There's No Place Like Home

Design and Implementation of a Segmented, Privacy-Centric Homelab Using Open-Source Platforms

Final Report

Student Name: Wolfgang Helnwein

Programme: SETU Higher Diploma

in Computer Science

Table of Contents

Table of Figures

Table of Tables

Declaration of Authenticity

I declare that the work which follows is my own, and that any quotations from any sources (e.g. books, journals, the internet) are clearly identified as such by the use of 'single quotation marks', for shorter excerpt and identified italics for longer quotations. All quotations and paraphrases are accompanied by (date, author) in the text and a fuller citation is the bibliography. I have not submitted the work represented in this report in any other course of study leading to an academic award.

Student..... Wolfgang Helmrein ..... Date 29/03/2026

Work Place Mentor..... Date .....

1. Introduction

1.1 Project Rationale

Home users of internet-connected systems increasingly face challenges around personal data protection and digital security. Modern home networks host growing numbers of Internet-of-Things (IoT) devices of varying trust levels alongside personal devices, while consumer-grade routers often provide limited support for effective segmentation or isolation. Beyond the home, reliance on Software-as-a-Service (SaaS) and cloud platforms introduces users to risks such as data breaches and privacy violations. The underlying technology stacks are also increasing in complexity and are often abstracted away from end users.

This project is motivated by an interest in exploring how a home network can be better designed to address these challenges, while also providing a practical way to gain a clearer understanding of how such systems work in practice and to develop skills in networking, cybersecurity, virtualisation, and self-hosting using open-source tools.

1.2 Project Aim

The aim of the project is to design and implement a secure, scalable, and logically segmented home network that mitigates risks posed by IoT devices and reduces reliance on cloud-hosted platforms, while also serving as a platform for further exploration of open-source technologies, networking, and cybersecurity.

1.3 Project Objectives

Breaking down the objectives necessary to achieve the project aim, we arrive at the following phases and items:

- **Decide on Network Backbone**: Assess the suitability of wired versus wireless networking, considering speed, reliability, ease of installation, and security.
- **Evaluate Hardware Needs:** Identify hardware requirements for the project and select the router/firewall, switch, access points and servers while considering affordability, ease of maintenance, flexibility, interoperability and scalability.
- **Explore Open-Source Software Alternatives:** Evaluate the project's software needs, from router/firewall platforms to self-hostable applications, based on usability, ease of deployment, and feature sets.
- **Architect the Network**: Plan the physical and logical network structure to minimise disruption to the home environment and adhere to security best practice.

- **Build out Network Backbone:** Put in place the physical infrastructure required to support the network, considering constraints such as noise, heat, and power availability.
- **Deploy Hardware:** Set up the router/firewall, switch, access point, camera, network video recorder, and server, then verify correct connectivity.
- **Software Deployment and Configuration:** Install and configure the router/firewall operating system, ensuring a stable, robust, and maintainable configuration.
- **Segment the Network:** Split the network into logical VLANs based on device trust levels and communication requirements.
- **Deploy Services:** Deploy open-source services and applications on the home server as alternatives to cloud-hosted platforms.

1.4 Scope and Constraints

This project's scope is primarily to serve as a proof-of-concept of an alternative model of home networking and internet usage. It covers a single household's end-user devices and assumes a technically inclined individual is available to implement and maintain the system. It is not intended to be easily replicated or applied at scale.

Where new devices such as cameras and servers are deployed, only a single instance of each is included as a foundation for future expansion.

The need to research, evaluate, and source a significant quantity of equipment was expected to introduce time constraints related to deliveries and potential shipping issues, as well as budget limitations.

2. Research Context

2.1 Digital Sovereignty, Autonomy and Self-Hosting

A key context for the project is growing concern at an EU level about over-reliance on digital platforms outside the bloc's direct control. This has prompted examination of whether open-source technologies can support greater technological sovereignty within the Union (Directorate-General for Communications Networks, Content and Technology, 2026). In parallel, there have been calls for a 'European cloud', although providers such as OVH (France) and Hetzner Cloud (Germany) remain limited in scale and capability.

While such regulatory and market responses aim to mitigate these concerns at an institutional level, they continue to assume reliance on third-party cloud platforms. This raises a more fundamental question: whether cloud-based solutions are necessary for many use cases,

particularly at a household scale where users may be willing to trade convenience for greater autonomy and control.

In this context, self-hosting has emerged as a viable alternative to mainstream SaaS platforms. Containerisation supports this approach by simplifying application deployment and dependency management, while community-driven resources such as AlternativeTo provide structured comparisons based on licensing, hosting model, and jurisdiction. Community forums further support knowledge sharing around deployment, maintenance, and long-term sustainability. However, this model is most applicable to technically inclined users or households where at least one individual is willing to assume an ongoing administrative role.

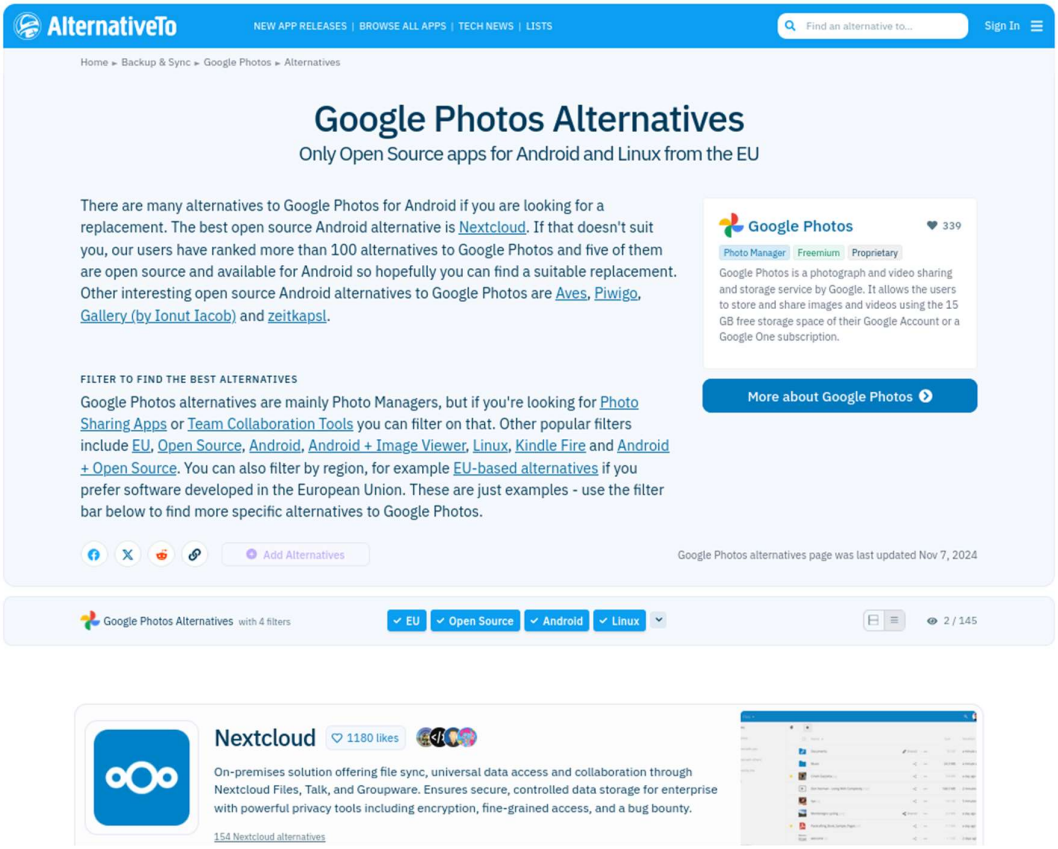

Figure 1: Example screenshot from AlternativeTo.net

2.2 Homelabbing

This project is informed by the “homelabbing” movement, which involves setting up a personal environment for experimenting with information technologies to develop practical skills in networking, systems administration, and familiarity with enterprise-level tools (r/homelab Community Members, 2025).

2.3 Sustainability and Repurposed Hardware

Sustainability also plays a role in the project, with a preference for repurposing used computers and networking equipment where possible rather than purchasing new devices.

2.4 Cybersecurity in Home Networks

The final strand of the project is cybersecurity, particularly in relation to network segmentation via Virtual Local Networks (VLANs), allowing untrusted IoT devices to be isolated from trusted systems such as workstations and servers.

3. Evaluation and Analysis

3.1 Desktop Operating System

EndeavourOS, an Arch-based distribution, was initially deployed on household desktops due to its flexibility and access to a wide range of software packages. However, its rolling-release model and reliance on user-managed security practices were deemed unsuitable for a shared home environment, particularly in light of recent supply-chain and ecosystem-level security incidents, including the xz/liblzma backdoor and malware discovered in Arch User Repository packages (Freund, 2024; Michaud, 2025).

Replacement distributions were therefore evaluated with greater emphasis on conservative security defaults, stability, and ease of recovery. Fedora Kinoite emerged as a strong candidate, offering SELinux enabled by default and an immutable, image-based design that reduces the risk of system modification and supports reliable rollback via atomic updates (Fedora Project, 2026).

These characteristics aligned more closely with the project's security and usability requirements for a shared home environment than a traditional rolling-release distribution (Fedora Project, 2025; Fedora Project, 2026). For these reasons, Fedora Kinoite was selected to replace EndeavourOS on all household desktops in anticipation of the project.

Table 1: Desktop Linux Comparison

Distribution	Technological Base	Availability of Applications	Ease of Use	Stability	Advanced Security Defaults
EndeavourOS	Traditional, package based	Excellent, both official and unofficial (Arch User Repository [AUR])	Complexity increases with customisation	Can be unstable, "rolling release"	No
Fedora Kinoite	"Atomic"/image based	Moderate	Excellent by default	Extremely stable, out of the box rollback capability	Yes, SELinux enabled by default

3.2 Virtualisation Layer

Proxmox Virtual Environment (Proxmox VE), a Debian-based open-source virtualisation platform, was selected as the project's virtualisation layer due to its integrated support for both

Kernel-based Virtual Machines (KVM) and Linux Containers (LXC) through a single management interface (Proxmox, 2019b; Proxmox, 2019c). This architecture was intended to support strong isolation between infrastructure and application workloads, with services hosted within virtual machines rather than directly on the host. As part of this design, a minimal openSUSE MicroOS instance was identified as a suitable candidate for the dedicated container host, enabling clear separation between the virtualisation layer and containerised application services. Based in Austria and Germany respectively, these projects also align with the project's broader context of EU digital sovereignty (openSUSE contributors, 2025).

3.3 Router/Firewall Platform

Several open-source firewall platforms were evaluated for the routing and firewall component of the network, with pfSense and OPNsense emerging as the primary contenders. Both platforms provide a feature set suitable for the project's requirements, including VLAN support, firewall rule management, Intrusion Detection and Prevention Systems (IDS & IPS), and virtual private networking.

While pfSense has a longer history of adoption in home and enterprise environments (Wikipedia Contributors, 2021), its increasing commercialisation under Netgate raised concerns regarding long-term transparency and alignment with open-source principles. In contrast, OPNsense maintains a community-driven development model with more frequent updates and a stronger emphasis on openness, aligning closely with the project's focus on digital autonomy (Conway, 2025; OPNsense, 2025). For these reasons, OPNsense was selected as the operating system for the custom router.

Table 2: Router/Firewall Operating System Comparison

Operating System	Base	Created	Frequent Patches	Active Community	Consistent Approach to Open Source
pfSense	BSD, m0n0wall	2004	Less so	Yes	No
OPNsense	BSD, pfSense	2014	Yes	Yes	Yes

3.4 Hardware Option Analysis

3.4.1 Infrastructure

The network backbone was designed around wired Ethernet, which offers more consistent performance, lower latency, and improved security than wireless alternatives when cabling is

contained within a controlled environment (Parrish, 2021). As suitable CAT6 cabling was already available at no additional cost, this approach was considered appropriate where feasible.

3.4.2 Router

Hardware capable of running OPNsense was evaluated across first-party appliances, third-party firewall devices, and repurposed PCs, using criteria such as system specifications and running costs, setup complexity, and upgradability. Hardware sizing guidance and compatibility recommendations were consulted to anchor a baseline specification suitable for IDS/IPS and other advanced OPNsense features (Deciso B.V., 2016a; HomeTechHacker, 2024)

Official OPNsense appliances from Deciso were considered due to first-party support and EU origin (Deciso B.V., 2024), but many models exceeded the project budget and offered limited user upgradability (Deciso B.V., 2022). Third-party appliances such as Protectli were also evaluated, offering coreboot support and greater upgrade headroom at comparable cost (Protectli, 2025).

A repurposed PC was selected to maximise flexibility and align with the project's sustainability goals, accepting the trade-offs of higher power use and noise (programming.dev Community Members, 2026). The Lenovo ThinkCentre M720q was selected due to its strong community adoption and practical expandability via a PCIe riser, allowing NIC expansion while remaining widely available at low cost (muffn_, 2024, Parallax, 2021).

3.4.3 Camera & Network Video Recorder

A wired Power-over-Ethernet (PoE) camera was selected over wireless alternatives due to its simplicity and reliability, allowing both power and data to be delivered over a single cable. Reolink devices were selected for their ability to operate fully offline without reliance on cloud services, as well as their support for community-maintained integrations with home automation platforms such as Home Assistant. A compatible network video recorder (NVR) was used to provide local video storage and ensure interoperability within the same ecosystem (r/reolink Community Members, 2024).

3.4.4 Switch & Access Point

As the Lenovo M720q provides limited built-in Ethernet connectivity, a managed PoE switch was required to support VLAN configuration, centralised management, and future expansion. Community discussion frequently highlighted MikroTik and TP-Link's Omada line as affordable, managed networking ecosystems; however, while MikroTik switches are highly regarded, feedback indicated weaker wireless offerings and a more complex multi-AP management experience (Cameron Gray, 2025; r/mikrotik Community Members, 2023). Based on this ecosystem consideration, a TP-Link Omada managed switch was selected alongside a

compatible TP-Link access point supporting PoE and VLAN features, allowing both wired switching and wireless infrastructure to be managed within the same platform.

4. System Architecture and Design

4.1 High-Level Network Architecture

At a high level, the system was designed around a custom router positioned downstream of the ISP-provided equipment, acting as the primary routing, firewall, and Dynamic Host Configuration Protocol (DHCP) component for the internal network. The router connects to a managed switch that serves as the central distribution point for downstream traffic, providing wired connectivity to core infrastructure such as servers and the network video recorder (NVR). Wireless connectivity is provided by an access point connected to the switch, with end-user devices connecting either via Wi-Fi or Ethernet.

4.2 Network Segmentation and VLANs

At the project's commencement, the network followed a typical flat home architecture, with the ISP router acting as the central point of connectivity. While common, this model relies primarily on separating internal and external networks (LAN/WAN) and provides limited protection against internal threats, meaning a single compromised device can enable lateral movement. Network segmentation using Virtual Local Networks (VLANs) was therefore identified as a key mitigation, allowing devices to be isolated based on trust level and function.

For example, migrating the OptiPlex server from the default LAN to a dedicated server subnet was intended to reduce the potential impact of vulnerabilities in self-hosted services. Similarly, given the weak security defaults often associated with consumer security cameras, as demonstrated by exposed feeds on platforms such as Insecam (Insecam, 2023) and several high-profile breaches (Yoon, 2025), cameras and the network video recorder (NVR) were planned to be placed on a dedicated VLAN with no internet access and limited communication with trusted devices.

The next page provides an overview of the network's initial architecture.

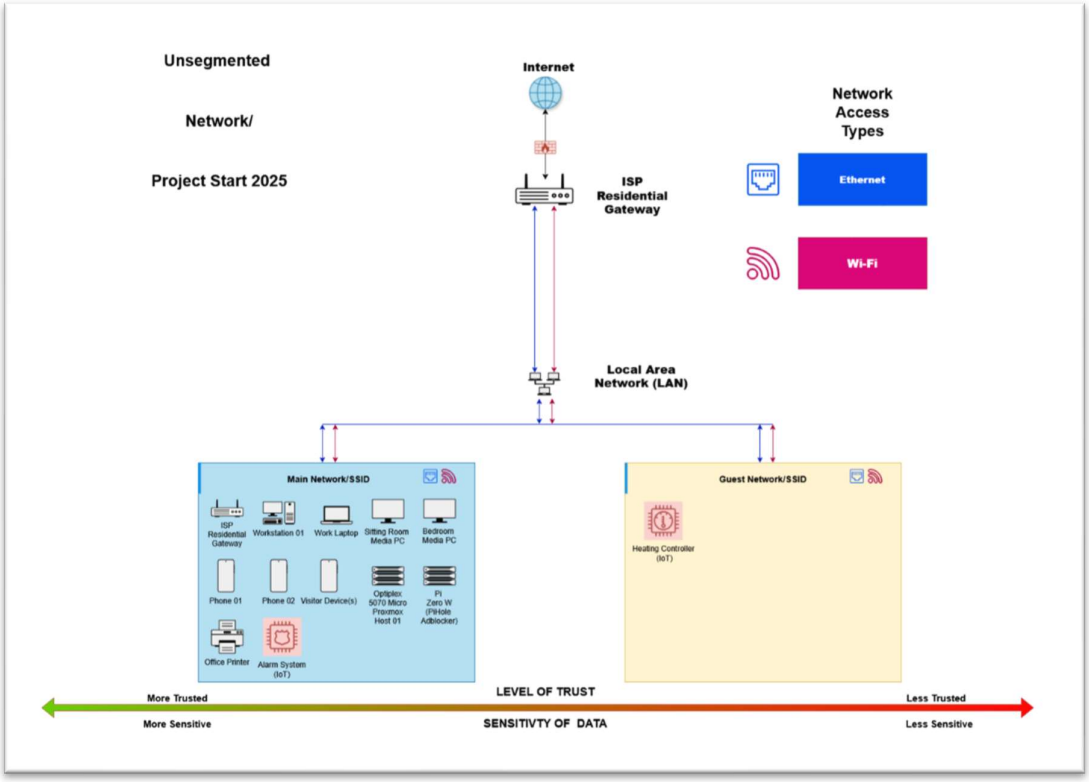

Figure 2: ISP Router Network (Before)

As shown in the diagram, the network featured limited and inconsistently applied segmentation, with the IoT heating controller isolated on the guest network while the IoT alarm system remained on the primary network. Servers, workstations, and media PCs also shared the same wired and Wi-Fi segments without segregation. Although the servers were not exposed externally, separating them into a dedicated VLAN would still be prudent, and personal and work devices likewise shared a common LAN.

After a review of common homelab network architectures (Quik Tech Solutions L.L.C, 2025), it was decided that 7 VLANs should be spun up, with the following roles:

1. **Management**: Network infrastructure (router, switch, access point).
2. **Work**: Work devices.
3. **Personal**: Household devices.
4. **Server**: Self-hosted servers and applications.
5. **Guest**: Guest devices.
6. **IoT**: IoT and other untrusted devices.
7. **Camera**: Cameras and network video recorder (NVR).

4.3 Server and Virtualisation Architecture

Although Proxmox supports running containers directly using LXC, the project documentation recommends nesting containers within a virtual machine to provide stronger isolation from the host and support non-Linux guest operating systems such as FreeBSD (Proxmox, 2025). The

server architecture was therefore designed around a dedicated container-host virtual machine, separating containerised application workloads from the underlying virtualisation layer.

4.4 Security Architecture

OPNsense's default inter-VLAN deny posture was taken as the baseline, with inter-VLAN connectivity to be permitted only where required. The Camera VLAN was intended to remain without internet access, while allowing limited access from the Personal VLAN for viewing and maintenance. IoT and Guest VLANs were designed to allow outbound internet access while remaining isolated from other VLANs. The Server VLAN was expected to reach the open internet and selected internal services, while the Personal VLAN was designed to access servers and the NVR but not the Work or Management VLANs. Threat-intelligence blocklists (e.g., CrowdSec) were identified as an additional layer of protection.

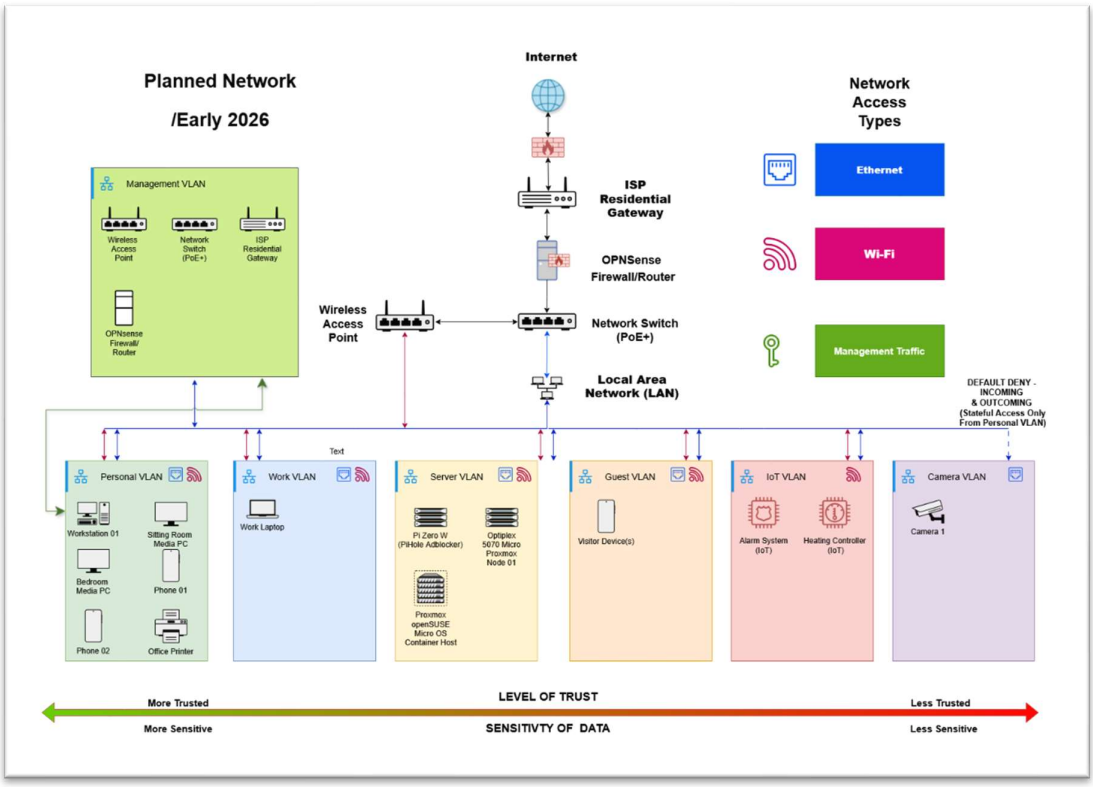

Figure 3: Logical Architecture (Design)

5. Methodology and Project Management

5.1 Project Methodology

Given the project's hardware-centric nature and lack of traditional software development phases, a Kanban methodology was selected. This approach was well suited to managing tasks with external dependencies, such as hardware procurement and staged deployment, while allowing work to progress incrementally as components became available. Kanban also supported flexible prioritisation and accommodated delays without disrupting overall project flow.

5.2 Planning and Task Management

Following definition of the target architecture, the project was broken down into a set of dependent tasks reflecting hardware procurement, configuration, deployment, and documentation activities. A Kanban board was used to track progress across states such as to-do, blocked, in-progress, and completed, allowing tasks affected by hardware lead times to be managed without stalling overall progress. Network-impacting work was planned to minimise household disruption, with temporary workarounds such as mobile connectivity identified where required.

5.3 Incremental Deployment and Testing

An incremental deployment approach was identified as appropriate, with core infrastructure components to be validated before proceeding to dependent systems. Configuration changes were intended to be tested at each stage to reduce the risk of misconfiguration or loss of connectivity, with recovery considerations incorporated into the deployment approach. Network-impacting changes were scheduled to minimise household disruption, and contingency access methods were identified where required.

5.4 Tools and Technologies Used

To support Kanban planning without reliance on a physical board, an offline, on-machine solution was prioritised. While tools such as Wekan and Vikunja were identified (AlternativeTo, 2026), options requiring self-hosting were unsuitable at the planning stage. Planify was selected as a lightweight local tool providing Kanban boards, timelines, and task tagging (Flathub, n.d.).

Its export functionality, combined with Git, enabled planning artefacts to be tracked and stored alongside project documentation.

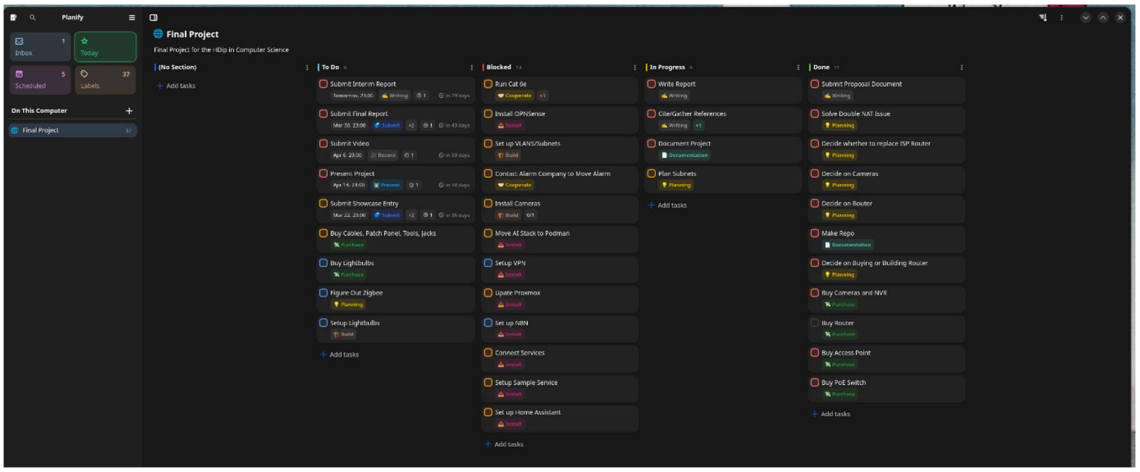

Figure 4: Planify's Board View

6. Implementation

6.1 Hardware Sourcing

This stage focused on sourcing the core routing, firewalling, switching, and wireless hardware required to support the homelab and segmented network architecture. Equipment was acquired from multiple vendors, with a preference for second-hand hardware in line with the project's sustainability focus.

A refurbished Lenovo M720q system was procured for use as the router platform, with a new PCIe riser and bracket purchased to support a second network interface. Although several recycled Realtek network interface cards were already available at no cost, a used Intel NIC was purchased separately, as FreeBSD (the base of OPNsense) is known to exhibit compatibility issues with Realtek cards that can require additional configuration to resolve (sysadmin102, 2025b).

Hardware deliveries occurred in stages, introducing schedule variability; this was mitigated through Kanban planning by progressing with tasks not dependent on outstanding components.

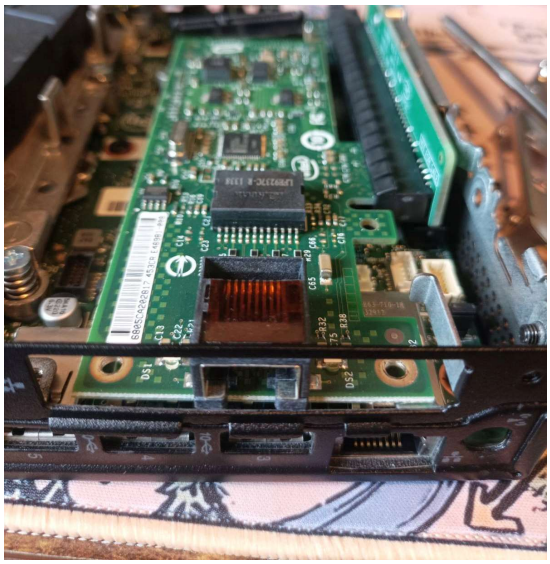

Figure 5: Riser and Intel NIC Fitted

6.2 Network Infrastructure Deployment

This stage involved physically connecting the core network infrastructure to establish basic connectivity. The router, managed switch, and access point were collocated and interconnected using short Ethernet runs to simplify cabling and troubleshooting during initial deployment. This temporary, centralised setup allowed the hardware to be powered on, interfaces verified, and cabling issues identified before any permanent placement or logical configuration was undertaken.

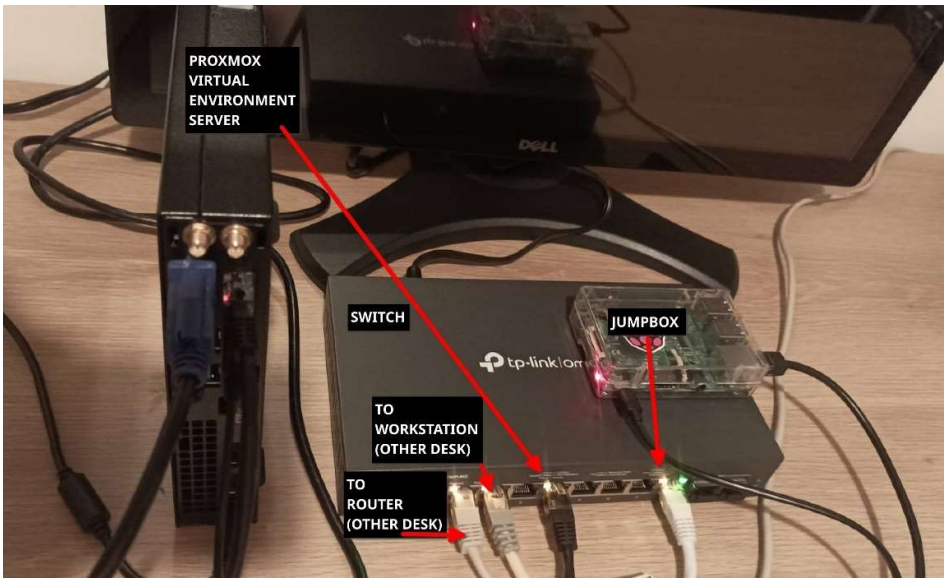

Figure 6: Initial Infrastructure Deployment

6.3 Router Setup

6.3.1 Installation

This phase covered the setup and configuration of the OPNsense router, including initial deployment, interface assignment, and activation of core network services. This work was informed by an established OPNsense setup guide (Casto, 2023).

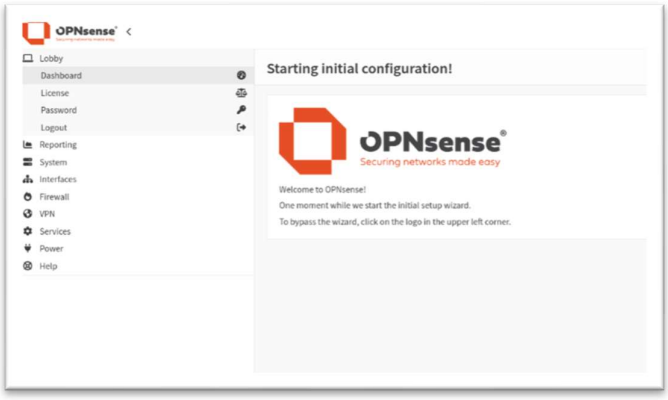

Figure 7: OPNsense Initial Configuration Wizard

The official OPNsense ISO installer image was sourced and verified using its sha256sum and the project's PGP key (Deciso B.V., 2016b). A similar verification process was followed for other operating system images used in the project.

The ISO was flashed to a USB key and used to boot the Lenovo system for installation. During setup, the onboard network interface was assigned as the WAN port and the PCIe NIC as the LAN port, after which the system was rebooted.

Following reboot, connectivity issues were encountered when attempting to access OPNsense directly. This was traced to DHCP being disabled on the LAN interface. Connectivity was restored by manually assigning an IP address in the 192.168.1.0/24 range and setting the gateway to 192.168.1.1.

After applying initial updates, core services were enabled to provide IP addressing and name resolution. Unbound was configured as the DNS resolver and Dnsmasq as the DHCP service, with Unbound forwarding local and reverse DNS lookups to Dnsmasq. DHCP functionality was verified by reconnecting the client and confirming lease assignment (Casto, 2025).

As part of the guide, a single proof of concept IoT VLAN was created with a dedicated interface address, DHCP range, and a minimal set of firewall rules.

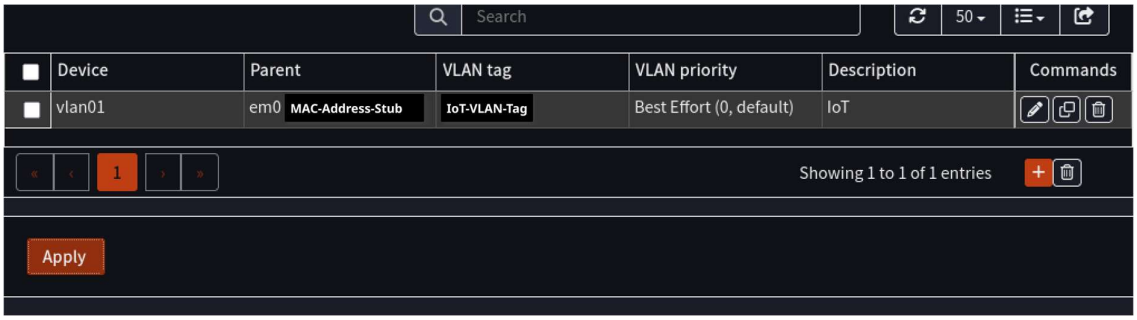

Figure 8: Setting up IoT VLAN

6.3.2 Router Hardening

In line with the principle of least privilege, the root account was disabled and a dedicated administrative user created. Multifactor authentication (TOTP) was configured and tested using OPNsense's built-in verification functionality before enforcement (Casto, 2022).

To further secure management access, the default self-signed certificate was replaced with a valid certificate for a personal domain. Split-brain DNS ensured the management subdomain resolved only internally while benefiting from a trusted Let's Encrypt certificate. A wildcard certificate was used to support consistent host-based naming.

Certificate issuance and renewal were automated using the OPNsense os-acme plugin, with post-installation hooks configured to support service restarts and certificate distribution (Campanale, 2025; Casto, 2020; Casto, 2022a; Casto, 2023b; Liv4IT, 2025; sysadmin102, 2023).

The os-cpu-microcode-intel plugin was enabled to ensure relevant CPU microcode mitigations were applied automatically (Deciso B.V., 2016a).

To reinforce this security baseline, the os-crowdsec plugin was enabled to apply crowdsourced threat-intelligence blocklists. On top of the default inbound threat-intelligence

blocking, floating rules were added to block outbound traffic to known malicious IP addresses, reducing the ability of compromised systems to establish external connections (Casto, 2022c; CrowdSec, 2022).

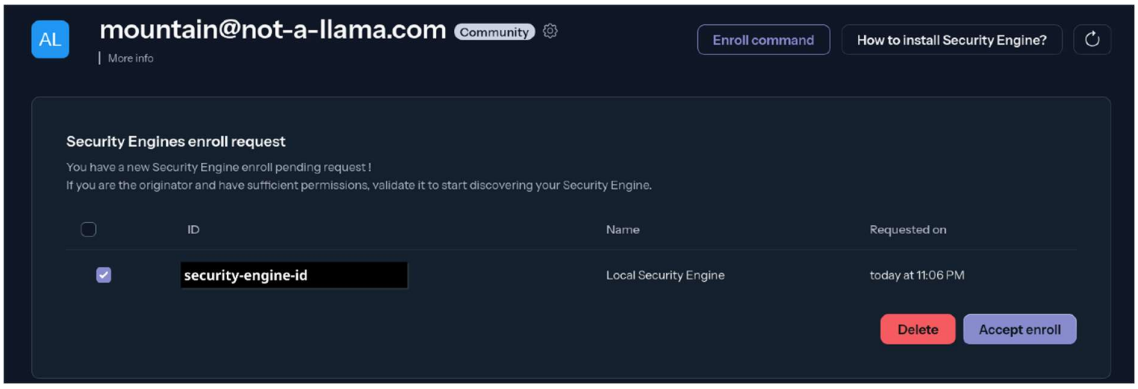

Figure 9: Accepting Crowdsec Security Engine Enrolment

6.4 VLAN, DHCP, and Firewall Configuration

The plan detailed in Section 4.2 was implemented on OPNsense through VLAN, DHCP, and firewall configuration. With the IoT VLAN already created in the prior iteration, seven additional VLANs were created in this phase, bringing the total VLAN count to eight. Each VLAN was created as an interface with the LAN interface as its parent, assigned a logical VLAN tag and description, and configured with a gateway address and subnet. DHCP ranges were then created for each VLAN, following the same pattern used for the initial IoT VLAN.

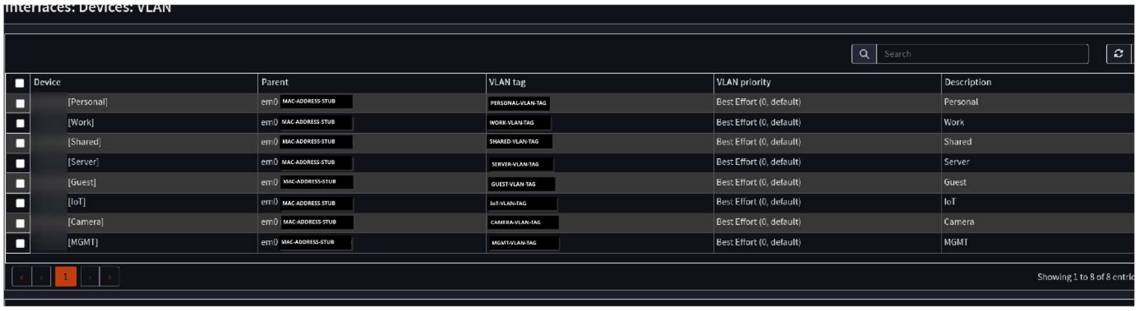

Figure 10: OPNsense VLAN Device List

At this stage, the implementation departed slightly from the original design with the addition of a Shared VLAN, used for printers and other shared devices needed by multiple VLANs but not treated as fully trusted.

Baseline firewall policy was derived from the IoT VLAN setup guide, with rules enforced by OPNsense's stateful firewall. An RFC1918-based rule blocking access to private address ranges was applied as a minimum baseline across all VLANs, preventing lateral movement between internal networks while permitting outbound internet access where appropriate. Additional rules were added to permit DNS access to the router and ICMPv4 for testing and

troubleshooting. For VLANs with stricter requirements, this baseline was further restricted: the Camera VLAN was configured without any firewall rules, allowing the default deny policy to apply to inbound and outbound traffic (Casto, 2018).

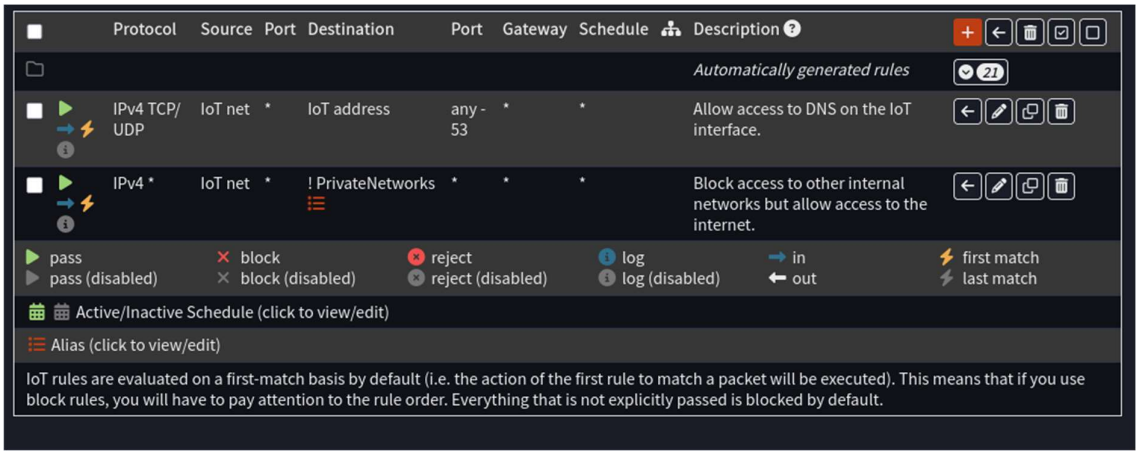

Figure 11: Example of VLAN Isolation Rule

6.5 Switch and Access Point Configuration

Once the router and VLAN architecture were in place, the managed switch was configured to support VLAN tagging and downstream connectivity. Initial configuration was performed via the switch's web interface using a direct workstation connection. As part of initial hardening, the default administrative password was changed, a new administrative user was created, and access verified before the default account was disabled.

Following this, the latest firmware was installed. VLANs were assigned to switch ports using Layer 2 configuration. The uplink port to the router was configured as a trunk, carrying all VLANs to enforce downstream traffic segmentation. As a precautionary measure, two ports were configured as untagged access ports on the Management VLAN to provide a recovery path in the event of misconfiguration. The switch management interface was assigned a static IP address within the Management VLAN and verified to be reachable after reboot.

To support wireless connectivity, a second trunk port was configured for the access point.

The access point was connected to the trunk port and configured with administrative hardening like the switch, including creation of a dedicated administrative account. An interface address was assigned on the Management VLAN. Two SSIDs were created for the Personal VLAN, operating on the 2.4 GHz and 5 GHz bands respectively. Test connections confirmed correct Wi-Fi operation, internet access, and DHCP address assignment from the expected VLAN (Casto, 2019).

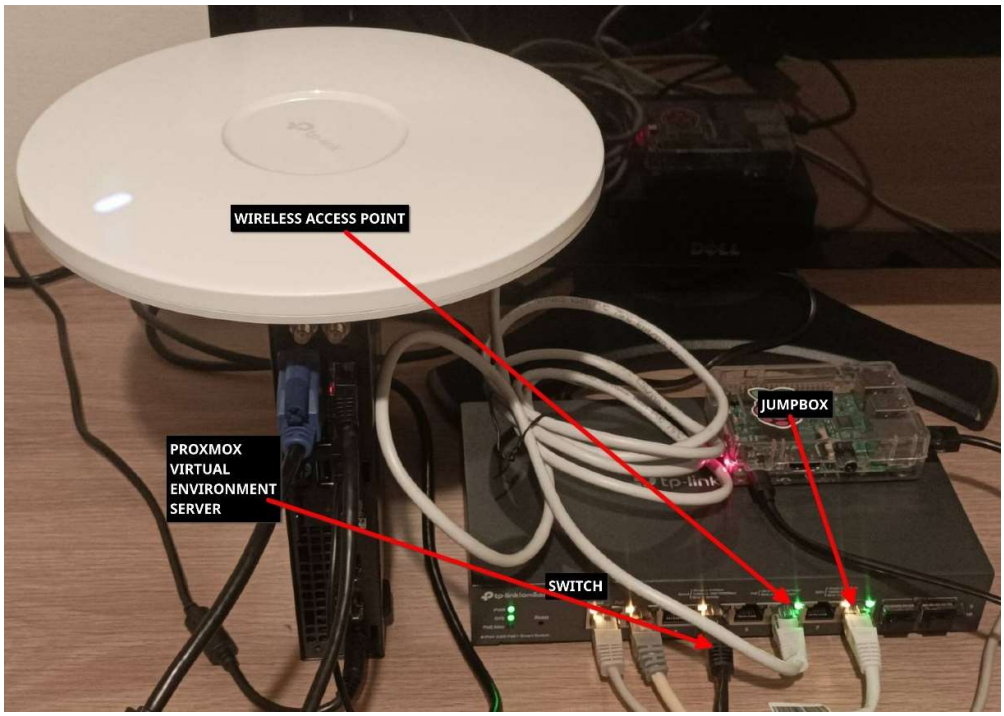

Figure 12: Secondary Infrastructure Deployment

6.6 Management Access Hardening

With the core VLAN structure in place and a dedicated Management VLAN established, a further phase of hardening focused on restricting administrative access to network infrastructure. Rather than allowing direct management access from the Personal VLAN, a minimal, dedicated jumpbox was deployed within the Management VLAN. This approach mirrors common enterprise practice, where bastion hosts serve as the sole access path into management networks (Casto, 2024).

To prevent accidental lockout during this transition, temporary rules were applied to allow the primary workstation to access the router, switch, and access point directly until the jumpbox was operational.

To further harden the jumpbox, additional guidance was followed covering SSH hardening, removal of unnecessary packages, and locking down local user accounts (Rampal, 2026).

To enable use of the jumpbox as the sole management entry point, an SSH-based reverse proxy was configured (Uche, 2024). This allowed management interfaces to be accessed via a web browser, either through manual proxy configuration or a browser extension such as FoxyProxy. The following alias was used to establish the connection:

```
ssh -i ~/.ssh/{ssh_key} -p {ssh-port} -D {port-to-use-for-proxy} -N -v
jumpboxuser@{jumpboxip}
```

As the network was operating in a double-NAT configuration, an additional rule allowed the jumpbox to access the ISP router's management interface, ensuring a single route for all management traffic (Casto, 2021).

Access from the Personal VLAN to the jumpbox was restricted to SSH only, with the jumpbox permitted to initiate SSH connections to management devices. This access path was further secured using hardware-backed multifactor authentication with YubiKey-based resident SSH keys, requiring PIN entry and physical user presence. As a final precaution, all general internet access from the Management VLAN was disabled, with only approved firmware and update endpoints permitted (Casto, 2026; Yubico, 2026).

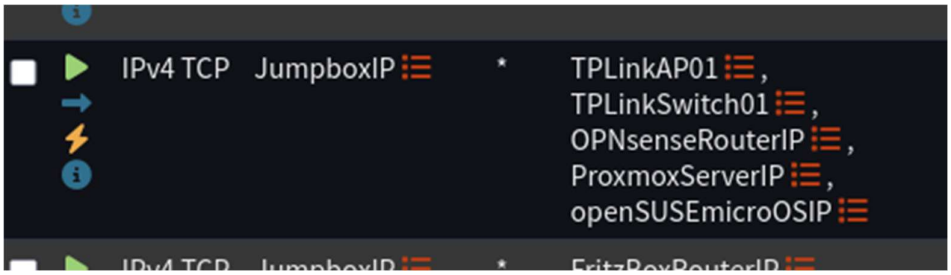

Figure 13: SSH Allowed from Jumpbox to Management Interface via Aliases

6.7 Server and Container Deployment

To enable application and container deployment, the existing OptiPlex-based Proxmox server was moved to the Server VLAN via an untagged switch port and upgraded in-place from Proxmox VE 8 to 9 (Proxmox, 2026).

Post-upgrade hardening focused on securing SSH access, enrolling Proxmox users in multi-factor authentication, and preventing direct root logins (alexandreravelli, 2025).

OPNsense ACME automation was leveraged to distribute a trusted certificate to the Proxmox interface; combined with a DHCP reservation, this enabled access via a predictable hostname e.g., 'proxmox-ve.homelab.mydomain.tld' (sysadmin102, 2025a; sysadmin102, 2026).

Figure 14: Proxmox User List with MFA Enabled

To reduce exposure, the Proxmox management interface was moved into the Management VLAN. Firewall rules and whitelist aliases were updated to permit required update domains without allowing general internet access, while hosted workloads continued to use Server VLAN addressing. Proxmox networking was updated to be VLAN-aware so tagged traffic for non-management VLANs could be carried as required (Proxmox, 2025a).

An openSUSE MicroOS virtual machine was deployed as the container host in line with the earlier platform selection, with backups configured once the environment stabilised (Beit-Halahmi, 2023; Thomas-Krenn AG, 2024; ProxmoxHHS, 2019).

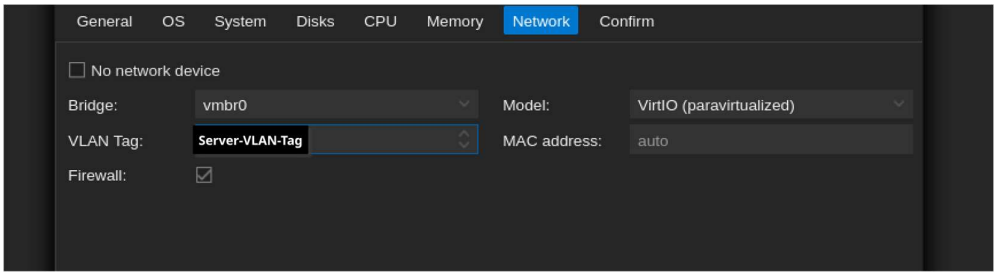

Figure 15: Establishing MicroOS VM on Server VLAN

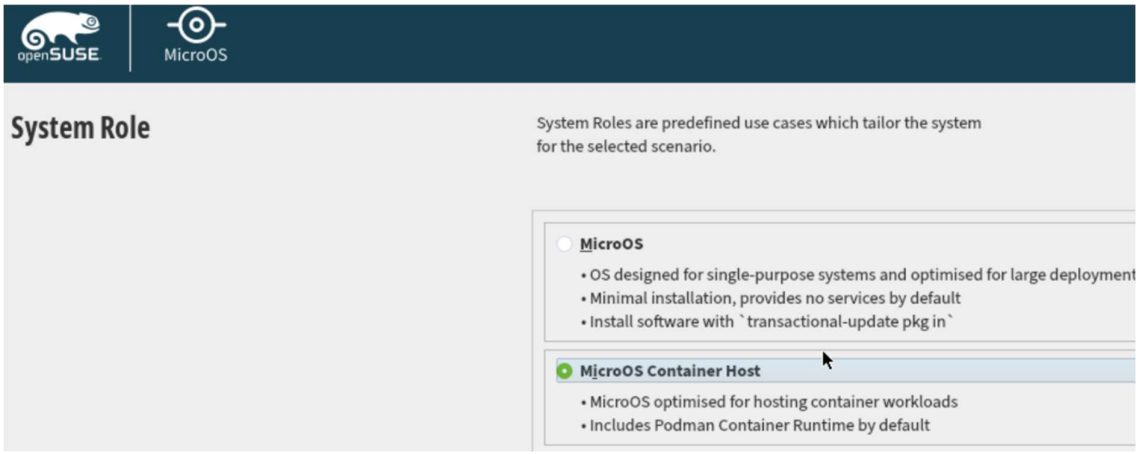

Figure 16: Setting up MicroOS as Container Host

As a final hardening measure, firewall controls were applied in layers across the stack (Proxmox datacenter, node, VM, and firewalld on MicroOS) to support defence-in-depth during service deployment (nbeam, 2016; Tkanov, 2025).

6.8 Self-Hosted Services and Applications

With the container host operational, a prototype stack was deployed to demonstrate self-hosting and service delivery. The application stack was based on an earlier assignment in the Computer Systems and Networks module, providing a familiar base. To suit the household context, the stack was reduced to a Traefik reverse proxy and a single application. Mealie, an open-source meal-planning and recipe-management application, was used due to its domestic focus.

Initial deployment used existing Docker Compose definitions executed via Podman Compose to confirm correct operation in the new environment.

Once baseline functionality was established, the deployment was migrated to a Podman-native approach. The existing Compose definitions were converted into

systemd-managed Podman Quadlets using the Podlet utility. This required modification and iterative troubleshooting to ensure reliable connectivity and access to both the Traefik dashboard and the Mealie service. This stack was developed and version controlled in a separate operations repository (Coletto, 2024; Kerkhof, 2022; mag37, 2024; Oliveira, 2023).

Following successful configuration, Traefik and Mealie were placed behind the reverse proxy and secured using TLS. To maintain separation between infrastructure components and user-facing services, application endpoints were assigned to a dedicated subdomain namespace distinct from the hardware-oriented local domain used elsewhere (for example, *.applications.mydomain.tld versus *.homelab.mydomain.tld).

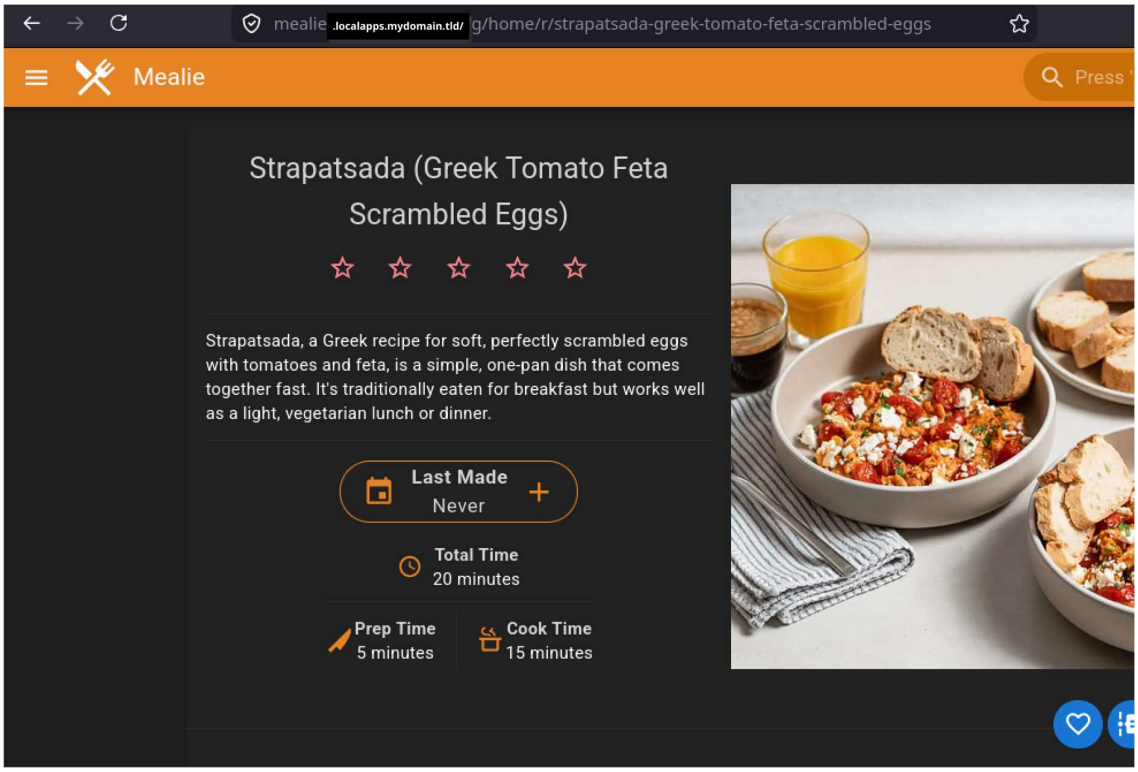

Figure 17: Self-hosted Mealie Instance

6.9 Home Automation and IoT

Given the overall scope of the project, home automation and IoT integration was limited to deploying a functional surveillance setup within the Camera VLAN. The network video recorder (NVR) was connected to an untagged switch port assigned to the Camera VLAN, with the camera connected directly to the NVR to receive power and network connectivity via its PoE ports.

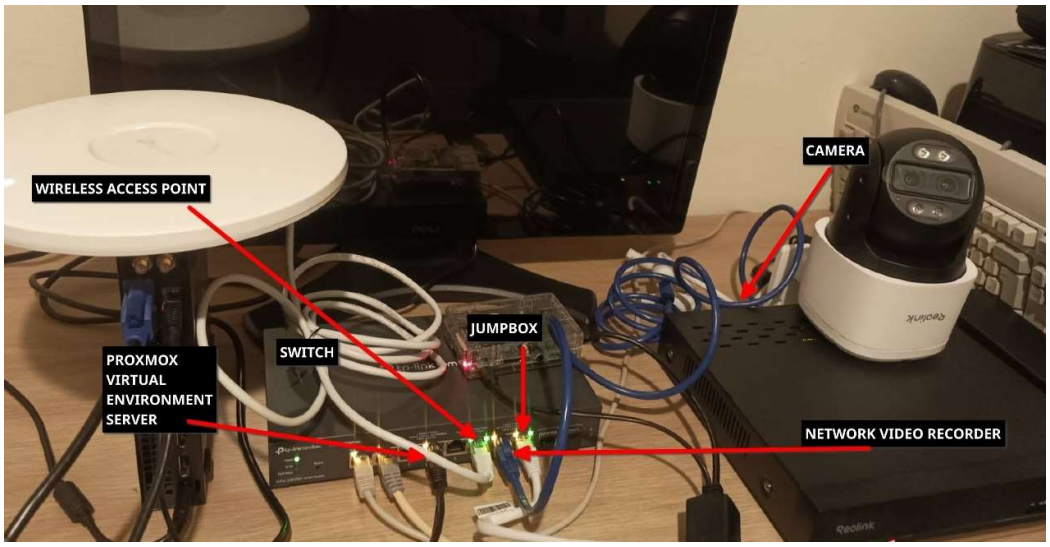

Figure 18: Final Infrastructure Deployment

Initial configuration of the camera and NVR was performed locally using a monitor and input devices until basic networking was established. Once operational, the NVR web interface and camera feed became accessible on the internal network. To enable access without exposing the Camera VLAN to the internet, firewall rules were configured to allow devices on the Personal VLAN to reach the NVR on the required ports. As OPNsense operates as a stateful firewall, the NVR and camera were able to respond to authorised inbound connections while remaining unable to initiate outbound connections.

Figure 19: View from Camera

7. Reflection

Though the components of this network and homelab are not novel or groundbreaking, I approached the project primarily as a learning exercise, with a focus on developing skills in systems architecture, networking, and cybersecurity. Tutorials informed the shape and direction of the project, but I made design decisions carefully and deliberately through a synthesis of best-practice guides and community recommendations, adapting or departing from tutorials where appropriate.

7.1 Achievements

Through the planning, design and implementation of this project, I have rearchitected a flat home network into a VLAN-segmented, hardened domestic network and homelab with a minimal but extensible self-hosted application stack.

The new network and access point support the same routine use cases of the previous ISP router-centric network, including browsing, messaging, streaming and email. From an operational standpoint, however, segmentation is enforced on a per SSID basis, with home users automatically being placed into the personal VLAN and being assigned an IP address on the personal subnet via DHCP. Firewall rules have been put in place to provide isolation from other VLANs and to selectively allow stateful communications with specific devices in other VLANs for service delivery.

I established a Proxmox and MicroOS-based virtualisation and container-hosting platform for application deployment via Podman. The platform hosts a Traefik reverse proxy, with Mealie deployed as an example of a web application for domestic use. Users on the personal VLAN can access this service through a fully qualified domain name and, thanks to a valid TLS certificate, there are no initial warnings about self-signed certificates. This local instance of Mealie provides a similar feature set to some of its cloud-hosted competitors, while reducing reliance on third-party data processors.

Through an iterative build, testing and improvement process, I hardened each layer of the network following best-practice guidelines. I set up new administrative accounts with strong passwords and enabled MFA where possible. I also disabled default administrator accounts and removed them, where supported by the platform. I implemented rules for each firewall layer, from host to datacentre level, applying a default-deny posture while maintaining the specific routes required for service delivery and management access.

At the router level, I have DHCP and DNS running cleanly across the required interfaces, so devices get addresses normally. Through Unbound, Dnsmasq, and the forwarding rules between them, local DNS records are set for each machine and service on the network. I also set up ACME automation to enable key nodes to have valid service-specific TLS certificates. As an additional layer of protection, I enabled the CrowdSec plugin to apply threat-intelligence blocklists, blocking inbound and outbound connections to known malicious IP addresses and providing visibility into blocked traffic.

I designed and set up access to the management plane of the network so that all access must be authorised via designated hardware security keys and touch verification, providing an extra barrier between it and any device attempting to establish a connection, which offers some deterrence against automated lateral movement attempts.

The network now has an operational NVR and camera stack, which via the Camera VLAN, is isolated from the wider internet and only accessible from machines in the Personal VLAN.

During the implementation phase, the system withstood misconfiguration, and I was able to recover quickly from lockouts. While the network was not in continuous use throughout this phase, once core services were established, services such as Wi-Fi were not disrupted during later iterations, and applications like Mealie remained available even as work was carried out in other layers of the network.

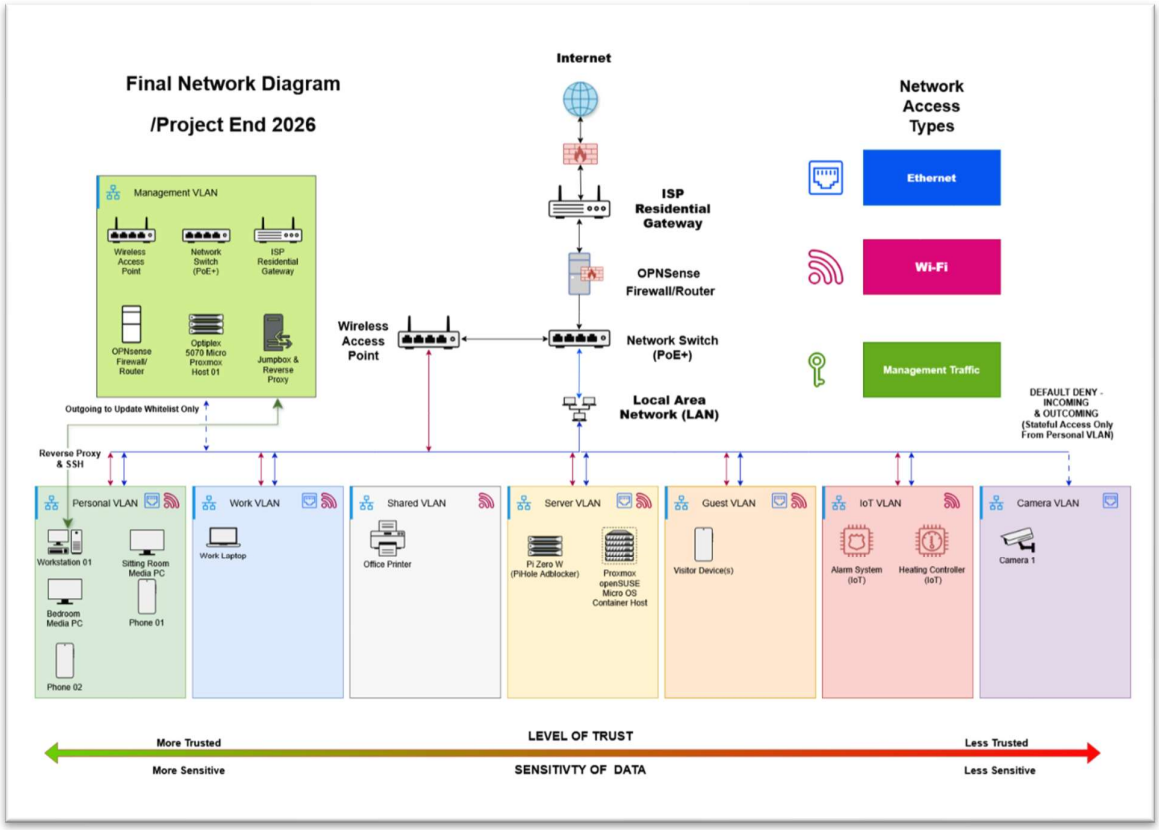

Figure 20: Logical Architecture (Implemented)

7.2 Challenges Encountered

Reflecting on the challenges I encountered, many arose during the early RAMP phase (Research, Analysis, Mapping and Planning), largely due to the volume of hardware options and the number of devices required for a network of this scale. Firewall hardware selection was

particularly difficult due to the range of viable options and the conflicting trade-offs emphasised by different sources.

Even after decisions were made, sourcing introduced delays that required adjustments to sequencing. For example, the Intel PCIe NIC delivery dragged on unusually long before being refunded without warning, forcing a late pivot to source an alternative used card.

Scheduling constraints also meant I could not rely on external assistance to lay Ethernet cabling within the project timeframe, which narrowed what could be achieved in terms of permanent infrastructure during the submission window. As a result, final placement of device, taking into account factors such as noise and heat, was not completed prior to submission.

Even when hardware arrived on time, unexpected setbacks occurred. The initial Lenovo router platform arrived poorly refurbished, with visible dirt and poorly applied thermal paste. When I attempted to clean the overflow, fibres caught in the CPU socket and bent pins out of alignment, preventing the CPU from functioning. I initially attempted to straighten the pins without magnification then, at a colleague's suggestion, revisited the socket using a document camera to inspect alignment more closely. This revealed that while many pins could be corrected, a small number had snapped and were missing entirely. Under time pressure, I sourced a second identical unit to keep the project moving, but this introduced a secondary complication: during the initial OPNsense install I accidentally used the wrong Lenovo, one that only had a spare Realtek card installed, leading to difficult-to-diagnose connectivity issues and extended troubleshooting.

Overall, despite having covered basic networking in earlier modules, building and securing a network of this complexity represented a steep learning curve. Troubleshooting was complicated by multiple firewall layers (OPNsense and Proxmox at datacenter/node/VM level, alongside host firewalls), and the combination of VLANs and stateful rules made getting locked out more likely, particularly given the use of long, password-manager-generated credentials.

An additional factor that added to cognitive load was balancing the need to demonstrate my work with the responsibility of protecting sensitive details of the internal network topology. While such information is often freely shared within the homelabbing community, I felt that unnecessary disclosure could provide a malicious reader with an initial advantage in mapping the network's attack surface.

Figure 21: Bent Pins on Router Socket

7.3 Solutions and Adaptations

For the challenges encountered during the project, I developed practical adaptations through trial and error that allowed the work to continue and reach a stable conclusion. The iterative workflow and Kanban approach helped here, as it made it easier to identify blocked tasks and shift focus to other work that could progress independently. In practice, sourcing delays were managed by continuing with whatever hardware was available and using waiting periods productively for report writing and incremental documentation.

Following the CPU socket damage described in Section 7.2, I took a sustainability-aligned approach rather than discarding the Lenovo chassis. After confirming that a small number of socket pins were snapped and missing, I researched repair options and sourced a replacement motherboard for the mini-PC at reasonable cost from China. This was also influenced by advice from the self-hosting community that a router platform benefits from having a like-for-like spare available for rapid recovery. Once the replacement motherboard arrived, I transferred the CPU, RAM, cooler, network card, CMOS battery, and other components, restoring the original chassis to a fully operational state.

To avoid further mix-ups between the two Lenovos, I noted the refurbisher's logo sticker on the production machine, versus the absence of any such sticker on the backup, Realtek model.

The absence of a preexisting wired backbone influenced how the network was deployed. By necessity, core devices were collocated on a single desk, which had the unexpected benefit of simplifying recovery from lockouts and troubleshooting. Physical access to equipment made it easier to verify cabling and quickly regain control during configuration errors.

To address issues with having to type long passwords when locked out of remote sessions where I could copy and paste, I repurposed an old Arduino-based "Malduino" Bad USB device, which I was able to flash successfully for the first time after years of ownership. This allowed scripted entry of these passwords via Duckyscript, leveraging the fact that such devices are recognised as standard human interface devices by most operating systems (Maltronics Limited, 2026).

Finally, in response to the concerns about disclosing network details within the context of my threat model, while security through obscurity is not considered a robust approach and the network design follows modern zero-trust principles, I erred on the side of caution. As a result, elements such as VLAN IDs, hostnames, domains, and internal IP addresses were redacted or deliberately disguised, and due to the project's hardware-heavy nature and the sensitivity of artefacts such as router configuration files, the report itself will serve as the primary vehicle for evidencing the work undertaken.

Figure 22: Router Repaired

7.4 Skills Developed

This project allowed me to develop practical skills in designing and deploying a segmented network architecture. By implementing multiple VLANs with distinct roles, interfaces, DHCP ranges, and subnets, I gained a clearer understanding of how network structure supports security and maintainability. Using a consistent numbering scheme for VLANs and subnets reinforced my understanding of gateway addresses, subnetting, and overall network design.

I also gained hands-on experience configuring stateful firewall policies and applying zero-trust principles in practice. By considering the role and trust level of each VLAN and device, I was able to write targeted firewall rules that permitted or restricted traffic as required. This included limiting internet access for less-trusted devices, such as cameras, and addressing concerns around devices "phoning home." Working through these configurations improved my understanding of firewall behaviour and my confidence in diagnosing connectivity issues.

Building and securing my own OPNsense router developed my systems administration skills. I configured core services alongside security features such as ACME certificate automation, account hardening, and CrowdSec. Applying similar hardening steps on Proxmox reinforced the importance of layering controls across the stack rather than relying on a single security boundary, and helped me understand how these tools complement each other in practice.

Once the core infrastructure was in place, I deepened my experience with container and service orchestration using openSUSE MicroOS and Podman. Converting Docker Compose files into Podman Quadlets and managing them through systemd, alongside refreshing my understanding of Traefik and reverse proxies, gave me a solid foundation for exploring these technologies at a more intermediate level.

7.5 Future Improvements

As the network deployed through this project represents a minimally viable product (MVP) of an advanced homelab, built using guided tutorials and improvisation within a fixed timeframe, there remains significant scope for further improvement and expansion.

Once additional testing has been undertaken with household members using the network and its services, I plan to configure Point-to-Point Protocol over Ethernet (PPPoE) and replace or bridge the ISP-provided router. This would remove the current double-NAT arrangement and reduce the potential for connectivity issues.

To enable more advanced control over the TP-Link infrastructure, I plan to deploy the software-based Omada Controller in a virtual machine. This would allow the use of pre-shared private keys (PPSKs) per VLAN, reducing the need for multiple SSIDs and avoiding unnecessary wireless congestion (Crawford, 2020; IPTel Solutions, 2025).

On the physical infrastructure side, Ethernet cabling still needs to be installed to connect rooms across both floors of the house, along with permanently mounting the external camera. To better organise the growing number of devices, I plan to introduce a small 10-inch rack to house core components such as the OPNsense router, Proxmox server, and switch, with larger devices like the NVR located separately.

Additional future extensions may include deploying a NAS and running Home Assistant in a virtual machine to further explore home automation and IoT integration. With appropriate hardware upgrades, the network could also be extended to support higher speeds, such as 2.5 Gbps routing or 10 Gbps links using the switch's SFP+ ports.

Finally, further functionality could be added through secure remote access, for example by deploying a WireGuard VPN or Tailscale, allowing access to internal services like Mealie without exposing ports (Casto, 2020b).

Some scalability considerations have emerged, particularly around switch port capacity if more devices are hardwired. Limited power outlets in several rooms may also require additional sockets or increased use of PoE-capable devices.

8 Conclusion

8.1 Project Outcomes

The aim of the project was to design and implement a scalable, logically segmented and secure home network that can serve as a platform for further explorations of open-source technologies, networking and cybersecurity.

This project successfully fulfilled the stated goal of deploying a segmented, privacy-centric home network and homelab for a single household. It established the core infrastructure required for daily use and future lab experimentation, including an OPNsense router, managed switch, dedicated access point, Proxmox virtualisation server, and the beginnings of a suite of self-hosted applications as an alternative to cloud-hosted platforms.

In practical terms, this delivered the intended outcomes of segmenting the network based on trust levels, deploying services and providing a stable configuration, which aligns with the objectives that were defined in Section 1.3 Project Objectives.

A preference for EU-based projects was applied where practical but was not always the primary deciding factor. Where EU options were limited, open-source and self-hosted alternatives still fulfilled the goal of greater digital autonomy.

8.2 Academic Value

In the context of digital autonomy and privacy, this project demonstrates the viability of catering to a household's needs through self-hosting as a means of reducing dependence on third-party SaaS for some use cases, so long as an individual can assume the role of the household's systems administrator.

From a cybersecurity perspective, the work shows how VLAN-based segmentation can isolate devices by trust levels to reduce internal risks in modern home networks, while stateful firewalls still allow one to implement access to specific locally hosted services.

Given the project's scope, the outcome is a proof-of-concept for a single household network, rather than a broadly replicable or enterprise-level solution. Some planned elements were ultimately not implemented prior to submission, such as rolling out PPPoE and laying Ethernet through the home, and the resulting system represents a stable foundation for these additions, rather than the final state of the network and homelab.

Overall, this project provides a secure and extensible baseline homelab that meets the project's original aims and can support further development beyond the submission window.

9 References

alexandreravelli (2025). *GitHub - alexandreravelli/Securing-SSH-Access-on-Proxmox-VE-9: Complete Guide to Secure SSH Access on Proxmox VE 9+ Using Key authentication, Least Privilege principle, and Hardened Configuration*. [online] GitHub. Available at: https://github.com/alexandreravelli/Securing-SSH-Access-on-Proxmox-VE-9 [Accessed 28 Mar. 2026].

AlternativeTo (2026). *Open Source Trello Alternatives for Linux*. [online] AlternativeTo. Available at: https://alternativeto.net/software/trello/?license=opensource&platform=linux [Accessed 15 Feb. 2026].

ArchWiki (2023). *Btrfs - ArchWiki*. [online] Archlinux.org. Available at: https://wiki.archlinux.org/title/Btrfs#Automatic_snapshots [Accessed 12 Feb. 2026].

Beit-Halahmi, D. (2023). *How to Create a VM in Proxmox*. [online] Linux Handbook. Available at: https://linuxhandbook.com/courses/proxmox/proxmox-create-vm/ [Accessed 28 Mar. 2026].

Cameron Gray (2025). *Building My Fully MikroTik Home Network! - Part 2: MikroTik CRS Switches*. [online] YouTube. Available at: https://www.youtube.com/watch?v=BGF8WogFODY [Accessed 13 Feb. 2026].

Campanale, P. (2025). *Why Your Homelab Needs a Domain*. [online] How-To Geek. Available at: https://www.howtogeek.com/why-your-homelab-needs-a-domain/ [Accessed 28 Mar. 2026].

Casto, D. (2018). *Need an Offline Local Network for a Home Lab or IP Video Cameras?* [online] Homenetworkguy.com. Available at: https://homenetworkguy.com/how-to/configure-opnsense-isolated-vlan/ [Accessed 28 Mar. 2026].

Casto, D. (2019). *How to Configure VLANs on TP-Link Switch for UniFi Access Points with VLAN per SSID*. [online] Homenetworkguy.com. Available at: https://homenetworkguy.com/how-to/configure-tp-link-switch-vlan-with-unifi-access-points-vlan-per-ssid/ [Accessed 28 Mar. 2026].

Casto, D. (2020a). *Disable Logging into OPNsense as the Root User*. [online] Homenetworkguy.com. Available at: https://homenetworkguy.com/how-to/disable-root-user-opnsense/ [Accessed 28 Mar. 2026].

Casto, D. (2020b). *How to Connect to Your Home Network via WireGuard VPN Server in OPNsense*. [online] Homenetworkguy.com. Available at: https://homenetworkguy.com/how-to/configure-wireguard-opnsense/ [Accessed 28 Mar. 2026].

Casto, D. (2021). *How to Access Your Modem's Web Interface with OPNsense*. [online] Homenetworkguy.com. Available at: https://homenetworkguy.com/how-to/access-your-modem-web-interface-with-opnsense/ [Accessed 28 Mar. 2026].

Casto, D. (2022a). *12 Ways to Secure Access to OPNsense and Your Home Network*. [online] Homenetworkguy.com. Available at: https://homenetworkguy.com/how-to/ways-to-secure-access-to-opnsense-and-your-home-network/ [Accessed 28 Mar. 2026].

Casto, D. (2022b). *How to Enable Multi-Factor Authentication in OPNsense*. [online] Homenetworkguy.com. Available at: https://homenetworkguy.com/how-to/enable-multi-factor-authentication-in-opnsense/ [Accessed 28 Mar. 2026].

Casto, D. (2022c). *How to Install and Configure CrowdSec on OPNsense*. [online] Homenetworkguy.com. Available at: https://homenetworkguy.com/how-to/install-and-configure-crowdsec-on-opnsense/ [Accessed 28 Mar. 2026].

Casto, D. (2023a). *Beginner's Guide to Set Up a Home Network Using OPNsense*. [online] Homenetworkguy.com. Available at: https://homenetworkguy.com/how-to/beginners-guide-to-set-up-home-network-using-opnsense/ [Accessed 28 Mar. 2026].

Casto, D. (2023b). *Replace the OPNsense Web UI Self-Signed Certificate with a Let's Encrypt Certificate*. [online] Homenetworkguy.com. Available at: https://homenetworkguy.com/how-to/replace-opnsense-web-ui-self-signed-certificate-with-lets-encrypt/ [Accessed 28 Mar. 2026].

Casto, D. (2024). *Set up a Management VLAN for OPNsense, a Network Switch, and a Wireless Access Point*. [online] Homenetworkguy.com. Available at: https://homenetworkguy.com/how-to/set-up-management-vlan-for-opnsense-network-switch-and-access-point/ [Accessed 28 Mar. 2026].

Casto, D. (2025). *How to Migrate from ISC DHCP to Dnsmasq or Kea DHCP in OPNsense*. [online] Homenetworkguy.com. Available at: https://homenetworkguy.com/how-to/migrate-from-isc-dhcp-to-dnsmasq-or-kea-dhcp-in-opnsense/ [Accessed 28 Mar. 2026].

Casto, D. (2026). *Write Better Firewall Rules in OPNsense Using Aliases*. [online] Homenetworkguy.com. Available at: https://homenetworkguy.com/how-to/write-better-firewall-rules-opnsense-using-aliases/ [Accessed 28 Mar. 2026].

Coletto, G. (2024). *How to Convert Any docker-compose.yml to Quadlets with Podlet*. [online] Giacomo Coletto. Available at: https://giacomo.coletto.io/blog/podman-podlet/ [Accessed 28 Mar. 2026].

Conway, A. (2025). *Why I use OPNsense over pfSense, and why I don't trust Netgate at all*. [online] XDA. Available at: https://www.xda-developers.com/why-use-opnsense-over-pfsense-dont-trust-netgate/ [Accessed 12 Feb. 2026].

Crawford, M. (2020). *Why restrict the number of SSIDs on an AP*. [online] MC Wireless. Available at: https://www.mcwireless.co.uk/post/why-restrict-the-number-of-ssids-on-an-ap [Accessed 28 Mar. 2026].

CrowdSec (2022). *OPNsense | CrowdSec*. [online] Crowdsec.net. Available at: https://docs.crowdsec.net/docs/getting_started/install_crowdsec_opnsense/ [Accessed 28 Mar. 2026].

Deciso B.V. (2016a). *CPU Microcode updates [AMD/Intel] — OPNsense documentation*. [online] Opnsense.org. Available at: https://docs.opnsense.org/manual/cpu-microcode.html [Accessed 28 Mar. 2026].

Deciso B.V. (2016b). *Hardware Sizing & Setup — OPNsense Documentation*. [online] Opnsense.org. Available at: https://wiki.opnsense.org/manual/hardware.html [Accessed 19 Jan. 2026].

Deciso B.V. (2016c). *Initial Installation & Configuration — OPNsense Documentation*. [online] Opnsense.org. Available at: https://docs.opnsense.org/manual/install.html#download-and-verification [Accessed 28 Mar. 2026].

Deciso B.V. (2016d). *VLAN and LAGG Setup — OPNsense Documentation*. [online] Opnsense.org. Available at: https://docs.opnsense.org/manual/how-tos/vlan_and_lagg.html [Accessed 19 Jan. 2026].

Deciso B.V. (2022). *DEC677 - OPNsense® Desktop Security Appliance*. [online] Opnsense.com. Available at: https://shop.opnsense.com/product/dec677-opnsense-desktop-security-appliance/ [Accessed 20 Jan. 2026].

Deciso B.V. (2024). *Let's Start a Conversation - Deciso*. [online] Deciso. Available at: https://www.deciso.com/get-in-touch/ [Accessed 20 Jan. 2026].

Deciso B.V. (2026). *Product Categories Hardware*. [online] Opnsense.com. Available at: https://shop.opnsense.com/product-categorie/hardware-appliances/ [Accessed 21 Jan. 2026].

Directorate-General for Communications Networks, Content and Technology (2026). *Page Restricted*. [online] Europa.eu. Available at: https://digital-strategy.ec.europa.eu/en/news/commission-opens-call-evidence-open-source-digital-ecosystems [Accessed 19 Jan. 2026].

Fedora (n.d.). *Welcome*. [online] Fedora Docs. Available at: https://docs.fedoraproject.org/en-US/iot/.

Fedora Project (2025). *Changes/KDEKinoiteAutoUpdateByDefault - Fedora Project Wiki*. [online] Fedoraproject.org. Available at: https://fedoraproject.org/wiki/Changes/KDEKinoiteAutoUpdateByDefault [Accessed 12 Feb. 2026].

Fedora Project (2026). *The KDE Plasma desktop, in an Atomic Fashion*. [online] Fedoraproject.org. Available at: https://fedoraproject.org/atomic-desktops/kinoite/.

Flathub (n.d.). *Planify*. [online] Flathub.org. Available at: https://flathub.org/en/apps/io.github.alainm23.planify [Accessed 15 Feb. 2026].

Freund, A. (2024). *oss-security - backdoor in upstream xz/liblzma leading to ssh server compromise*. [online] www.openwall.com. Available at: https://www.openwall.com/lists/oss-security/2024/03/29/4.

GeeksforGeeks (2023). *What is Mesh Network?* [online] GeeksforGeeks. Available at: https://www.geeksforgeeks.org/computer-networks/what-is-mesh-network/.

geerlingguy (2025). *GitHub - geerlingguy/mini-rack: Miniature rack builds, for portable or compact Homelabs*. [online] GitHub. Available at: https://github.com/geerlingguy/mini-rack?tab=readme-ov-file#racks [Accessed 19 Jan. 2026].

HomeTechHacker (2024). *7 Great Choices for OPNsense Hardware - HomeTechHacker*. [online] HomeTechHacker. Available at: https://hometechhacker.com/great-choices-for-opnsense-hardware/ [Accessed 19 Jan. 2026].

Insecam (2023). *FAQ on insecure cameras*. [online] Insecam.org. Available at: http://www.insecam.org/en/faq/.

IPTel Solutions (2025). *Why Too Many SSIDS Are Bad for Your Wi-Fi*. [online] Iptel.com.au. Available at: https://blog.iptel.com.au/why-too-many-ssids-are-bad-for-your-wi-fi [Accessed 28 Mar. 2026].

Kerkhof, R. (2022). *Autostarting, auto-updating, Rootless Podman Containers*. [online] Nano's Blog. Available at: https://www.nanosector.nl/posts/podman-rootless-autostart/ [Accessed 28 Mar. 2026].

Liv4IT (2025). *How to Install Let's Encrypt Certificate on OPNsense Using ACME Client*. [online] YouTube. Available at: https://www.youtube.com/watch?v=YdbKXYMTRKs [Accessed 28 Mar. 2026].

mag37 (2024). *Guide: Getting Started with Podman Quadlets*. [online] mag37.org. Available at: https://mag37.org/posts/guide_podman_quadlets/ [Accessed 28 Mar. 2026].

Maltronics Limited (2026). *Maltronics*. [online] Maltronics. Available at: https://maltronics.com/pages/malduino-help [Accessed 28 Mar. 2026].

Maltronics Limited (2022). *MalDuino 3*. [online] Maltronics. Available at: https://maltronics.com/products/malduino-3 [Accessed 28 Mar. 2026].

Michaud, Q. (2025). *[SECURITY] firefox-patch-bin, librewolf-fix-bin and zen-browser-patched-bin AUR packages contain malware - Aur-general - lists.archlinux.org*. [online] Archlinux.org. Available at: https://lists.archlinux.org/archives/list/aur-general@lists.archlinux.org/thread/7EZTJXLIAQLARQNTMEW2HBWZYE626IFJ/.

MikroTik (2026). *MikroTik*. [online] Mikrotik.com. Available at: https://mikrotik.com/aboutus/company [Accessed 13 Feb. 2026].

muffn_ (2024). *OPNsense On A Lenovo M720q With 10Gb*. [online] Muffn.io. Available at: https://blog.muffn.io/posts/m720q-opnsense-firewall/ [Accessed 13 Feb. 2026].

nbeam (2016). *Secure Proxmox Install – Sudo, Firewall with IPv6, and more – How to Configure from Start to Finish*. [online] KiloRoot. Available at: https://www.kiloroot.com/secure-proxmox-install-sudo-firewall-with-ipv6-and-more-how-to-configure-from-start-to-finish/ [Accessed 28 Mar. 2026].

Netgate (n.d.). *About Us*. [online] www.netgate.com. Available at: https://www.netgate.com/about-us [Accessed 22 Jan. 2026].

Oliveira, A. (2023). *Configure a Container to Start Automatically as a Systemd Service*. [online] Redhat.com. Available at: https://www.redhat.com/en/blog/container-systemd-persist-reboot [Accessed 28 Mar. 2026].

openSUSE contributors (2025). *openSUSE MicroOS*. [online] Get openSUSE. Available at: https://get.opensuse.org/microos/?type=server#download [Accessed 28 Mar. 2026].

OPNsense (2025). *OPNsense® - OPNsense*. [online] OPNsense. Available at: https://opnsense.org/opnsense/.

OPNsense Forum Community Members (2026). *Starting Homelab Network - Hardware Choices*. [online] OPNsense Forum. Available at: https://forum.opnsense.org/index.php?topic=50357.0 [Accessed 19 Jan. 2026].

Parallax (2021). *Lenovo Thinkcentre/ThinkStation Tiny (Project TinyMiniMicro) Reference Thread*. [online] ServeTheHome Forums. Available at: https://forums.servethehome.com/index.php?threads/lenovo-thinkcentre-thinkstation-tiny-project-tinyminimicro-reference-thread.34925/ [Accessed 13 Feb. 2026].

Parrish, K. (2021). *Ethernet vs. Wi-Fi: Is It Really Better to Go Wireless?* [online] HighSpeedInternet.com. Available at: https://www.highspeedinternet.com/resources/ethernet-vs-wifi.

pfSense (n.d.). *Official pfSense Hardware, Appliances, and Security Gateways*. [online] www.pfsense.org. Available at: https://www.pfsense.org/products/ [Accessed 22 Jan. 2026].

programming.dev Community Members. (2026). *Homelab Hardware Choices - programming.dev*. [online] Programming.dev. Available at: https://programming.dev/post/43932823 [Accessed 19 Jan. 2026].

Protectli (2025). *Buyer's Guide - Protectli Knowledge Base*. [online] Protectli Knowledge Base. Available at: https://kb.protectli.com/buyers-guide/#workload [Accessed 21 Jan. 2026].

Protectli (2026). *VP2430 - 4x 2.5G Port Intel® N150*. [online] Protectli EU. Available at: https://eu.protectli.com/product/vp2430/ [Accessed 15 Feb. 2026].

Proxmox (2019a). *Careers*. [online] Proxmox. Available at: https://www.proxmox.com/en/about/about-us/careers [Accessed 12 Feb. 2026].

Proxmox (2019b). *Features*. [online] Proxmox. Available at: https://www.proxmox.com/en/products/proxmox-virtual-environment/features.

Proxmox (2019c). *Proxmox Virtual Environment*. [online] Proxmox. Available at: https://www.proxmox.com/en/products/proxmox-virtual-environment/overview.

Proxmox (2025a). *Network Configuration - Proxmox VE*. [online] pve.proxmox.com. Available at: https://pve.proxmox.com/wiki/Network_Configuration.

Proxmox (2025b). *Proxmox VE Administration Guide*. [online] Proxmox.com. Available at: https://pve.proxmox.com/pve-docs/pve-admin-guide.html#chapter_pct [Accessed 14 Feb. 2026].

Proxmox (2026). *Upgrade from 8 to 9 - Proxmox VE*. [online] Proxmox.com. Available at: https://pve.proxmox.com/wiki/Upgrade_from_8_to_9#In-place_upgrade [Accessed 28 Mar. 2026].

ProxmoxHHS (2019). *[TUTORIAL] - Proxmox Beginner Tutorial: How to set up your first virtual machine on a secondary hard disk*. [online] Proxmox Support Forum. Available at: https://forum.proxmox.com/threads/proxmox-beginner-tutorial-how-to-set-up-your-first-virtual-machine-on-a-secondary-hard-disk.59559/ [Accessed 28 Mar. 2026].

Quik Tech Solutions L.L.C (2025). *The 5 VLANs Every Home Network Should Have!* [online] YouTube. Available at: https://www.youtube.com/watch?v=JFvzBBUxRYM [Accessed 19 Jan. 2026].

r/fedora Community (2020). *Reddit - The heart of the internet*. [online] Reddit.com. Available at: https://www.reddit.com/r/Fedora/comments/g3h2bh/getting_started_with_coreos_ignition_file/ [Accessed 14 Feb. 2026].

r/homelab Community Members (2025). *r/homelab Wiki: Your Guide to Building a Homelab*. [online] Reddit.com. Available at: https://www.reddit.com/r/homelab/wiki/introduction/.

r/mikrotik Community Members (2023). *Reddit - The heart of the internet*. [online] Reddit.com. Available at: https://www.reddit.com/r/mikrotik/comments/13u7u6e/do_you_guys_think_mikrotik_wireless_aps_are_good/ [Accessed 13 Feb. 2026].

r/reolink Community Members (2024). *Reddit - The heart of the internet*. [online] Reddit.com. Available at: https://www.reddit.com/r/reolinkcam/comments/1gl0ow0/homeassistant_users_i_need_ideas_and_inspiration/ [Accessed 28 Mar. 2026].

Rampal, V. (2026). *Hardening a Raspberry Pi Trixie Edition - rampal.io*. [online] rampal.io. Available at: https://rampal.io/2026/01/hardening-a-raspberry-pi-trixie-edition/ [Accessed 28 Mar. 2026].

sysadmin102 (2023). *Create Let's Encrypt Wildcard Certificates on OPNsense with ACME Client – SYSADMIN102™*. [online] Sysadmin102.com. Available at: https://sysadmin102.com/2023/05/create-lets-encrypt-wildcard-certificates-on-opnsense-with-acme-client/ [Accessed 28 Mar. 2026].

sysadmin102 (2025a). *How to Automatically Deploy SSL Certificates to Proxmox on OPNsense with ACME Client – SYSADMIN102™*. [online] Sysadmin102.com. Available at: https://www.youtube.com/watch?v=Qz8e8iu4JNA [Accessed 28 Mar. 2026].

sysadmin102 (2025b). *OPNSense – Fix Realtek Ethernet NIC Issues – SYSADMIN102™*. [online] Sysadmin102.com. Available at: https://sysadmin102.com/2025/01/opnsense-fix-realtek-ethernet-nic-issues/ [Accessed 28 Mar. 2026].

sysadmin102 (2026). *Deploy SSL Let's Encrypt Certificates to Proxmox on OPNsense with ACME Client*. [online] Youtu.be. Available at: https://youtu.be/Qz8e8iu4JNA?si=YKxcq1b9fl0UHSR [Accessed 28 Mar. 2026].

Thomas-Krenn AG (2024). *Proxmox Upload ISO Image*. [online] Thomas-krenn.com. Available at: https://www.thomas-krenn.com/en/wiki/Proxmox_upload_ISO_image [Accessed 28 Mar. 2026].

Tkanov, I. (2025). *Proxmox Firewall Layers in Simple Terms – {IT}*. [online] {IT}. Available at: https://igortkanov.com/proxmox-firewall-layers-in-simple-terms/ [Accessed 28 Mar. 2026].

Uche, R. (2024). *How to Access Web Pages via SSH*. [online] Baeldung on Linux. Available at: https://www.baeldung.com/linux/ssh-tunnel-access-web-pages [Accessed 28 Mar. 2026].

Wikipedia Contributors (2021). *pfSense*. [online] Wikipedia. Available at: https://en.wikipedia.org/wiki/PfSense.

Wikipedia Contributors (2022). *TP-Link*. [online] Wikipedia. Available at: https://en.wikipedia.org/wiki/TP-Link.

Wikipedia Contributors (2025). *m0n0wall*. [online] Wikipedia. Available at: https://en.wikipedia.org/wiki/M0n0wall.

Wikipedia Contributors (2026). *Self-hosting (network)*. [online] Wikipedia. Available at: https://en.wikipedia.org/wiki/Self-hosting_(network).

Yoon, J. (2025). 120,000 Home Cameras Were Hacked for Sexual Videos, South Korean Police Say. *The New York Times*. [online] 2 Dec. Available at: https://www.nytimes.com/2025/12/02/world/asia/south-korea-cameras-hacked.html.

Yubico (2026). *Securing SSH with FIDO2*. [online] Yubico.com. Available at: https://developers.yubico.com/SSH/Securing_SSH_with_FIDO2.html [Accessed 28 Mar. 2026].

10 Use of AI Declaration

Following guidance from the project supervisor to use AI strictly in an editorial role, Microsoft Copilot was used during the preparation of this report to support editing, structural review, and word-count reduction, but not to generate any new technical content, analysis, or conclusions.

To ensure the Large Language Model (LLM) adhered to this role, I used opening prompts for all conversations that specifically outlined limitations for what Copilot should and should not do:

"I need you to act as my editor for the final report for my Higher Diploma in Computer Science. I will first provide you with our project handbook. It lays out exactly how our project, and project report should be carried out. The report here is meant to be the 8,000 words total, not counting the references or appendices, but it is currently x words too long. After you have reviewed the Project Handbook, please review my text with the project handbook and its description of how the report should be written in mind. Do not write anything for me, just act as my guide and editor. I will attach the handbook to this message."

Later, similar prompts were used to further reduce the wordiness of some passages:

"Shorten this without deviating from my original too much or changing the references or logic."

When on occasion Copilot diverged from this editorial role, the LLM was instructed once more not to generate any content and to return to the role of editor and simulated examiner.

All recommendations proposed by Copilot were reviewed and appraised to ensure that academic integrity was strictly adhered to.

Copilot and other AI platforms such as DuckDuckGo's duck.ai were also used during the research phase, primarily as a stand-in search engine to help find references about specific devices or technologies. They were also used to simplify text from references for better understanding using prompts like: *"Explain this to me as if I am a beginner at networking"*.

Though an experiment was conducted to see whether LLMs could compose hardware comparison tables based on their own search findings, this was abandoned after the author found multiple incorrect specifications listed.

Prior to using Copilot, in accordance with the open-source and self-hosted context of the project, an attempt was made to use open-source local LLMs like Llama 3.1 and Qwen 3 via Ollama and the Open WebUI frontend. However, as the machine this was attempted on only had 16 GB of RAM and 16 GB of VRAM available, only low-parameter models of around 8B

parameters could be used and these proved more prone to frequent hallucinations and loss of context (such as the project handbook or draft report).

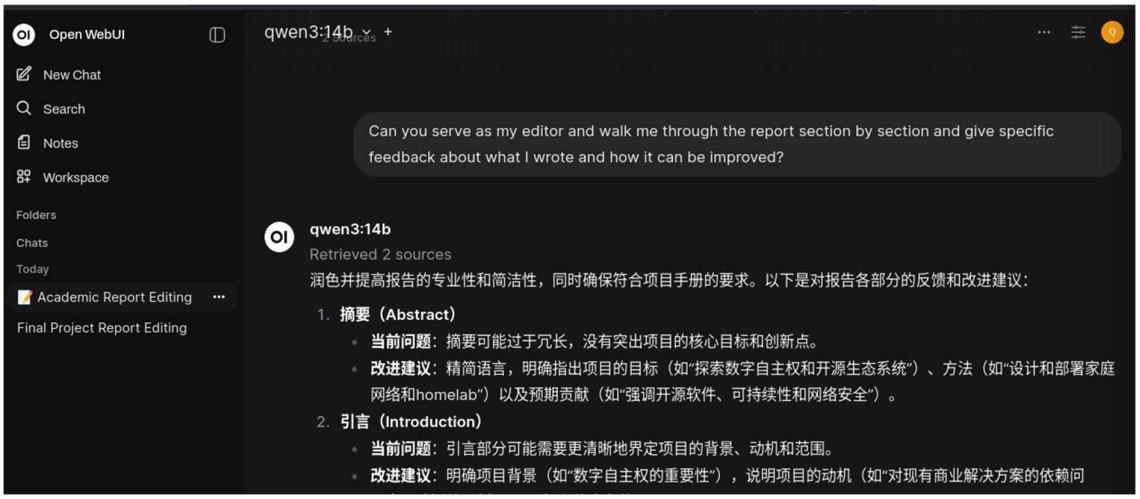

Figure 23: An example of a malfunction - Qwen 3 switched to Chinese

Google's Gemini 3 was also used to troubleshoot some issues, such as when errors occurred during the switchover from Docker to Podman for the container workloads. This worked relatively well, but Gemini seemed to be very prone to losing context entirely, sometimes within 1-2 messages from the start of a conversation. This could however have been due to not being logged in at the time.

Overall, AI was useful to me in finding references and articles to read and perhaps most useful as an editorial assistant. However, though I use AI in this capacity often, personally I feel that this offloading of cognitive load seems to come with a penalty a subtle atrophying of knowledge and ability, and for this reason, I aim to reduce my use of it in less time-sensitive situations.

11 Appendices

Appendix A.1 Overview

This appendix contains detailed hardware evaluation tables which were composed during the selection process for the hardware router/firewall. These tables support the summary presented in Section 3.4 and provide further technical detail on the rationale behind the decision.

Appendix A.2 Deciso Firewall Models

Table 3: Deciso Firewall Model Comparison

MODEL	CPU	Memory	Internal Storage	Typical Applications	Threat Protection Throughput	Price
DEC677	AMD G-Series SOC 4 (max frequency 1.8Ghz)	4GB DDR3	32GB Solid State Flash [integrated uSD]	Firewall / Routing & VPN	N/A	€588.00
DEC697	AMD G-Series SOC, 4 (max frequency 1.8Ghz)	8GB DDR3	256GB NVMe Solid State Flash	Firewall / Routing, VPN, IDPS & Webproxy	~540Mbps	€678.00

(Deciso B.V., 2022)

Appendix A.3 Deciso vs Protectli

The Protectli Shop allows interested buyers to configure the amount of RAM, additional hard drives, the type of BIOS (AMI or coreboot) as well as features like an optional Trusted Platform Module (TPM), but for comparison purposes, the VP2430 was configured as close to the Deciso DEC697 as possible, with only 8GB of RAM and no additional storage.

Table 4: Deciso vs Protectli Firewall Comparison

MODEL	CPU	Memory	Internal Storage	Typical Applications	Threat Protection Throughput	Price
VP2430	Intel® N150 Quad Core CPU (6MB Cache, up to 3.6GHz)	8GB Crucial DDR5	32GB eMMC + 256GB Protectli NVMe	Firewall / Routing, VPN, IDPS & Webproxy	? (Not advertised by Protectli)	€623.61
DEC697	AMD G-Series SOC, 4 (max frequency 1.8Ghz)	8GB DDR3	256GB NVMe Solid State Flash	Firewall / Routing, VPN, IDPS & Webproxy	~540Mbps	€678.00

(Deciso B.V., 2022; Protectli, 2026)