



Software Defined Systems

Week 1 Introduction to Software Defined Systems

Lecturer: Biniam Behailu

Addis Ababa Science and Technology

Addis Ababa, Ethiopia

Course Description

- This course explores the transformative approach of Software Defined Systems (SDS) in IT infrastructure.
- Key topics include: Software Defined Networking (SDN), Software Defined Storage, Software Defined Servers, Software Defined Datacenters, Software Defined Security (SDSec), Software Defined Clouds (SDCloud).
- Students will learn how these technologies function individually and integrate seamlessly. The course covers essential virtualization concepts, including hypervisors, and their role in enabling SDS.

Contents

- 01 Overview of Software-Defined Systems
- 02 Deep Dive into Key Elements
- 03 Characteristics that Define SDS
- 04 Comprehensive Benefits Analysis
- 05 Diverse Use Cases
- 06 Real-World Challenges
- 07 Industry Adoption

Introduction to Software Defined Systems

Learning objectives

- Articulate the key concepts and definitions of software-defined systems.
- Examine the evolution from traditional to software-defined architectures.
- Recognize and explain key components.
- Assess characteristics such as abstraction, automation, and scalability.
- Explain the benefits and use cases of SDS in various applications.
- Identify challenges in implementing software-defined systems.

Introduction – Conventional Systems

- Conventional systems refer to traditional computing architectures where hardware and software are tightly integrated.
- In conventional systems, hardware is designed to work with specific software, which means the hardware and software are often optimized for each.
- Each application or service typically runs on dedicated hardware, leading to underutilization of resources.

Introduction – Conventional Systems

- Conventional control and management platforms are facing considerable challenges regarding flexibility, dependability and security that next generation systems will require.

Limited Flexibility

Scalability Issues

Hardware lock-in

Incremental Scaling

Physical Constraints

Higher Capital Expenditures

Security

Resource Underutilization

Complex Management



Software Defined Systems

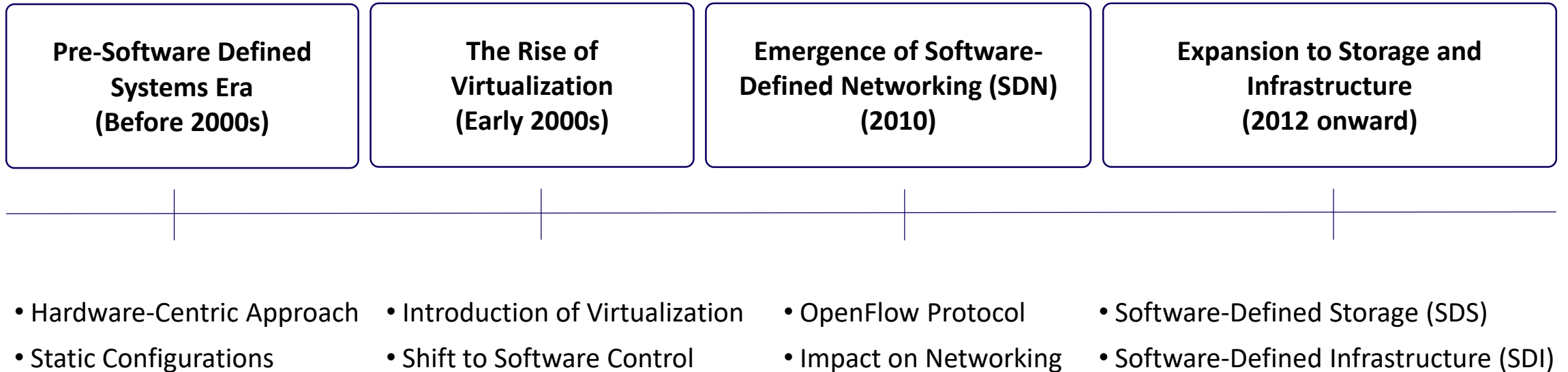
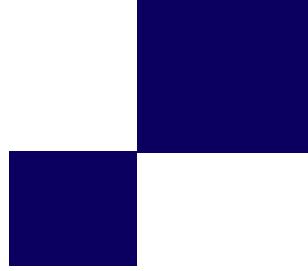
What are Software Defined Systems?

- A software-defined system is an approach to designing and managing computer systems where the **functionality** and **behavior** of the system are controlled primarily through software.
- The term “***software defined***” can refer to either the software that defines the architecture, or the architecture itself.

What are Software Defined Systems?

- It involves virtualizing and abstracting the underlying hardware resources, such as processors, storage, and networking, and providing a software interface for managing and controlling these resources.
- In a software-defined system, the control and management of the system are decoupled from the underlying hardware.
- This separation allows for greater flexibility, scalability, and agility in managing and provisioning resources.

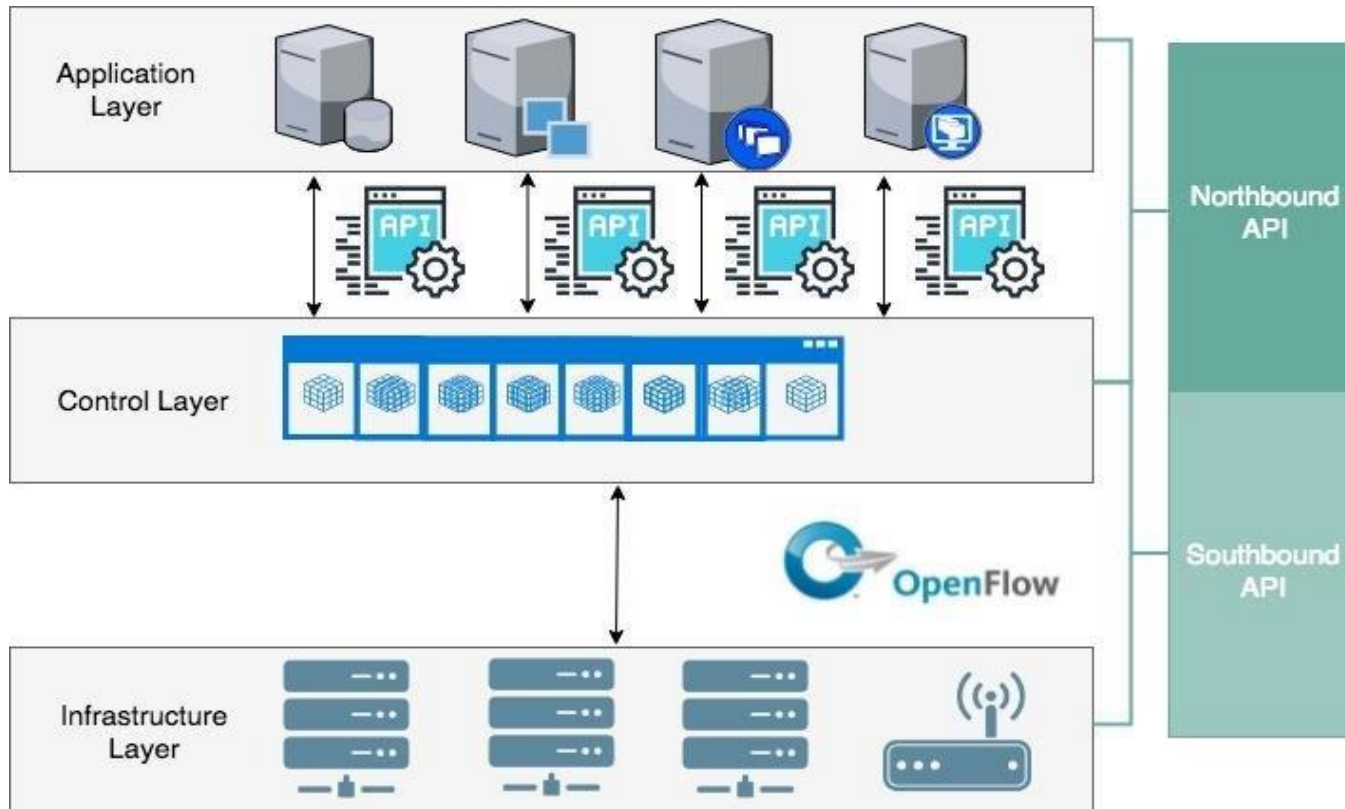
Evolution of IT Infrastructure



Software Defined Architecture

- The Software Defined Architecture gives large cloud services like Amazon and Microsoft Azure the ability to operate at **Web scale**, serving massive numbers of consumers while adapting to rapidly changing needs.
- Software-defined Architecture is the Foundation of IT Infrastructure Transformation.

Software Defined Architecture



- SDA creates a layer of virtualization between the **infrastructure** and the **application** layers.

Source: https://www.researchgate.net/figure/Diagram-of-software-Define-Architecture_fig1_327281564

Terminologies in Software Defined Systems

- Software-defined systems are built on several fundamental components that work together to create a flexible, efficient, and manageable IT infrastructure.
 - Software-Defined Networking (SDN)
 - Software-Defined Storage (SDS)
 - Software-Defined Compute (SDC)
 - Virtualization
 - Orchestration
 - Automation Tools

Software-Defined Networking (SDN)



- SDN abstracts the network infrastructure, separating the control plane from the data plane.
- It provides a centralized software controller that manages and configures network devices, enabling more flexible and dynamic network management.
- Enables dynamic network configuration, traffic management, and automated provisioning of network services.

Software-Defined Storage (SDS)

- Abstracts storage resources from the underlying hardware, allowing for flexible management and allocation.
- It enables the dynamic allocation and management of storage resources based on application requirements, improving scalability and efficiency.
- Enables automated storage provisioning, and data replication based on performance needs.

Software-Defined Compute (SDC)

- SDC virtualizes the compute resources, such as processors and memory, providing the ability to dynamically allocate and manage computational capacity.
- It allows for the creation and management of virtual machines or containers that can be provisioned as needed.
- It decouples physical hardware from the software that utilizes it, enabling dynamic provisioning and orchestration of computing resources.

Virtualization

- Virtualization is the technology that allows you to create virtual instances of physical computing resources, such as servers, storage devices, and networks.
- It abstracts the underlying hardware, enabling multiple virtual environments to run simultaneously on a single physical machine.
- It maximizes resource utilization, reduces hardware costs, and enhances flexibility in resource management.

Orchestration



- Orchestration is the automated management and coordination of complex IT systems and services.
- It involves integrating multiple processes and tasks to streamline workflows, manage resources, and automate the deployment and operation of applications across various environments.
- Streamlines workflows, automates deployment processes, and ensures consistency in resource allocation.

Automation Tools



- Automation tools are essential for enhancing efficiency in software-defined systems.
- These tools automate repetitive tasks such as provisioning, configuration, and monitoring, reducing the need for manual intervention.
- Software solutions that automate repetitive tasks and processes within the IT infrastructure.
- Improves efficiency by reducing manual intervention and speeding up deployment cycles.

Key Characteristics of SDS

- Decoupling of Hardware and Software
- Centralized Control
- Automation
- Dynamic Resource Allocation
- Programmability
- Scalability
- Virtualization
- Interoperability

Abstraction in SDS

- Abstraction in SDS involves separating the high-level management of resources from the underlying hardware, allowing users to interact with simplified interfaces rather than complex physical components.

Simplified Management

Resource Pooling

Dynamic Provisioning

Interoperability

Automation in SDS

- Automation in SDS refers to the use of software tools and processes to manage and orchestrate IT resources without human intervention, streamlining operations and enhancing efficiency.

Provisioning and Deployment

Configuration Management

Monitoring and Alerts

Scaling and Load Balancing

Scalability in SDS

- One of the major characteristics of Software Defined Systems (SDS) is their inherent scalability.
- This scalability stems from the core principle of separating the control plane (software) from the data plane (hardware).

Independent Scaling

Horizontal Scalability

Vertical Scaling

Dynamic Resource Allocation

Improved Management in SDS

- Improved management in software-defined systems refers to the enhanced capabilities for monitoring, controlling, and optimizing IT resources through centralized and automated processes.

Centralized Control

Automated Processes

Real-Time Monitoring

Enhanced Visibility



What are advantages of using Software Defined Systems?

Advantages of Software Defined Systems

- Increased Agility
- Cost Efficiency
- Enhanced Resource Utilization
- Simplified Management
- Improved Security
- Scalability
- Interoperability

Real-World Applications

- Data Centers
- Networking
- Cloud Computing
- Internet of Things (IoT)

SDS in Cloud Computing

- SDS allows for real-time allocation of resources based on workload demands, ensuring efficient use of cloud infrastructure.
- Organizations can easily scale resources up or down, responding to changing needs without significant downtime.
- By optimizing resource usage, SDS minimizes unnecessary expenditures, enabling a pay-as-you-go model in cloud environments.

Case study: Netflix

Scenario

- In 2008, Netflix faced a significant challenge when a database corruption caused a three-day halt in DVD shipments. This incident highlighted the limitations of traditional, vertically scaled data centers and inspired Netflix to migrate to the cloud. They partnered with Amazon Web Services (AWS) and began a seven-year transformation to become a fully cloud-native company.

The Netflix logo is displayed in a bold, red, sans-serif font. The letters are thick and have a slightly irregular, hand-drawn appearance. The logo is positioned in the upper right quadrant of the slide, below a decorative graphic of two dark blue squares.

Source: <https://www.techdaily.com.au/b2/making-the-most-of-netflix-in-australia-legally/>

Case study: Netflix

Objective

- To enhance service reliability, scalability, and innovation by transitioning from traditional data centers to a cloud-based infrastructure.
- Building infrastructure that can handle Netflix's ever growing user demand and requests.

Case study: Netflix

Process and Strategy for Implementing

1. **Cloud-Native Approach:** Instead of replicating existing systems in the cloud, Netflix re-architected its systems from the ground up.
2. **Elasticity:** By leveraging the cloud, Netflix gained the ability to instantly add thousands of virtual servers and petabytes of storage to handle rapid growth.

Case study: Netflix

Process and Strategy for Implementing

3. **Resilience Building:** Designed systems with redundancy and graceful degradation principles.
4. **Team Transformation:** Shifted to a DevOps environment, empowering engineering teams to make independent decisions.

Case study: Netflix

Scalability

Netflix experienced a massive surge in its user base and the amount of content being streamed. Created capability to scale its infrastructure based on need.

Global Expansion

Enabled service deployment in 130+ new countries, making Netflix a global internet TV network.

Improved Availability

Achieved near four nines (99.99%) of service uptime by designing fault-tolerant systems.

Case study: Netflix

Cost Optimization

Lowered streaming costs per start by leveraging the elasticity and economies of scale offered by the cloud.

Innovation Growth

Rebuilt their infrastructure and adopted cutting-edge practices that positioned Netflix for continuous improvement and expansion.

SDS in IoT



- SDS facilitates the seamless integration of various IoT devices, enabling them to communicate and operate effectively within a unified system.
- As the number of IoT devices grows, SDS allows for easy scaling to accommodate increased data and device management needs.
- SDS handles the influx of data from multiple sources, allowing for real-time analysis and decision-making.

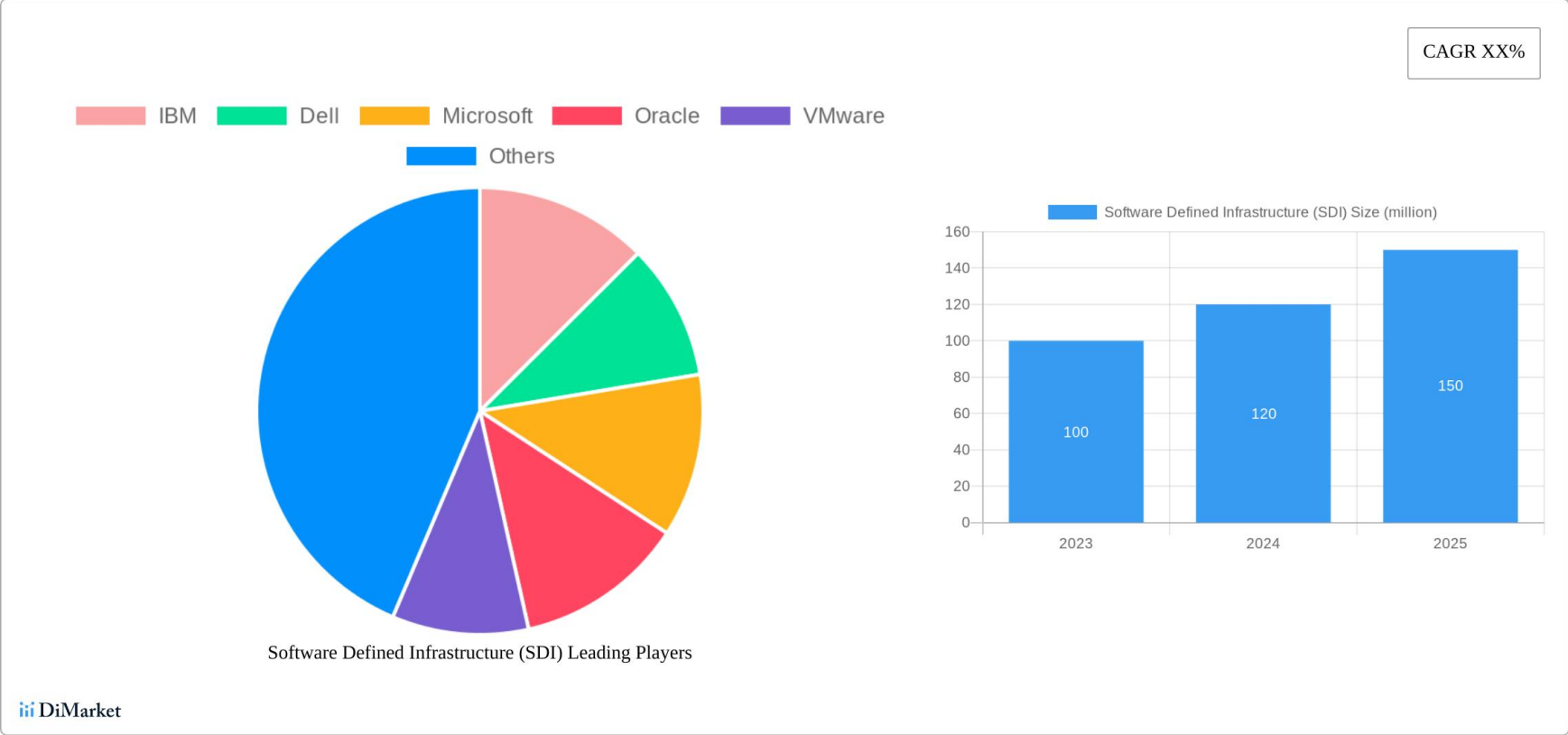
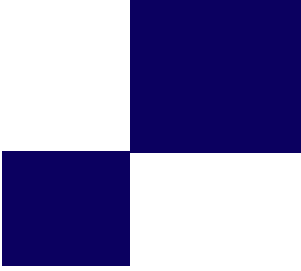
Challenges of Software-Defined Systems

- Security Concerns
- Complexity of Management
- Integration with Legacy Systems
- Performance Overheads
- Regulatory Compliance
- Skill Gaps

Industry Adoption for SDS

- Recent surveys indicate that over 75% of enterprises are investing in or planning to invest in software-defined technologies within the next 2 years.
- The Software Defined Infrastructure (SDI) market is projected to reach a value of USD 57.85 billion by 2033, exhibiting a CAGR of 25.6% during the forecast period from 2025 to 2033 [\[1\]](#).

Industry Adoption for SDS



Software Defined Infrastructure (SDI) Leading players

Source: <https://www.datainsightsmarket.com/>



Software Defined Everything

Software Defined Everything



- Software Defined Everything (SDE) is an architectural approach that extends the principles of Software Defined Networking (SDN) beyond networking to encompass all aspects of IT infrastructure, including storage, compute, and security.
- By decoupling software from hardware, SDE enables more flexible, programmable, and efficient management of resources across an entire IT ecosystem.

Software Defined Everything

- Software Defined Networking (SDN)
- Software Defined Storage (SDS)
- Software Defined Compute (SDC)
- Software Defined Security (SDSec)



Future Trends in Software-Defined Systems

Future Trends in SDS

- Artificial Intelligence (AI) and Machine Learning (ML) Integration
- Enhanced Security
- Hybrid and Multi-Cloud Environments
- Edge Computing
- Network Function Virtualization (NFV) Evolution
- Programmability and Open APIs

Test Your Knowledge

1. Which of the following is a key benefit of software-defined networking (SDN)?
 - A) Increased hardware dependency
 - B) Static resource allocation
 - C) Centralized management
 - D) Complex configurations

Test Your Knowledge

1. Which of the following is a key benefit of software-defined networking (SDN)?
 - A) Increased hardware dependency
 - B) Static resource allocation
 - C) Centralized management**
 - D) Complex configurations

Reason: One of the primary benefits of software-defined networking (SDN) is the ability to manage network resources centrally, simplifying administration and configuration.

Test Your Knowledge

2. What does "software-defined" primarily refer to in software-defined systems?
 - A) Hardware-based solutions
 - B) Separation of hardware from software
 - C) Manual configuration
 - D) Legacy systems

Test Your Knowledge

2. What does "software-defined" primarily refer to in software-defined systems?
- A) Hardware-based solutions
 - B) Separation of hardware from software**
 - C) Manual configuration
 - D) Legacy systems

Reason: Software-defined systems emphasize the decoupling of hardware and software, allowing for greater flexibility and easier management.

References

- [1] Data Insights Market, Jan. 2025. [Online]. Available: <https://www.datainsightsmarket.com/reports/software-defined-infrastructure-sdi-537645>.
- [2] Sadiku, M & Nelatury, Sudarshan & Musa, Sarhan. (2017). 2017 software defined everything. 4. 48-50.



Thank you!

Lecturer: Biniam Behailu

Addis Ababa Science and Technology

Addis Ababa, Ethiopia