TECHNISCHE HOCHSCHULE MITTELHESSEN | THM | CAMPUS FRIEDBERG | MND Mathematik, Naturwissenschaften und Datenverarbeitung

Seminararbeit
im Studiengang Wirtschaftsinformatik
am Fachbereich Mathematik, Naturwissenschaften und Datenverarbeitung
der Technischen Hochschule Mittelhessen

Migration von On-Premise auf Hybrid- und Multi-Cloud Umgebung

vorgelegt von:
Cong Chanh Vinzenz Nguyen
Matrikelnummer: 5291018
cong.chanh.vinzenz.nguyen@mnd.thm.de

Referent: Dipl.-Inform. Adrian Lidwin

Wintersemester 2023/2024

Abstract

This seminar paper, submitted for the Business Informatics program at the Technische Hochschule Mittelhessen, addresses the migration from on-premise systems to hybrid and multi-cloud environments. In light of the increasing reliance of companies on cloud-based systems, this work highlights the challenges and strategies associated with migrating custom software to the cloud. The focus is on both the technical and strategic aspects of this transformation.

The paper begins with an introduction to the fundamental concepts of cloud computing, followed by a detailed examination of market dynamics and leading cloud providers. It then discusses the technical challenges of cloud migration, such as system compatibility, hardware integration, data protection, and regulatory compliance. Strategic considerations for selecting and managing cloud services, including the exploration of serverless computing and multi-cloud strategies, are also discussed.

A significant part of the paper is dedicated to custom development in the cloud, examining various approaches such as rehosting, replatforming, refactoring, and rearchitecting. The paper also sheds light on the importance of Service-Oriented Architecture (SOA) and its transition to cloud-native architectures, including the role of microservices and serverless architecture.

In conclusion, the paper provides a comprehensive overview of the challenges and opportunities associated with migrating to cloud-based systems and offers insights into the future of cloud development. The aim is to provide organizations with a deeper understanding of the complexity and strategic decisions required for a successful cloud migration.

Table of Contents

Table of Figures

1 Introduction

1.1 Problem statement

The rise of computers led to a big demand for software, which caused the software industry to grow a lot. This growth got even faster because of new technologies and the internet, which made people rely more on cloud-based systems instead of software on their own computers. But moving software to the cloud, especially custom-made software, is not easy. It's important to make sure the software can work well in the cloud.

Changing from software on your own computers to using cloud systems is a big issue in the IT world today, affecting all kinds of companies. These companies need to update their IT systems to get benefits like being able to grow easily, be more flexible, and possibly save money. But moving to the cloud is complicated. There are technical challenges like making old systems work with new ones, making sure data is safe, and following rules about data. There are also business challenges like managing costs, helping employees adjust to the changes, and making sure everything fits with the company's goals. Cloud technology is always changing, which makes things even more complicated.

1.2 Objectives and research focus

The main goal of this paper is to describe a basic plan for moving from on-site computer systems to cloud systems that use both multi-cloud and hybrid approaches. We will give special attention to understanding how to change existing programs so they can work in the cloud. This involves both the technical and strategic aspects of making this change.

1.3 Structure of the paper

This paper, which is set up like a review of existing research, is part of a seminar in the fifth semester of the Business Informatics program. We expect readers to have a basic understanding of topics like software development, information systems, and managing IT, as it's intended for students at this point in their academic studies. The paper starts by talking about important ideas and explaining key words. Then, it goes into detail about why moving to the cloud is a big deal and how it works in general. We focus especially on moving custom-made software to the cloud and the unique problems that come with it. Finally, we'll summarize what we've learned and talk about how these changes are affecting software development today.

1.4 Motivation

This research originates from the intricate landscape of organizational IT (Information Technology) systems, which include a variety of software and hardware components crucial for daily operations. Understanding this complex area is essential for comprehending why an increasing number of organizations are adopting cloud-based systems. Historically, organizations have developed a diverse mix of IT systems, essential for their routine functions and achieving long-term goals. The decision to transition to cloud services is closely connected to the functionalities and historical development of these existing systems.

In many organizational settings, one commonly finds standard software suites, such as office productivity tools, alongside more complex systems like ERP (Enterprise Resource Planning), SCM (Supply Chain Management), and CRM (Customer Relationship Management) software. These systems are often central to an organization's operational framework. Therefore, migrating them to cloud-based environments can significantly impact their effectiveness and operational efficiency. [1, p. 20]

Additionally, numerous organizations utilize bespoke software solutions, tailored to their specific needs. These include both commercially acquired applications and in-house developed software. The migration of these customized software applications to the cloud presents unique challenges, primarily due to their deep integration within the organization's operational processes and potential compatibility issues arising from their non-cloud-centric original design.

The IT ecosystem also comprises various hardware components, such as personal computers, mobile devices, and possibly IoT (Internet of Things) devices or elements of a smart factory. [2, p. 8]

When considering the migration to cloud-based systems, it is crucial to ensure the seamless interaction of these hardware components with cloud infrastructures. This consideration should include aspects like ease of access, security enhancements, and performance optimization.

Another critical aspect involves the role of traditional on-premises server infrastructure. Transitioning from localized server infrastructure to cloud-based solutions represents a significant shift in data storage, accessibility, and management approaches. This shift requires careful planning to maintain data integrity, security measures, and uninterrupted service continuity.

The motivation behind this research is to navigate the complex details of IT systems in the context of cloud migration. Organizations are faced with the task of balancing the potential benefits of cloud computing against the multifaceted challenges and inherent risks involved in migrating their existing IT systems. This paper aims to provide a comprehensive guide, aiding organizations in understanding these challenges and enabling them to make informed decisions during their transition to cloud-based environments.

2 Fundamental Concepts

2.1 Definition of Cloud Computing

In this chapter, we delve into the key terms associated with cloud computing, drawing upon definitions provided by the National Institute of Standards and Technology (NIST), supplemented with explanations of on-premise systems and an overview of the current cloud market. [3]

2.1.1 On-Demand Self-Service

On-Demand Self-Service refers to the ability of a cloud user to automatically provision and manage computing resources like server time and network storage, without interacting with the service provider. This allows users to acquire and configure resources anytime they need, offering high flexibility and rapid scalability in using cloud services.

2.1.2 Broad Network Access

Broad Network Access means that services are widely accessible over the network, typically the Internet, and can be used across various platforms such as laptops, tablets, smartphones, and workstations. This ensures that cloud services are accessible from any location at any time, provided there is an Internet connection, maximizing the accessibility and connectivity of cloud services.

2.1.3 Resource Pooling

Resource Pooling indicates that the cloud provider pools its resources (like memory, processors, storage, and network components) for multiple customers in a multi-tenant model. Resources are dynamically assigned and reassigned based on customer demand, without the customer knowing the exact location. This leads to efficient use of resources and enables scalability and cost-efficiency.

2.1.4 Rapid Elasticity

Rapid Elasticity is about the quick and flexible adjustment of resources to meet changing demand. It means that resources can be elastically provisioned and released, appearing to the user as unlimited and available at any time in any quantity. This characteristic is crucial for handling peak loads and ensuring the scalability of services.

2.1.5 Measured Service

Measured Service means that the use of cloud services is monitored, controlled, and reported, enabling transparent billing and utilization of services. Resource usage is measured and can be billed according to various metrics such as storage space, processing time, or the number of active users. This ensures that customers only pay for what they actually use, promoting efficient resource usage.

2.2 Service Models

Cloud computing includes various service models, each with different levels of scope and control. Each of these service models offers different levels of flexibility, control, and management, catering to the diverse needs of users in the cloud computing ecosystem.

2.2.1 Infrastructure as a Service (IaaS)

Infrastructure as a Service offers access to basic computing resources like physical or virtual servers and storage. This model provides the fundamental building blocks for cloud services, where users can rent virtual machines, storage, or networks. For example, a company might use IaaS to host their website's data on a virtual server rather than buying and maintaining their own server hardware.

2.2.2 Platform as a Service (PaaS)

Platform as a Service (PaaS) in addition to providing the infrastructure, also offers platforms for the development, testing, and deployment of applications. This service is particularly useful for developers as it provides a framework they can use to build upon. An example of PaaS is a cloud-based platform that allows developers to create and deploy web applications without worrying about underlying software or hardware maintenance.

2.2.3 Software as a Service (SaaS)

Software as a Service (SaaS) allows users to access application software hosted on the cloud platform by the provider. These applications are accessible over the internet, often through a web browser. SaaS eliminates the need for installing and running applications on individual computers, simplifying maintenance and support. A common example is an email service accessed through a web browser, where the email software and user data are stored on the cloud, not on the user's personal computer.

2.3 Deployment Models

There are various deployment models for cloud services, each suited to different organizational needs. Each deployment model offers unique advantages and can be tailored to meet specific business requirements, with considerations for factors like security, scalability, and cost.

2.3.1 Private Cloud

Private Cloud is designed exclusively for a single organization. It can either be managed internally or by a third party. A private cloud offers enhanced security and control, making it ideal for businesses with sensitive data or specific regulatory requirements. In practice, a private cloud can be realized through on-premises data centers or by using dedicated infrastructure provided by cloud service providers. For example, a financial institution might use a private cloud to manage customer data, employing strict access management controls to ensure data security.

2.3.2 Community Cloud

Community Cloud is a model used by multiple organizations with shared concerns, such as security, compliance, and performance requirements. This type of cloud can be hosted either by one of the organizations in the community, a third-party provider, or a combination of both. In practice, community clouds offer a collaborative environment where resources are shared, but with more control and security than public clouds. For instance, several government agencies might use a community cloud to share infrastructure and applications while meeting specific security standards.

2.3.3 Public Cloud

Public Cloud is provided by a cloud service provider for the general public or a large industry group. It's based on the standard cloud computing model, where service providers offer resources like applications and storage on the Internet. Public clouds are easy to access and flexible, making them suitable for a wide range of applications. An example is a business using public cloud services for non-sensitive operations like email hosting or customer relationship management (CRM) systems. These services often include robust access management features to control user permissions and data access.

2.3.4 Hybrid Cloud

Hybrid Cloud combines two or more of the above-mentioned cloud models, typically private and public clouds, creating a flexible and scalable environment. In practice, a hybrid cloud allows businesses to store protected or privileged data on a private cloud, while taking advantage of the computational power of a public cloud for less sensitive tasks. For example, a retail company might use a private cloud to store customer data securely, while utilizing a public cloud for its e-commerce website, allowing it to scale resources during high traffic periods.

2.4 On-Premise

On-premise refers to IT systems and applications that are physically located within a company's premises. This arrangement typically involves the company owning and maintaining its own server cabinets or rooms. Unlike cloud solutions, these systems are operated and managed locally on the company's own servers. The maintenance and administration of these systems are usually handled by an internal IT department or a designated IT professional.

One of the key characteristics of on-premise infrastructure is its limited scalability. Due to the physical nature of the hardware, there is a ceiling on how much the system can be expanded in terms of storage and computing power. This limitation often means that companies with on-premise solutions may face challenges in rapidly scaling their IT capabilities in response to business growth or fluctuating demands.

Furthermore, running applications on in-house hardware allows organizations to have full control over their data and network. This level of control can be crucial for companies with stringent data security requirements or those that operate in industries with strict regulatory compliance needs. However, this complete control also comes with the responsibility of ensuring the security and efficiency of the systems, which can be both resource-intensive and costly.

In summary, while on-premise systems offer the benefit of complete control over IT infrastructure, they also come with challenges such as limited scalability, significant maintenance requirements, and the need for dedicated administrative support.

2.5 Market Overview

The cloud computing market is characterized by its diversity and dynamism, encompassing a wide array of providers who offer varied services and models. The complexity of this market extends beyond the mere size and scope of services; it also includes geographical presence and specialization in specific domains. A detailed examination of the current market landscape reveals the following insights:

2.5.1 Amazon Web Services

Amazon Web Services (AWS) stands as a prominent leader in the cloud computing sector, providing a comprehensive array of services including computing power, storage options, and networking capabilities. Renowned for its robust and scalable infrastructure, AWS caters to a broad spectrum of business requirements, making it a favored choice for entities ranging from small startups to large multinational corporations. [4]

2.5.2 Microsoft Azure

Microsoft Azure represents another key player in this field, offering an extensive suite of cloud services. It is seamlessly integrated with Microsoft's suite of software products, making it an optimal solution for businesses heavily reliant on Microsoft's ecosystem. Azure's services are diverse, covering areas such as artificial intelligence (AI), machine learning, and the Internet of Things (IoT), in addition to its foundational cloud services. [5]

2.5.3 Google Cloud Platform

Google Cloud Platform distinguishes itself with its capabilities in high-performance computing and advanced data analytics. It emphasizes open-source technologies and specializes in data and analytics solutions, catering to businesses that prioritize data-driven strategies and decision-making processes. [6]

2.5.4 Alibaba Cloud

Dominating the Asian market, Alibaba Cloud offers a robust portfolio of tools and services in e-commerce, AI, and cloud computing. While it has a strong foothold in Asia, Alibaba Cloud is also progressively expanding its global presence. [7]

2.5.5 OVH Cloud

Focused primarily on the European market, OVH Cloud provides public cloud services that emphasize data security and privacy, aligning with Europe's stringent data protection regulations. This makes it a suitable option for businesses seeking a provider that adheres to European compliance standards. [8]

2.5.6 Emerging and Specialized Cloud Providers

Beyond these market leaders, the cloud computing landscape is enriched by a growing number of smaller, specialized providers. These entities often concentrate on niche areas, offering unique services that cater to specific market segments. Examples include:

DigitalOcean: Known for its simplicity and ease of use, DigitalOcean offers cloud services tailored for developers, startups, and small to medium-sized businesses. They provide straightforward solutions for hosting web applications and managing infrastructure.

Rackspace: Specializes in hybrid cloud solutions, offering both public and private cloud services with a strong emphasis on customer support and managed services. Rackspace is ideal for businesses looking for tailored cloud solutions with comprehensive support.

Heroku: A cloud platform focusing on supporting app development, Heroku simplifies the deployment and scaling of applications. It's particularly popular among software developers for its ease of use and integration with various development tools.

Cloudflare: While primarily known for its content delivery network (CDN) and internet security services, Cloudflare also provides cloud services that emphasize web performance and security.

Known for delivering more personalized and tailored services, these smaller providers offer a contrast to the broader service range provided by larger corporations.

This multifaceted ecosystem of cloud providers mirrors the evolving requirements of businesses and organizations. Large providers deliver a vast array of services on a global scale, while smaller, specialized firms like DigitalOcean, Rackspace, Heroku, and Cloudflare address specific needs or compliance demands. This diversity ensures that businesses of varying types and sizes can find and adopt cloud solutions that align with their unique operational requirements.

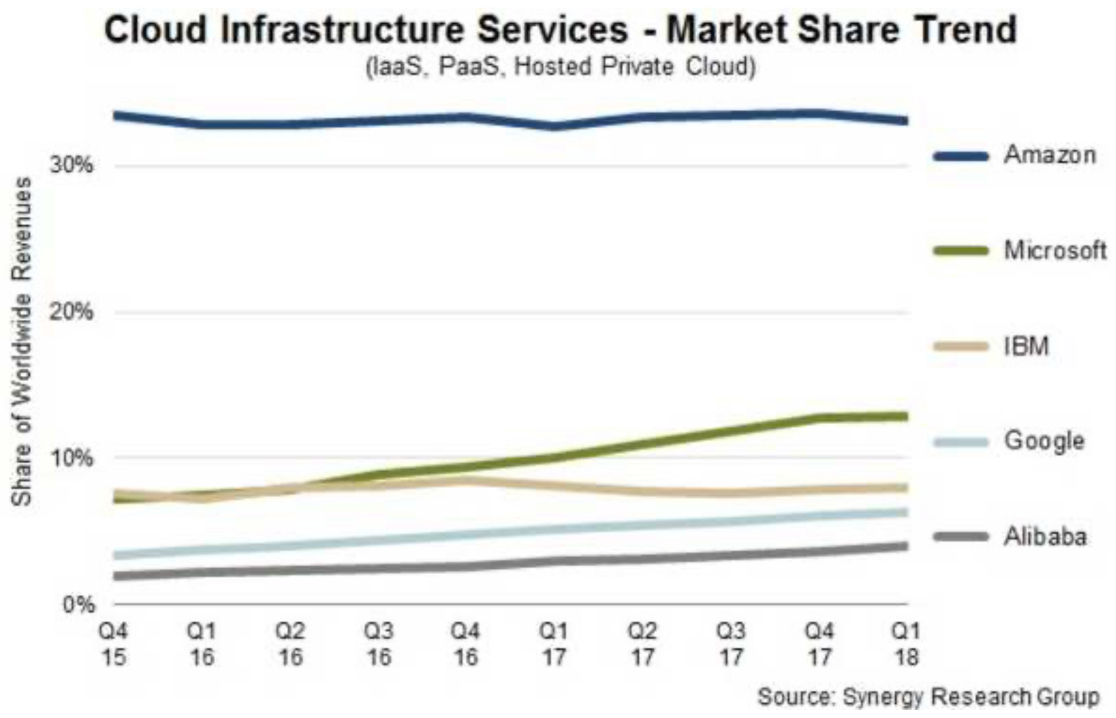

Figure 1: Market share cloud provider [9]

3 Technical Challenges in Cloud Migration

The migration of IT systems to cloud environments involves a range of technical challenges that are crucial to address for a successful transition. These challenges vary from compatibility issues of current systems to concerns about data protection, adherence to regulations, and maintaining or improving efficiency in complex, established structures and software architectures.

3.1 Ensuring System Compatibility

A key technical challenge in cloud migration is making sure that existing IT infrastructures are compatible with the cloud environment. This requires a thorough evaluation of current systems, including both software and hardware, to check if they can be adapted for cloud integration. For example, standard software suites like Google Workspace or Microsoft Office 365 often need to be switched to their cloud-based versions, which can change how data is stored and accessed. Similarly, complex systems such as ERP (Enterprise Resource Planning), SCM (Supply Chain Management), and CRM (Customer Relationship Management), like those on platforms such as SAP4HANA, need careful planning. The migration process for these systems has to ensure that all functions are kept and data integrity is maintained during the move.

3.2 Hardware Integration

Hardware components are also vital in the migration strategy. Transitioning desktops and smartphones to use Progressive Web Apps (PWAs) and web interfaces can help integrate them with cloud systems. In the Internet of Things (IoT) domain, devices in smart factories or other smart environments might use edge computing. This approach processes data closer to its source, reducing latency and bandwidth needs. For server infrastructure, organizations might consider hybrid solutions like Azure Stack, which combine cloud and on-premise resources, offering a balanced approach to data management and processing.

3.3 Data Protection and Regulatory Compliance

A crucial part of cloud migration is ensuring data protection and adhering to regulatory requirements, such as the General Data Protection Regulation (GDPR) in Europe. This challenge isn't just about securing data during migration; it also involves ensuring that cloud service providers maintain high data security standards and comply with relevant laws.

3.4 Addressing Efficiency in Grown Structures

Often, the aim of cloud migration is to improve efficiency. However, achieving this in organizations with established structures and intricate software architectures can be challenging. The migration process must be strategically planned and executed to streamline operations and optimize resource use, all while minimizing disruption to existing workflows.

In summary, the technical challenges of cloud migration are diverse and complex. They require a strategic approach that includes thorough planning and careful execution. Addressing each aspect, from software and hardware compatibility to maintaining data security and operational efficiency, is crucial for a successful shift to cloud computing. While challenging, this process offers significant potential benefits in terms of scalability, flexibility, and overall organizational agility.

4 Strategic Considerations in Cloud Service Selection and Management

In the realm of cloud migration and management, making strategic choices about service providers and their specific offerings, such as serverless architectures and multi-cloud setups, is crucial. This decision-making process is influenced by various factors, including data protection policies, geopolitical considerations, and dependencies on specific cloud service providers. This discussion aims to provide a deeper exploration of these elements, covering aspects like user and access management and the issue of vendor lock-in.

4.1 Provider Specifics in Serverless Computing

Serverless computing, represented by services like AWS Lambda and Azure Functions, represents a significant shift in application development and management. This model allows organizations to run code without managing servers, improving scalability and cost-effectiveness. However, each cloud provider offers different features in terms of execution environments, pricing structures, and integration capabilities. Understanding these differences is key to optimizing performance and cost in serverless computing environments.

4.2 Strategies for Multi-Cloud Deployment

Adopting multi-cloud environments, which involve using resources from multiple cloud providers, has several benefits. These include reduced risk of vendor lock-in, better disaster recovery capabilities, and the chance to use the unique strengths of each provider. However, this approach can complicate managing different platforms, especially in maintaining consistent user and access management across these varied environments.

4.3 Public and Private Cloud Models in the Context of Data Protection

Choosing between public and private cloud models often depends on data protection policies. For sensitive data, organizations might prefer a private cloud or a hybrid model, combining features of both private and public clouds. This choice is usually driven by the need to comply with strict data protection regulations like GDPR in Europe. [10] Hybrid clouds offer the flexibility of public clouds while keeping the security and control of private cloud infrastructures.

4.4 Geopolitical Influences and Provider Dependencies

Geopolitical dynamics significantly affect cloud computing strategies. For example, using Alibaba Cloud might be popular in Asian markets but viewed cautiously in Western regions due to data sovereignty and privacy concerns. Similarly, the dominance of American cloud providers like AWS and Azure has led to discussions about European data sovereignty and the need for a Europe-focused cloud infrastructure to comply with local data protection laws. [11]

4.5 The Challenge of Vendor Lock-In and Deployment Decisions

Vendor lock-in is a major issue in cloud computing. It occurs when an organization becomes too dependent on a single provider, making it difficult and costly to switch to other providers. When making deployment decisions, organizations need to weigh the convenience and advanced features of proprietary services against the risks of vendor lock-in. Strategies that promote some independence from providers, such as using containerization and open standards, are crucial to reduce these risks.

4.6 User and Access Management in Cloud Environments

Managing user identities and access rights is essential in cloud environments, especially in multi-cloud setups. Organizations must establish strong policies and use advanced tools to efficiently manage user access across different cloud platforms. This is not only important for operational efficiency but also for maintaining high security standards and complying with data protection regulations.

5 Custom Development in Cloud Computing

In this chapter, we embark on an in-depth exploration of diverse methodologies employed in the development of bespoke software solutions tailored for cloud environments. Our discussion encompasses a range of strategic approaches, each meticulously examined to provide clarity on their application in various scenarios:

5.1 Rehosting (Lift and Shift)

The concept of Rehosting, commonly referred to as 'Lift and Shift', entails the direct migration of existing applications or systems to a cloud setting without substantial alterations to their underlying architecture. This chapter will critically analyze the merits and limitations of the Rehosting strategy, delineating scenarios where it proves most effective. Further elaboration on this approach will be presented in the subsequent chapter, offering a granular perspective on its implementation. [12]

5.2 Replatforming

Replatforming is defined as the transition of an application to a cloud environment, leveraging cloud-native services and resources. This strategy aims to harness the potential of the cloud platform for optimizing application functionality. The ensuing sections will detail various Replatforming strategies, shedding light on their respective advantages. A comprehensive treatment of this topic is reserved for the next chapter, where specific instances of Replatforming will be dissected.

5.3 Refactoring

Refactoring involves the systematic restructuring or modification of existing applications to enhance their suitability for cloud deployment. This chapter will explore a spectrum of refactoring techniques, examining how they contribute to augmenting application performance and scalability in cloud contexts. An in-depth exploration of these methodologies will be continued in the following chapter, providing a nuanced understanding of practical Refactoring scenarios.

5.4 Rearchitecture

Rearchitecture represents a more profound evolution in cloud application development, entailing a fundamental redesign of applications to fully leverage cloud capabilities. This may involve transitioning to entirely new architectural frameworks, such as Microservices or Serverless architectures. The subsequent discussion will delve into the applicability and implications of Rearchitecture, setting the stage for an extensive analysis in the next chapter, which will expound on specific use cases and strategic considerations.

5.5 Monolith Architecture

This section presents a detailed analysis of monolithic software applications, focusing on their defining characteristics, inherent challenges, and potential migration strategies for cloud integration.

Monolithic applications are often older software systems developed as a single, indivisible unit. An example of this architecture is Windows Forms. These applications are typically characterized by a lack of or inadequate separation into distinct software layers, which poses significant challenges in maintenance and scalability. The architecture does not readily support parallelization or distribution across different systems, thereby limiting compatibility with modern, distributed system environments. Although these systems, due to their low level of abstraction, are easier to develop and debug, they may exhibit limitations in performance and scalability. However, they do offer certain advantages in specific contexts, such as performance optimization due to close hardware proximity or improved data privacy and network security, especially since they often operate in isolated environments on local computers.

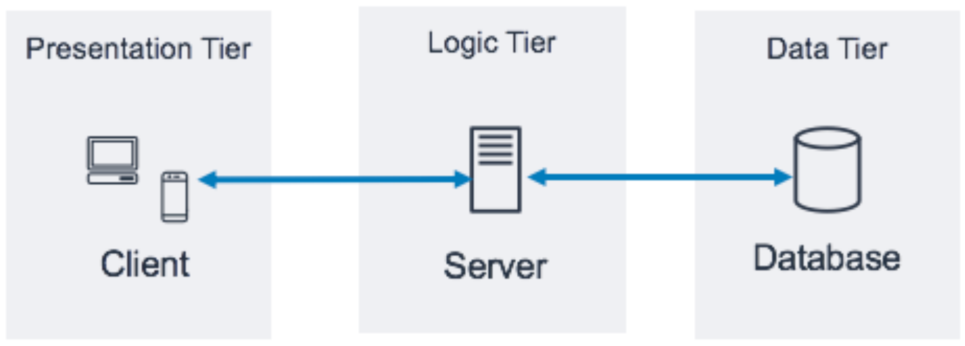

Figure 2: 3 layer architecture [13]

The graphical representation breaks down the architecture of a monolithic application into three primary layers: the Presentation Layer (GUI), the Business Logic Layer (Core), and the Persistence Layer (Database). In practice, these layers are often tightly interwoven, complicating efforts towards code modularity and maintainability.

For a future-proof migration strategy, an immediate "Rehosting" for quick cloud compatibility (Infrastructure as a Service, IaaS) is recommended. This pragmatic approach involves transferring the existing application to a cloud infrastructure without fundamentally altering its architecture. In the medium term, "Replatforming" is advised, such as moving to a cloud database platform (Platform as a Service, PaaS). This step involves adapting the application to the cloud environment without a complete overhaul of the code.

These considerations hold particular significance for organizations aiming to modernize their legacy software systems and harness the advantages offered by cloud computing, such as scalability, flexibility, and cost efficiency. The illustration and ensuing discussion in this chapter serve as a foundation for exploring the most effective strategies to modernize monolithic systems in the digital transformation landscape.

5.6 Server-Client Architecture

This section outlines a comprehensive, step-by-step approach to transitioning traditional server-client architectures to cloud-based solutions. It highlights three critical strategies for this transformation: Rehosting, Replatforming, and Refactoring.

The server-client architecture is a conventional model for delivering IT services and applications. In this model, clients (such as desktops, Android, iOS devices) communicate with a central server via an API. This architecture, known for its robustness and reliability, is often utilized in tandem with frameworks like ASP.Net. [14]

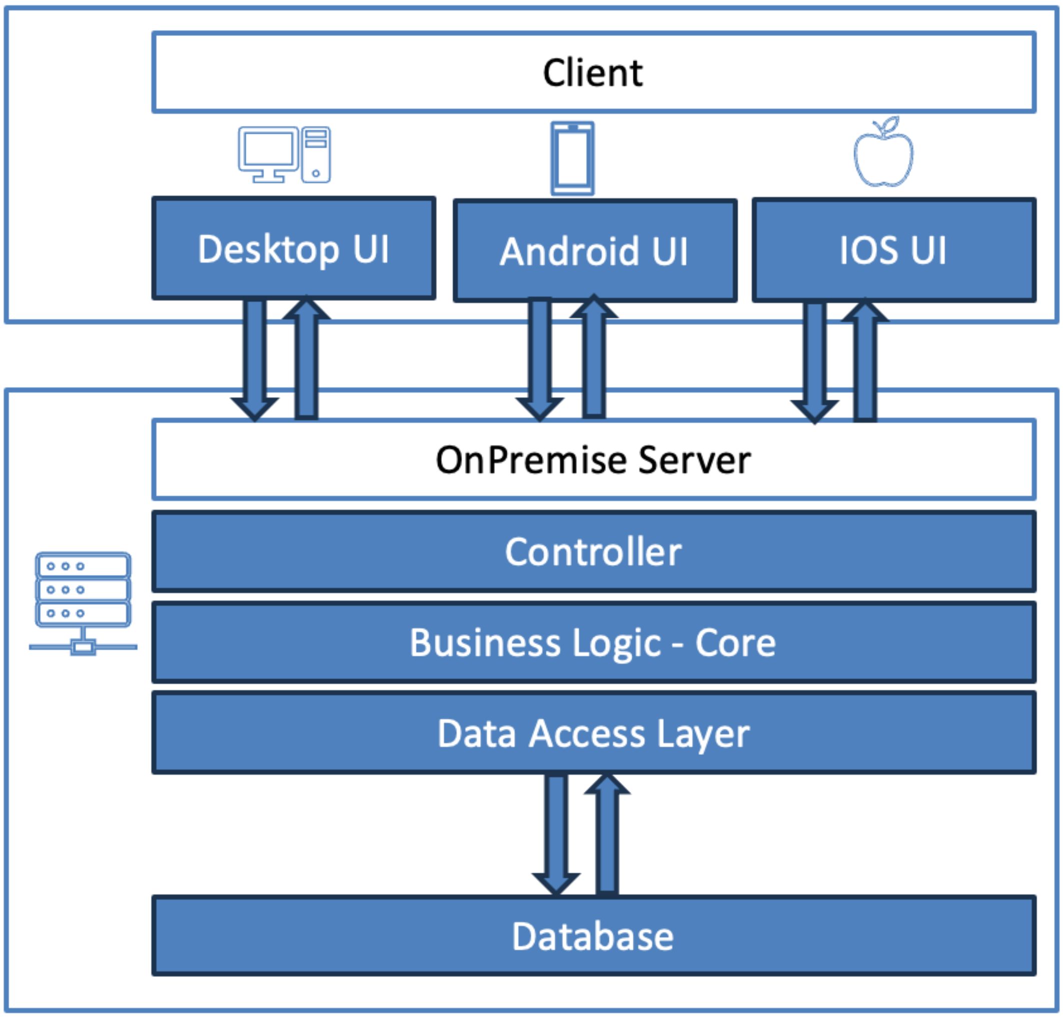

Figure 3: Server-Client architecture

5.6.1 Replatforming Database

Replatforming the database involves migrating it from an on-premises system to a cloud-based environment. Essential prerequisites for this process include a well-defined layer separation, a clear architecture and codebase, and the effective implementation of Object-Relational Mapping (ORM) and migration scripts. This transition reduces direct database access and shifts data management to the cloud, resulting in enhanced scalability and simplified monitoring.

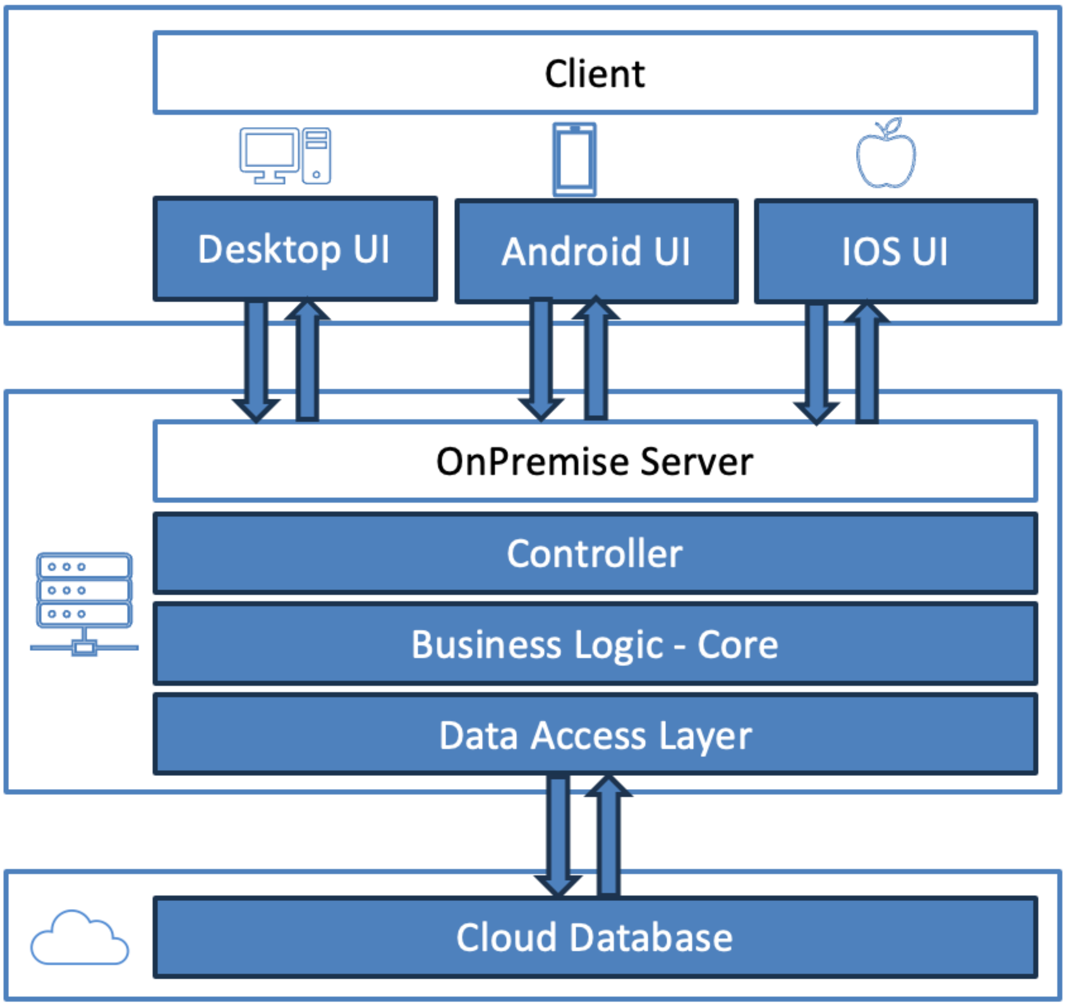

Figure 4: Server-Client architecture using a cloud database

5.6.2 Rehosting Server

Rehosting entails relocating the entire server infrastructure to the cloud. This shift eliminates the need for physical servers and the associated administrative and network maintenance tasks. Server configuration and monitoring are then managed entirely in the cloud, leading to a decrease in local infrastructure requirements and potential cost savings.

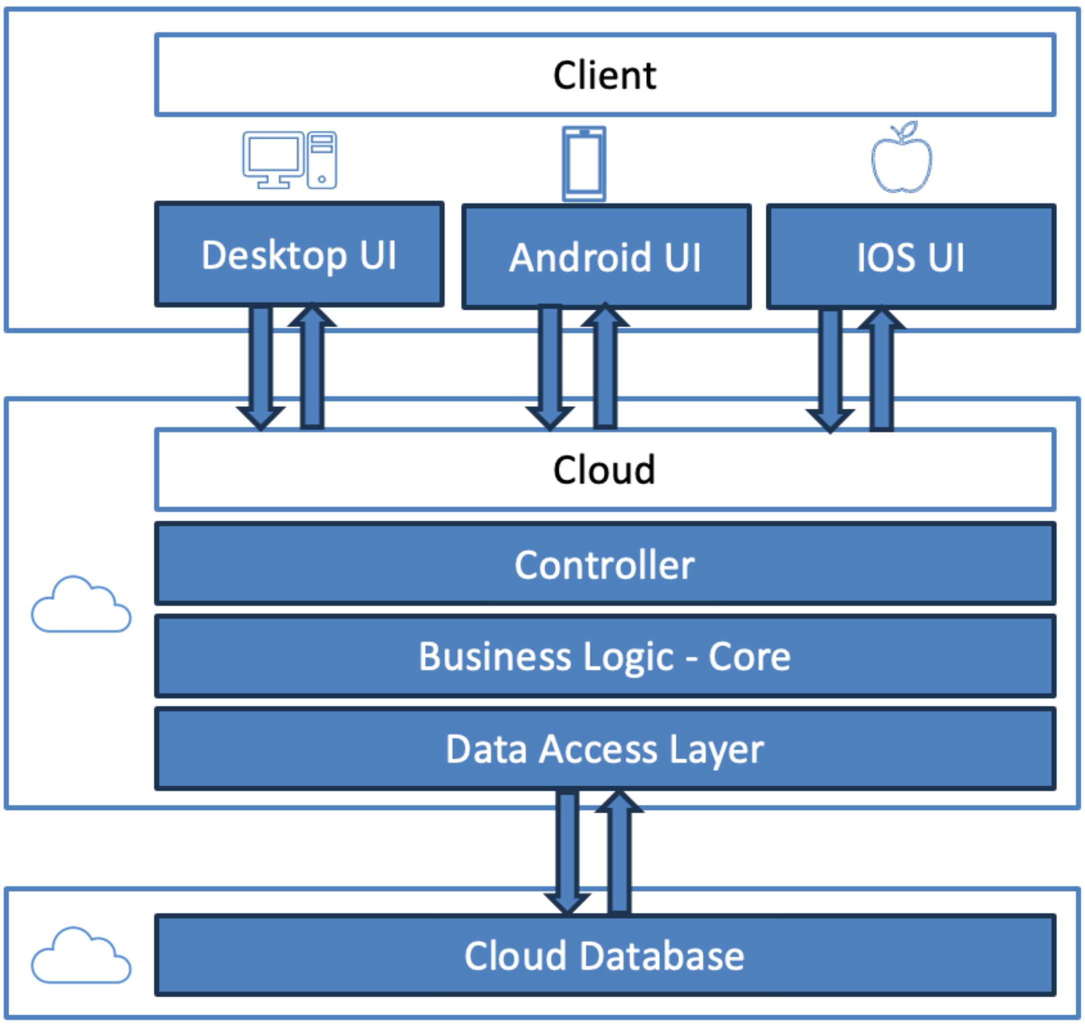

Figure 5: Server-Client architecture using a cloud database and running on a cloud plattform

5.6.3 Refactoring Individual Functions

Refactoring individual functions involves a more detailed migration process, where specific functions are outsourced to the cloud and re-implemented as cloud functions. This approach allows for the individual scaling of these functions and the utilization of advanced cloud services.

In summary, these steps portray an evolutionary approach to cloud migration. They range from the straightforward transfer of existing infrastructures to a more fundamental redesign of individual components to maximize the benefits of cloud computing, such as scalability, availability, and cost efficiency. Each step in this process must be meticulously planned and executed, considering the unique requirements and constraints of the organization. [15]

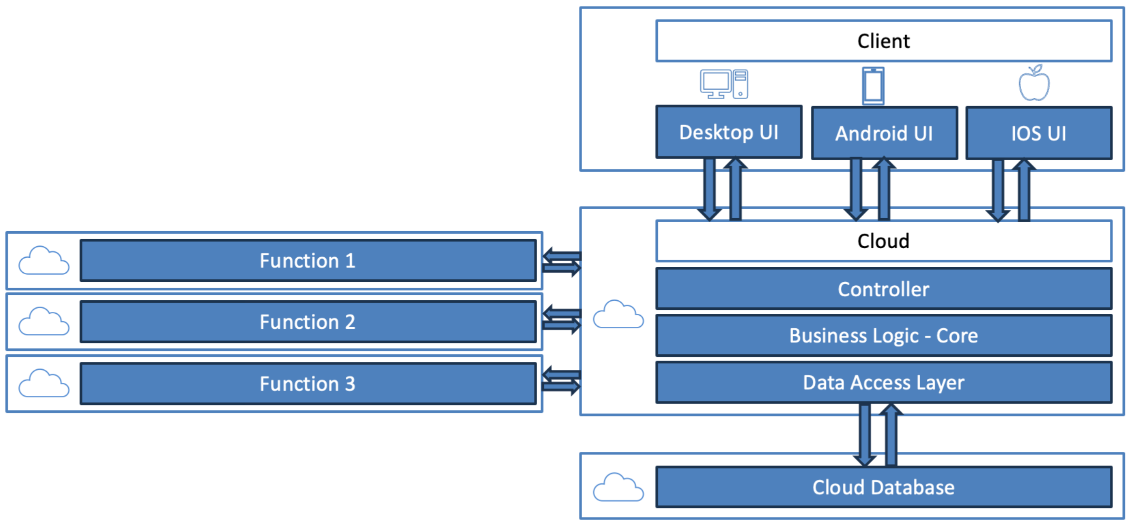

Figure 6: Server-Client architecture using external cloud functions

This method represents a systematic and strategic approach to transition from legacy server-client architectures to contemporary, cloud-based solutions. It enables organizations to capitalize on the advantages of cloud computing while maintaining the reliability and functionality of their existing systems.

5.7 Understanding Service-Oriented Architecture (SOA)

Service-Oriented Architecture (SOA) is a software design pattern where services are offered to other components by application components, through a communication protocol over a network. Key principles of SOA include the abstraction of services into distinct units, reusability, interoperability, and composability. [16]

5.7.1 SOA Characteristics and Design Principles

SOA encourages reusing established services across various platforms, promoting a service-centric design approach. This design paradigm enables the use of shared resources, such as databases or common code functionalities, to ensure consistent service delivery across different application domains. The architecture supports a modular development framework where services are loosely coupled and can be orchestrated to create comprehensive business processes.

5.7.2 Challenges in SOA and the Role of Middleware

Communication is crucial in SOA, requiring middleware technologies like Enterprise Service Buses (ESB) and Message-Oriented Middleware (MOM). These technologies, such as BizTalk [17] and RabbitMQ [18], serve as the foundation of SOA, allowing different services to communicate seamlessly. However, the increasing complexity in communication and shared resources may lead to challenges in maintainability and scalability.

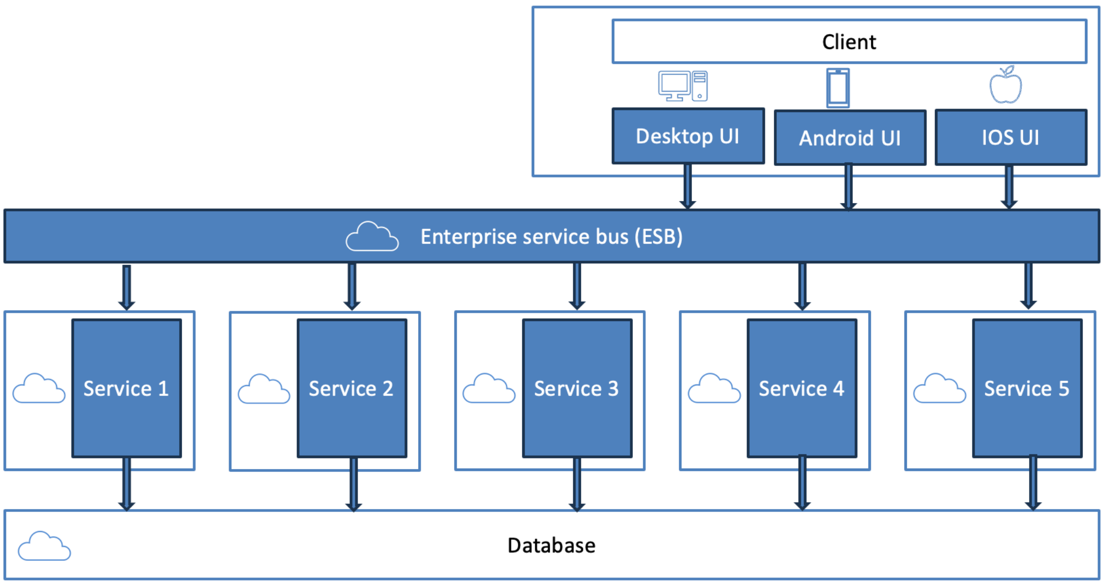

Figure 7: Service oriented architecture

5.7.3 SOA as a Precursor to Cloud Computing

SOA can be viewed as an early form of cloud computing, where services are provided on-demand to clients. Developments such as Microsoft's Cloud optimized services illustrate this transition, indicating a shift from on-premises architecture to services orchestrated in a cloud environment.

5.8 Transitioning from SOA to Cloud-Native Architectures

The shift from Service-Oriented Architecture (SOA) to cloud-native architectures like microservices has been driven by the need for more detailed control, better scalability, and increased resilience. Cloud-native architectures make greater use of containerization and orchestration, highlighted by technologies such as Docker and Kubernetes. [19]

5.8.1 Advantages of Cloud-Native Solutions

Cloud-native solutions have several benefits compared to traditional SOA. They provide improved scalability, as they can be spread out across multiple servers or cloud regions. Also, cloud-native architectures make full use of the cloud's capabilities, offering resources on demand and better solutions for disaster recovery.

5.8.2 Containerization and Orchestration

Containerization packs an application's code, configurations, and dependencies into a single package, while orchestration automates the deployment, management, scaling, and networking of these containers. Kubernetes is a leading system for orchestration, simplifying the deployment and scaling of applications and supporting a microservices architecture. [20]

5.8.3 Focusing on Kubernetes

Kubernetes plays a crucial role in cloud-native architectures. It is an open-source platform that manages containerized services and workloads. This system makes it easier for organizations to deploy, scale, and operate application services in cloud environments. It is designed to be extensible and fault-tolerant, scaling with no downtime and providing developers with an easy way to deploy their applications.

5.8.4 Mitigating Vendor Lock-In with Cloud-Native Technologies

An important aspect of cloud-native architectures is their support for open standards and the ability to avoid being tied to a single vendor. By using containerization and Kubernetes, organizations can make sure their applications are portable and can be moved across different cloud providers, avoiding dependency on any single vendor. [21]

5.8.5 Implications for IT Strategy and Development

The move towards cloud-native solutions brings significant changes to an organization's IT strategy. This transition means a drastic shift in IT infrastructure requirements, with a strong focus on embracing cloud-centric technologies and practices.

5.8.6 Enhanced Flexibility and Deployment Speed

Adopting SOA and evolving to cloud-native solutions allows organizations to be more agile in their development and deployment processes. This increased flexibility lets them respond more quickly to market changes and customer demands.

The journey from SOA to cloud-native solutions shows a trend towards greater abstraction, modularity, and a focus on interoperability. As the cloud becomes a fundamental part of organizational IT strategies, the insights gained from SOA are crucial in developing future-proof, resilient, and scalable cloud architectures. The next sections will go into more detail about the practicalities of adopting these architectures, focusing on their real-world applications and strategic considerations for organizations on this transformative path.

5.9 Microservices and Cloud-Native Patterns

Delving deeper into modern software architectures, this chapter concentrates on the microservices architectural style, a key element in cloud-native application development. We'll explore the details of microservices, their benefits, challenges, and their significant role in helping organizations achieve agility and scalability in the cloud.

5.9.1 The Microservices Architecture

Microservices architecture is a method of developing a single application as a collection of small services. Each of these services runs in its own process and communicates using lightweight mechanisms, typically an HTTP-based API. These services, focused on specific business capabilities, are independently deployable through fully automated deployment processes. [22]

Key Characteristics of Microservices

Fine-Grained: Microservices are more detailed and specific than those in SOA, with each microservice handling a distinct, small part of business functionality.

Cloud-Ready and Scalable: Designed for cloud environments, microservices can efficiently scale up or down as required. They are deployable across various servers, cloud regions, or even across different cloud providers.

Configuration-Intensive: Deploying microservices involves considerable configuration effort to manage interactions between service instances.

Demanding on Software Architecture: Microservices require a sophisticated approach to software architecture, including strategies for breaking down applications, defining APIs, managing distributed data, and orchestrating services. [23]

5.9.2 Communication in Microservices

Increased Communication: The distributed nature of microservices leads to a significant amount of network communication.

Patterns and Strategies: Techniques like API Gateways, Backend for Frontends (BFFs), and Service Meshes are commonly used for effective communication management.

Event-Driven Architecture [24]: Microservices often utilize asynchronous, event-driven communication, sometimes using Event Sourcing and CQRS patterns to maintain data consistency.

5.9.3 Microservices Versus SOA and Monoliths

Comparison with SOA: Although both SOA and microservices are service-based, microservices advocate for smaller, more specialized services that are independently developed, deployed, and scaled.

Advancement over Monoliths: Unlike monolithic architectures, microservices enable independent updates, leading to more frequent updates and enhanced resilience.

Microservices offer significant flexibility in development and deployment, allowing the use of different programming languages, databases, and software environments. They also provide fault isolation, as issues in one service don’t necessarily impact others.

5.9.4 Cloud-Native Technologies Enabling Microservices

Containerization: Technologies like Docker package microservices into containers, ensuring portability and consistency across various environments.

Orchestration: Tools such as Kubernetes manage containerized services, taking care of deployment, scaling, and management, thereby streamlining operations.

5.9.5 Monitoring and Telemetry

With the complexity of managing multiple microservices, effective monitoring and telemetry are essential. [25]

Practices and tools for logging, monitoring, and tracing calls across services are critical to maintain an understanding of the system’s state and to identify issues promptly. For many organizations, transitioning to microservices means transforming their existing architectures.

From SOA: This involves refactoring existing SOA services into finer microservices, often by dividing larger services and applying domain-driven design principles.

From Monoliths: The process includes breaking down a monolithic architecture into microservices, typically starting with the parts of the system that change most frequently.

Microservices provide a pathway to create scalable, flexible, and resilient cloud-native applications that meet modern business needs for quick innovation and deployment. However, this approach also demands thoughtful consideration of design principles, operational complexity, and the need for a cultural shift within development teams. As organizations navigate through digital transformation, microservices emerge as a key component of modern architecture, aligning with the cloud-native approach and enabling businesses to excel in a dynamic, competitive environment.

6 The Emergence of Serverless Architecture and PaaS

Moving further into the world of cloud-native development, this chapter will investigate the significant shift from traditional infrastructure management to the newer models of Serverless computing and Platform as a Service (PaaS). We will examine the benefits they offer, the challenges they present, and their impact on the future of cloud computing.

6.1 Understanding PaaS and Serverless

Platform as a Service (PaaS) and Serverless are two essential cloud computing models that hide the complexity of the underlying infrastructure, letting developers focus more on coding and managing their applications. [26]

PaaS: Offers a platform that allows customers to develop, run, and manage applications without worrying about the complexity of building and maintaining infrastructure. PaaS includes services like databases, messaging queues, and storage options, managed by the cloud provider.

Serverless: Goes a step further by automatically managing the allocation of machine resources. Costs are based on the actual resources used by an application, rather than pre-purchased capacity units. It's similar to how electricity usage is measured and billed, treating compute power as a utility. [27]

6.2 Transitioning to Serverless

Moving to Serverless architecture is motivated by the need for high scalability, cost efficiency, and simplified operational management. Serverless architectures can scale up or down automatically in response to real-time demand, without manual intervention.

6.2.1 Cloud-Native Technologies

This chapter will also look at the range of cloud-native technologies that support the deployment, scaling, and management of applications:

Containers and Orchestration: Technologies like Docker and Kubernetes are crucial for cloud-native applications, providing tools to create, deploy, and manage scalable services.

Provisioning and Automation: Automation and configuration tools like Ansible, Terraform, and Chef help quickly provision cloud resources.

Continuous Integration and Continuous Deployment (CI/CD): Systems like Jenkins and GitLab CI/CD enable automated testing and deployment, promoting continuous improvement and agility.

6.2.2 DevSecOps and Cloud-Native Operations

DevSecOps integrates security practices into the DevOps process, emphasizing security decisions at the same pace as development and operations decisions, ensuring ongoing security. [28]

Infrastructure as Code (IaC): Managing infrastructure through machine-readable files rather than through physical hardware or interactive tools. [29]

Monitoring and Telemetry: Robust monitoring and telemetry are essential for understanding the performance and security of applications.

Serverless: Simplifying Deployment

Serverless computing focuses on making deployment and operation easier:

Reduced Overhead: It takes away the need for developers to handle server provisioning, maintenance, and scaling.

Integration and Development: Tools like AWS Lambda and Azure Functions enable seamless integration with the cloud ecosystem, often using familiar IDEs and development frameworks.

6.2.3 Challenges and Considerations

Despite their benefits, Serverless and PaaS have challenges:

Complexity of Configuration: The simplicity can be misleading, as there can be a lot of configuration work, especially with Kubernetes.

Vendor Lock-in: Relying on specific cloud providers' tools and services can lead to vendor lock-in, which may limit flexibility and increase costs if changing providers is needed. [21]

7 Conclusion

In conclusion, this seminar paper provides an in-depth analysis of the transition from traditional on-premise systems to more advanced cloud configurations, such as hybrid and multi-cloud environments. This transition is increasingly relevant for contemporary businesses. The paper methodically addresses the intricate challenges and strategic considerations essential for a successful migration to cloud-based systems.

From a technical standpoint, the paper explores various critical aspects of cloud migration. These include ensuring compatibility between existing and new systems, integrating legacy hardware with modern cloud infrastructures, safeguarding data security, and adhering to regulatory standards. These elements underscore the complexity and detailed nature of migrating to cloud environments, highlighting the necessity for thorough planning and technical precision.

On the strategic front, the paper delves into the decision-making process involved in selecting and managing cloud services. It discusses the nuances of serverless computing and the adoption of multi-cloud strategies. This section emphasizes the importance of strategic foresight in leveraging the distinct benefits offered by various cloud models, while also being mindful of potential pitfalls such as vendor lock-in.

A substantial portion of the paper is devoted to custom development within cloud environments. It examines various methodologies, including rehosting, replatforming, refactoring, and rearchitecting, providing valuable insights into tailoring cloud migration strategies to meet specific organizational needs. The transition from Service-Oriented Architecture (SOA) to cloud-native architectures, particularly the integration of microservices and serverless architectures, is highlighted. This transition marks a significant shift in application development and management, promising enhanced scalability, flexibility, and operational efficiency.

The paper concludes by offering a forward-looking perspective on cloud development. It posits that migrating to cloud-based systems represents not just a technological upgrade but a strategic transformation with significant implications for a company's operational dynamics. As cloud technologies continue to evolve, it becomes imperative for businesses to remain informed and adaptable, ready to capitalize on emerging opportunities and innovations in the cloud computing sphere.

Overall, this seminar paper serves as a comprehensive guide for organizations embarking on their cloud migration journey. It provides a detailed overview of the challenges, strategies, and considerations crucial for a successful transition to cloud-based systems, aiming to equip organizations with the knowledge and understanding necessary to navigate the complex landscape of cloud computing effectively.

8 Bibliography

Hier sind die angegebenen Quellen im IEEE-Stil formatiert:

1. D. Hirsch, "An overview of the cloud migration process: a case study of a migration from legacy application to cloud-based application," Luleå University of Technology, Luleå, 2023.

2. N. Antonopoulos, *Cloud Computing Principles, Systems and Applications*. Springer, 2017.

3. National Institute of Standards and Technology, "National Institute of Standards and Technology," September 2011. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-145.pdf. [Accessed: Jan. 26, 2024].

4. Amazon, "Amazon AWS." [Online]. Available: https://aws.amazon.com. [Accessed: Jan. 26, 2024].

5. Microsoft, "Microsoft Azure." [Online]. Available: https://azure.microsoft.com/en-gb/. [Accessed: Jan. 26, 2024].

6. Google, "Google Cloud." [Online]. Available: https://cloud.google.com/?hl=de. [Accessed: Jan. 26, 2024].

7. Alibaba, "Alibaba Cloud." [Online]. Available: https://eu.alibabacloud.com/en. [Accessed: Jan. 26, 2024].

8. Deloitte, "Deloitte," July 2021. [Online]. Available: https://www2.deloitte.com/content/dam/Deloitte/de/Documents/public-sector/Deloitte-Cloud-in-der-oeffentlichen-Verwaltung-2021.pdf. [Accessed: Jan. 26, 2024].

9. Synergy Research Group. [Online]. Available: https://www.sans.org/blog/doing-cloud-in-china/. [Accessed: Jan. 26, 2024].

10. European Union, "GDPR." [Online]. Available: https://gdpr.eu. [Accessed: Jan. 26, 2024].

11. Liongate, "Liongate," Nov. 22, 2022. [Online]. Available: https://www.liongate.de/de/news/eu-us-privacy-shield-cloud-act-dsgvo-und-die-cloud/. [Accessed: Jan. 26, 2024].

12. Microsoft, "Microsoft Learn." [Online]. Available: https://learn.microsoft.com/de-de/azure/cloud-adoption-framework/digital-estate/5-rs-of-rationalization. [Accessed: Jan. 26, 2024].

13. Amazon, "Amazon AWS." [Online]. Available: https://docs.aws.amazon.com/whitepapers/latest/serverless-multi-tier-architectures-api-gateway-lambda/three-tier-architecture-overview.html. [Accessed: Jan. 26, 2024].

14. Cloud Native Computing Foundation, "Cloud Native Computing Foundation." [Online]. Available: https://glossary.cncf.io/client-server-architecture/. [Accessed: Jan. 26, 2024].

15. Microsoft, "Microsoft," 2017. [Online]. Available: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwjQjJf36fqDAxXu8gIHHcRICmwQFnoECA8QAQ&url=https%3A%2F%2Fdownload.microsoft.com%2Fdownload%2FD%2FD%2FF%2FDDF9069B-0189-473D-87A2-7016A7DCA140%2FModernize_existing_net_applications_wit. [Accessed: Jan. 26, 2024].

16. Amazon, "Amazon AWS." [Online]. Available: https://aws.amazon.com/de/what-is/service-oriented-architecture/. [Accessed: Jan. 26, 2024].

17. Microsoft, "Microsoft." [Online]. Available: https://learn.microsoft.com/de-de/biztalk/core/introducing-biztalk-server. [Accessed: Jan. 26, 2024].

18. RabbitMQ, "RabbitMQ." [Online]. Available: https://rabbitmq.com/documentation.html. [Accessed: Jan. 26, 2024].

19. Google, "Google Cloud." [Online]. Available: https://cloud.google.com/containers?hl=en. [Accessed: Jan. 26, 2024].

20. Kubernetes, "Kubernetes." [Online]. Available: https://kubernetes.io/de/. [Accessed: Jan. 26, 2024].

21. O.-M. et al., "Critical analysis of vendor lock-in and its impact on cloud computing migration: a business perspective," *Journal of Cloud Computing: Advances, Systems and Applications*, 2016.

22. M. Fowler, "Martin Fowler's Blog," Aug. 21, 2019. [Online]. Available: https://martinfowler.com/microservices/. [Accessed: Jan. 26, 2024].

23. Amazon, "Amazon AWS." [Online]. Available: https://aws.amazon.com/de/compare/the-difference-between-soa-microservices/#:~:text=Die%20Microservices%2DArchitektur%20ist%20eine,auf%20eine%20einzige%20Aufgabe%20spezialisiert. [Accessed: Jan. 26, 2024].

24. M. Fowler, "Martin Fowler's Blog," Dec. 12, 2005. [Online]. Available: https://martinfowler.com/eaaDev/EventSourcing.html. [Accessed: Jan. 26, 2024].

25. Basta Konferenz, "YouTube." [Online]. Available: https://www.youtube.com/watch?v=sk1NPkAzYSM. [Accessed: Jan. 26, 2024].

26. GoTo Conference, "YouTube." [Online]. Available: https://youtu.be/dUXDuxqTxmo?si=XmMONpeC7axnctmR. [Accessed: Jan. 26, 2024].

27. M. Fowler, "Martin Fowler's Blog," May 22, 2018. [Online]. Available: https://martinfowler.com/articles/serverless.html. [Accessed: Jan. 26, 2024].

28. Amazon AWS, "YouTube." [Online]. Available: https://youtu.be/xCyRCBGfaqk?si=Y2krD-s4ZG0byVw7. [Accessed: Jan. 26, 2024].

29. Cloud Native Computing Foundation, "Cloud Native Computing Foundation." [Online]. Available: https://glossary.cncf.io/infrastructure-as-code/. [Accessed: Jan. 26, 2024].

Ehrenwörtliche Erklärung

Ich versichere hiermit ehrenwörtlich, dass ich die vorliegende Hausarbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder unveröffentlichten Quellen entnommen worden sind, sind als solche kenntlich gemacht. Dazu zählen auch Abbildungen, Grafiken, Tabellen, einzelne Zahlenwerte, Skizzen oder bildliche Darstellungen. Diese Arbeit lag in gleicher oder ähnlicher Form – auch nicht in Teilen – noch keiner anderen Prüfungsbehörde vor und wurde bisher noch nicht veröffentlicht.

Frankfurt, 26.01.2024

[illegible]