

Table of Contents

<i>PiNAS v2 - Design Progress (HDD Cage & Caddies Validated)</i>	3
<i>When It's Not the Network: An RDP Investigation That Led Elsewhere</i>	4
<i>Rediscovering Engineering Through 3D Printing</i>	6
<i>Script</i>	7
<i>IPv4 Rules</i>	7
<i>Let's Talk - Building a Modular PHP Framework part 3</i>	9
<i>Let's Talk - Building a Modular PHP Framework part 2</i>	10
<i>Let's Talk - Why and How I Use Artificial Intelligence</i>	11

Home



PiNAS v2 - Design Progress (HDD Cage & Caddies Validated)

Author(s): Louis Ouellet

PiNAS is back — and this time, it's more than just an iteration.

Over the past few weeks, I've been deep into redesigning my Raspberry Pi-based NAS from the ground up. What started as a simple idea quickly turned into a series of design challenges, print tests, and unexpected lessons.

This article isn't just about what works — it's about what didn't, what changed, and why.

[Read more](#)

2026/03/24 20:18 · Louis Ouellet · 0 Comments

[3d-printing](#), [project](#), [raspberrypi](#), [pinas](#), [raspberry pi](#), [nas](#), [homelab](#), [3d printing](#), [3d design](#), [diy](#), [hardware design](#), [prototyping](#), [modular design](#), [storage](#), [sata](#)



When It's Not the Network: An RDP Investigation That Led Elsewhere

Author(s): Louis Ouellet

In a recent deployment, I was asked to investigate unstable Remote Desktop (RDP) sessions to a remote server accessed over a site-to-site VPN.

At first, the explanation sounded simple: the VPN was unstable. Users were being disconnected, the target server was remote, and all the symptoms seemed to point in the same direction.

On paper, the environment was straightforward:

- Local network: 192.168.115.0/24
- Remote network: 192.168.201.0/24
- Original VPN: IPsec
- Temporary replacement VPN: WireGuard
- Target server: 192.168.201.100
- Multiple users connecting via RDP

But as is often the case in infrastructure work, the first explanation turned out to be the most convenient one — not the most accurate.



This is one of those cases where everything looks like a network issue — until you start proving what is, and is not, actually failing.

[Read more](#)

2026/03/22 20:36 · Louis Ouellet

[general](#), [blog](#), [rdp](#), [network](#), [wireguard](#), [investigation](#), [security](#), [windows](#), [linux](#)



Rediscovering Engineering Through 3D Printing

Author(s): Louis Ouellet

I didn't buy a 3D printer to print toys. I bought it to prototype systems.

In early October, I added a BambuLab P1S to my lab. Most people start with a Benchy — a calibration cube, a cable clip, something small and forgiving. I opened FreeCAD and started designing a NAS enclosure from scratch.

Probably not the most subtle first print.

But I wasn't interested in printing trinkets. I wanted constraints. I wanted tolerances. I wanted airflow questions, structural rigidity concerns, screw alignment problems, and the kind of iterative refinement that forces you to think before you click "print." I wanted to push engineering muscles I hadn't used since high school — the ones that make you consider load

distribution, clearances, material thickness, and assembly order.

For years, most of what I build has been invisible: servers, scripts, infrastructure diagrams, automation pipelines. Systems that live in racks or in the cloud. With 3D printing, for the first time in a long time, I could design something in the morning and physically hold the result in the evening.

Something functional. Something structural. Something real.

That decision — to start with a NAS enclosure instead of a novelty print — set the tone for everything that followed.

[Read more](#)

2026/02/14 12:22 · Louis Ouellet
[general](#), [blog](#), [p1s](#), [3d-printing](#), [diy-kits](#)

Script

[Read more](#)

2025/10/13 12:16 · Louis Ouellet · [0 Comments](#)

IPv4 Rules

```
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT ACCEPT [0:0]

# Always allow loopback
-A INPUT -i lo -j ACCEPT

# Allow established/related to talk back in
-A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
-A FORWARD -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT

# ICMP (ping) from LAN/DMZ (adjust to taste)
-A INPUT -i br2 -p icmp -j ACCEPT
-A INPUT -i br3 -p icmp -j ACCEPT
-A INPUT -i br4 -p icmp -j ACCEPT
```

```
# SSH to the router from LAN only (adjust/lock down as needed)
-A INPUT -i br2 -p tcp --dport 22 -j ACCEPT
-A INPUT -i br4 -p tcp --dport 22 -j ACCEPT

# DNS & DHCP to the router from LAN/DMZ (dnsmasq)
-A INPUT -i br2 -p udp --dport 67:68 -j ACCEPT
-A INPUT -i br3 -p udp --dport 67:68 -j ACCEPT
-A INPUT -i br4 -p udp --dport 67:68 -j ACCEPT
-A INPUT -i br2 -p tcp --dport 53 -j ACCEPT
-A INPUT -i br2 -p udp --dport 53 -j ACCEPT
-A INPUT -i br3 -p tcp --dport 53 -j ACCEPT
-A INPUT -i br3 -p udp --dport 53 -j ACCEPT
-A INPUT -i br4 -p tcp --dport 53 -j ACCEPT
-A INPUT -i br4 -p udp --dport 53 -j ACCEPT

# Forwarding policy:
# - LAN -> WAN: allow
# - DMZ -> WAN: allow
# - WLAN -> WAN: allow
# - WAN -> LAN/DMZ: block unless established/related
# - LAN <-> DMZ: default block (tight). Uncomment the next line if you
want LAN to reach DMZ.
-A FORWARD -i br2 -o br0 -j ACCEPT
-A FORWARD -i br4 -o br0 -j ACCEPT
-A FORWARD -i br3 -o br0 -j ACCEPT
# Allow LAN to reach DMZ (optional)
-A FORWARD -i br2 -o br3 -j ACCEPT
-A FORWARD -i br4 -o br3 -j ACCEPT
# Allow LAN and WLAN to reach each other (optional)
-A FORWARD -i br2 -o br4 -j ACCEPT
-A FORWARD -i br4 -o br2 -j ACCEPT

COMMIT

*nat
:PREROUTING ACCEPT [0:0]
:INPUT ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]

# NAT (masquerade) LAN+DMZ out of WAN
-A POSTROUTING -o br0 -j MASQUERADE

# Example: Port-forward 80 on WAN to a DMZ host 192.168.60.10
```

```
# (and allow the forward)
#-A PREROUTING -i eth0 -p tcp --dport 80 -j DNAT --to-destination
192.168.60.10:80
#-A FORWARD -i eth0 -p tcp -d 192.168.60.10 --dport 80 -j ACCEPT

COMMIT
```

```
# manual
sudo iptables -I FORWARD 1 -i br1 -o eth0 -j ACCEPT
sudo iptables -I INPUT 1 -i br1 -p udp --dport 67:68 -j ACCEPT
sudo iptables -I INPUT 1 -i br1 -p udp --dport 53 -j ACCEPT
sudo iptables -I INPUT 1 -i br1 -p tcp --dport 53 -j ACCEPT
sudo iptables -I INPUT 1 -i br1 -p icmp -j ACCEPT
sudo iptables -I INPUT 1 -i br1 -p tcp --dport 22 -j ACCEPT
sudo netfilter-persistent save
```

2025/09/17 20:23 · Louis Ouellet · [0 Comments](#)



Let's Talk - Building a Modular PHP Framework part 3

Author(s): Louis Ouellet

In this third installment of our **Building a Modular PHP Framework** series, we will cover:

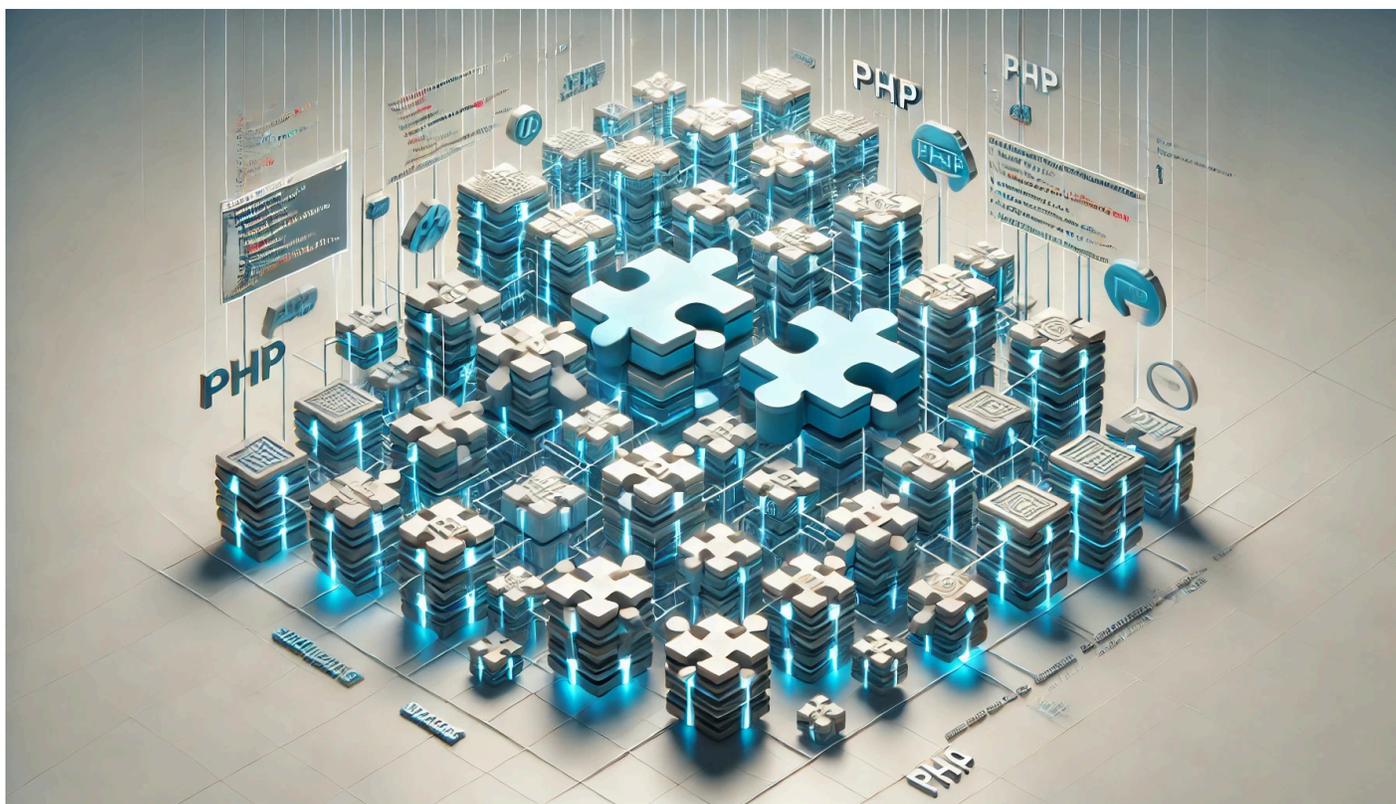
- Cross-Site Request Forgery (CSRF)
- API and Endpoint creation
- Database Handling including:
 - Managing the database structure using Schema
 - Managing database queries using Query

We'll build upon the groundwork set up in the previous parts, focusing on security (CSRF), setting up API routes, and more advanced database handling.

[Read more](#)

2025/02/03 16:18 · Louis Ouellet

[general](#), [core-framework-en](#), [framework](#), [core](#), [modularity](#), [modular](#), [module](#), [lightweight](#), [application](#), [php](#), [cakephp](#), [symfony](#), [learn](#), [leaning](#), [maintainability](#), [maintain](#)



Let's Talk - Building a Modular PHP Framework part 2

Author(s): Louis Ouellet

Time for part 2! In our previous part, we set up the groundwork of our modular PHP framework. This time, we will focus on expanding its capabilities to support the following objectives:

- Add support for extensions
- Begin implementing a Command-Line module
- Add support for models to be used to create shared methods that require a database
- Add support for helpers to be used to create shared methods that do not require a database

These enhancements will provide the flexibility we need to build modular, maintainable, and extensible applications. Let's walk through each update step by step.

[Read more](#)

2025/01/28 08:53 · Louis Ouellet

[general](#), [core-framework-en](#), [framework](#), [core](#), [modularity](#), [modular](#), [module](#), [lightweight](#), [application](#), [php](#), [cakephp](#), [symfony](#), [learn](#), [leaning](#), [maintainability](#), [maintain](#)



Let's Talk - Why and How I Use Artificial Intelligence

Author(s): Louis Ouellet

So, if you haven't lived under a rock, you should know by now about the existence of AI (Artificial Intelligence) and LLMs (Large Language Models). They have become integral parts of many creative and technical workflows, and they're here to stay.

[Read more](#)

2025/01/27 16:46 · Louis Ouellet

[general](#), [ai](#), [llm](#), [artificial intelligence](#), [large language modal](#)

[Older entries >>](#)

From:

<https://laswitchtech.com/> - **LaswitchTech**

Permanent link:

<https://laswitchtech.com/index>

Last update: **2024/10/30 20:37**

