# The Virtual Computer Networks Lab: On the Design and Implementation of a Location Independent Networks Laboratory in Higher Education

Christian Baun[a], Martin Kappes[b], Henry-Norbert Cocos[c], Malte Koch and Maurizio Petrozziello[d]

*Department of Computer Science and Engineerung, Frankfurt University of Applied Sciences, Nibelungeplatz 1, Frankfurt am Main D-60318, Germany*

*{christianbaun, kappes, cocos, maltekoch, petrozziello}@fb2.fra-uas.de*

Keywords: Virtual Teaching, Proxmox VE, CEPH, Teaching Computer Science, Asynchronous Teaching.

Abstract: This paper describes a novel virtual platform for university teaching, which in particular allows the creation and use of complex IT infrastructures even for non-experts. Until now, complex network infrastructures in teaching have usually not been realizable virtually or physically with reasonable effort. However, they are essential to fundamental courses such as computer networks, distributed systems, or IT security. The platform described in this work enables a flexible and practical offer for lecturers, scientists, and students simultaneously by making complex IT infrastructures virtually available quickly and easily. It thus offers new possibilities for designing and utilizing virtual resources and is based exclusively on free software components. The project uses the hypervisor technology Proxmox Virtual Environment, which, together with the distributed storage solution CEPH as an object storage service, offers a virtualization platform for providing virtual resources. The successful use in computer science courses is demonstrated and evaluated. It is shown that the effort for setting up practical laboratories is signifficantly reduced by the virtual learning platform in contrast to the physical lab.



## 1 INTRODUCTION

The COVID-19 pandemic has highlighted the urgent need for digitalization, particularly in education, where flexible, high-quality teaching solutions are essential. Combining innovative face-to-face courses with digital elements requires user-friendly IT infrastructures that adapt to changing demands.

Digitalization enriches university teaching through modern methods like online tests and media content, complementing traditional formats (Erpenbeck and Sauter, 2017). Approaches include digitized teaching concepts and online courses (Beckmann, 2020), with key factors being course suitability, content selection, and digitized elements (Handke, 2015).

While tools like Moodle (Athaya et al., 2021) address specific needs like participant registration, they overlook practical, location-independent labs. There is a need for virtual teaching solutions that expand learning beyond face-to-face constraints (Handke and Schäfer, 2012). Current solutions lack options for designing and using complex IT infrastructures.

This paper addresses these gaps, proposing a platform for intuitive configuration and use of complex IT scenarios via virtualization, even for users with limited expertise. It discusses the infrastructure's design, functionality, scalability, and security, illustrating its application in teaching and outlining future directions.

## 2 BACKGROUND AND RELATED WORK

Projects in digitized infrastructures for academia are not new, with examples like the Fed4Fire+ project(Serrano et al., 2022) and SLICES-RI(Demchenko et al., 2023). Both focus on large testbeds for research in networking and cloud-to-edge

[a] https://orcid.org/0009-0004-9955-3752
[b] https://orcid.org/0000-0002-8768-8359
[c] https://orcid.org/0009-0001-7573-0361
[d] https://orcid.org/0009-0003-7916-2001



Baun, C., Kappes, M., Cocos, H.-N., Koch, M. and Petrozziello, M.
The Virtual Computer Networks Lab: On the Design and Implementation of a Location Independent Networks Laboratory in Higher Education.
DOI: 10.5220/0013199400003932

In *Proceedings of the 17th International Conference on Computer Supported Education (CSEDU 2025) - Volume 1*, pages 199-207
ISBN: 978-989-758-746-7; ISSN: 2184-5026

computing. However, these initiatives target research rather than teaching needs in higher education.

For network education, tools like Packet Tracer(Abdul Rashid et al., 2019), mininet(Lantz et al., 2010), and GNS3 (Emiliano and Antunes, 2015) are widely used. Packet Tracer, by Cisco, offers an intuitive drag-and-drop interface for simulated network topologies but is tied to Cisco hardware. Mininet enables software-defined network simulations using Linux-Kernel virtualization but lacks a user-friendly interface, making it less accessible to students. GNS3 combines virtual and real devices for complex network simulations but is limited to networking scenarios and lacks extensibility to broader applications.

Dietz's work (Dietz, 2023) on a virtual lab for cybersecurity demonstrates the potential of cloud-based tools like IONOS Data Center Designer (DCD) for user-friendly IT scenario setups. However, relying on public cloud service providers leads to high costs and limited control. The SKILL/VL platform addresses these issues by using open-source components and operating on-premises, ensuring extensibility and complete infrastructure control.

# 3 DESIGNING A VIRTUAL LAB

The decision to design a virtual lab to simulate networking scenarios stems from the necessity of extending the physical lab for computer networks since the physical lab only has space for a maximum number of 24 students in a class. Another reason for designing such a virtual lab stems from the experiences during the COVID-19 pandemic. Asynchronous teaching methods using digital platforms were a pivotal element for teaching in higher education during that period and essential for successfully implementing courses without the need for face-to-face teaching. The use of a virtual teaching platform is beneficial out of logic reasons like:

- **Time and Location-Independent Offers.** The increasingly problematic nature of housing costs in the metropolitan area makes it increasingly difficult for students to live where they study. As a result, many students accept long traveling times and distances to study at their preferred university. The partial virtualization of courses can partially compensate for these problems.
- **Improving Collaboration Capatilities.** Virtualization makes scenarios possible that would otherwise be difficult to implement, such as students at partner universities working together on the same project, for example, in computer science, where laboratories can be virtually interconnected over large distances (continents). Examples are IT security and IoT, where we already have had good experiences.
- **Virtual Sessions and Labs.** Providing virtual workspaces for students, regardless of location, is a major step towards reconciling work and study. Students can attend courses and labs in the evenings, making studying possible for many today.

Table 1: Characteristics of physical and virtual labs (Alam and Mohanty, 2023).

| Characteristics | Physical Lab | Virtual Lab |
|---|---|---|
| Realism | High | High |
| Content | Stable | Dynamic |
| Focus on study | Lecturer | Student |
| Form | Synchronous | Asynchronous |
| Number of students | Limited | No (physical) limits |
| Time | Scheduled | Anytime |
| Focus of course | Group | Individual |
| Accessibility | Low | High |
| Cost | Very high | Low |
| Maintenance effort | Very high | Low |
| Remote work | Impossible | Possible |

Therefore, the focus was to implement a virtual lab that is flexible and powerful enough for the operation of networking courses. Table 1 presents characteristics demonstrating the benefits of using a virtual lab in higher education. However there are still limitations to the setup of virtual labs like the hazard of a lack of encouragement in communication and isolation of students. We have seen this issues in the COVID-19 pandemic and do not advocate to steer away from learning spaces in presence. However we advocate to extend the physical resources (in this work a computer networks lab) by virtual resources and extend the possibilities for location independent learning.

At the start of the project, open-source tools for simulating networking courses were evaluated. Section 2 introduced three alternatives, and Table 2 summarizes the evaluation criteria: open source, simulator, and automatic setup. These were critical for the platform's development, with open source being key due to cost-effectiveness in higher education.

Table 2: Comparison of SKILL/VL to exisitng solutions.

| Criteria | SKILL/VL | Cisco Packet Tracer | Mininet | GNS3 |
|---|---|---|---|---|
| Open Source | Yes | No | Yes | Yes |
| Simulator | Yes | Yes | No | Yes |
| Emulator | No | No | Yes | Yes |
| Scalability | Yes (by extending web service) | No | Limited (using more processes) | No |
| Automatic setup | Yes | No | No | No |
| Graphical User Interface | Yes | Yes | No | Yes |
| Extensible beyond networks | Yes | No | No | No |

- **Open Source.** Most inspected solutions, except Cisco, are free to use. This was crucial to avoid high licensing fees and maintain flexibility.
- **Simulator.** The platform needed to realistically represent real-world scenarios.
- **Automatic Setup.** Integration into the university's IT infrastructure was essential for seamless operation.

Among the tools, GNS3 stood out for being open-source and offering a graphical user interface, making it accessible to students with limited networking experience. However, its applicability is restricted to computer networking courses, limiting its versatility for broader educational needs.

Therefore, we developed SKILL/VL (Strategic Competence Platform - Innovative Learning and Teaching/Virtualization of distributed environments for teaching) as an extensible and comprehensive platform not limited to the computer networks' application field. It was intended to be capable of use in classes of distributed systems or software engineering, for example.

Section 4 outlines the infrastructure developed for the platform, focusing on the selection of open-source components and a design optimized for reliability. Section 5 details the implementation of these components and their internal communication, while Section 6 showcases a practical use case of the platform and compares it to the traditional physical computer networks lab.

# 4 INFRASTRUCTURE SKILL/VL

The virtualization platform operates on 12 high-performance servers in a clustered network, designed for virtual teaching across all disciplines and eventually the entire university. This requires high usability, scalability, and security standards.

The platform exclusively uses open source software, ensuring flexibility and avoiding licensing issues. It is developed as an on-premise cloud service (Baun et al., 2011), managed within the university's infrastructure but adaptable for public cloud deployment. At its core is Proxmox (Algarni et al., 2018; Chen et al., 2017), running on Debian Linux with KVM/QEMU for virtualization. Proxmox supports clustering and provides extensive control tools, making it ideal for professional virtual machine operations. The platform employs the VXLAN protocol (Mahalingam et al., 2014) to encapsulate virtual machines in isolated network enclosures, enabling secure and flexible virtual network management.

The user-friendly web interface (see Figure 1), built with NextJS and ReactFlow (Thakkar, 2020), allows intuitive drag-and-drop setup for virtual machines, networks, and learning environments. Templates simplify the creation and configuration of virtual labs, making the platform accessible even to users with limited technical expertise.

The interface provides students, academics, and teaching staff with easy access to the system, enabling the rapid development and use of complex IT scenarios. It simplifies course design and modernizes existing modules, allowing students to engage with practical content remotely. Currently, no comparable free system offers these features.

The platform's infrastructure is hyper-converged, where all required IT components are centrally managed and mapped to hardware using software-defined components (Meneses et al., 2021).

Figure 1 highlights key functionalities:

- Figure 1a: A graphical room creator for designing networking scenarios using drag-and-drop.
- Figure 1b: Real-time interaction with network components, showing IP and MAC addresses.
- Figure 1c: Permission configuration for virtual resources using Keycloak.
- Figure 1d: Virtual machine views integrated into a browser, eliminating the need for additional software.

The platform's goals are the exclusive use of open-source software and the creation of an intuitive virtual environment to enhance practical computer science education. Details of the implementation are in Section 5.

# 5 IMPLEMENTATION OF INDIVIDUAL COMPONENTS

The platform prioritizes free software, using established frameworks for reliability. Figure 2 illustrates the infrastructure and interactions. Keycloak manages user authentication, linking to the university's service and enabling a centralized, expandable single sign-on solution.

User interaction is facilitated via a NextJS-based web interface, where users can create and share templates for virtual learning spaces. Resource sharing is handled through Keycloak using the UMA protocol (Maler et al., 2019). The platform's backend, developed in Python with the FastAPI framework, communicates with the Proxmox hypervisor and manages the automatic creation of virtual machines and VXLANs via a REST interface.

Users can select pre-configured ISOs for operating systems, which come with the QEMU-Guest-Agent pre-installed to apply configurations, such as static IP addresses. The backend enforces user permissions, ensuring controlled access to virtual machines (e.g., restricting VNC access to specific VMs). The backend is designed for easy expansion, supporting potential integration with cloud providers like AWS, Google Cloud, or Azure for hybrid or multi-cloud scenarios.

# 6 PRACTICAL APPLICATION IN LABORATORIES

This section presents a practical application of the computer networks lab. The use case involves setting up a network with four hosts: two workstations and two routers. Figure 3 shows the network topology, serving as a reference for computer networks and IT security experiments in university labs. This setup simulates a basic real-world network configuration, chosen for comparison between the physical lab (Figure 4) and the virtual lab environment (Figure 5).

## 6.1 Network Topology

The hosts in the institution network are connected to the public network (`172.16.2.0/24`) through routers. Institution A (`192.168.1.0/24`) is connected via Router 1, and Institution B (`10.2.4.0/24`) is connected via Router 2. After installing the Linux OS, students configure network parameters, including IP addresses, gateways, and router settings. A JavaScript-based web service is set up on the machines, which communicates with clients via browser or `curl` using TCP sockets. Another host on the public network is used for testing and monitoring.

## 6.2 Setup in Physical Lab

The physical computer networking lab consists of 24 workstations, each with five Ethernet interfaces, organized into groups of four. Each workstation is connected to a switch, with additional network components provided by teaching staff based on the experiment. Workstations are linked to a management server via DHCP for network parameter distribution.

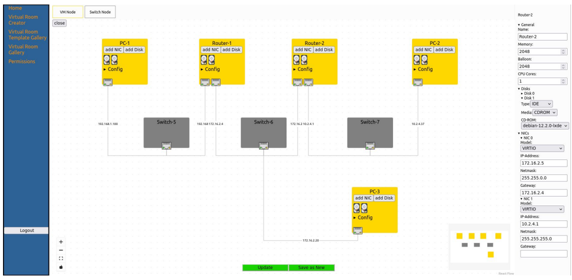
The management server uses the open-source imaging software linuxmuster (Steffen Auer, 2024), which provides different OS images, such as Linux or Windows, and automatically deploys images to workstations. Students or teachers select the appropriate image, which configures the host for the exercise, either automatically or manually. This functionality was adapted for the SKILL/VL platform, replicating the setup in a virtual form.

The network topology setup in the physical lab is time-consuming, taking students an average of four hours to complete. Much of this time is spent on setting up the workstations and networking equipment, diverting attention from the core task of machine configuration and routing.

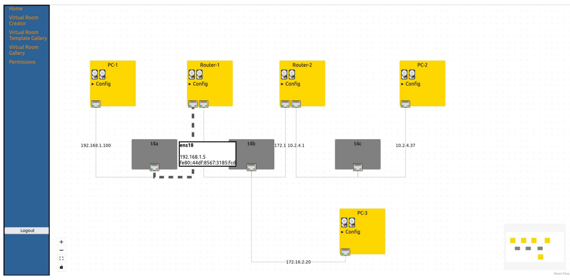
Additionally the lab is limited to 24 workstations, restricting the number of students who can work in parallel, which reduces the depth of learning and increases the setup overhead for teaching staff. The imaging system linuxmuster, while robust, is also time-consuming. It takes 30 minutes to install an image on one workstation, and up to 6 hours for the entire lab. This makes virtualization a more efficient alternative, especially for managing multiple scenarios across different exercises (further discussed in section 6.4).

## 6.3 Setup in Virtual Lab

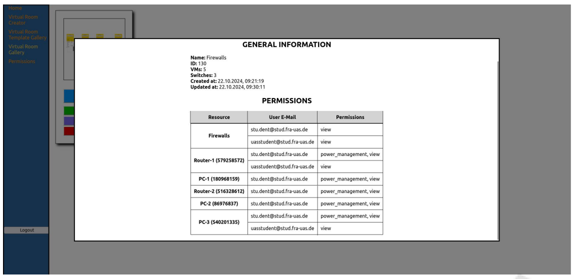
In class, students implemented the network topology described in section 6.1 in the virtual lab, complet-



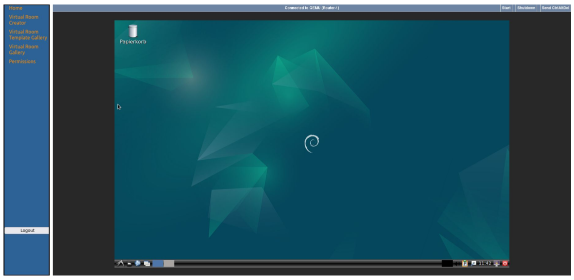
(a) Template for a learning room.



(b) Interactive selection of interfaces.



(c) Permission management for resources.



(d) Virtual machine running in browser.

Figure 1: The user web interface of the platform.

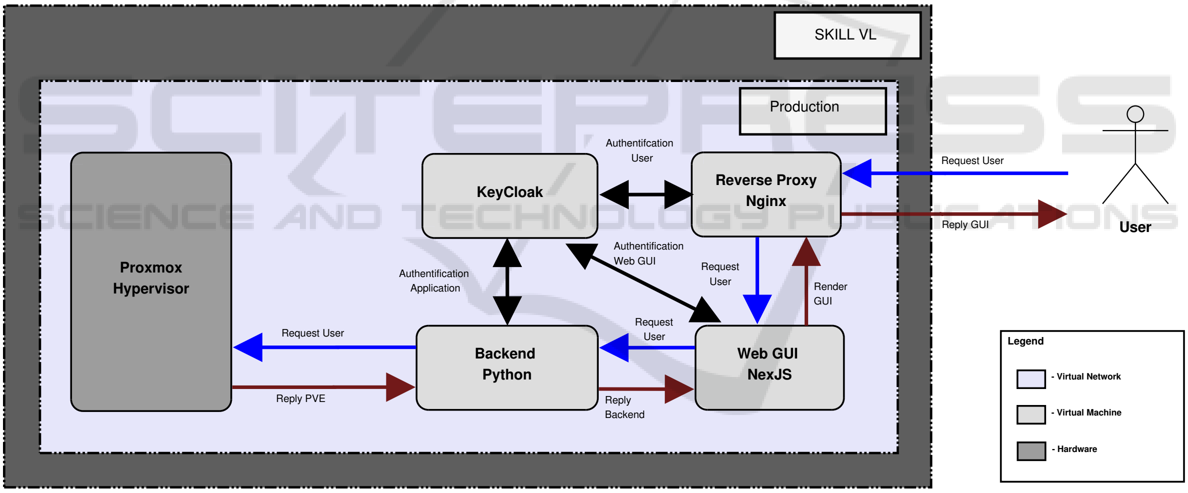



Figure 2: Software components of the SKILL/VL infrastructure and interaction.

ing the task in less time than in the physical lab. We also increased the complexity by adding a physical Raspberry Pi into the scenario. The hybrid approach of combining virtual hosts with a physical Raspberry Pi allowed us to extensively test the environment. SKILL/VL enables us to offer complex experiments to larger groups with minimal location restrictions, providing significant benefits to the students.

Our platform allows us to logically separate project and student groups to support independent work while enabling connections through virtual switches and routers when needed. This setup facilitates the creation of sophisticated network scenarios for teaching and research. The demonstrated lab example has been evaluated as part of a computer networks module.

Figure 6 illustrates how virtual learning rooms are implemented using VXLANs for physical separation. Each virtual room is isolated with its own VXLAN bridge, ensuring automatic and invisible network configuration. This approach offers individual address spaces for each user.

Unlike the physical lab, the infrastructure supports many students, reducing operational burdens. It simplifies the simulation of large corporate networks, which can be used for tasks like troubleshoot-

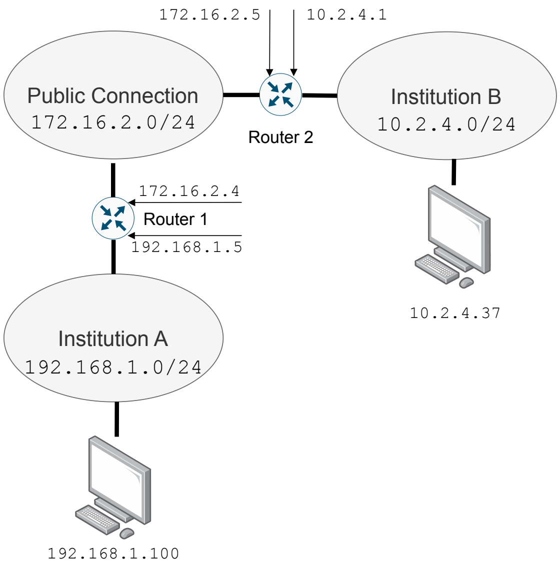



Figure 3: Topology of IT-Security class (Kappes, 2023).

Figure 4: Physical Lab for computer networks.

ing and testing security mechanisms. During the summer semester of 2023, a test run was conducted with computer science bachelor students, where they completed lab tasks using the platform, such as analyzing network traffic between virtual machines with Wireshark.

The test run provided valuable insights for further developing the platform in terms of user-friendliness, performance, and applicability. A challenge for many students was adapting to virtual cabling, as it requires greater abstraction. Some students also struggled with the web interface, which was still in an experimental phase. Additionally, delays (lags) were observed when interacting with virtual machines via the noVNC client, which streams the graphical output to the browser. While we rely on subjective feedback from students, technical feasibility showed no major impediments to user experience, and students successfully used the platform for networking tasks.

Table 3: Comparison of physical lab vs. virtual lab using SKILL/VL.

| **Characteristics** | **Physical Lab** | **SKILL/VL** |
|---|---|---|
| Images | LINBO | QEMU/KVM |
| Rollout time for images | 30 Minutes | 5 Minutes |
| Rollout of scenario | Time dependent | Time independent |
| Rollout of multiple parallel scenarios | Impossible | Possible |
| Snapshots | Impossible | Possible |
| Time | Scheduled | Anytime |
| Logical separation of networks | Whole lab | Individual group or student |
| Separation on network layer | Layer 3 (IP) | Layer 2 (VXLAN) |
| Focus of course | Group | Individual |
| Accessibility | Low | High |
| Remote Work | Impossible | Over ordinary web browser |
| Maintenance effort | Very high | Low |

The SKILL/VL platform emphasizes extensibility, making it suitable for various computer science courses. This is a key advantage over other tools like Mininet and GNS3, which are limited to computer network classes. Section 7 discusses the platform's potential in different application scenarios.

## 6.4 Comparison Physical vs Virtual Lab

The characteristics of the SKILL/VL platform make it very beneficial for use in the lab environments of the computer network classes. Table 3 shows the comparison of the different characteristics of the physical lab and the SKILL/VL platform and its benefits.

One key advantage of the SKILL/VL platform is the reduction in setup time. In the physical lab, setting up an experiment typically takes over an hour, whereas in SKILL/VL, this is significantly reduced, allowing students to focus more on configuring networks and monitoring traffic. The platform also allows for the parallel rollout of multiple scenarios, such as introducing configuration errors or vulnerabilities, which the physical lab cannot accommodate.

Additionally, SKILL/VL's use of VXLAN provides logical separation of networks at layer 2 of the OSI stack, unlike the physical lab, which separates

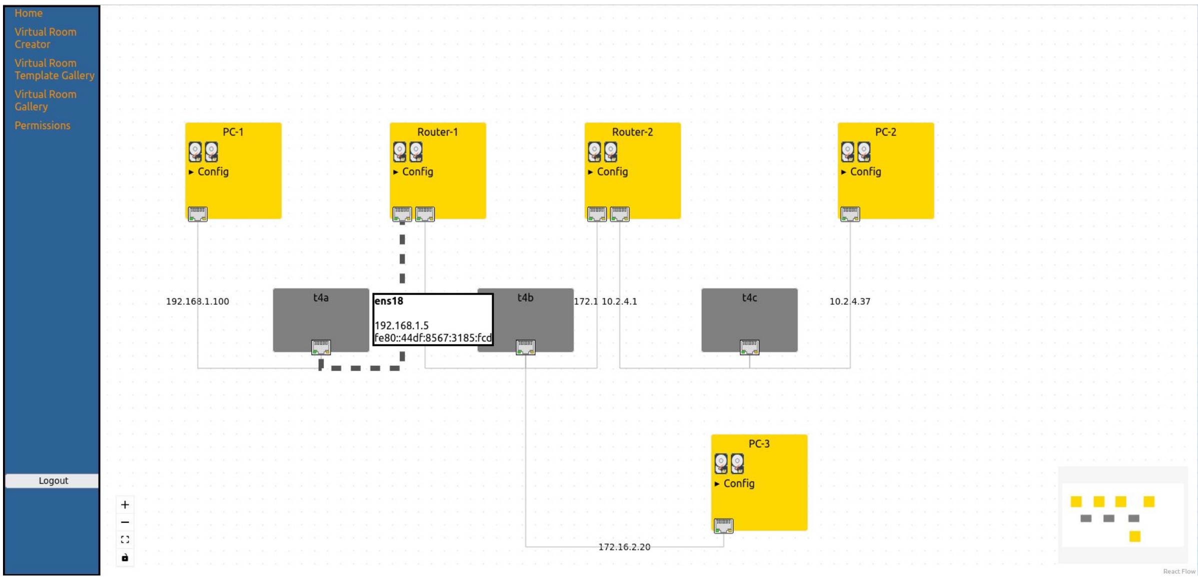



Figure 5: Virtual representation of network topology from Figure 3.

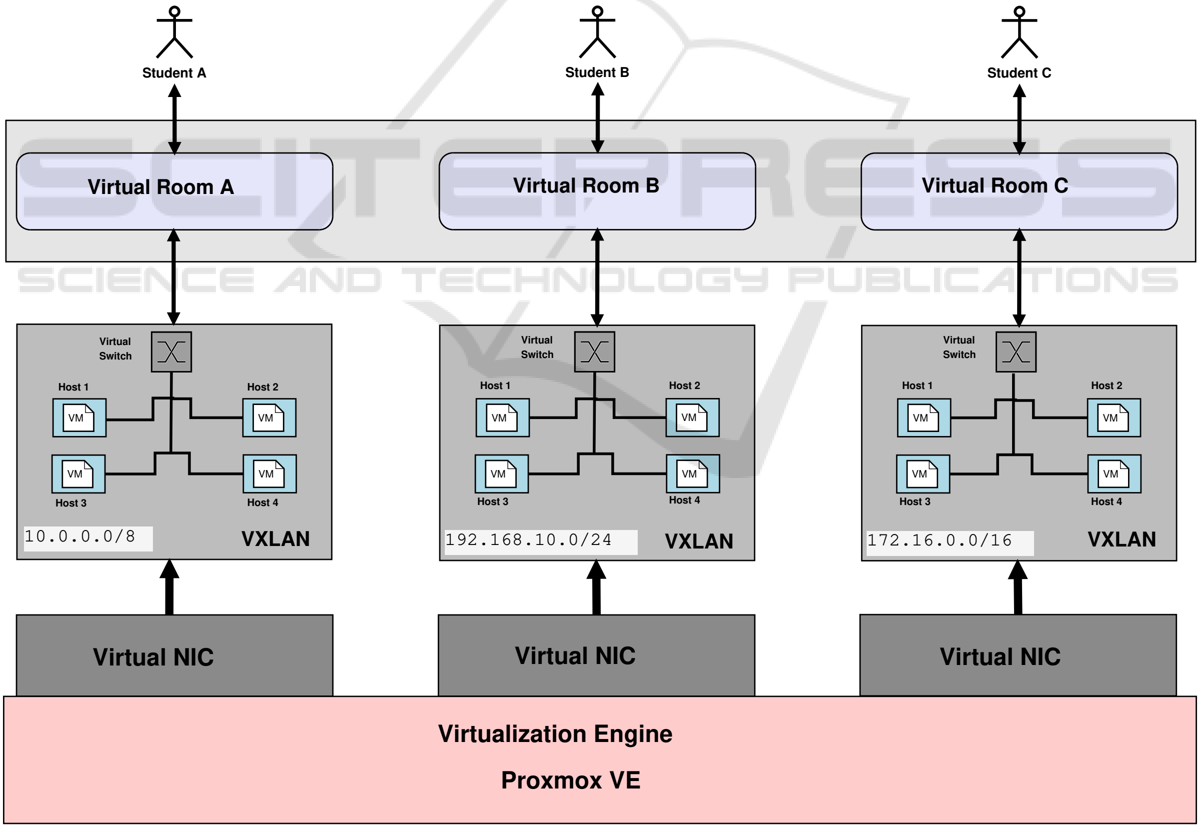





Figure 6: Mapping virtual learning spaces to physical components.

networks at the physical layer. This prevents address collisions and offers greater flexibility in network configurations. Another major benefit is the accessibility of the platform through a web browser, enabling remote work for both students and teachers.

A broader analysis of the platform's scalability has yet to be conducted. The setup, using Proxmox and SDN on 12 powerful servers (each with 64 CPU cores and 1 TB of RAM), is expected to support a whole class and more. Based on available resources, a single server can theoretically run 32 virtual machines or up to 256 VMs with sufficient memory. Equation 1 and 2 show the calculation for the assumptions. ¨With 12 servers, the theoretical capacity ranges from 384 to 3072 VMs. Realistically, with a 2:1 overprovisioning of CPU cores, reliable operation of up to 768 VMs is expected, enabling the simultaneous operation of over 150 experimental setups, compared to the 24 workstations in the physical lab. This increases the lab capacity by a factor of 6.25.

$$\frac{64\ physical\ CPU\ cores}{2\ vCPU\ cores\ per\ VM} = 32\ VMs\ per\ server \quad (1)$$

$$\frac{1024\ GB\ physical\ RAM}{4\ GB\ vRAM\ per\ VM} = 256\ VMs\ per\ server \quad (2)$$

However, VXLAN's limitation on the Maximum Transfer Unit (MTU) to 1450 bytes (compared to 1500 bytes in regular Ethernet networks) could impact performance in network benchmarks, though it does not affect configuration tasks

## 7 FUTURE WORK

The virtual environment demonstrated in section 6.3 shows great potential for computer network courses. However, its applications extend beyond networking. For example, in distributed applications courses, it allows the design of various distributed architectures, from basic three-layer to complex service-oriented architectures (Richards and Ford, 2020).

Another example is a virtual Internet of Things (IoT) learning space, where physical sensors communicate with virtual nodes. This enables students to quickly build and test IoT applications, visualize cyberphysical systems (CPS)(Wolf, 2009), and simulate Industry 4.0 scenarios using a graphical editor (see Figure1a).

The project has proven its feasibility, with ongoing improvements, including a web interface update for enhanced user experience. A prototype following the Web Content Accessibility Guidelines 2.1 has been developed to improve accessibility for users with disabilities.

Expanding the platform's capabilities includes offering various Linux distributions. While integrating Windows VMs is possible, the priority lies with open-source components. Refactoring software toward a container platform will boost efficiency and speed up release cycles using a cloud-native setup.

## 8 CONCLUSION

The virtual learning platform for university teaching, enabling the creation and use of complex IT structures for non-experts, has been successfully implemented using only free software. Students have evaluated the web interface and the underlying infrastructure setup with success. The next step is to use the platform productively across various courses, with performance measurements and evaluations planned to assess technical characteristics such as reliability and scalability. User feedback will also be gathered for improvement and incorporated into future platform versions.

By leveraging online solutions, virtualization in educational institutions fosters innovation, provides flexible learning environments, and prepares students for a technology-driven future. Our platform allows for the logical separation of project and student groups to support independent work. At the same time, virtual switches and routers can be used to connect these groups, enabling the creation of sophisticated networking scenarios for teaching and research.

## ACKNOWLEDGEMENTS

The SKILL/VL project described in this paper is part of the SKILL project network (Strategic Competence Platform - Innovative Learning and Teaching), which is funded by the Initiative Hochschullehre durch Digitalisierung stärken der Stiftung Innovation in der Hochschullehre, at Frankfurt University of Applied Sciences.

## REFERENCES

Abdul Rashid, N. b., bin Othman, M. Z., bin Johan, R., and bin Hj. Sidek, S. F. (2019). Cisco packet tracer simulation as effective pedagogy in computer networking course. *International Journal of Interactive Mobile Technologies (iJIM)*, 13(10):pp. 4–18.

Alam, A. and Mohanty, A. (2023"). Discerning the Application of Virtual Laboratory in Curriculum Transaction of Software Engineering Lab Course from the Lens of Critical Pedagogy. In *Sentiment Analysis and Deep Learning*, pages 53–68, Singapore. Springer Nature Singapore.

Algarni, S. A., Ikbal, M. R., Alroobaea, R., Ghiduk, A. S., and Nadeem, F. (2018). Performance evaluation of xen, kvm, and proxmox hypervisors. *International Journal of Open Source Software and Processes*, 9(2):39–54.

Athaya, H., Nadir, R. D. A., Indra Sensuse, D., Kautsarina, K., and Suryono, R. R. (2021). Moodle implementation for e-learning: A systematic review. In *Proceed-*

*ings of the 6th International Conference on Sustainable Information Engineering and Technology*, SIET '21, page 106–112, New York, NY, USA. Association for Computing Machinery.

Baun, C., Kunze, M., Nimis, J., and Tai, S. (2011). *Cloud Computing - Web-Based Dynamic IT Services*. Springer.

Beckmann, A. (2020). Digitalisierung in der hochschullehre: Erfahrungen mit dem mathedu digital-lehrkonzept und zur akzeptanz digitaler lehrelemente durch die studierenden. *MedienPädagogik: Zeitschrift für Theorie und Praxis der Medienbildung*, page 1–20.

Chen, L., Huang, W., Sui, A., Chen, D., and Sun, C. (2017). The online education platform using proxmox and novnc technology based on laravel framework. In *2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)*, pages 487–491.

Demchenko, Y., Gallenmüller, S., Fdida, S., Rausch, T., Andreou, P., and Saucez, D. (2023). Slices data management infrastructure for reproducible experimental research on digital technologies. In *2023 IEEE Globecom Workshops (GC Wkshps)*, pages 1–6.

Dietz, M. (2023). The internet of digital twins: Advances in hyperscaling virtual labs with hypervisor- and container-based virtualization. In Auer, M. E., Pachatz, W., and Rüütmann, T., editors, *Learning in the Age of Digital and Green Transition*, pages 574–586, Cham. Springer International Publishing.

Emiliano, R. and Antunes, M. (2015). Automatic network configuration in virtualized environment using gns3. In *2015 10th International Conference on Computer Science & Education (ICCSE)*, pages 25–30.

Erpenbeck, J. and Sauter, W. (2017). *Handbuch Kompetenzentwicklung im Netz Bausteine einer neuen Lernwelt*. Schäffer-Poeschel, Stuttgart, Germany, 1 edition.

Handke, J. (2015). *Handbuch Hochschullehre Digital*. Tectum Wissenschaftsverlag, Marburg an der Lahn, Germany, 1 edition.

Handke, J. and Schäfer, A. M. (2012). *E-Learning, E-Teaching und E-Assessment in der Hochschullehre*. Oldenbourg Wissenschaftsverlag Verlag, München.

Kappes, M. (2023). *Netzwerk- und Datensicherheit*. Springer Vieweg, Wiesbaden, Germany, 3 edition.

Lantz, B., Heller, B., and McKeown, N. (2010). A network in a laptop: rapid prototyping for software-defined networks. In *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, Hotnets-IX, New York, NY, USA. Association for Computing Machinery.

Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and Wright, C. (2014). Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks. RFC 7348.

Maler, E., Machulak, M., Richer, J., and Hardjono, T. (2019). Federated Authorization for User-Managed Access (UMA) 2.0. Internet-Draft draft-maler-oauth-umafedauthz-00, Internet Engineering Task Force. Work in Progress.

Meneses, S., Maya, E., and Vasquez, C. (2021). Network design defined by software on a hyper-converged infrastructure. case study: Northern technical university fica data center. In Botto-Tobar, M., Zamora, W., Larrea Plúa, J., Bazurto Roldan, J., and Santamaría Philco, A., editors, *Systems and Information Sciences*, pages 272–280, Cham. Springer International Publishing.

Richards, M. and Ford, N. (2020). *Fundamentals of software architecture*. O'Reilly Media, Sebastopol, CA.

Serrano, M., Isaris, N., and Schaffers, H., editors (2022). *Building the future internet through FIRE*. Taylor & Francis, London, England.

Steffen Auer, A.-F.-S. E. (2024). linuxmuster.net – documentation. accessed March 10, 2025, https://www.linuxmuster.net/en/documentation/.

Thakkar, M. (2020). *Building React Apps with server-Side Rendering: Use React, redux, and Next to build full server-Side Rendering applications*. APRESS, New York, NY.

Wolf, W. (2009). Cyber-physical systems. *Computer*, 42(3):88–89.