintainable monolithic system. The system contains several business logical aspects which results as an additional challenge for documenting the EA. To improve the transparency of the monolithic architecture the enterprise decomposed the system into modules to gain information about the dependencies to other applications and/or systems.

Business Continuity Management

Driven by the ISO standard 22301 the german insurance company of this case study is also obliged to fulfill this regulation through a BCM project. The company is exposed to increasing risks that endanger the continuous and timely provision of its services to the customer. Due to the increasing complexity of business processes, their increasing dependence on information technology and external service providers, external events such as fire, flood, the failure of information technology or external service providers can have a major impact.

The Business Continuity Management (BCM) is a management process with the aim of early identification of serious risks for the enterprise, which endanger the existence of the enterprise, and to establish measures against them. In order to ensure the viability and thus the existence of the company, appropriate preventive measures must be taken, which on the one hand increase the robustness and reliability of the business processes and on the other hand enable rapid and targeted response in an emergency or crisis.

The goal of the BCM is to ensure that important business processes are not or only temporarily interrupted, even in critical situations, and that the economic existence of the enterprise remains secured even in the event of a major loss event. A holistic view is therefore crucial. To consider are all aspects that are required to continue critical business processes when a claim event occurs, not just the Information Technology resource. IT emergency management is part of the BCM. Critical business processes in the sense of emergency management means "time-critical", so that this process requires a faster resumption of activity, otherwise a high level of damage can be expected. The high damage can result from financial losses, violations of laws or contracts, from

image damage or other damage scenarios. A business process classified as "uncritical" does not mean that it is unimportant to the enterprise, but merely that it has a lower priority in recovery.

Decommissioning project

The goal of the decommissioning project is to withdraw traditional data centers from active service to mainly reduce costs. The decommission of applications or systems have different reasons. The first reason is to decommission systems due to lack of support available or the purpose to remove old legacy systems. The data of the removed system has to be migrated to the new system. Another reason for decommissioning systems is to withdraw traditional data centers from active service. The efficient redistribution of IT resources can lead to migrations of systems from one data center to another to reduce the amount of running data centers and consequently reduce the costs of running data centers. The other approach of migrating applications to withdraw servers is the migration to the cloud. The elasticity of the cloud enables the efficient use of computational efficiently and has as a consequence that traditional server can be withdrawn.

Cloud migration project

The migration to the cloud of the IT infrastructure is a central and strategic issue. The expansion of the IT architecture, including cloud solutions, forms the basis for digitalization and increases the productivity and efficiency of IT processes. The migration improves the availability and scalability of IT services, thereby increasing the growth potential of all IT structures. The duration of the total migration amounts to an estimated 4 years and is divided into quarterly projects. The migrated applications are put into production at the end of every quarter.

The arguments for migrating to the cloud are divided into two aspects. The arguments covering the first aspect of operations are:

- Time to market: Cloud infrastructures reduce the time-to-market due to an integration of an automated build and deployment pipeline.
- Performance and Availability: Redhat's OpenShift³ and Pivotal's CloudFoundry⁴ are proven solutions that are used in many large organizations and designed for high availability.

³<https://www.openshift.com/>

⁴<https://www.cloudfoundry.org/>

- Scalability: OpenShift and CloudFoundry enables fast and easily scalable automation of applications based on demand and load.
- Cost Efficiency: Moving from traditional virtualization technology to container technology will bring better utilization of existing resources and, in the long run, cost savings.
- Security: OpenShift and CloudFoundry provide easy deployment of security patches for platform and applications. The updates can be imported without downtime for the end user. Both providers also offer easy deployment of security patches for platform and applications. The updates can be imported without downtime for the end user.

The following migration arguments cover the second aspect of application development:

- Developer Satisfaction: Developers can leverage cutting edge technologies to build innovative solutions.
- Developer Efficiency: CloudFoundry is designed as a self-service platform. Developers get more freedom in a well-defined framework to get to their destination faster.

The project objectives are:

- Full application migration without interruption to ongoing operations (24x7 applications, sales, support and night jobs). These means the technical migration of more than 400 applications.
- Create a migration blue print and best practices for dealing with the new infrastructure.
- Creation of the migration framework including half / tools to carry out the migration
- Supplementing the operational concept for post-migration operations. This means the monitoring and error analysis of the migrated applications after the production has been set for the duration of the project.

The project does not include the migration of fat clients and non-insurance-owned applications.

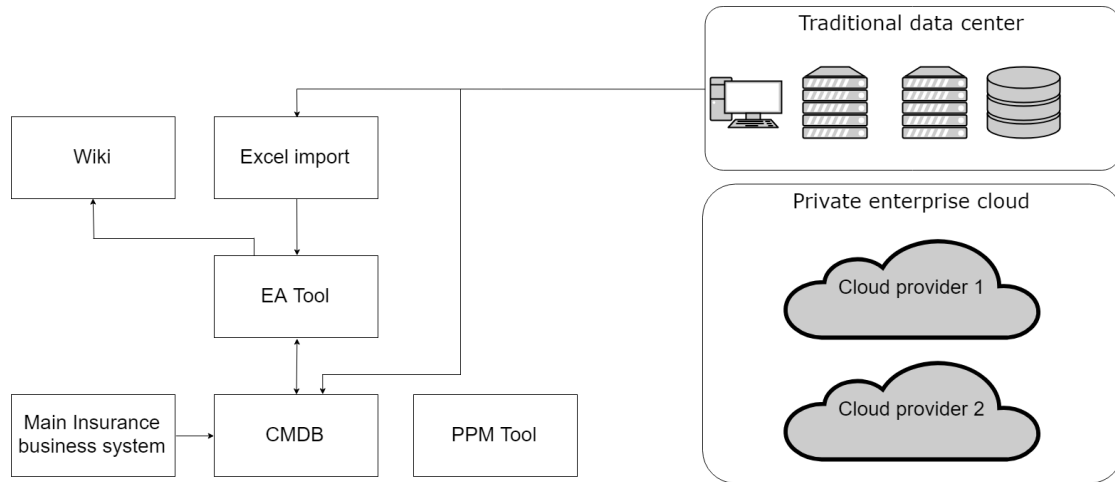


Figure 6.1.: AS-IS IT landscape

6.3. AS-IS IT landscape

This section will describe the IT landscape of the company. The following picture shows the main components of the IT landscape.

The company consists of two environments: the traditional environment which contains traditional data center hardware and its private cloud infrastructure which is divided into two different cloud service providers. The first cloud infrastructure service provider is Redhat's OpenShift⁵ and the second provider is Pivotal's CloudFoundry⁶.

The traditional data centers host the legacy systems of the company and all other kind of applications.

OpenShift is a container platform. The PaaS provider OpenShift offers the possibility to deploy docker containers to its platform and enables the orchestration by Kubernetes⁷.

CloudFoundry (CF) is an open source, multi-cloud application platform as a service. Similar to Openshift, CF has a container-based architecture which also enables the deployment of applications written in any programming language.

The main components of the IT landscape shown in figure 6.1 in regard to EAD are the EA tool, the CMDDB, the PPM tool, the Wiki is a collaboration software program and the main business system.

⁵<https://www.openshift.com/>

⁶<https://www.cloudfoundry.org/>

⁷<https://kubernetes.io/>

6.3.1. Current EA documentation

The EAD integrates several tools as shown in figure 6.1.

Excel import

The first EA documentation process was driven by the external mentioned ISO 22301 standard. The enterprise decided to collect the information about the IT landscape. The usage of Microsoft Excel sheets for keeping data is still important for enterprises since many organizations still rely on them for information storing. [15]

Initially the systems or applications that are critical for the business were documented. This was mainly driven by the BCM project and the BIA of IT failures. Ensuring that important business process are not interrupted and have no economic impact to the company is essential that the enterprise remains secure. Due initiative a team was created to retrieve this information. The documentation process was done fully manually. Therefore the document containing the information was inconsistent and contained redundant information.

The continuation of the project is mainly driven by the VAIT regulation since enterprise will have to have a complete application inventory.

Integration of the CMDB

Due to the BCM regulation mentioned in section 6.2.2, an integration between the EA tool and the CMDB was implemented. The EA tool integrates data from the CMDB. The application developed for the integration retrieves data from the CMDB and imports the data to the EA tool in regular intervals. The import was unidirectional. Only Business Services were imported from the CMDB. A Business Service is a metamodel object in the CMDB. No infrastructure data was integrated into the EA tool. The import did not analyze the data, meaning that if the imported data was redundant there was no merge conflict solved. The integration was further developed to improve these challenges. The Integration on the CMDB contains now a conflict solution and harmonizes the redundant imported applications with a unique identifier. The import mechanism also differentiates between a flag set in the CMDB metamodel to import the business services as an application or a technical components. This differentiation is done because in the CMDB everything was modeled as a business service.

This integration was the pilot integration project for information sources to the existing EA tool. Further integrations of different information sources will be implemented due to the successful development of this integration.

The main business system of the enterprise was initially documented in the CMDB. For that reason the information about the main business system was retrieved via the

CMDB to the EA tool.

The integration of the CMDB to the EA tool is seen as semi-automated since the information of the CMDB was retrieved manually.

Export to Wiki

The export of the collected data in the EA tool was mainly driven by the integration of the CMDB. The goal of the data export to the Wiki was to analyze and evaluate the data of the EA tool.

Enterprise Wikis enhances sharing and collaboration of employees knowledge. The employees can easily edit the wikis content and provide their knowledge to the information center. The unstructured information can contain file attachments, multimedia content and allows the interlinkage of wiki pages. Referencing EA relevant information such as documents is supported by wikis. To link this references wikis are more appropriate than an EA tool. [18]

To enable the collaboration and knowledge sharing of the employees knowledge a Wiki was integrated into the IT landscape. A wiki page is created for each application of the EA tool. This enables to collect more data than the metamodel of the EA tool allows. The wiki page contains several characteristics of an application. Some of the extended characteristics of an application are architectural decision, users, architectural diagrams, decommissioning information driven by the decommission concern, operational manual and other notes regarding the application.

Cloud integration

As illustrated in figure 6.1 there is no integration of the applications running on the cloud-based environments. There is no defined process of the EAD of applications deployed to the cloud.

In the context of the cloud migration project a new defined process for the deployment of the new developed applications was introduced. The major goal of this defined process is to establish a standardized process for agile teams. The process is divided into a build process and a deployment process.

The build process is shown in figure 6.2. The DevOps team first checks out the latest state of the code from the version control service (VCS). After developing the team commits and pushes the new state back to the VCS repository. The CD tool is constantly monitoring the repository to track changes in the code. If new changes are tracked, the CD tool automatically builds an artifact of the repository and registers the artifact into the binary repository manager tool. That means the built artifact is hosted in the tools and can be downloaded from it.

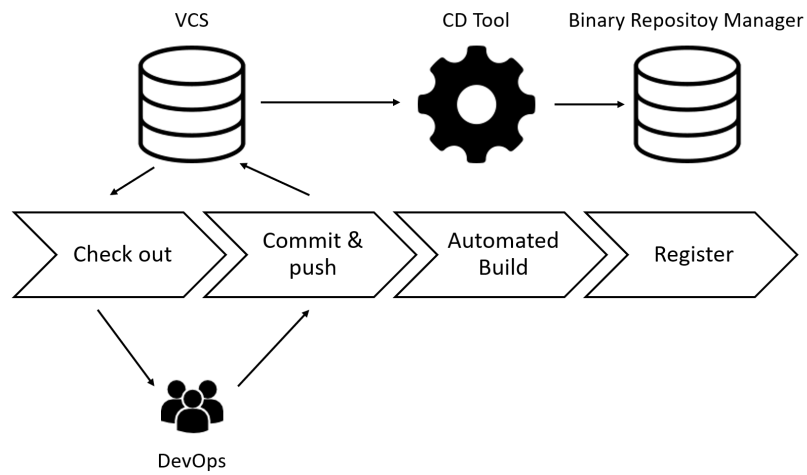


Figure 6.2.: Build process

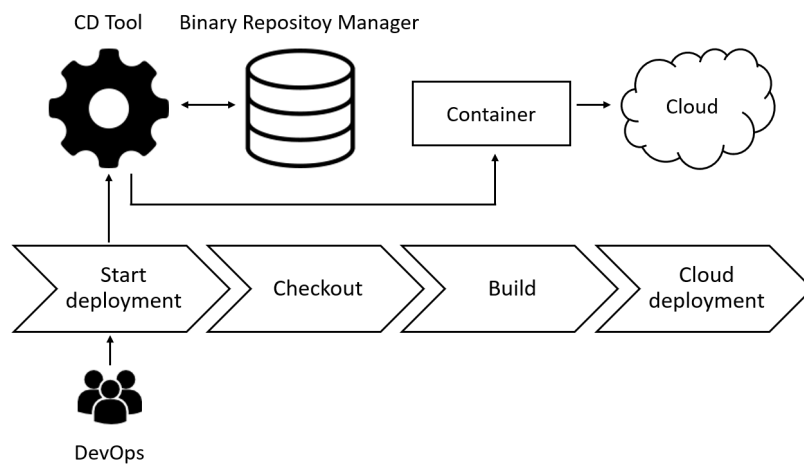


Figure 6.3.: Deployment process

The second part of the defined process is the deployment process as shown in figure 6.3. The DevOps team triggers manually the deployment process within the CD tool. The tool downloads the latest pushed artifact of the binary repository manager. After downloading the artifact, it build automatically a container out of the artifact. The CD tool pushed then the container to the cloud.

The first approach for a documentation process of the applications running on a cloud-based environment is the establishment of a defined process and a toolchain for building and deploying to the cloud infrastructure. Every application is tracked in the binary repository manager but there is still no integration of the cloud application inventory to the EA tool.

PPM Tool integration

As figure 6.1 shows, there is no integration of a PPM Tool. Projects and business information can not be aligned to applications during the EAD.

6.4. Target IT landscape

This section will explain the target IT landscape of the company. The target IT landscape is mainly driven by the concerns. The following picture shows the target IT landscape in regard to EAD.

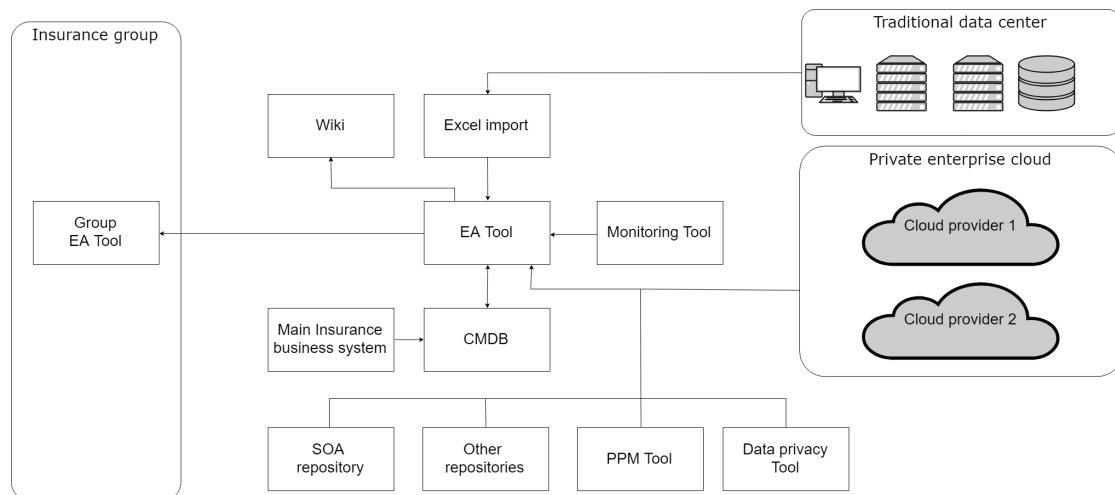


Figure 6.4.: Target IT landscape

Integration of data privacy tool

Driven by the GDPR the enterprise has developed a tool which contains a list of applications that are related to this regulation. Therefore an integration of this tool is planned in order to have this specific information in the EA tool.

Integration of service-oriented architecture repositories

An integration of a service-oriented architecture (SOA) repository is intended. The SOA repository is used for managing services such as WSDLs and XML schema definitions, access rights, information related to the service level agreements and transactional operations of the services. [31]

Integration of other repositories

The enterprise goal of the enterprise is to interlink different tools containing different EA relevant information. This goal enables a federated architecture to allow information sharing between different data sources. Therefore it will also integrate other repositories such as license management tools and change management tools.

Integration of the PPM tool

The main goal of the integration of a PPM Tool is to enable a mapping between the projects and the applications or systems in the enterprise. The EA tool in place contains in its metamodel the entity "project". This allows an integration of the PPM tool to the existing EA tool. The projects of the PPM tool are imported into the EA tools as the entity "project". A manual mapping between the projects and applications is then still required. This EAD process of projects can be seen as semi-automated.

Integration to the group EA tool

The insurance company is divided into several subsidiaries around the world. The VAIT regulation is applied to every german insurance company. The company group is based Germany so the regulation implies an IT inventory for every subsidiary of the group. Therefore every operating entity (subsidiary) has to export the EA information to the Group entity.

Cloud integration

As shown in figure 6.4 the applications running on the cloud-based environments will be integrated to the EA tool. The main driver for this integration is the VAIT regulation.

As mentioned in the regulation, an insurance company has to be able to deliver an application inventory of the IT landscape. That includes also applications running on private enterprise clouds.

Monitoring tool integration

The introduction of a monitoring tool is planned. The reason for establishing a monitoring tool is to gain advantage over the competitors. Monitoring applications allow to retrieve extra information such as individual requests and transactions, resource consumption, reporting and alerting, etc.

The business benefits of monitoring tools for the enterprise are many. One of the benefits is that the enterprise can react quicker to an application failure reducing the revenue loss due to the breakdown of the respective applications or systems. Knowing the failures and obtaining the dependencies from the monitoring tool the enterprise can derive what other systems will be affected by the breakdown. This matches the requirements from the BCM project.

6.5. Derived requirements

In the previous sections the as-is and target landscape of an insurance company were presented. These sections cover various aspects of the EAD within the studied company. The case study shows different outlooks for future integrations. From the introduced case study, requirements can be derived for an automated EAD. The following table presents the derived requirements:

Table 6.1.: Automated EAD requirements derived the case study

Id	Requirement
RC1	Integration of a PPM tool
RC2	Integration of different information sources
RC3	Integration of cloud infrastructure
RC4	Integration of a monitoring tool

RC1: The integration of a PPM tool is desired by the enterprise to relate projects to applications.

RC2: An integration of different information sources is required by the enterprise to relate information and propagate information sharing.

RC3: The integration of cloud infrastructure is needed due to a full application inventory required by law and due to strategic decisions as migrating to cloud infrastructure.

RC4: An integration of a monitoring tool is planned at the enterprise to increase the reaction time between the failure of a system and the enterprise. Thus, the monetary impact can be reduced.

The requirements from the insurance company can be aligned to the requirements derived during the literature research. **RC1** and **RC2** can be mapped to **RL1** since the requirements propose the integration of different information sources. **RC1** is just a special requirement of the integration of various data sources. **RC3** and **RL5** can be aligned since both requirements demand an integration of cloud environments. **RC4** and **RL6** require an enhancement of runtime KPIs through monitoring tools. In summary, the enterprise requirements can be aligned to the findings of the literature research and the respective derived requirements.

6.6. Case study evaluation

This section will evaluate the implemented prototype of chapter 5 within the studied enterprise. The first subsection will introduce the methodology and the overall goal of the evaluation. The second subsection will present the results of the prototype evaluation.

6.6.1. Evaluation goal and methodology

The goal is to evaluate the presented approach and the implemented prototype in the case to see whether the approach and the prototype are a valuable solution for improving the automation of EAD. To test the prototype stakeholder were interviewed. The interview was structured in three parts. The first part was an introduction of the approach. The second part of the interview was a presentation of the tool. The third part was the actual interview of the stakeholder based on an evaluation questionnaire of the developed solution. The evaluation questionnaire is attached in Appendix A.1.

Nine enterprise architects and three product owners were interviewed to obtain beneficial feedback. Table 6.2 shows the interviewed industry partners, their roles, the years of experience (YoE) as enterprise architects or product owners and the enterprises (E) they work for. The table shows that the maturity of most of the interviewed enterprise architects is high. Different to the enterprise architects, most of the product owners were not that experienced but could deliver some innovative ideas. All of the interviewed industry partners work in insurance companies.

Table 6.2.: Interviewed industry partners evaluation

No	Pseudonym	Role	YoE	E
1	EA1	Enterprise Architect and Chief Architect	20	E1
2	EA2	Enterprise Architect	2	E1
3	EA3	Enterprise Architect	17	E1
4	EA4	Enterprise Architect and Product Owner	10	E1
5	EA5	Enterprise Architect	5	E1
6	EA6	Enterprise Architect and IT Management Expert	20	E2
7	EA7	Enterprise Architect	18	E1
8	EA8	Enterprise Architect	16	E3
9	EA9	Enterprise Architect	30+	E4
10	PO1	Product Owner and Head of Product Architecture	11	E1
11	PO2	Product Owner	1	E1
12	PO3	Product Owner	3	E1

6.6.2. Approach evaluation

As mentioned before a part of the interviews was the presentation of the approach on how to automate the EAD. This subsection will give the overall results of the evaluation questionnaire regarding the approach.

At the beginning of the interview some general questions regarding EAD were asked. When the experts were asked about the effort for documenting the IT landscape of their company, 10 experts strongly agreed and 2 of them agreed. This means that the effort to document the landscape is still seen as an exhausting task.

A similar answer was given when the experts were asked about the actuality of the EA information. 10 of 12 experts answered that the EA information is outdated and 2 of them replied that it is partially outdated.

All of the interviewed experts replied that their company follow a CD approach (questions 2.1) and over 80 percent of the experts stated that their enterprises use cloud environments for deploying new developed applications or for the migration of legacy systems (question 1.6 and 1.7).

The answered of the experts during the case study and the interviews were validated during the case study.

In regard to automated EAD, the definition is not perceived the same way by the experts. 6 experts denied to have an automated process for documenting the EA while 6 agreed to the question if the company has an automated EAD. All experts know that the data from the CMDB is imported to the EA tool but the opinions on the import as

an automated EAD disagree because the data in the CMDB was gathered manually.

During the interview the experts were asked (question 2.2) to determine the responsible person for documenting new developed applications in the EA Tool. Table 6.3 shows that the role of documenting the EA is not clearly defined.

Table 6.3.: Question: Who collects the EA information of a new developed application/service for the EA Tool?

Pseudonym	EA	PO	DevOps	Other
EA1			X	
EA2	X			
EA3		X		Application owner
EA4	X	X	X	Application owner
EA5				Application owner
EA6	X	X		Application owner
EA7			X	
EA8	X			
EA9	X			Application owner
PO1	X			
PO2	X			
PO3	X			

More than half of the enterprise architects responded that the "application owner" should document the new developed applications. As a result of that, the interviewed experts were inquired (question 2.10) to explain how the application owner is determined. The application owner is the responsible person from the business side. This answer supports the business and IT alignment of the EA documentation since the experts struggle to determine the adequate person of the business part.

One of the experts replied to this questions that either you find the right information in the wiki or by "knowing the organization very well." This answer sums up and reinforces again the need for a business alignment.

Information sources

The experts were also interviewed about the information sources they use and would like to use and/or integrate. In question 2.3 the experts were asked if they use any of the mentioned information sources to retrieve EA relevant information. Figure 6.5 shows the answers of the experts regarding the usage of the EA relevant information sources according to the information sources mentioned by literature. [15]

The reason why the CMDB reached a high number of answers is because most of the interviewed industry partners belong to the german insurance company used for the evaluation which is depicted in the table as E1. The same reason explains the null value of the ESB. E1 has no established ESB in the enterprise. According to the results the other enterprises (E2, E3, E4) do not have an ESB in place. However, an ESB is considered as a n EA relevant information source. [25] Most of the experts use a PPM tool as a EA relevant information source but document the information manually in the EA tool.

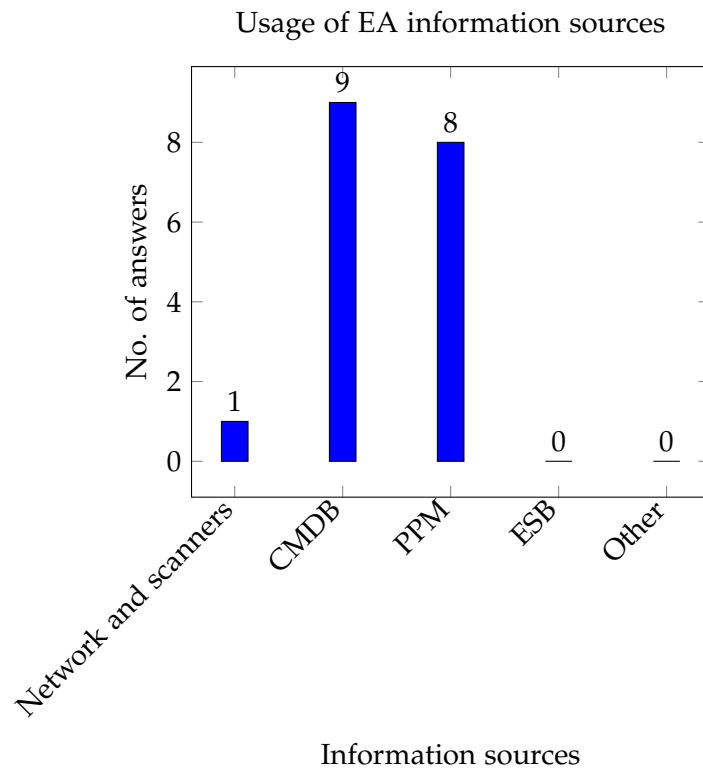


Figure 6.5.: Question 2.3: Do you use any of the mentioned information sources to retrieve EA relevant information?

In addition to question 2.3 the experts were also asked if the information sources shown in figure 6.6 also contain EA relevant data. Over 80 percent of the experts agreed. Figure 6.6 shows how often these additional information were named.

Moreover, only 3 of the experts integrate the EA relevant information of the tools into the EA tool manually (question 2.5). The rest of the experts did not give an answer to that question and only PO3 expects that the enterprise architects should document

Information sources containing possible EA information

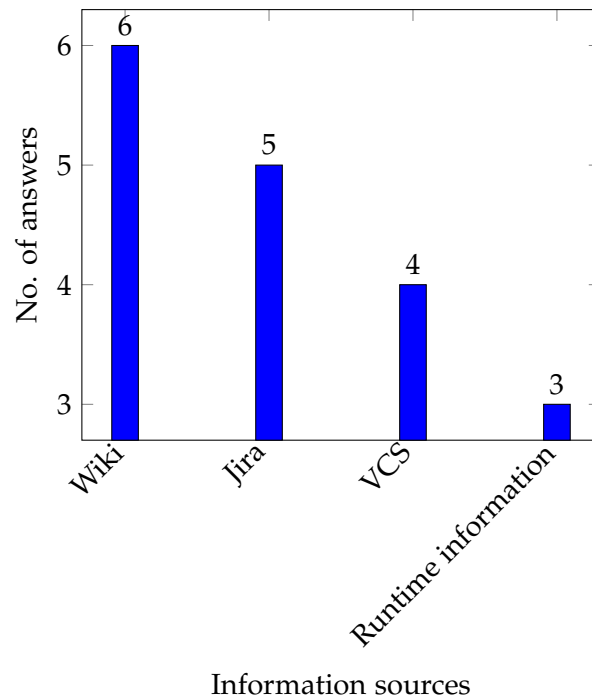


Figure 6.6.: Question 2.4: Do you think this information sources could contain relevant EA information?

the EA relevant information of the information sources mentioned in figure 6.6.

Application development pipeline

The experts were asked if their company had an established toolchain. More than 80 percent of the interviewed industry partners confirmed it. Figure 6.7 shows the approach of this work in comparison to the pipeline of the case study of section 6.1.

There are some differences regarding both approaches. The first difference is that in the pipeline of the case study the build process is triggered automatically after new changes were committed to the repository. The build process of the presented approach is triggered manually because not every change in the repository needs to generate a new artifact. After the build process is triggered in the pipeline of the presented approach the business information of the PPM tool is retrieved. This supports the statements of the interviewed experts to aggregate the information of PPM tool to an artifact. Another difference is that in the pipeline of the presented case study the built artifact is registered and stored in a Binary Repository Manager (BRM). Before starting the deployment process a container is built to wrap the application. The container is then deployed to the cloud. In the presented approach no container is built to wrap the application and the deployment process is automatically triggered. After the deployment the runtime information is gathered and it is aggregated to the other information collected throughout the pipeline. The aggregation of that information is then pushed to the tool used for an automated EAD.

All of the industry partners confirmed that an established toolchain improves the process of EA documentation (question 2.7). The most frequently named reasons why it improves the EAD are standardization and automation (question 2.7). An established toolchain automates the EAD by introducing a standardized process. This reduces the manual effort mentioned in literature. [24] It also increases the data quality (DC4) and the data actuality (DC2).

Further to question 2.6 the industry partners demanded to confirm the application of the approach mentioned in chapter 4, specially the imposition of the script for a predefined pipeline and the predefinition of the toolchain. Figure 6.8 shows the reactions to the statement if imposing the team to incorporate a pipeline-script in the repository is easy to establish.

Figure 6.9 shows the answers in regard to the imposition of a predefined toolchain. The disagreed interviewed were asked about their disagreement and the disagreement concerns rather the establishment of a toolchain within a large enterprise, not about imposing the agile development teams to use a predefined toolchain.

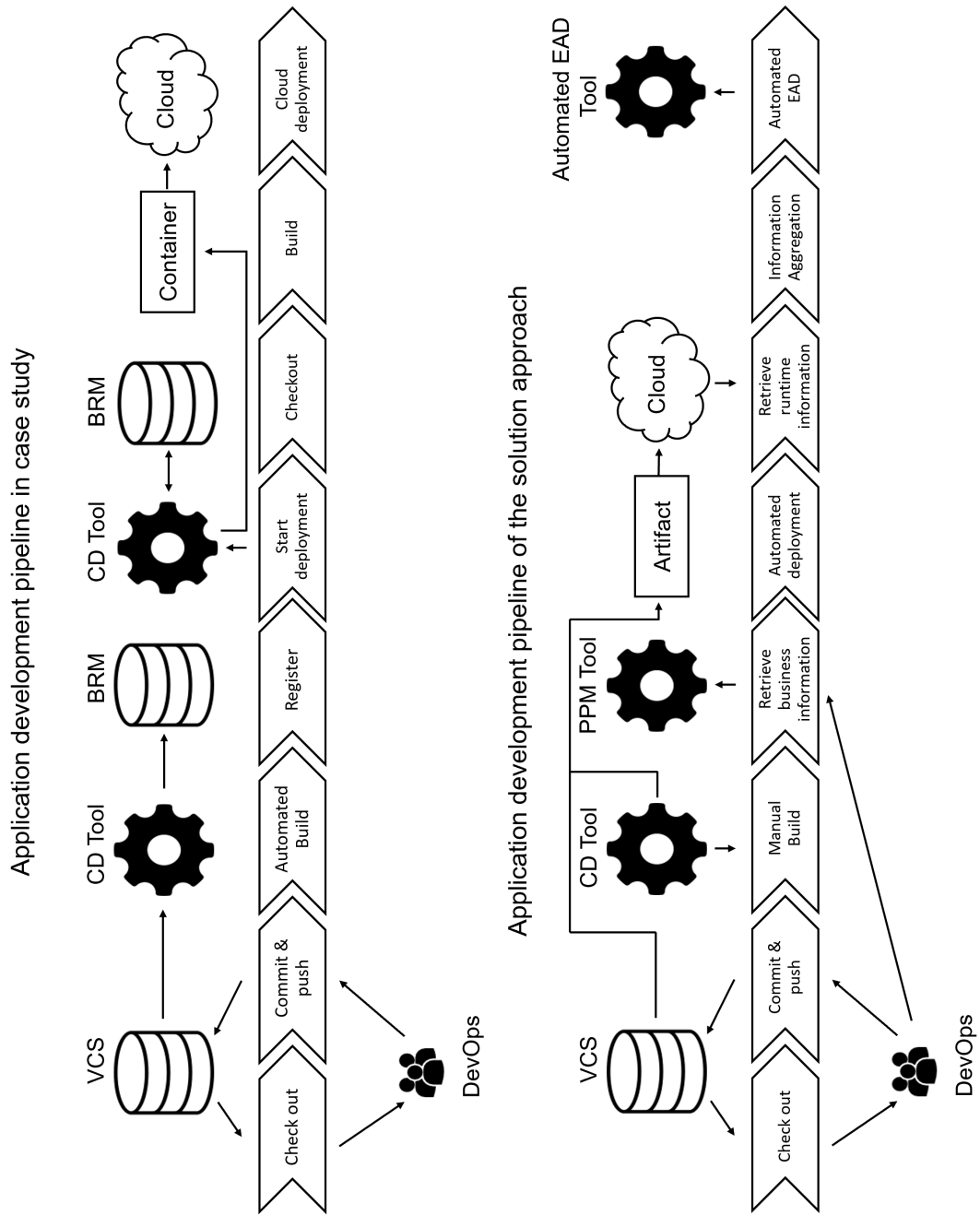


Figure 6.7.: Comparison of approaches

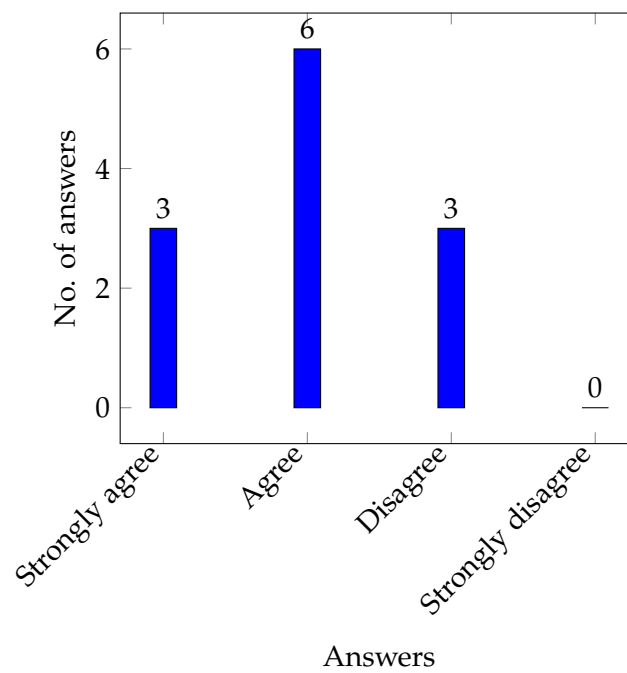


Figure 6.8.: Question 2.8: Imposing the team to incorporate a pipeline-script in the repository is easy to establish

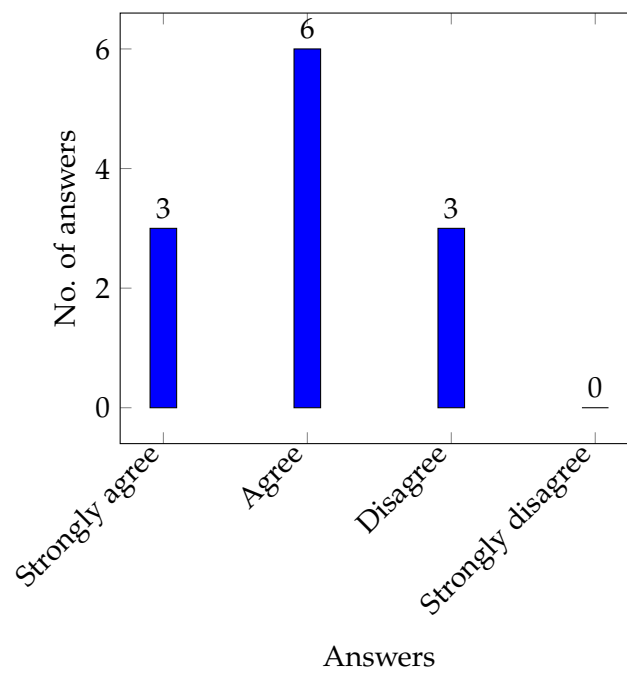


Figure 6.9.: Question 2.9: Imposing the team to use a predefined toolchain for the application development is easy to establish

Federated EA

Regarding a federated enterprise architecture all experts confirmed that the approach of this work enables it. 50 percent of the industry partners agreed to it and 50 percent strongly agreed.

Cloud EAD

8 of the interviewed industry partners use technologies for monitoring applications and the rest do not use technologies for monitoring applications or do not know about the usage of these technologies (question 3.5).

During the interviews the experts were asked to list the monitoring technologies. The following "monitoring technologies" were listed:

- Grafana⁸
- Tivoli⁹
- Prometheus¹⁰
- ELK stack¹¹
- Dynatrace¹²

Grafana was the most mentioned technology. This shows that the experts do not know the definition of monitoring technologies. Grafana is a platform that allows the user to visualize, alert on and understand the applications metrics. Grafana does not monitoring the application itself. The ELK Stack is consists three open-source products: Elasticsearch, Logstash, and Kibana. Elasticsearch is a NoSQL database, Logstash is a log pipeline tool and Kibana is a visualizes the Elasticsearch data. None of the mentioned products monitors the applications. However, the other listed technologies do monitor applications. Tivoli is a set of products from IBM that monitors the performance and availability of operating systems and applications. Dynatrace monitors real user data, application performance, infrastructure and cloud environments. Prometheus is an systems monitoring and alerting toolkit.

When it comes to the EAD of cloud applications 4 out of 8 industry partners using technologies for monitoring applications still document cloud applications manually.

⁸<https://grafana.com/>

⁹https://www.ibm.com/support/knowledgecenter/en/SSTFXA_6.3.0/com.ibm.itm.doc_6.3/welcome_63.htm

¹⁰<https://prometheus.io/>

¹¹<https://www.elastic.co>

¹²<https://www.dynatrace.com/>

All of the interviewed experts confirmed that the presented process automates the EA documentation of applications/services running on a cloud-based environment (question 5.1). 3 experts strongly agreed to the statement and 9 agreed.

6.6.3. Tool evaluation

This subsection will give an overview of the results of the evaluation regarding the implemented prototype. The same experts of table 6.2 were consulted about the implemented prototype for further development and improvement.

Solution architecture of prototype

Figure 6.10 shows the solution architecture of the prototype in the current IT environment of the insurance company.

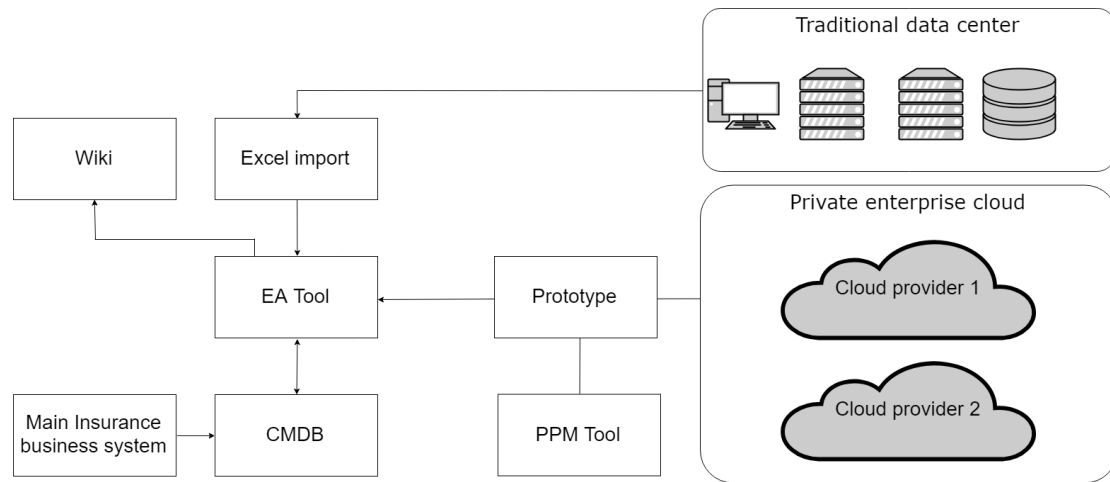


Figure 6.10.: Solution architecture of the prototype in the current IT landscape of the enterprise

Challenges of the prototype

The challenges of the prototype include governance aspects and implementation issues.

The first problem during the establishment of the tool was that the cloud where the prototype was deployed, did not allow to deployment and set-up of a MongoDB instance. Due to this reason a MySQL¹³ database component was used in the case

¹³<https://www.mysql.com/>

study. The consequences of this was the adaptation the server component. The web component also needed some adjustments. The web-browser allowed at the company did not ECMAScript 6¹⁴. Therefore the visualizations such as the diagrams shown in the visualizations view and the filtertree were not displayed.

To achieve a fully discovery of the cloud environment and the documentation of it a access to the whole environment needs to be ensured. During the establishment of the prototype this was not ensured therefore a complete discovery and documentation of the cloud infrastructure could not be fulfilled.

General section

This section shows the results of the interviews regarding the general section. Figure 6.11 shows what the experts considered not useful in the general section.

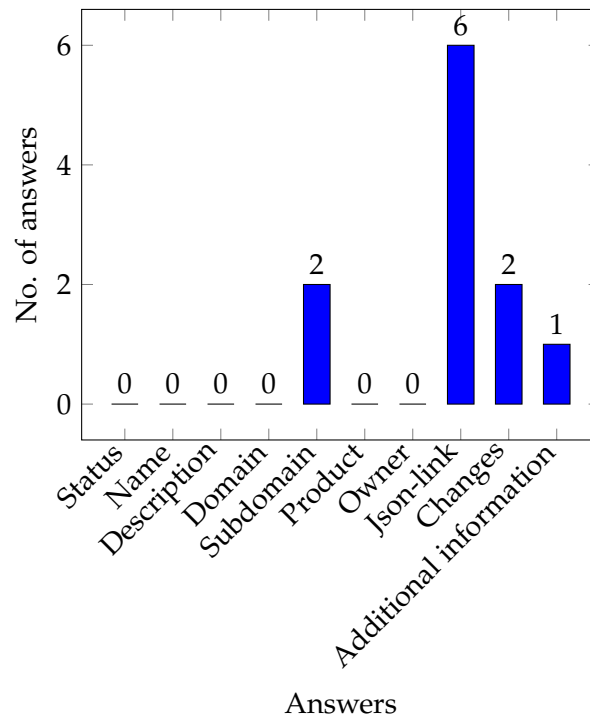


Figure 6.11.: Question 4.4: The following information displayed in the general section of the detailed view of an application/service is NOT useful

Json-link: 50 percent of the experts considered the json-link as not useful. This attribute showed the json-response of the database. It contained every key-value pair

¹⁴<http://www.ecma-international.org/>

stored in the database. This result was expected since enterprise architects and product owners are not interested in every detail of a database entry. Therefore the attribute was removed from the general section.

Changes: EA7 and PO3 replied that the changes of an artifacts regarding the last documentation status and deployment status are not relevant. When the experts were asked what attributes they consider useful EA1 and EA3 did not mark the changes attribute. In question 1.3 more than 80 percent stated that the information in a EA tool is outdated and the rest considered the information as partially outdated. If the experts do not consider the changes as useful how can an enterprise architect determine the state (actual or outdated) of the EA information? This result was not expected.

Runtime section

9 experts considered every information as useful. 2 experts marked the instances as useful and everything else as not useful. The running instances of an application running on the cloud is indicates an increase in user load and concurrent requests ergo it is an indicator for the traffic on the application. Therefore 2 experts were only interested in the instances. EA7 was not interested in any of the KPIs shown in this section but expected the running costs to be displayed since the costs are calculated based on the resources consumption. Considering that enterprise architects and product owners demanded costs during the interviews, the running costs of the application was included to this section. The amount of http-calls was also included to this section because it was also demanded during the interviews.

Metrics monitoring section

8 of the 12 experts interviewed contemplate the section as useful. 2 experts considered the response time and the url as important. PO2 only considered the response time as valuable and PO1 marked as well the response time as the Prometheus¹⁵ link (monitoring agent) as relevant.

Services section

11 experts see this section as useful since it shows the communication to other services running in the cloud environment. Only 1 of the experts did not give any statements regarding the usage of this section.

¹⁵<https://prometheus.io/>

Software dependencies section

All of the experts stated that this section is important due to several reasons:

- Indicator for the complexity of the software
- Indicator for dependencies to third party software
- Management of software frameworks and technologies throughout the whole enterprise
- Detection of outdated components

This results were not expected from the enterprise architects since enterprise architects are mostly interested in a holistic view of the IT landscape. As well enterprise architects as product owners pointed this section as crucial.

Jira monitoring section

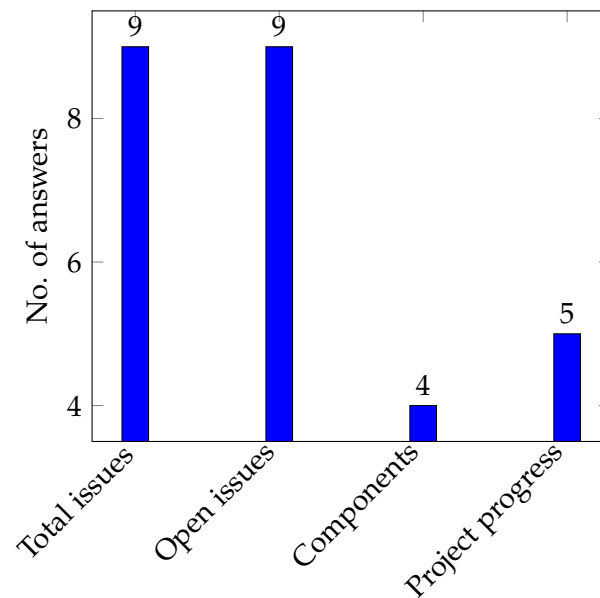
The opinions about the usage of the Jira monitoring section were different. Figure 6.12 shows the different answers. The reason why "project progress" was not very often perceived as useful is that there was no added value since the total issues and open issues were already displayed. More information about the individual issues were required such as links or status of the issues. Depending on the size of the project the visualization of individual issue information can result in an enormous amount of information just in this section.

Github monitoring section

3 the interviewed experts regard the display of the contributors as delicate due to compliance reasons. 2 of the industry partners did not see the "lines of code" as a valuable indicator. The static analysis of code quality and continuous inspection to perform automatic reviews was named several times. Other metrics mentioned were information on programming languages, cyclomatic complexity and more complex indicators.

Jenkins section

More than half of the interviewed experts replied that all metrics displayed in the Jenkins section are relevant. The result of the last job and the duration were the most ticked answers.



Information displayed in Jira monitoring section

Figure 6.12.: Question 4.14: The following information in the Jira monitoring section is useful

Visualizations section

The presented visualizations of the tool were perceived as useful. EA5 mentioned that an additional cluster diagram would be useful as well. PO1 also mentioned a pie chart as an additional visualization. Both requirements are justified but the main idea of the tool is to export the data to the EA tool in place for additional visualizations.

Additional requirements

During the evaluation additional requirements for the tool were named. The evaluation questionnaire was extended some the questions regarding the requirements. Table 6.4 lists the additional requirements.

Table 6.4.: Automated EAD requirements derived the case study

Id	Requirement
AR1	Business Impact Analysis of applications
AR2	Data privacy compliance
AR3	Automated verification of the 12 factor app
AR4	Automated verification of a resilience pattern
AR5	Integration of the architecture belt
AR6	Additional KPIs

AR1: A Business Impact Analysis of applications is not only required by the enterprise, it is also required due to regulations. The enterprise can analyze in case of a sinister what applications are affected and which related systems are implied by a failure. It can quantify the economic impact of failures.

AR2: Tools have to be compliant with the data privacy regulation. An enterprise needs to know what applications store personal data.

AR3: The presented approach and tool enables an automated EAD of cloud applications. Therefore a static analysis of the code quality regarding the suitability for deployment on cloud platforms was required. The required methodology for verifying this suitability is the 12 factor methodology¹⁶. 90 percent of the experts declared that this methodology would indicate if the application is cloud-ready or not (question 4.22).

Due to time constraints a fully implementation of the 12 factor app criteria was not possible. The following paragraphs show the status of the implemented criteria. An improved implementation is seen as valuable for future implementations.

¹⁶<https://12factor.net/>

- **I. Codebase:** Is the artifact linked to a repository? Since the build and deployment pipeline is only possible if a repository is linked the CD tool this criteria is always fulfilled.
- **II. Dependencies:** Configuration files are searched within the repository. The configuration files should declare software-dependencies, if this is not the case, this criteria is not fulfilled.
- **III. Config:** This criteria is fulfilled if configuration files are found. Depending on the programming language or framework specific configurations files are predetermined. The search of the configuration files depends on that. Example configuration files are yaml-files, application.properties, package.json, etc.
- **IV. Backing Services:** In some cloud environments the backend services are attached to an artifact. In Cloudfoundry, the specific environment used for this work, the attached services can be queried. If an attached services is a database the criteria is fulfilled.
- **V. Build, release, run:** Version numbers and release numbers are search within the repository. This search does not accord the specification of the criteria but there is no standard process for versioning and release management in the evaluated enterprise. The search for a version number or a release number was considered useful.
- **VI. Processes:** Cloudfoundry is used as the cloud infrastructure. This environment does not allow that the the app is executed as one or more processes. Therefore this criteria is always fulfilled.
- **VII. Port binding:** The declaration of static ports within the code is not allowed. A text search was implemented to very this criteria. If static ports are declared within the code, this criteria is not fulfilled.
- **VIII. Concurrency:** Not implemented.
- **IX. Disposability:** Not implemented.
- **X. Dev/prod parity:** The configuration files for the development, staging, and production environment are searched within the repository. These files should be as similar as possible. Depending on the programming language there are some conventions for the specification of these files. For example, in the case of Springboot applications, the configurations files are usually stored with the prefix "dev" and "prod". If the files are found, the similarity has to be compared. A text

similarity can be implemented for the comparison. The comparison of the attached services or tools has not been implemented.

- **XI. Logs:** This criteria is fulfilled if the artifact does not use log dependencies.
- **XII. Admin processes:** Not implemented.

AR4: Due to a connection to the code repository of the developed artifact a continuous inspection is enabled. The code quality can be automatically reviewed with static code analysis. Therefore enterprise architects required an integration of the following criteria defined by the resilience pattern of the studied enterprise:

- Is the application running on the enterprise cloud?
- Redundancy checks
- Zero downtime deployment verification
- Retry verification
- Isolation verification
- Caching verification
- Fallback verification
- Loose coupling degree calculation

All answers of question 4.23 affirm that an automated test of the resilience pattern is helpful to determine the elasticity of an application.

AR5: The architecture belt is a web application that enables and support the a collaborative approach to establish architecture principles and guidelines. The product owners expressed during the interviews that it is challenging for the team to follow architecture guidelines because these are not well transmitted by the enterprise architects. All experts of the case study asserted that a combination with this tool would also display the overall status of an application regarding the compliance with the architecture guidelines.

Figure 6.13 shows an additional section implemented due to **AR3**, **AR4** and **AR5**.

AR6: Additional KPIs were mentioned during the interviews. Some of the most mentioned KPIs are: Total costs of ownership, mean time between failures, mean time to repair, mean time to failure and real time data of users on the individual applications such as average time per user on app, individual API calls, etc.

Governance Monitoring	
The twelve-factor methodology I. Codebase Yes II. Dependencies Yes III. Configuration Yes IV. Backing services Yes V. Build, release, run Yes VI. Processes Yes VII. Port binding No VIII. Concurrency Not implemented IX. Disposability Not implemented X. Dev/prod parity No XI. Logs Yes XII. Admin processes Not implemented	
Architecture belt No architecture belt for this application	
Resilience I. Enterprise Cloud No II. Redundancy No III. Zero Downtime deployment No IV. Retry No V. Isolation No VI. Caching No VII. Fallback No VIII. Loose Coupling Not implemented	

Figure 6.13.: Additional implemented section: Governance monitoring section

6.6.4. Evaluation summary

As mentioned before 100 percent of the interviewed industry partners stated that the presented process automates the EA documentation of the artifacts running a cloud infrastructure (question 5.1).

Regarding the presented tool the opinions about the different sections were not always aligned. The sections cover different information for different stakeholder. This statement was supported by question 4.1. Over 90 percent of the experts confirmed this.

The experts were asked to value the approach in combination with the tool. The overall score 4.17 was achieved.

7. Limitations

This chapter describes the limitations emerged during this work.

7.1. Technical challenges

During the establishment of the prototype in the enterprise analyzed during the case study some technical challenges prevented an easy integration.

The first challenge was the web browser used in the company did not support ECMAScript 6. The visualizations implemented in the web component needed to be adapted to plain javascript.

The server component also needed some adaptations since the actual prototype uses a Nodejs server connected to a MongoDB database. This database was not supported by the enterprise so the server was adapted to be connected to a MySQL database.

7.2. Governance challenges

After solving the technical challenges some governance challenges emerged that avoided a full cloud discovery.

To accomplish a full discovery of the cloud environment and thus enable an EA documentation of the infrastructure, access rights to the whole environment needs to be guaranteed. This was not ensured therefore a complete discovery and EA documentation of the cloud infrastructure could not be fulfilled.

The integration of a pipeline script to enable an automated EAD of a new developed application was not tested in a pilot project due to time and governance constraints.

8. Conclusion

This final chapter of this work presents the summary of this thesis and gives an outlook for further work.

8.1. Summary

This section presents the results for the research questions introduced in section 1.2.

RQ1. How to obtain EA relevant information from the runtime behavior of cloud based environments? Most cloud infrastructures provide runtime information without the installation of additional monitoring agents. Thus, Shadow IT is prevented. However installing additional monitoring agents unveil further information like the total amount of API requests.

RQ2. How to assign the application landscape to business domains? To enable the assignment of the application landscape to the business domains there are two possibilities. The first one is to add the business specific information to a configuration file so the information of the configuration file is used for the assignment. The developers have to manually maintain the configuration file. The second possibility is to integrate the PPM tool in the Build-Deployment pipeline. For this possibility the PPM tool needs to provide the business domain information. This also has to be maintained manually by the developers or the product owners using the PPM tool.

RQ3. How to automate the assignment process with an integrated toolchain? The automation of the assignment process can be achieved through a configuration file which contains the links to the relevant tools or through name mapping. The usage of the configuration file results in producing further overhead. Therefore the mapping via name is more popular, however the names need to be stable and unique.

RQ4. How does a prototype implementation of the automated documentation process of cloud applications look like? The prototype implementation of the automated documentation process of cloud applications is explained and illustrated in chapter 5.

8.2. Future work

This section presents the possible extensions of the prototype that were not implemented due to time constraints. The future use cases were demanded by the industry partners during the evaluation.

The first additional implementation is to include other cloud environments to enable the transferability to other companies.

During this work the PPM tool was integrated during the build and deployment pipeline to include the business domain information. An further extension would be to include a mapping of business processes and capabilities for an Business Impact Analysis of the applications.

During the evaluation two new code related use cases were requested by the experts. The first use case was the automated verification of the cloud readiness of an application by verifying the 12 factor app criteria. The other use cases was an automated elasticity evaluation through a complete implementation of resilience pattern. To improve both automated verifications an integration a continuous inspection tool to perform automatic reviews with static code analysis would enhance both use cases. A possible tool for this could be Sonarqube¹.

Another requested extension of the prototype is to include an automated Data privacy compliance (GDPR compliance) analysis. Knowing what applications actually store information enables the opportunity to analyze the data that is stored.

To prove that the approach and prototype can automate the EAD of enterprises, access rights to the whole cloud infrastructure needs to be guaranteed and the concept needs to be tested in several pilot projects.

¹<https://www.sonarqube.org/>

A. Appendix

A.1. Evaluation questionnaire

1. General data

Company: _____ Date and time: _____

Interviewer: _____ Questionnaire-Nr.: _____

Years of experience: _____ Industry sector: _____

1. Which of the below roles are applicable for you?

☐Enterprise Architect ☐Product Owner ☐DevOps ☐Other: _____

2. It is a lot of effort to document the IT landscape of your company

☐Strongly agree ☐agree ☐disagree ☐Strongly disagree

3. The information in the EA tool is outdated

☐Yes ☐No

4. Does your company have an automated process for documenting the EA?

☐Yes ☐No

5. If your company has an automated process for documenting the EA, please describe the process:

6. Does your company run applications/services in a cloud-based environment?

☐Yes ☐No

7. Does your company consider migrating legacy systems into a cloud-based environment?

☐Yes ☐No

2. Agile methodologies

8. Does your company use a Continuous Delivery approach?

☐Yes ☐No

9. Who collects the EA information of a new developed application/service for the EA Tool?
☐Enterprise Architect ☐Product Owner ☐DevOps ☐Other: _____
10. Do you use any of the mentioned information sources to retrieve EA relevant information?
☐Network scanners and monitors ☐CMDB ☐PPM ☐ESB ☐Other: _____
11. Do you use any of the mentioned information sources to retrieve EA relevant information?
☐Runtime environment ☐Jira ☐Github ☐Jenkins ☐Wiki
12. Do you think this information sources could contain relevant EA information? Why?

13. How do you collect this information?

14. Does your company have an established toolchain for the development pipeline?
☐Yes ☐No
15. Do you think an established toolchain improve the process of EA documentation?Why?
☐Yes ☐No

16. Imposing the team to incorporate a pipeline-script in the repository is easy to establish
☐Strongly agree ☐agree ☐disagree ☐Strongly disagree
17. Imposing the team to use a predefined toolchain for the application development is easy to establish
☐Strongly agree ☐agree ☐disagree ☐Strongly disagree
18. How do you determine an application owner?

19. Do you follow any guidelines when developing a new product?

3. Cloud environments

20. Does your company use any technologies for automating the EA documentation process?

☐Yes ☐No

21. How does your company collect the applications running in a cloud-based environment?

☐Yes ☐No

22. Does your company have any architecture guidelines for cloud applications?

☐Yes ☐No

23. Are you familiar with the 12-factor-app criteria for determining if an application/product is cloud-ready?

☐Yes ☐No

24. Do you use any technologies for monitoring applications?

☐Yes ☐No

25. If yes, please give a short answer regarding the technologies used for monitoring the applications/services:

4. Tool Evaluation

26. The presented prototype covers different views for the different stakeholders such as product owners, enterprise architects and devops-teams

☐Strongly agree ☐agree ☐disagree ☐Strongly disagree

27. The presented prototype enables a federated approach linking the tools used during an application development pipeline

☐Strongly agree ☐agree ☐disagree ☐Strongly disagree

28. The following information displayed in the general section of the detailed view of an application/service is useful

☐Status ☐Name ☐Description ☐Domain ☐Subdomain
☐Product ☐Owner ☐Json link ☐Changes ☐Additional information

29. The following information displayed in the general section of the detailed view of an application/service is NOT useful.Why?

☐Status ☐Name ☐Description ☐Domain ☐Subdomain
☐Product ☐Owner ☐Json link ☐Changes ☐Additional information

30. I would like to see the following information displayed in the general section:

31. The following information in the runtime section is useful

☐Instances ☐Ram ☐CPU ☐Disk ☐Host

32. The following information in the runtime section is NOT useful. Why?

☐Instances ☐Ram ☐CPU ☐Disk ☐Host

33. I would like to see the following information displayed in the runtime section:

34. The following information displayed in the metrics section is useful

☐URL ☐Prometheus metrics endpoint ☐Response time

35. The following information displayed in the metrics section is NOT useful. Why?

☐URL ☐Prometheus metrics endpoint ☐Response time

36. I would like to see the following information displayed in the metrics section:

37. The information displayed in the Application/Services section is useful.If no, why?

☐Yes ☐No

38. The information displayed in the software dependencies section is useful. Why?

☐Yes ☐No

39. The following information in the Jira monitoring section is useful:

☐Total issues ☐Open issues ☐Components ☐Project progress

40. The following information in the Jira monitoring section is NOT useful. Why?

☐Total issues ☐Open issues ☐Components ☐Project progress

41. I would like to see the following information displayed in the jira section:

42. The following information in the github monitoring section is useful:

☐Contributors ☐Lines of Code ☐Commit activity

43. The following information in the github monitoring section is NOT useful. Why?

☐Contributors ☐Lines of Code ☐Commit activity

44. What information would you like to see in the Github monitoring section regarding the repository of the application?

45. The following information displayed in the Jenkins job monitoring section is useful:

☐Build number ☐Duration ☐Estimated duration ☐Result ☐Timestamp

46. The following information displayed in the Jenkins job monitoring section is NOT useful. Why?

☐Build number ☐Duration ☐Estimated duration ☐Result ☐Timestamp

47. An automated 12-factors evaluation of the application/service is helpful to evaluate if the application/service is cloud-ready:

☐Yes ☐No

48. An automated test of the resilience pattern is helpful to determine the elasticity of an application/service:

☐Yes ☐No

49. An integration of the architecture belt tool would improve the governance monitoring section:

☐Yes ☐No

50. The presented sections represent relevant EA information sources

☐Yes ☐No

51. Would you include other information sources?

☐Yes ☐No

52. If yes please describe briefly why:

53. The visualizations displayed in the visualizations tab is useful:

☐Yes ☐No

54. I would like to see the following visualizations:

55. Other comments:

5. Overall

56. The presented process will automate the EA documentation of application-s/services running on a cloud-based environment

☐Strongly agree ☐agree ☐disagree ☐Strongly disagree

57. Overall score of the tool: Being 5 the best

☐1 ☐2 ☐3 ☐4 ☐5

Bibliography

- [1] F. Ahlemann, E. Stettiner, M. Messerschmid, and C. Legner. *Strategic Enterprise Architecture Management Challenges, Best Practices, and Future Developments*. 2012. ISBN: 9783642242229. DOI: 10.1007/978-3-642-24223-6.
- [2] V. Anforderungen. "Rundschreiben 10/2018." In: (2018), pp. 1–23.
- [3] L. Badger, T. Grance, R. Patt-corner, and J. Voas. *Cloud Computing Synopsis and Recommendations Recommendations of the National Institute of Standards and Technology*. Tech. rep. 2012. DOI: 2012.
- [4] A. Balalaie, A. Heydarnoori, and P. Jamshidi. "Microservices Architecture Enables DevOps." In: *IEEE Software* 33.3 (2016), pp. 42–52. ISSN: 0740-7459. DOI: 10.1109/MS.2016.64. arXiv: 1606.04036.
- [5] J. Bogner and A. Zimmermann. "Towards Integrating Microservices with Adaptable Enterprise Architecture." In: *Proceedings - IEEE International Enterprise Distributed Object Computing Workshop, EDOCW 2016-Septe* (2016), pp. 158–163. ISSN: 15417719. DOI: 10.1109/EDOCW.2016.7584392.
- [6] S. Buckl, F. Matthes, C. Neubert, and C. M. Schweda. *A Wiki-Based Approach to Enterprise Architecture Documentation and Analysis*. Tech. rep. 2009.
- [7] S. Buckl, F. Matthes, R. Ramacher, and C. M. Schweda. "Towards a Language for Enterprise Architecture Documentation and Analysis—Extending the Meta Object Facility." In: *The 4th International Workshop on Vocabularies, Ontologies and Rules for The Enterprise (VORTE 2009)* (2009). ISSN: 2078-0958. DOI: 10.1016/S0377-2217(03)00296-0. arXiv: 0302031 [math].
- [8] M. Buschle, S. Grunow, F. Matthes, M. Ekstedt, and S. Roth. *Automating Enterprise Architecture Documentation using an Enterprise Service Bus Americas Conference on Information Systems Automating Enterprise Architecture Documentation using an Enterprise Service Bus*. Tech. rep. 1. 2012.
- [9] M. Canat, N. P. Català, A. Jourkovski, S. Petrov, M. Wellme, and R. Lagerström. "Enterprise Architecture and Agile Development - Friends or Foes?" In: *IEEE 22nd International Enterprise Distributed Object Computing Workshop*. 2018.

- [10] L. Chen. "Continuous Delivery: Overcoming adoption challenges." In: *Journal of Systems and Software* 128 (2017), pp. 72–86. ISSN: 01641212. DOI: 10.1016/j.jss.2017.02.013.
- [11] L. Chen. "Towards Architecting for Continuous Delivery." In: *Proceedings - 12th Working IEEE/IFIP Conference on Software Architecture, WICSA 2015* (2015), pp. 131–134. ISSN: 1754-2189. DOI: 10.1109/WICSA.2015.23.
- [12] L. Chen. "Towards Architecting for Continuous Delivery." In: *Proceedings - 12th Working IEEE/IFIP Conference on Software Architecture, WICSA 2015*. 2015. ISBN: 9781479919222. DOI: 10.1109/WICSA.2015.23.
- [13] P. Drews, I. Schirmer, B. Horlach, and C. Tekaat. "Bimodal enterprise architecture management: The emergence of a New EAM function for a BizDevOps-based fast IT." In: *Proceedings - IEEE International Enterprise Distributed Object Computing Workshop, EDOCW 2017-October* (2017), pp. 57–64. ISSN: 15417719. DOI: 10.1109/EDOCW.2017.18.
- [14] M. Farwick, B. Agreiter, R. Breu, M. Häring, K. Voges, and I. Hanschke. "Towards living landscape models: Automated integration of infrastructure cloud in Enterprise Architecture Management." In: *Proceedings - 2010 IEEE 3rd International Conference on Cloud Computing, CLOUD 2010* (2010), pp. 35–42. ISSN: 2159-6182. DOI: 10.1109/CLOUD.2010.20.
- [15] M. Farwick, R. Breu, M. Hauder, S. Roth, and F. Matthes. "Enterprise architecture documentation: Empirical analysis of information sources for automation." In: *Proceedings of the Annual Hawaii International Conference on System Sciences* (2013), pp. 3868–3877. ISSN: 15301605. DOI: 10.1109/HICSS.2013.200.
- [16] M. Farwick, C. M. Schweda, R. Breu, and I. Hanschke. "A situational method for semi-automated enterprise architecture documentation (SoSyM abstract)." In: *2015 ACM/IEEE 18th International Conference on Model Driven Engineering Languages and Systems, MODELS 2015 - Proceedings* (2015), p. 448. ISSN: 16191374. DOI: 10.1109/MODELS.2015.7338278.
- [17] M. Farwick, C. M. Schweda, R. Breu, and I. Hanschke. "A situational method for semi-automated enterprise architecture documentation (SoSyM abstract)." In: *2015 ACM/IEEE 18th International Conference on Model Driven Engineering Languages and Systems, MODELS 2015 - Proceedings* (2015), p. 448. ISSN: 16191374. DOI: 10.1109/MODELS.2015.7338278.
- [18] M. Fiedler, M. Hauder, and A. W. Schneider. "Foundations for the Integration of Enterprise Wikis and Specialized Tools for Enterprise Architecture Management." In: *11th International Conference on Wirtschaftsinformatik Student Track (WI-SC) March 2013* (2013), pp. 1–15.

- [19] L. Filion, N. Daviot, J. P. Le Bel, and M. Gagnon. "Using Atlassian tools for efficient requirements management: An industrial case study." In: *11th Annual IEEE International Systems Conference, SysCon 2017 - Proceedings*. 2017. ISBN: 9781509046225. DOI: 10.1109/SYSCON.2017.7934769.
- [20] U. Frank, D. Heise, and H. Kattenstroth. "Use of a Domain Specific Modeling Language for Realizing Versatile Dashboards." In: *Proceedings of the 9th OOPSLA workshop on domain-specific modeling (DSM)* (2009), p. 8.
- [21] I. Hadar, S. Sherman, E. Hadar, and J. J. Harrison. "Less is more: Architecture documentation for agile development." In: *2013 6th International Workshop on Cooperative and Human Aspects of Software Engineering, CHASE 2013 - Proceedings*. 2013. ISBN: 9781467362900. DOI: 10.1109/CHASE.2013.6614746.
- [22] I. Hanschke. "Enterprise Architecture Management - einfach und effektiv : ein praktischer Leitfaden für die Einführung von EAM." In: *Enterprise Architecture Management - einfach und effektiv : ein praktischer Leitfaden für die Einführung von EAM*. 2011. ISBN: 978-3-446-42694-8. DOI: doi:10.3139/9783446429369.fm.
- [23] S. Hanschke, J. Ernsting, and H. Kuchen. "Integrating agile software development and enterprise architecture management." In: *Proceedings of the Annual Hawaii International Conference on System Sciences 2015-March* (2015), pp. 4099–4108. ISSN: 15301605. DOI: 10.1109/HICSS.2015.492.
- [24] M. Hauder, F. Matthes, and S. Roth. "Challenges for automated enterprise architecture documentation." In: *Lecture Notes in Business Information Processing 131 LNBIP* (2012), pp. 21–39. ISSN: 18651348. DOI: 10.1007/978-3-642-34163-2_2.
- [25] M. Hauder, S. Roth, F. Matthes, A. Lau, and H. Matheis. "Supporting Collaborative Product Development through Automated Interpretation of Artifacts." In: *Proceedings of the BMSD 2013 – Third International Symposium on Business Modeling and Software Design, Noordwijkerhout, The Netherlands, 8-10 July 2013* 19 (2013), pp. 151–156.
- [26] H. Holm, M. Buschle, R. Lagerström, and M. Ekstedt. "Automatic data collection for enterprise architecture models." In: *Software and Systems Modeling* 13.2 (2014), pp. 825–841. ISSN: 16191374. DOI: 10.1007/s10270-012-0252-1.
- [27] Isaac Odun-Ayo, Frank Agono and S. Misra. "Cloud Migration - Issues and Developments." In: *Proceedings of the International MultiConference of Engineers and Computer Scientists 2018* (2018).

- [28] P. Johnson, M. Ekstedt, and R. Lagerstrom. "Automatic Probabilistic Enterprise IT Architecture Modeling: A Dynamic Bayesian Networks Approach." In: *Proceedings - IEEE International Enterprise Distributed Object Computing Workshop, EDOCW 2016-Septe* (2016), pp. 122–129. issn: 15417719. doi: 10.1109/EDOCW.2016.7584351.
- [29] H. Jonkers, M. M. Lankhorst, H. W. Ter Doest, F. Arbab, H. Bosma, and R. J. Wieringa. "Enterprise architecture: Management tool and blueprint for the organisation." In: *Information Systems Frontiers* 8.2 (2006), pp. 63–66. issn: 13873326. doi: 10.1007/s10796-006-7970-2.
- [30] S. Kaisler, F. Armour, and M. Valivullah. "Enterprise Architecting: Critical Problems." In: *Proceedings of the 38th Annual Hawaii International Conference on System Sciences* 00.C (2005), 224b–224b. issn: 1530-1605. doi: 10.1109/HICSS.2005.241.
- [31] D. Krafzig, K. Banke, and D. Slama. *Enterprise SOA: Service-Oriented Architecture Best Practices*. 2005. isbn: 9780131465756. doi: Doi10.1097/00007890-200009150-00004.
- [32] J. Landthaler, Ö. Uludağ, G. Bondel, A. Elnaggar, S. Nair, and F. Matthes. "A machine learning based approach to application landscape documentation." In: *Lecture Notes in Business Information Processing* 335 (2018), pp. 71–85. issn: 18651348. doi: 10.1007/978-3-030-02302-7_5.
- [33] C. Lucke, S. Krell, and U. Lechner. "Critical Issues in Enterprise Architecting – A Literature Review." In: *16th Americas Conference on Information Systems (AMCIS)* June 2015 (2010), p. 11. issn: 0370-1662. doi: 10.1098/rsp1.1856.0144.
- [34] F. Matthes, M. Hauder, and N. Katinszky. "Enterprise Architecture Management Tool Survey 2014 Update." In: (2014).
- [35] P. F. Matthes, S. Buckl, J. Leitel, C. M. Schweda, E. Denert-stiftungslehrstuhl, and F. Matthes. *Enterprise Architecture Management Tool Survey 2008 About sebis*. 2008. isbn: 9783000245206.
- [36] F. Menge. "Enterprise Service Bus." In: (2007), pp. 1–6.
- [37] M. Meyer, R. Zarnekow, and L. M. Kolbe. "IT-Governance - Begriff, Status quo und Bedeutung." In: *Wirtschaftsinformatik* (2003). issn: 09376429.
- [38] A. Nicolaescu, D.-I. Andreas Steffens Andrej Dyck, K. Fögen, M. Firdaus Harun, and S. Hacks. *FsSE / AoST 2017*. Tech. rep. 2017.
- [39] W. W. Osterhage. "Notfallmanagement in Kommunikationsnetzen." In: (2016), pp. 5–15. doi: 10.1007/978-3-662-45660-6.
- [40] P. v. d. M. Paul Homan Matthew Rouse Tom van Sante Mike Turner. *ArchiMate® 3.0.1 Specification*. 2013. isbn: 9789087530945. doi: 10.1111/jmg.12104.

- [41] T. Rabl, M. Sadoghi, H.-A. Jacobsen, S. Gómez-Villamor, V. Muntés-Mulero, and S. Mankowskii. "Solving Big Data Challenges for Enterprise Application Performance Management." In: 5.12 (2012), pp. 1724–1735. ISSN: 21508097. DOI: 10.14778/2367502.2367512. arXiv: 1208.4167.
- [42] S. Roth, M. Hauder, M. Farwick, R. Breu, and F. Matthes. "Enterprise architecture documentation: Current practices and future directions." In: *Wi 2013* (2013), pp. 912–925. ISSN: 00219673. DOI: 10.1148/rg.327125019.
- [43] P. Runeson and M. Höst. "Guidelines for conducting and reporting case study research in software engineering." In: *Empirical Software Engineering* (2009). ISSN: 13823256. DOI: 10.1007/s10664-008-9102-8. arXiv: 9809069v1 [gr-qc].
- [44] J. Schekkerman. *How to survive in the jungle of Enterprise Architecture Frameworks*. 2004. ISBN: 141201607X. DOI: 10.1073/pnas.1205276109.
- [45] M. Shahin, M. Ali Babar, and L. Zhu. "Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices." In: *IEEE Access* 5.Ci (2017), pp. 3909–3943. ISSN: 21693536. DOI: 10.1109/ACCESS.2017.2685629. arXiv: 1703.07019.
- [46] H. Ter Doest and M. Lankhorst. *Tool Support for Enterprise Architecture-A Vision*. Tech. rep. 2004.
- [47] The Open Group. *TOGAF Version 9.1*. 2011. DOI: 10.1111/j.1365-2702.2009.02827.x. arXiv: arXiv:1211.6894v1.
- [48] M. Välja, R. Lagerström, M. Ekstedt, and M. Korman. "A requirements based approach for automating enterprise IT architecture modeling using multiple data sources." In: *Proceedings of the 2015 IEEE 19th International Enterprise Distributed Object Computing Conference Workshops and Demonstrations, EDOCW 2015* (2015), pp. 79–87. ISSN: 2325-6583. DOI: 10.1109/EDOCW.2015.33.
- [49] J. E. Van Aken. "Management research as a design science: Articulating the research products of mode 2 knowledge production in management." In: *British Journal of Management* (2005). ISSN: 10453172. DOI: 10.1111/j.1467-8551.2005.00437.x.
- [50] B. Van Der Raadt and H. Van Vliet. "Designing the enterprise architecture function." In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 2008. ISBN: 3540878785. DOI: 10.1007/978-3-540-87879-7-7.

- [51] J. Wang, J. Li, Q. Wang, D. Yang, H. Zhang, and M. Li. "Can requirements dependency network be used as early indicator of software integration bugs?" In: *2013 21st IEEE International Requirements Engineering Conference, RE 2013 - Proceedings* (2013), pp. 185–194. issn: 1090-705X. doi: 10.1109/RE.2013.6636718.
- [52] J. Webster. "&R. Watson (2002).“Analyzing the Past to Prepare for the Future: Writing a Literature Review.”” In: *MIS Quarterly* (2002). issn: 0960-0035. doi: 10.1007/s11576-006-0057-3. arXiv: ECIS2009-0566.R1.