Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 3, July 2026
DOI : https://doi.org/10.33395/sinkron.v10i3.16296
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Tailscale Overlay Network for Secure Remote Management of Proxmox VE

Fiqih Akbari
Politeknik Negeri Sambas, Indonesia
fiqihakbari@poltesa.ac.id

Submitted: May 29, 2026 | **Accepted**: June 27, 2026 | **Published**: July 9, 2026

Abstract: Remote management of virtualized infrastructure introduces security risk when management services are exposed directly to the public internet. This risk is amplified when testbeds are intended to support sovereign edge computing workloads that require secure, isolated infrastructure. This study designs and evaluates a secure remote management architecture for a Proxmox VE node using a Tailscale overlay network and interface-specific firewall hardening, establishing a foundational infrastructure baseline for sovereign edge computing. The research follows Design Science Research supported by a network engineering evaluation procedure. The artefact was developed through problem identification, topology design, implementation, measurement, and evaluation. Data were collected from Tailscale status checks, Proxmox VE observation, ping latency testing, relay netcheck output, iptables verification, and external port scanning before and after firewall hardening. The Tailscale path achieved an average round-trip time of 0.434 milliseconds with zero packet loss, comparable to the public Internet Protocol path at 0.540 milliseconds with zero packet loss. Before hardening, public scanning detected management ports 22, 2222, and 8006. After applying interface-specific firewall rules, the external scan reported no open ports among the top 1000 ports, while private access to Proxmox VE through the Tailscale interface remained available. The proposed architecture demonstrates that overlay networking must be combined with firewall hardening to remove public management exposure without disrupting authorized remote administration. The result establishes a replicable foundational infrastructure baseline for sovereign edge computing, providing the first stage toward deployment of secure edge computing systems in resource-limited environments.

Keywords: firewall hardening; overlay network; Proxmox VE; sovereign edge computing; Tailscale

INTRODUCTION

Virtualized infrastructure is widely adopted by educational institutions, research laboratories, and small organizations to provide private cloud services and experimental computing environments. Proxmox Virtual Environment (Proxmox VE) is one of the platforms commonly adopted for this purpose because it integrates virtual machine, container, storage, and network management in a single administrative interface. Prior studies have demonstrated that Proxmox VE supports adequate performance for cloud-based workloads in resource-constrained settings (Kondoj et al., 2022). In edge computing research, this type of infrastructure is useful because several experimental nodes can be emulated on one physical host before being expanded into distributed deployment.

The administrative flexibility of Proxmox VE also creates a security problem. Management services such as SSH and the Proxmox VE web interface are often reachable through public IP addresses during initial deployment. If these services remain publicly visible, the infrastructure becomes more exposed to port scanning, brute-force attempts, misconfiguration abuse, and exploitation of vulnerable services. This condition is particularly undesirable for research testbeds that may later host sensitive workloads, private datasets, or sovereign AI experiments. As an initial step toward sovereign edge computing, establishing a secure remote management architecture constitutes a foundational prerequisite before any higher-layer AI or federated workloads can be deployed.

Overlay networks offer a practical mechanism to separate administrative access from direct public exposure. Tailscale, which is built on WireGuard, provides identity-based device membership, private addressing, NAT traversal, and encrypted peer-to-peer connectivity. However, deploying an overlay network alone does not

*Corresponding author

e-ISSN : 2541-2019
p-ISSN : 2541-044X

automatically close services that are already exposed on the public interface. The security benefit depends on whether public access is explicitly restricted while private overlay access is preserved.

Previous studies have discussed WireGuard performance, VPN comparison, zero trust architecture, and edge computing testbeds. However, a practical gap remains in small-scale private cloud environments: there is limited empirical documentation showing how a Tailscale overlay can be combined with interface-specific firewall hardening to secure Proxmox VE remote management while maintaining operational access. This gap is particularly relevant for sovereign edge computing research, where the infrastructure layer must be demonstrably secure before any AI, federated learning, or sensitive data workloads are introduced. Many implementation reports stop at connectivity success and do not verify whether public management exposure has actually been removed.

This study addresses that gap by designing and evaluating a secure remote management architecture for Proxmox VE using Tailscale overlay networking and iptables-based firewall hardening. The research question is: can a Tailscale-based remote management architecture remove public exposure of Proxmox VE management services while preserving private administrative access? The contribution of this paper is a tested architecture, a chronological implementation method, before-and-after security evidence from external port scanning, and a baseline design for edge computing testbed development in resource-limited environments.

LITERATURE REVIEW

Private cloud computing provides compute, network, and storage resources under controlled access for a particular organization. Compared with public cloud services, private cloud deployment offers stronger control over data location, network configuration, and administrative access policies. The NIST definition of cloud computing describes essential characteristics such as resource pooling, broad network access, and measured service, which remain relevant when a private cloud is implemented at a smaller organizational scale (Mell & Grance, 2011).

Virtualization enables a single physical server to host multiple virtual machines or containers. This capability is important for edge computing research because multiple logical nodes can be deployed without requiring many physical devices. Edge computing moves computation closer to the data source to reduce latency, optimize bandwidth, and support context-aware processing (Satyanarayanan, 2017; Shi et al., 2016). For research environments with limited resources, virtualization can serve as an initial testbed layer before physical edge nodes are deployed.

Remote management security is closely related to the principle of reducing public exposure. Zero trust architecture emphasizes that access should not be implicitly trusted based only on network location; instead, access should be continuously controlled through identity, policy, and verification mechanisms (Rose et al., 2020). In the context of Proxmox VE, management interfaces and SSH access should therefore be reachable only through authorized administrative paths.

WireGuard has been widely discussed as a lightweight VPN protocol. Donenfeld (2017) introduced WireGuard as a modern kernel network tunnel with a simpler cryptographic design. Performance comparisons also report that WireGuard can provide lower latency and better throughput than several traditional VPN approaches in many scenarios (Mackey et al., 2020; Anyam et al., 2025). Tailscale builds on WireGuard and simplifies overlay network deployment by adding device identity, coordination, and NAT traversal.

Edge computing testbeds require secure and manageable infrastructure. Open-source testbeds have been used to evaluate edge and 5G security scenarios (Pepito & Dutta, 2021), while federated and serverless edge-cloud architectures show the need for distributed, manageable, and secure experimental environments (Yang et al., 2019; Bonawitz et al., 2019; Kjorveziroski et al., 2025). Based on these studies, the present work positions Proxmox VE and Tailscale as a practical baseline for small-scale secure edge testbed development.

METHOD

This study uses Design Science Research as the methodological foundation because the main output is an implemented technical artefact rather than a survey model or predictive statistical model. Design Science Research is appropriate for information systems and engineering studies that design, build, and evaluate artefacts intended to solve practical problems. In this study, the artefact is a secure remote management architecture that combines Proxmox VE, Tailscale overlay networking, and iptables firewall hardening.

The research procedure consists of five stages: problem identification, artefact design, implementation, measurement, and evaluation. Problem identification defines the operational need: Proxmox VE must remain remotely manageable while public management exposure must be minimized. Artefact design translates this requirement into a logical architecture consisting of an administrator client, a VPS control node, a Tailscale tailnet, a Proxmox VE host, and virtual workloads. Implementation deploys Tailscale connectivity and firewall rules on the operational environment. Measurement collects connectivity, status, and port exposure data. Evaluation compares the system before and after hardening.

*Corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 3, July 2026
DOI : https://doi.org/10.33395/sinkron.v10i3.16296
e-ISSN : 2541-2019
p-ISSN : 2541-044X

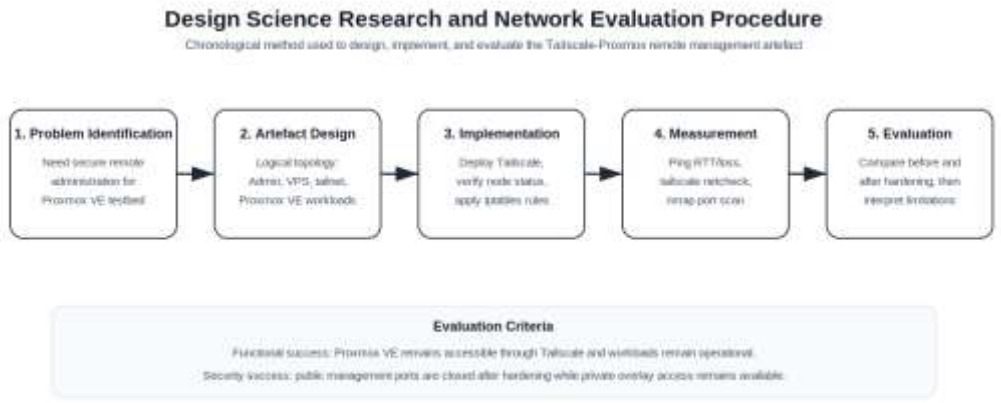

Figure 1. Research method workflow based on Design Science Research and network evaluation

Figure 1 maps the study to an artefact-based evaluation with defined stages, measured parameters, and explicit success criteria.

The testbed consists of one Proxmox VE host named `pve-new`, one administrator client, one VPS control node, and four virtual workloads running on the Proxmox VE host. The Proxmox VE host uses the Tailscale private address `100.93.33.43` and also has a public IP address used during the exposure comparison. The virtual workloads represent services typically required in edge-oriented environments: a reverse proxy and web application security container, a deployment platform container, a personal AI agent container, and a virtual router.

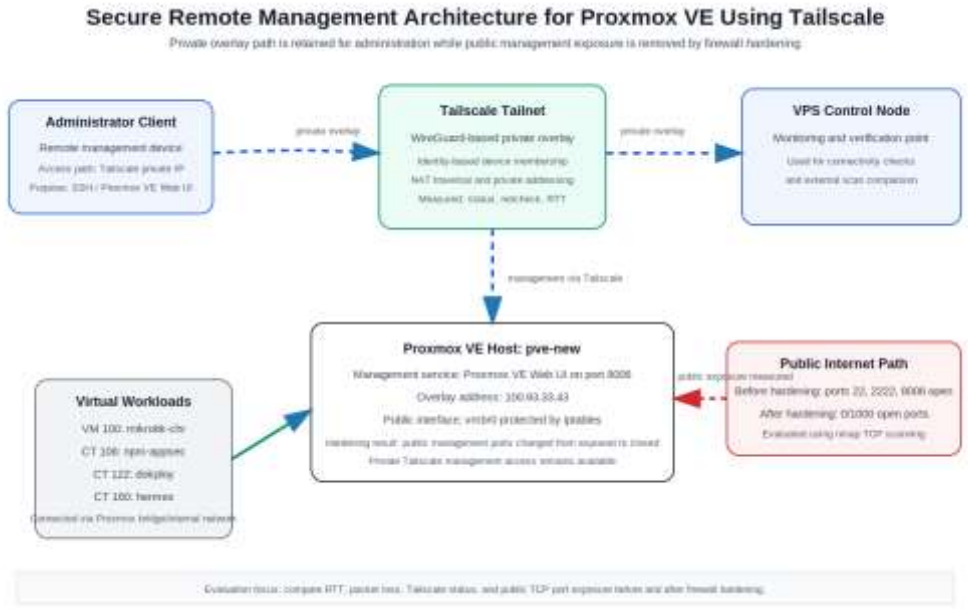

Figure 2. Secure remote management architecture for Proxmox VE using Tailscale

Figure 2 shows the proposed architecture. The key design principle is the separation between the private administrative path and the public internet path. Administrative access is directed through the Tailscale overlay, while the public interface is hardened using iptables rules. This design allows Proxmox VE management to remain available to authorized devices without keeping management ports openly visible to external scanners.

Data were collected using direct system observation and network measurement tools. Functional evidence was obtained from the Proxmox VE dashboard, Tailscale machine status, `tailscale status`, and `tailscale netcheck`. Connectivity data were collected using ICMP ping with 20 packets for each tested path. Security exposure data were collected using `nmap -sT --top-ports 1000 -Pn <public-ip>` executed from the VPS control node. Firewall verification was performed using `iptables -L INPUT -n --line-numbers` to confirm that public-interface blocking and private-interface allowance were applied as intended.

*Corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 3, July 2026
DOI : https://doi.org/10.33395/sinkron.v10i3.16296
e-ISSN : 2541-2019
p-ISSN : 2541-044X

The evaluation used two scenarios. Scenario 1 represents the initial condition after Tailscale was deployed but before public firewall hardening was applied. Scenario 2 represents the hardened condition after iptables rules blocked management ports on the public interface while retaining private access through the Tailscale interface. The architecture is considered successful if the Proxmox VE management interface remains accessible through Tailscale, the node remains registered in the tailnet, virtual workloads remain operational, and external scanning shows that public management ports are no longer exposed.

RESULT

The proposed architecture was implemented on an operational Proxmox VE environment. The Proxmox VE host `pve-new` ran four workloads during the evaluation. Table 1 presents the workloads and their functions in the testbed.

Table 1. Virtual workloads running on the Proxmox VE host

VMID	Name	Type	vCPU	RAM	Function
:---:---	---	:---:	:---:	---	
100	mikrotik-chr	QEMU VM	1	512 MB	Network routing experiments
106	npm-appsec	LXC container	2	2 GB	Reverse proxy and web application security
122	dokploy	LXC container	3	4 GB	Docker-based deployment platform
160	hermes	LXC container	2	2 GB	Personal AI agent workspace

The workload composition shows that the host was not only configured for a single service, but was used as a small edge-oriented private cloud node. The virtual router, deployment container, reverse proxy container, and AI agent container represent different functional roles that can support later edge computing experiments. The resource footprint is deliberately modest: the four workloads collectively consume 8 vCPUs and 6.5 GB RAM, consistent with the resource-limited environments referenced in the abstract and conclusion. Redacted dashboard screenshots from the operational Proxmox VE and Tailscale environments serve as supporting evidence of a real testbed. Account names, internal hostnames, tailnet addresses, public addresses, and activity timestamps are masked to reduce publication risk.

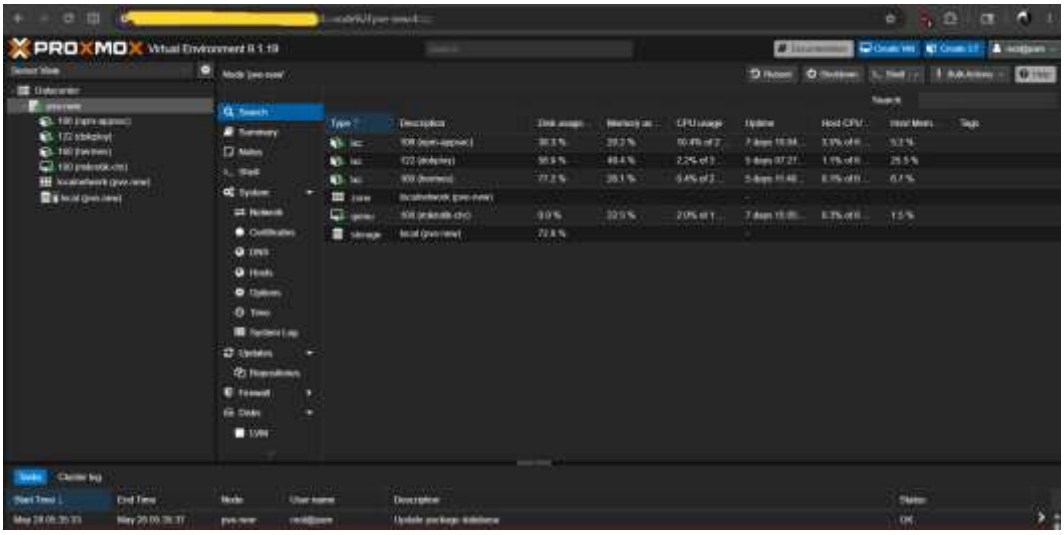

Figure 3. Redacted Proxmox VE dashboard showing operational virtualized workloads

Figure 3 shows the Proxmox VE dashboard used in the experiment. The figure confirms that the architecture was implemented on an active virtualization environment with running workloads and visible resource utilization

*Corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 3, July 2026
DOI : https://doi.org/10.33395/sinkron.v10i3.16296
e-ISSN : 2541-2019
p-ISSN : 2541-044X

metrics. Sensitive identifiers are intentionally redacted, but the dashboard structure and resource metrics remain visible as physical implementation evidence.

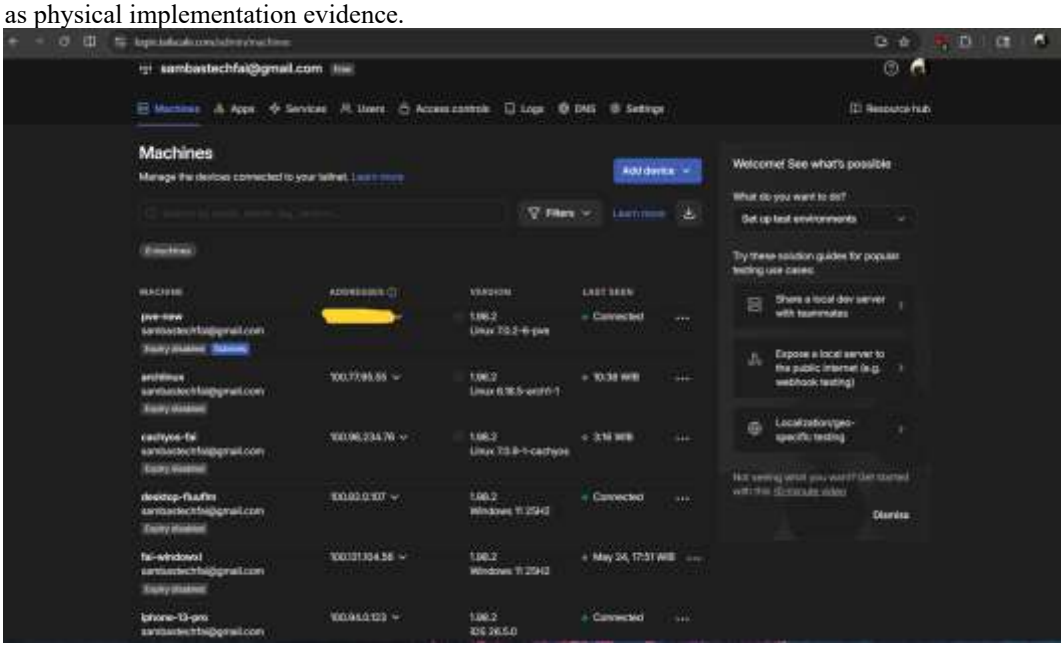

Figure 4. Redacted Tailscale machines dashboard showing tailnet-based device management

Figure 4 shows the Tailscale machines dashboard used to verify that the remote management environment was connected through a tailnet. The figure supports the claim that overlay-based access was not only conceptual, but deployed in a real administrative environment. Machine names, account information, tailnet addresses, and activity timestamps are redacted for security and privacy.

Connectivity testing compared the Tailscale private path and the public IP path. The test measured average RTT, minimum RTT, maximum RTT, packet loss, and packet count. Table 2 presents the result.

Table 2. Connectivity performance comparison between Tailscale and public IP paths

Parameter	Tailscale private path	Public IP path
Destination	100.93.33.43	45.127.35.124
Average RTT	0.434 ms	0.540 ms
Minimum RTT	0.253 ms	0.304 ms
Maximum RTT	0.774 ms	2.267 ms
Packet loss	0%	0%
Packets sent	20	20
Packets received	20	20

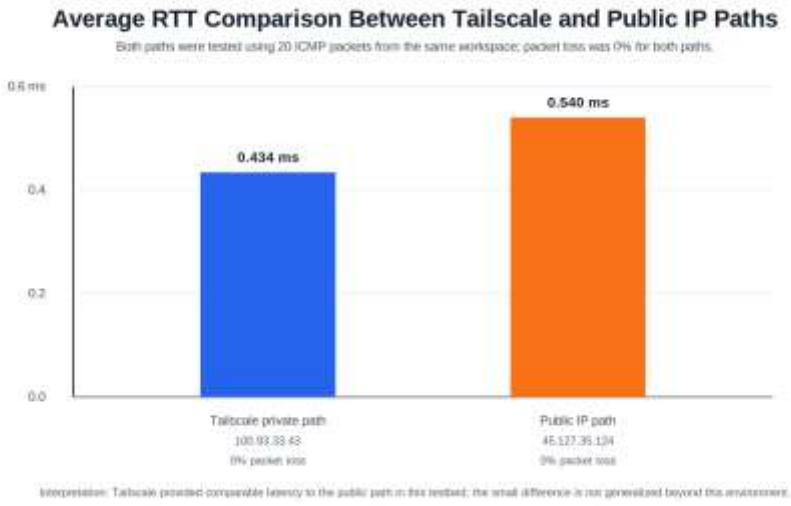

Figure 5. Average RTT comparison between Tailscale and public IP paths

*Corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 3, July 2026
DOI : https://doi.org/10.33395/sinkron.v10i3.16296
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Figure 5 visualizes the same measurement as Table 2. The revised figure only includes the two paths discussed in the evaluation, so the visual evidence is aligned with the narrative. Both paths achieved 0% packet loss, and the Tailscale path produced slightly lower average RTT in this environment.
Tailscale status checking confirmed that the host was reachable through the overlay network. The `tailscale netcheck` result confirmed UDP connectivity over IPv4 and identified Singapore as the nearest DERP relay with 700 µs latency. Table 3 summarizes the netcheck output relevant to the evaluation.

Table 3. Tailscale connectivity and DERP netcheck result

Parameter	Value
UDP connectivity	Available
IPv4 connectivity	Available
IPv6 connectivity	Not available
Nearest DERP relay	Singapore
Nearest DERP latency	700 µs
Additional DERP latency	Hong Kong 33.2 ms, Bangalore 41.6 ms, Tokyo 86.1 ms, Sydney 114.5 ms

The security evaluation compared public port exposure before and after firewall hardening. Before hardening, external scanning found that public management ports were still visible. After hardening, iptables rules were applied to block management-related ports on the public bridge interface while preserving the required management access through the Tailscale interface. Table 4 presents the before-and-after result.

Table 4. TCP management port exposure before and after firewall hardening

Access path	Before hardening	After hardening	Interpretation
Public IP path	Ports 22, 2222, and 8006 open	0 open ports among top 1000 TCP ports	Public management exposure removed
Tailscale private path	Private access available	Port 8006 available through tailscale0	Authorized remote management preserved

Figure 6. TCP management port exposure before and after firewall hardening

Figure 6 presents the main security result of the study. The figure directly reflects the measured before-and-after condition: public management services were initially visible, but after firewall hardening they were no longer detected by external scanning. At the same time, private Proxmox VE access through Tailscale remained available.

Firewall verification showed that TCP port 22 and TCP port 8006 were dropped on the public bridge interface, while access to port 8006 through the Tailscale interface was retained. This confirms that the result was not achieved by disabling Proxmox VE management access entirely, but by separating the permitted private management path from the blocked public path.

The firewall hardening rules applied to the Proxmox VE host are listed below:

*Corresponding author

e-ISSN : 2541-2019
p-ISSN : 2541-044X

```
# Check current INPUT chain before changes
iptables -L INPUT -n --line-numbers
# Drop TCP ports 22 and 8006 on the public bridge interface
iptables -A INPUT -i vmbr0 -p tcp --dport 22 -j DROP
iptables -A INPUT -i vmbr0 -p tcp --dport 8006 -j DROP
# Accept TCP port 8006 specifically on the Tailscale interface
iptables -A INPUT -i tailscale0 -p tcp --dport 8006 -j ACCEPT
# Verify the rules are in place
iptables -L INPUT -n --line-numbers
```

The rules are interface-specific: they drop management traffic on the public bridge (vmbr0) while selectively accepting the same port on the overlay interface (tailscale0). This design ensures that the Proxmox VE management service remains running and reachable through Tailscale, but invisible to external scanners on the public internet.

DISCUSSIONS

The Tailscale path achieved a lower average round-trip time (0.434 ms) compared to the public IP path (0.540 ms). Since both paths traverse the same physical network infrastructure, the difference requires a specific explanation rather than a general statement about overlay efficiency. The most likely cause is jitter: the public IP path recorded a maximum round-trip time of 2.267 ms against 0.774 ms for the Tailscale path. With only 20 ICMP samples, a small number of high-latency packets on the public path is sufficient to raise the average by approximately 0.1 ms. WireGuard's kernel-level UDP encapsulation also avoids some per-packet overhead present on the public IP path, and the DERP netcheck confirmed that a direct peer-to-peer UDP connection was established, not a relayed path. This is consistent with prior studies reporting WireGuard's efficient performance characteristics (Mackey et al., 2020; Anyam et al., 2025; Ruhault et al., 2024).

The more important finding is the change in public management exposure. Deploying an overlay network does not automatically remove public service exposure: ports 22, 2222, and 8006 remained visible on the public IP path even after the overlay was available, showing a common operational gap where connectivity tools are deployed but the public interface remains insecure. After iptables hardening, external scanning reported zero open ports among the top 1000 TCP ports on the public IP path, while private access through the Tailscale interface remained fully available. This result supports the argument that overlay networking and firewall hardening should be treated as complementary controls: Tailscale provides private identity-based access while firewall rules enforce exposure reduction on the public interface, consistent with zero trust principles (Rose et al., 2020). Interface-specific hardening is a lightweight security control compatible with containerized workloads on resource-limited hosts (Yang et al., 2022).

Compared with broader edge computing and security testbed studies, this research contributes a smaller but operationally concrete baseline. Prior works discuss edge computing concepts, federated learning systems, or open-source security testbeds at a broader architectural level (Satyanarayanan, 2017; Pepito & Dutta, 2021; Kjorveziroski et al., 2025). Network-aware container scheduling has also been explored for edge environments where resource efficiency is critical (Qiao et al., 2025). This study focuses on the infrastructure layer that must exist before those workloads can be tested safely: a manageable virtualization host with reduced public attack surface. The study used one Proxmox VE host and one VPS control node, representing a baseline implementation rather than a multi-site deployment. The evaluation covered ICMP latency and TCP port exposure but did not include throughput testing, jitter analysis under load, or formal penetration testing.

CONCLUSION

This study designed and evaluated a secure remote management architecture for Proxmox VE using a Tailscale overlay network and interface-specific firewall hardening. The result answers the research question by showing that public exposure of management services can be removed while private administrative access remains available through Tailscale. This architecture constitutes the foundational infrastructure stage for sovereign edge computing, establishing the necessary secure baseline before higher-layer AI or federated workloads can be deployed.

The connectivity test showed that the Tailscale private path achieved 0.434 ms average RTT with 0% packet loss, comparable to the public IP path at 0.540 ms with 0% packet loss. The security test showed a clearer contribution: before hardening, the public IP exposed ports 22, 2222, and 8006; after hardening, an external scan of the top 1000 TCP ports found no open ports on the public IP path, while Proxmox VE remained reachable through the Tailscale private interface.

*Corresponding author

e-ISSN : 2541-2019
p-ISSN : 2541-044X

The main contribution is a replicable foundational architecture for secure remote management of small-scale private cloud and edge computing testbeds. The study demonstrates that Tailscale should not be treated as a standalone security solution; it must be combined with firewall rules that close unnecessary public exposure. As the first stage of a sovereign edge computing pipeline, this baseline directly supports subsequent research on federated learning workloads, secure edge services, and resilient digital systems in resource-limited or border-region contexts. Future work should evaluate multi-node Proxmox clusters, iperf3 throughput testing, failover behavior under network impairment, and deployment of federated learning workloads on the secured edge testbed.

REFERENCES

Anyam, J., Singh, R. R., Larijani, H., & Philip, A. (2025). Empirical performance analysis of WireGuard vs. OpenVPN in cloud and virtualised environments under simulated network conditions. *Computers*, *14*(8), 326. https://doi.org/10.3390/computers14080326

Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., Kiddon, C., Konečný, J., Mazzocchi, S., McMahan, H. B., Van Overveldt, T., Petrou, D., Ramage, D., & Roselander, J. (2019). Towards federated learning at scale: System design. *Proceedings of Machine Learning and Systems*, *1*, 374–388.

Deutschmann, J., Jahandar, S., Hielscher, K.-S., & German, R. (2023). Internet via satellite: GEO vs. LEO, OpenVPN vs. WireGuard, and CUBIC vs. BBR. *Proceedings of the 1st ACM MobiCom Workshop on Satellite Networking and Computing*. https://doi.org/10.1145/3614454.3622998

Donenfeld, J. A. (2017). WireGuard: Next generation kernel network tunnel. *Proceedings of the Network and Distributed System Security Symposium (NDSS)*.

Kjorveziroski, V., Filiposka, S., Kocarev, L., & Prodan, R. (2025). Federated architecture for serverless platforms aimed at transparent execution in the edge-cloud continuum. *International Journal of Cloud Computing*, *14*(2), 145–166. https://doi.org/10.1504/IJCC.2025.145664

Kondoj, M., Langi, H., & Putung, Y. (2022). Performance analysis of cloud computing based e-commerce server using Proxmox virtual environment. *Proceedings of the 5th International Conference on Applied Science and Technology on Engineering Science (iCAST-ES)*. https://doi.org/10.5220/0011876000003575

Mackey, S., Mihov, I., Nosenko, A., Vega, F., & Cheng, Y. (2020). A performance comparison of WireGuard and OpenVPN. *Proceedings of the Tenth ACM Conference on Data and Application Security and Privacy*, 162–164. https://doi.org/10.1145/3374664.3379532

Mell, P., & Grance, T. (2011). The NIST definition of cloud computing. *National Institute of Standards and Technology, Special Publication 800-145*.

Pepito, R., & Dutta, A. (2021). Open source 5G security testbed for edge computing. *2021 IEEE 4th 5G World Forum (5GWF)*. https://doi.org/10.1109/5GWF52925.2021.00075

Proxmox Server Solutions GmbH. (2025). Proxmox Virtual Environment documentation. https://pve.proxmox.com/pve-docs/

Qiao, Y., Xiong, J., & Zhao, Y. (2025). Network-aware container scheduling in edge computing. *Cluster Computing*, *28*(3), 477–492. https://doi.org/10.1007/s10586-024-04733-8

Rose, S., Borchert, O., Mitchell, S., & Connelly, S. (2020). Zero trust architecture. *National Institute of Standards and Technology, Special Publication 800-207*. https://doi.org/10.6028/NIST.SP.800-207

Ruhault, S., Lafourcade, P., & Mahmoud, D. (2024). A unified symbolic analysis of WireGuard. *Proceedings of the Network and Distributed System Security Symposium (NDSS)*. https://doi.org/10.14722/ndss.2024.24364

Satyanarayanan, M. (2017). The emergence of edge computing. *Computer*, *50*(1), 30–39. https://doi.org/10.1109/MC.2017.9

Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, *3*(5), 637–646. https://doi.org/10.1109/JIOT.2016.2579198

Tailscale Inc. (2025). Tailscale documentation. https://tailscale.com/kb/

Varghese, B., Wang, N., Barbhuiya, S., Kilpatrick, P., & Nikolopoulos, D. S. (2016). Challenges and opportunities in edge computing. *Proceedings of the IEEE International Conference on Smart Cloud*, 20–26.

Yang, N., Chen, C., & Yuan, T. (2022). Security hardening solution for Docker container. *2022 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, 44–49. https://doi.org/10.1109/cyberc55534.2022.00049

Yang, Q., Liu, Y., Chen, T., & Tong, Y. (2019). Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology*, *10*(2), Article 12. https://doi.org/10.1145/3298981

*Corresponding author