

The logo for CDLAN, featuring the letters 'CDLAN' in white. The letter 'A' is stylized with a red outline and a red dot above it.

CDLAN

Integrating Wireguard
into Cloudstack VR

speaker



Marco Ziglioli

Circle Lead Cloud

About **CDLAN**

We fuel the growth and competitiveness of companies with a robust network infrastructure and two state-of-the-art data centers. Our hallmark is an **unwavering commitment to customer focus and clear, direct offerings.**

We view digitalization not just as a mere trend but as an accelerator of technological and cultural transformations. We trust **human relationships** because we believe they are the lifeblood of technological innovation.

Proud owners of **C21, the Tier IV Compliant Data Center** located in the heart of Caldera Park, Milan, and **E100**, our Data Center in Rome.



Our services



DATA CENTER



CLOUD



CONNECTIVITY



VOICE



CYBER SECURITY



Current VPN Integration



IPsec is a network protocol used for the encryption of IP traffic.

IPsec is frequently used as the secure communication protocol for business VPNs, most commonly with a tunneling protocol like L2TP.

IPsec is supported on many operating systems and device types, including embedded devices and network routers.

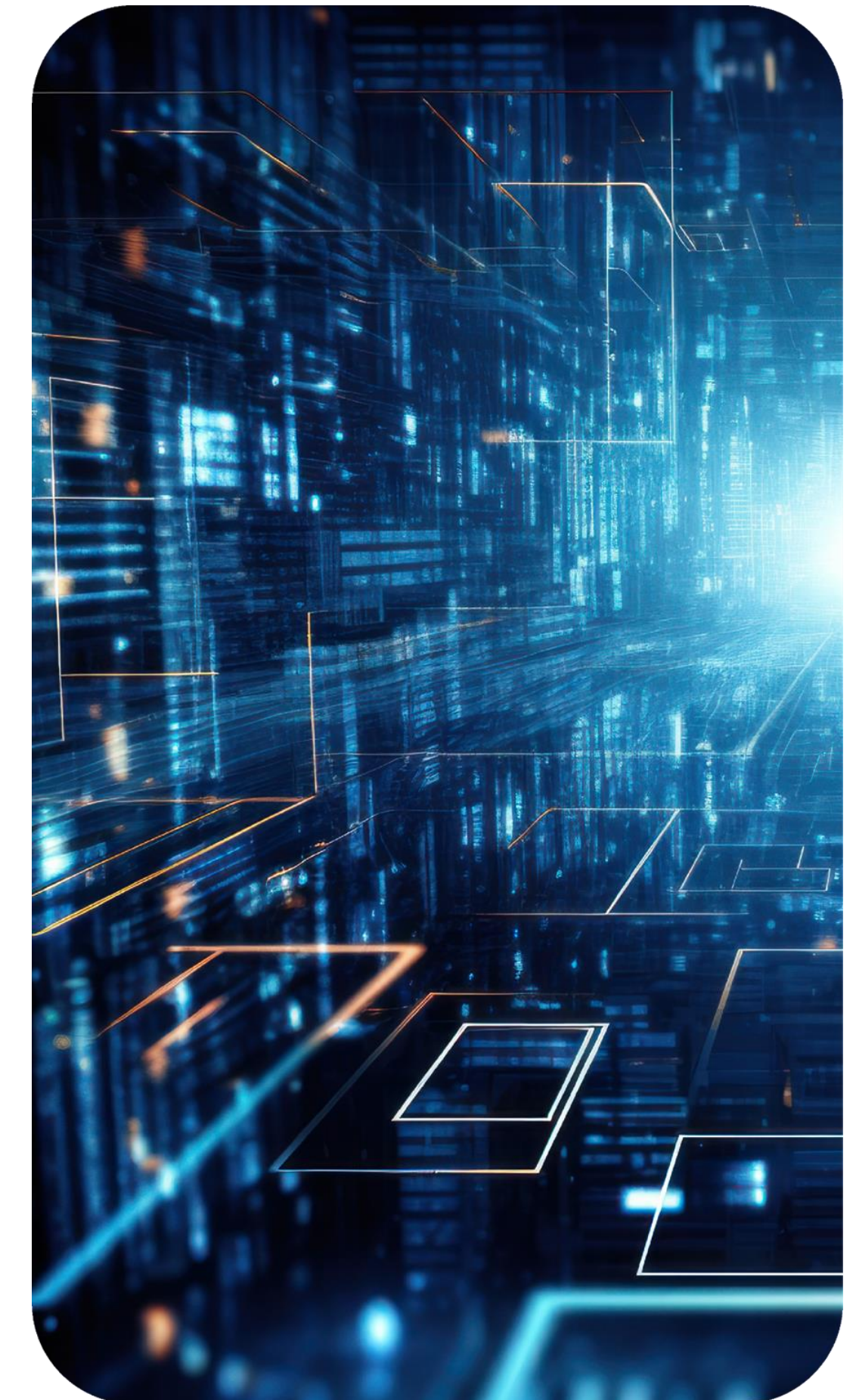
- site2site IPsec VPN in VPC networks
- L2TP over IPSEC for “road-warrior” VPN in isolated networks

What is L2TP?

L2TP is a tunneling protocol, often used to support VPNs, which encapsulates data for secure transmission over public networks.

L2TP (Layer 2 Tunneling Protocol) works by encapsulating data packets within a tunnel over a network. Since the protocol does not inherently encrypt data, it relies on IPsec (Internet Protocol Security) for confidentiality, integrity, and authentication of the data packets traversing the tunnel. This combination, known as L2TP/IPsec, is widely adopted for its enhanced security measures.

Since Layer 2 Tunneling Protocol does not offer encryption by itself, **its primary role is to create a tunnel for data to pass through securely**. The security of the data within this tunnel relies entirely on IPsec. The combination provides a dual layer of protection by first creating a tunnel and then securing the data with encryption.



What is IPsec?

In computing, **Internet Protocol Security (IPsec)** is a secure network protocol suite that authenticates and encrypts packets of data to provide secure encrypted communication between two computers over an Internet Protocol network. It's commonly used in virtual private networks.

IPsec includes protocols for establishing mutual authentication between agents at the beginning of a session and negotiation of cryptographic keys to use during the session. IPsec can protect data flows between a pair of hosts (**host-to-host**), between a pair of security gateways (**network-to-network**), or between a security gateway and a host (**network-to-host**).

What is Wireguard?

WireGuard is a modern **VPN protocol** that is simple to use and easy to implement on both new and existing networks. WireGuard is free and open-source, and WireGuard implementations are available for major operating systems.

A VPN connection is made simply by exchanging very simple public keys – exactly like exchanging SSH keys – and all the rest is transparently handled by WireGuard.

WireGuard uses **state-of-the-art cryptography**, like the Noise protocol framework, Curve25519, ChaCha20, Poly1305, BLAKE2, SipHash24, HKDF, and secure trusted constructions. It makes conservative and reasonable choices and has been reviewed by cryptographers.



Difference with IPsec?

Comparison table (<https://tailscale.com/compare/ipsec#overview-of-ipsec>)

	IPsec	WireGuard
Open source	Yes	Yes
End-to-end encryption	Yes	Yes
Encryption options	Many encryption options present the possibility of using insecure settings	Fewer encryption options, focused on modern encryption solutions with more secure defaults
Key change	Uses Internet Key Exchange (IKE)	Uses Noise Protocol
Maintains an active connection	Yes	No

Why should we change?

- 1 IPsec can be insecure if incorrectly configured while WireGuard limits the available choices to modern, secure encryption methods.
- 2 IPsec supports using the RSA algorithm and pre-shared keys for authentication which are no longer considered secure.
- 3 IPsec is quite slow, 15% slower than WireGuard and have 20% more latency.

Why choose WireGuard?



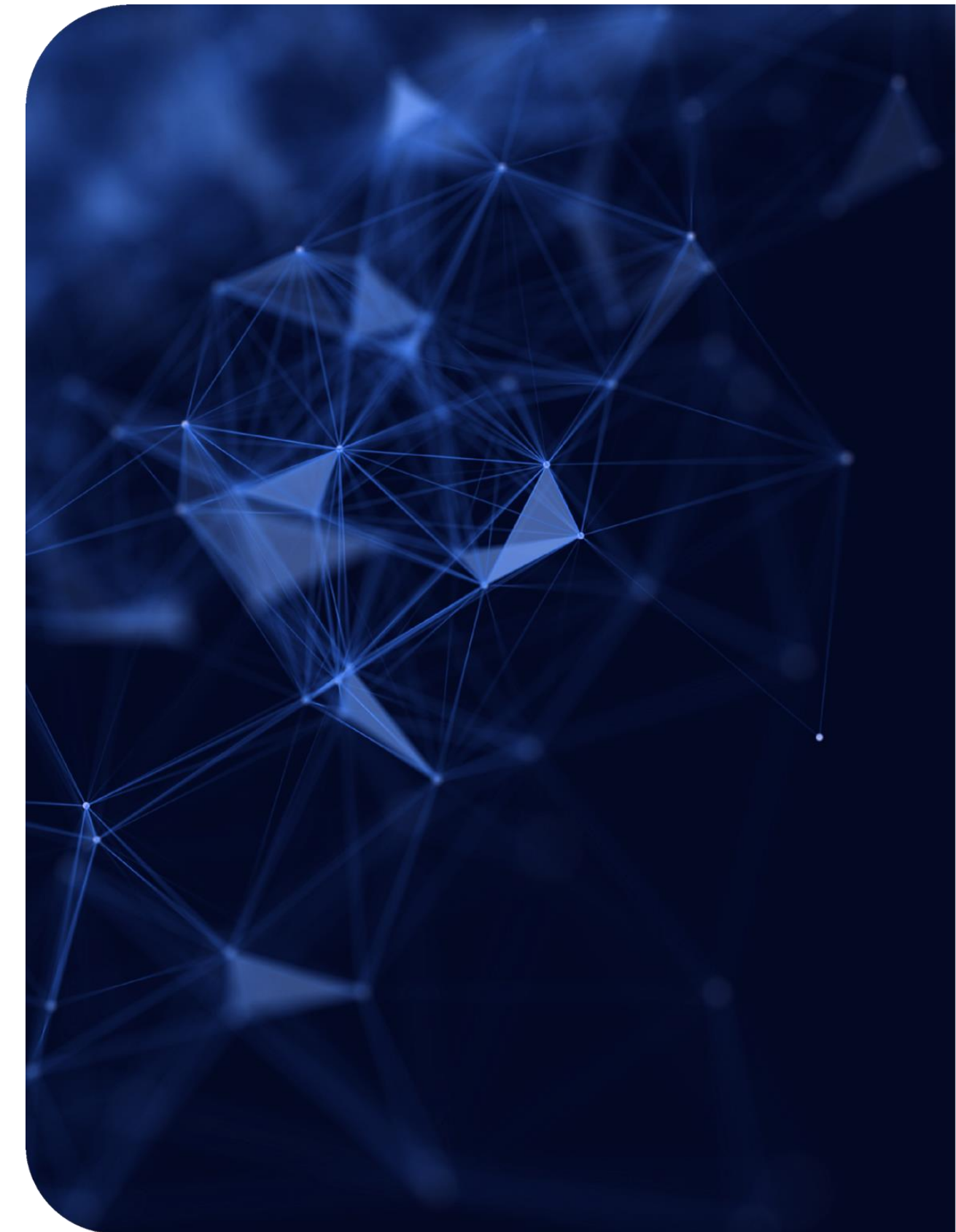
WireGuard is currently one of the fastest VPN protocols on the market due to its encryption algorithms as well as the less overall code that goes into WireGuard.



It also utilizes ChaCha20 for encryption which is substantially more modern and faster than IPsec used by L2TP or other algorithm like AES-256 used by protocols like OpenVPN.



The smaller codebase of Wireguard also reduces the overall complexity and makes it much easier to use because the configuration process is not that complicated. This makes it also a much better option for those who are not that familiar with VPN networks.



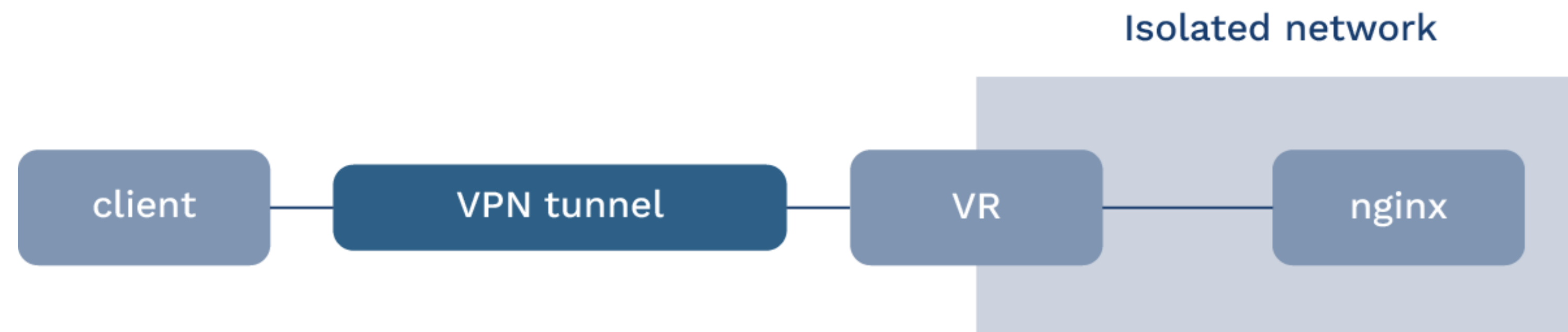
Test: a simple http file transfer (setup)

We made a simple test by downloading a 50G file from an nginx server in a VXLAN isolated network.

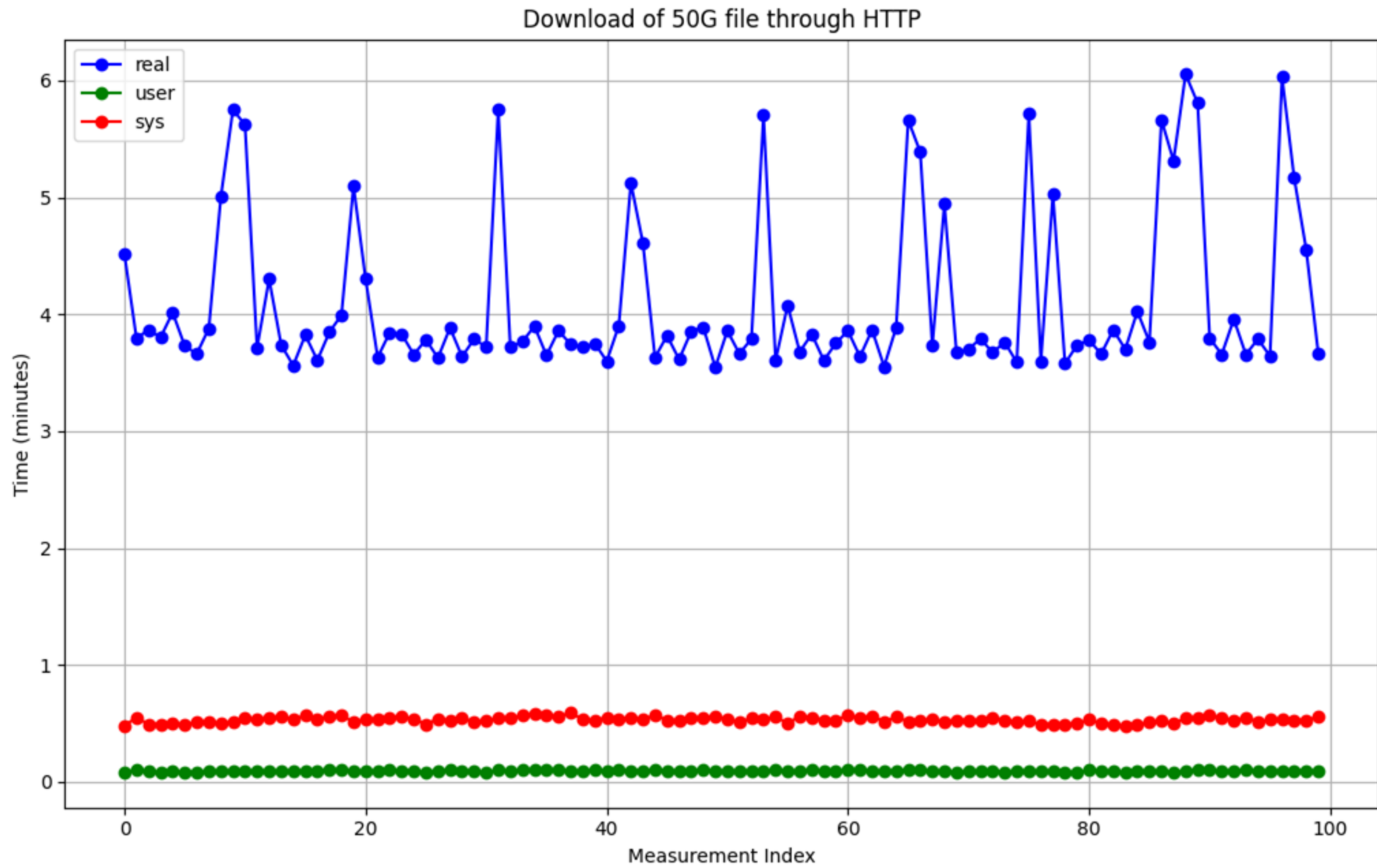
We registered the download times in the following scenarios:

- Client inside the isolated network (LAN)
- Client connected through IPsec to the isolated network's VR (IPS)
- Client connected through Wireguard to the isolated network's VR (WG)

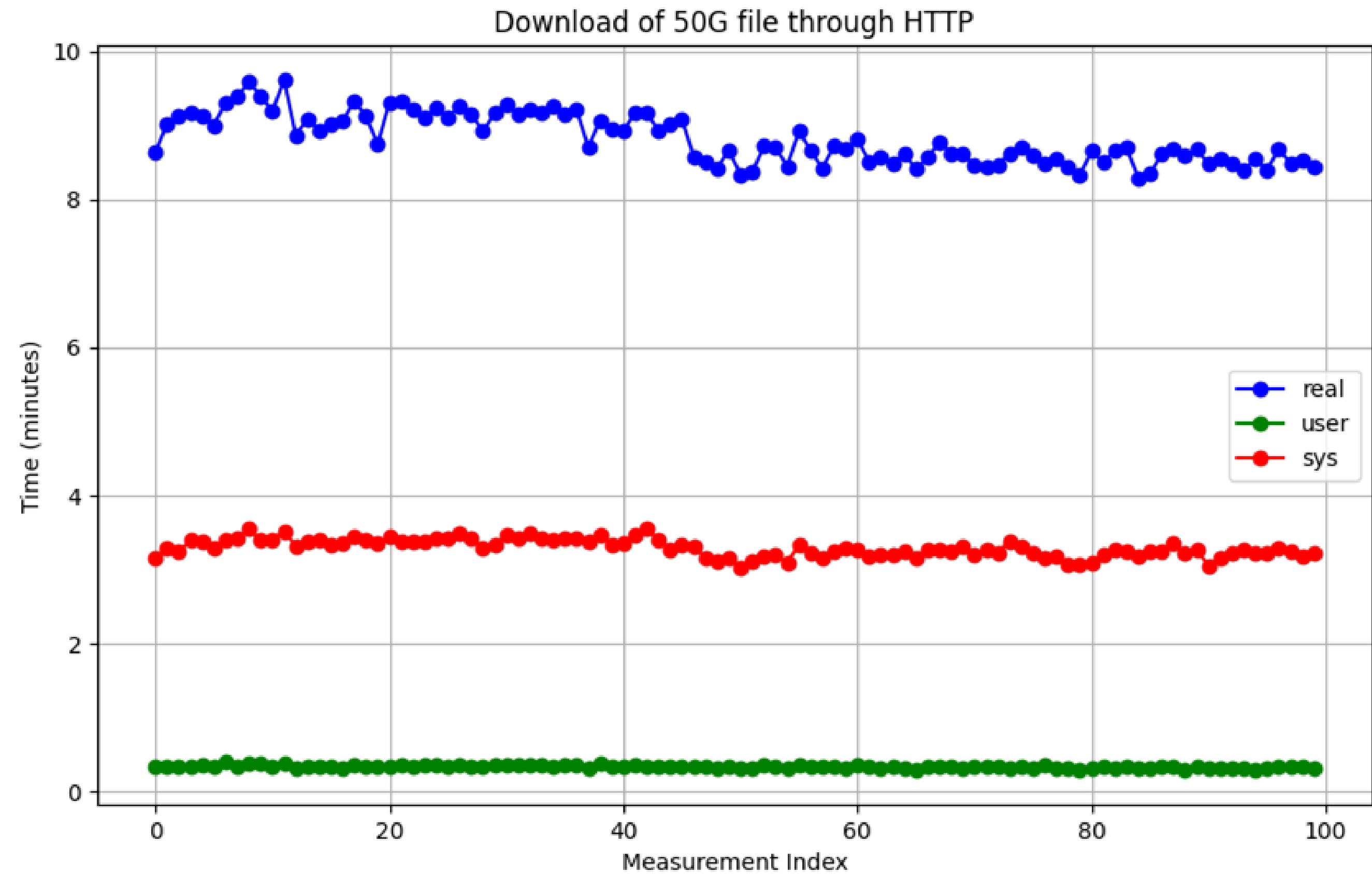
All the VMs (VR included) are hosted on the same host in order to minimize the effect of the physical network.



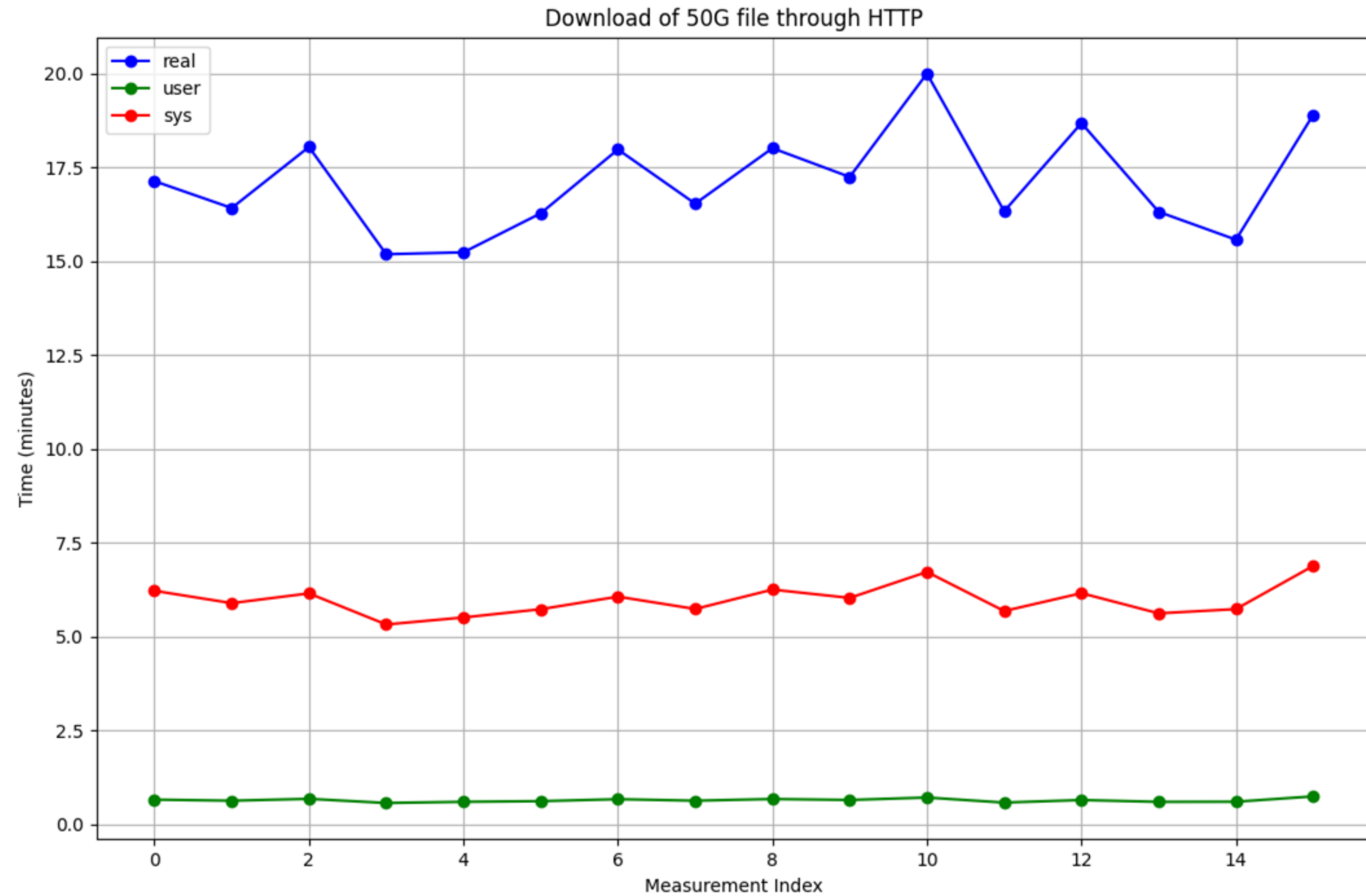
Test: a simple http file transfer (LAN)



Test: a simple http file transfer (WireGuard)

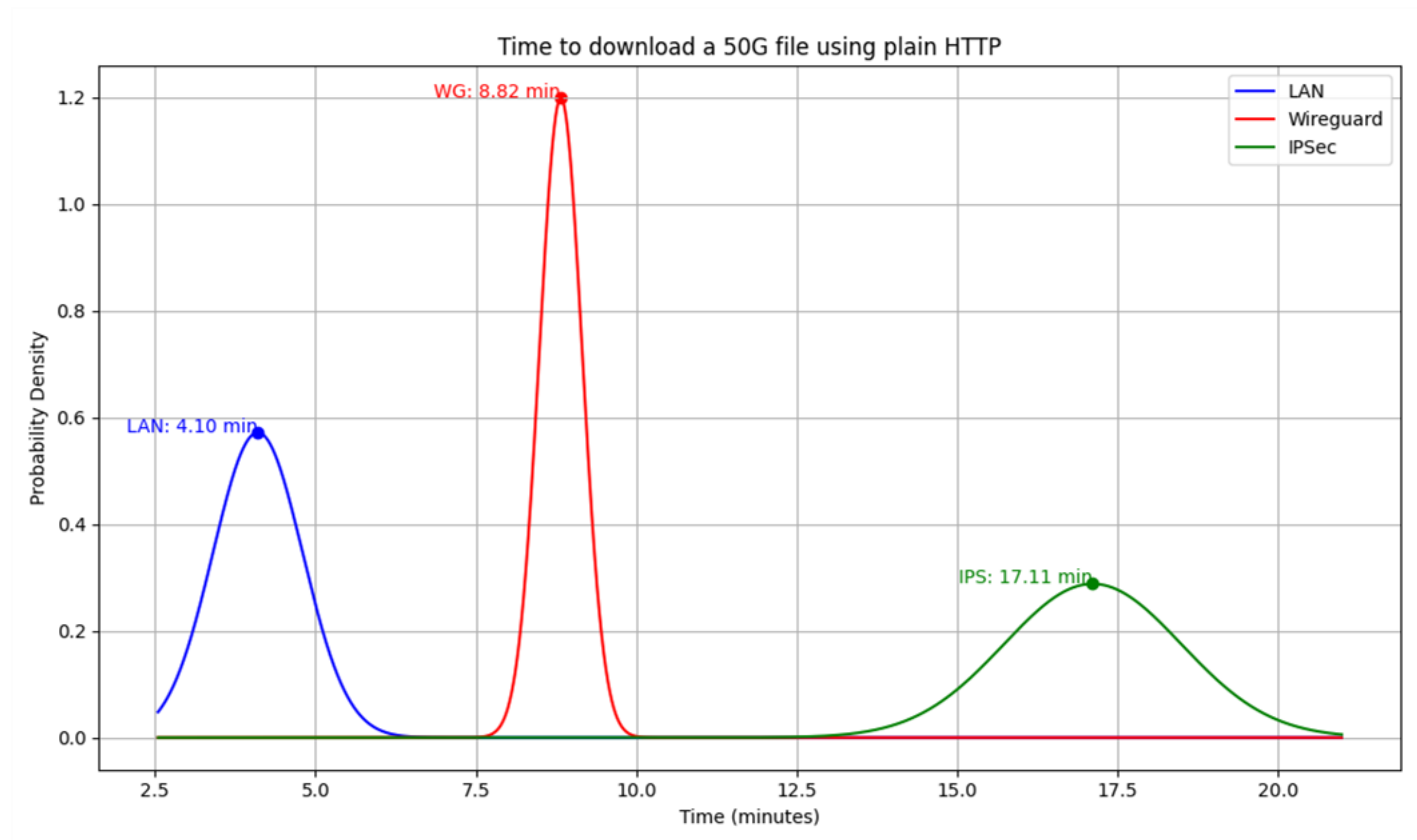


Test: a simple http file transfer (IPsec)



Test: a simple http file transfer (comparing real times)

The data is represented using normal distributions to highlight key statistical properties such as the mean transfer speed, variability, and overlap between the configurations.



Some WireGuard UIs examples we found (1)

<https://github.com/ngoduykhanh/wireguard-ui>



WIREGUARD UI

vpn_ui_administrator

MAIN

- Wireguard Clients
- Wireguard Server

SETTINGS

- Global Settings

UTILITIES

- Status
- WoL Hosts

Wireguard Clients

+ New Client ✓ Apply Config ↶ Logout

<p>Download QR code Email More ▾</p> <p>Grigory S (mobile)</p> <p>📧</p> <p>🕒 2022/07/03 14:47:24</p> <p>🕒 2022/07/03 14:47:24</p> <p>☰ DNS enabled</p> <p>IP Allocation</p> <p>10.0.0.1/32</p> <p>Allowed IPs</p> <p>0.0.0.0/0</p>	<p>Download QR code Email More ▾</p> <p>Grigory S (laptop, linux)</p> <p>📧</p> <p>🕒 2022/07/03 14:47:43</p> <p>🕒 2022/07/03 14:47:43</p> <p>☰ DNS enabled</p> <p>IP Allocation</p> <p>10.0.0.2/32</p> <p>Allowed IPs</p> <p>0.0.0.0/0</p>	<p>Download QR code Email More ▾</p> <p>Alexey S (laptop, windows)</p> <p>📧</p> <p>🕒 2022/07/03 15:21:18</p> <p>🕒 2022/07/03 15:21:18</p> <p>☰ DNS enabled</p> <p>IP Allocation</p> <p>10.0.0.3/32</p> <p>Allowed IPs</p> <p>0.0.0.0/0</p>
<p>Download QR code Email More ▾</p> <p>Alexey S (mobile)</p> <p>📧</p> <p>🕒 2022/07/03 15:21:42</p> <p>🕒 2022/07/03 15:21:42</p> <p>☰ DNS enabled</p> <p>IP Allocation</p> <p>10.0.0.4/32</p> <p>Allowed IPs</p> <p>0.0.0.0/0</p>	<p>Download QR code Email More ▾</p> <p>Zhahangir (mobile)</p> <p>📧</p> <p>🕒 2022/07/03 16:04:41</p> <p>🕒 2022/07/03 16:04:41</p> <p>☰ DNS enabled</p> <p>IP Allocation</p> <p>10.0.0.5/32</p> <p>Allowed IPs</p> <p>0.0.0.0/0</p>	<p>Download QR code Email More ▾</p> <p>Zhahangir (laptop, windows)</p> <p>📧</p> <p>🕒 2022/07/03 16:05:04</p> <p>🕒 2022/07/03 16:05:04</p> <p>☰ DNS enabled</p> <p>IP Allocation</p> <p>10.0.0.6/32</p> <p>Allowed IPs</p> <p>0.0.0.0/0</p>

Copyright © 2022 Wireguard UI. All rights reserved. Version v0.3.7

Some WireGuard UIs examples we found (2)

<https://www.wireguardconfig.com/>

Wireguard Config Generator

This tool is to assist with creating config files for a WireGuard 'road-warrior' setup whereby you have a server and a bunch of clients. Simply enter the parameters for your particular setup and click Generate Config to get started.

All keys, QR codes and config files are generated client-side by your browser and are never seen by our server.

Random Seed

G/U2QGwPnZuRmz5yQUugUH9xa3QvyFn4avIPDoVvnaF8wYBZwKvDtp4q90cKIPivfunQVhgNwkt07vvutaDEC6hvWhjlpGimYB2nEGwGTcqsRCO/SzJhq65c

Listen Port

51820

Number of Clients

3

CIDR

10.0.0.0/24

Client Allowed IPs

0.0.0.0/0, ::/0

Endpoint (Optional)

myserver.dyndns.org:51820

DNS (Optional)

Post-Up rule

iptables -A FORWARD -i %i -j ACCEPT; iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE

Post-Down rule

iptables -D FORWARD -i %i -j ACCEPT; iptables -t nat -D POSTROUTING -o eth0 -j MASQUERADE

Use Pre-Shared Keys (Enhanced Security)

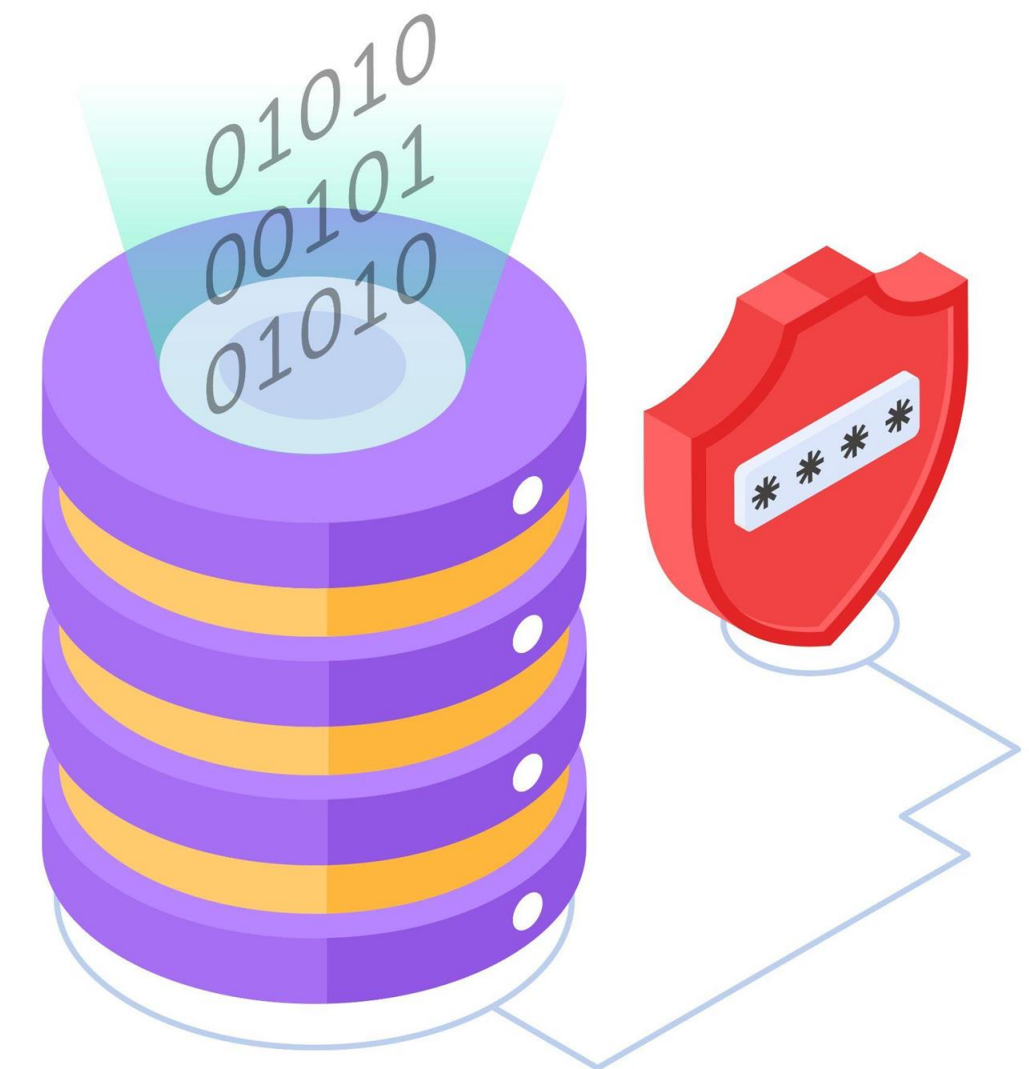
Generate Config

Our target

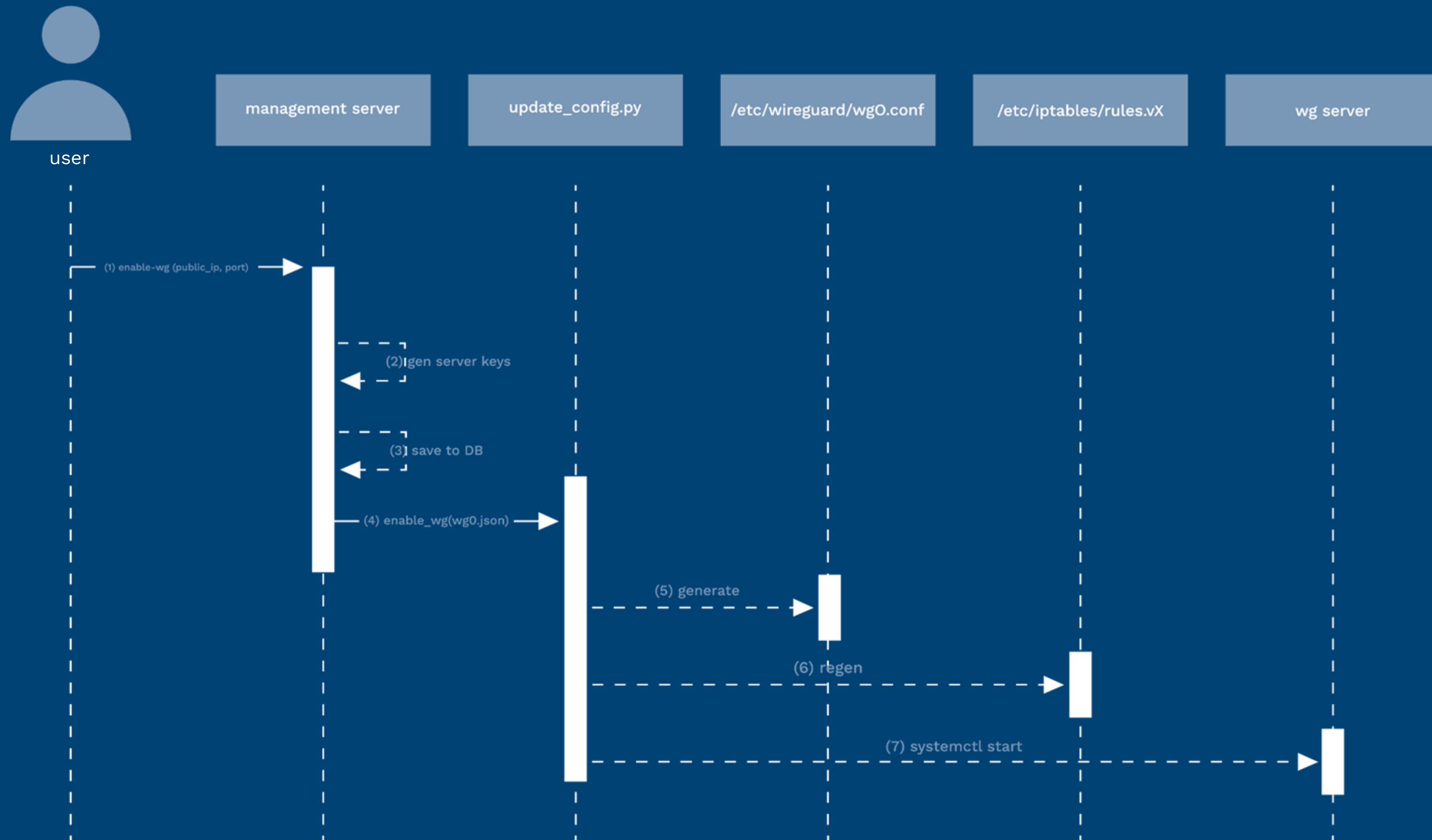
Our idea is to implement Wireguard as **another VPN option** alongside IPsec.

The implementation of the configuration for the user must be as simple as possible, giving the possibility to have a working configuration in few steps even to those who are not very experienced in creating and configuring a VPN.

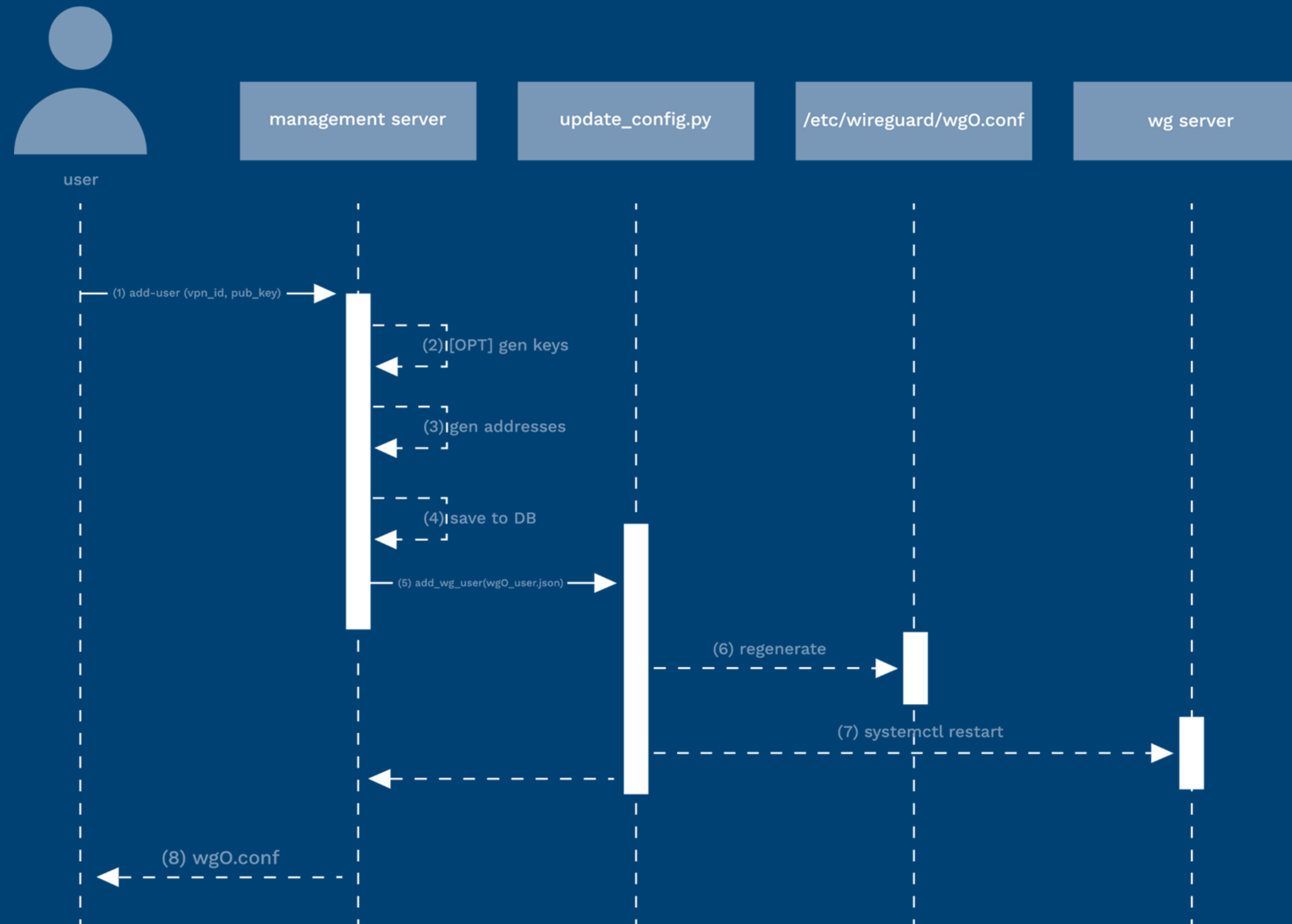
Directly from the ACS UI the user will be able to create and manage VPNs and related users (wireguard peers) in just a few clicks.



High Level Idea (tunnel creation)



High Level Idea (user creation)



Proposal: New APIs to manager WG in ACS

APIs:



● {Create|List|Delete}WgVpn



● {Create|List|Delete}WgUser

API: CreateWgVpn

Name	Type	Default	Notes
public_ip_id*	int64		public ip address id of the vpn server
ip4_enable	bool	true	
ip4_range	string		IPv4 network (CIDR)
ip6_enable	bool	false	
ip6_range	string		IPv6 network (CIDR)
open_firewall	bool	true	If firewall rule for source/end public port is automatically created
for_display	bool	true	
account_id	int64		
domain_id	int64		

API: ListWgVpn (request)

Name	Type	Default	Notes
id	int64		List wireguard vpn with the specified ID
network_id	int64		
page	int		
page_size	int		
public_ip_id	int64		Public ip address id of the vpn server
domain_id	int64		
account_id	int64		
list_all	bool	false	If set to false, list only resources belonging to the command's caller

API: DeleteWgVpn

Name	Type	Default	Notes
id*	int64		Id of the vpn to delete

API: CreateWgUser

Name	Type	Default	Notes
vpn_id*	int64		id of the wireguard vpn for this user
public_key	string		public key for the new user, if not provided will be generated
domain_id	int64		
account_id	int64		
for_display	bool	true	
split_tunnel	bool	true	if false, user will have all its traffic routed through the VPN

The response will contain the config file (and perhaps the qr code for mobile users).

API: ListWgUser (request)

Name	Type	Default	Notes
id	int64		List wireguard vpn with the specified ID
vpn_id	int64		List all user for given wg_vpn
page	int		
page_size	int		
domain_id	int64		
account_id	int64		
list_all	bool	false	If set to false, list only resources belonging to the command's caller

API: ListWgUser (response)

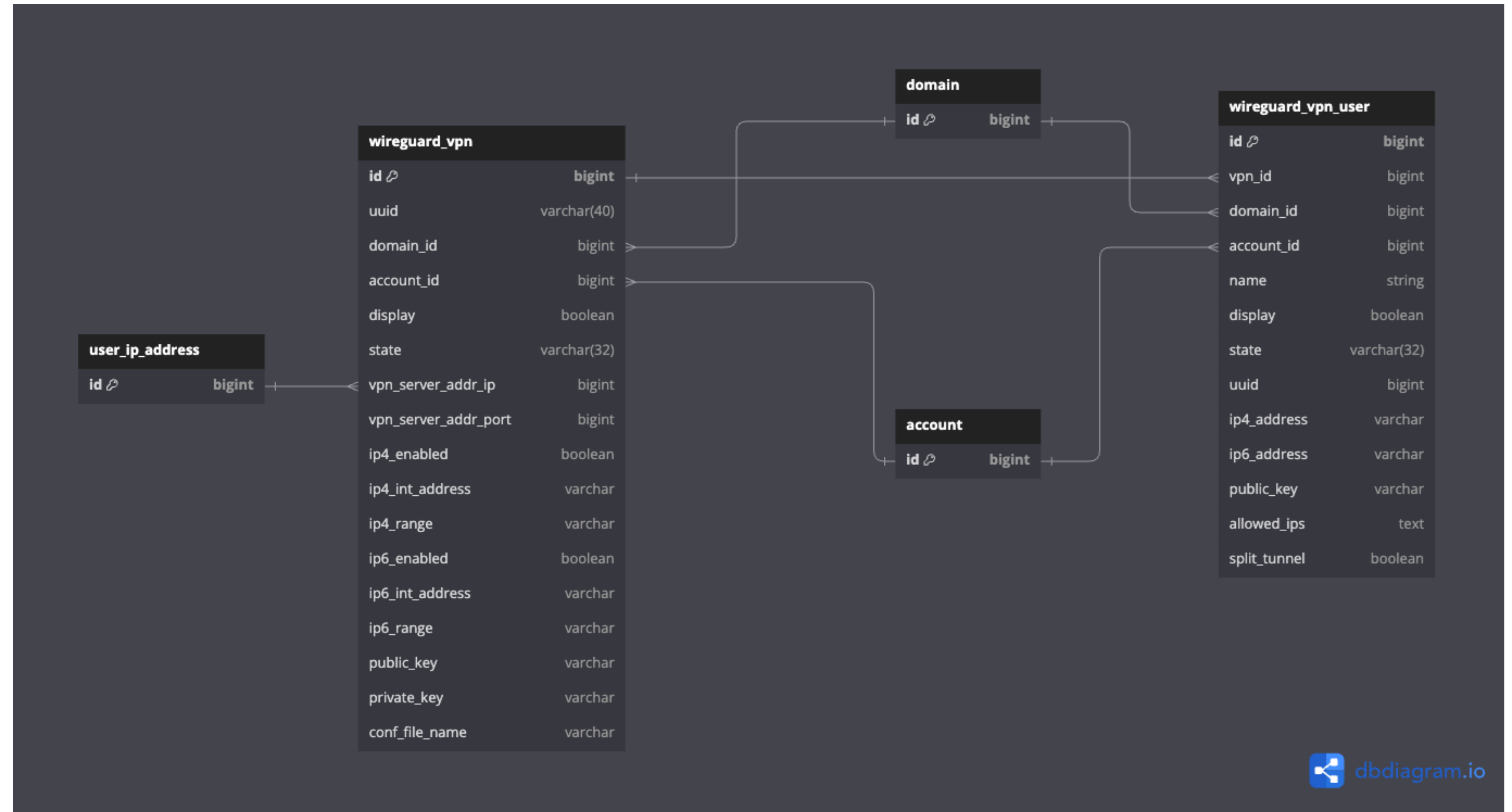
Name	Type	Notes
id	int64	
vpn_id	int64	
state	string	
public_key	string	
split_tunnel	bool	
account_id	int64	
domain_id	int64	
for_display	bool	

Name	Type	Notes
ip4_address	string	
ip6_address	string	

API: DeleteWgUser

Name	Type	Default	Notes
id*	int64		Id of the vpn user to delete

Database



Changes in IPsec rules to enable WG VPN

When enabling wireguard on the virtual router, some rules must be added to iptables.

In this presentation we will show the ones just for IPv4 but they will need to be extended to accommodate also the ipv6 case.

We needed to add rules in the mangle and filter tables.

Changes in IPsec rules (mangle table)



1 • create new chain WG_XXX.XXX.XXX.XXX

- A WG_XXX.XXX.XXX.XXX -p udp -m udp --dport <wg_port> -j ACCEPT
- A WG_XXX.XXX.XXX.XXX -j RETURN



2 • add rule to redirect to chain WG_XXX.XXX.XXX.XXX if the destination ip matches:

- A PREROUTING -d XXX.XXX.XXX.XXX/32 -j WG_XXX.XXX.XXX.XXX

Changes in IPsec rules (filter table)



1. Add rule to allow incoming packets

```
-A INPUT -d XXX.XXX.XXX.XXX/32 -i eth2 -p udp -m udp --dport  
<wg_port>  
-j ACCEPT
```




2. Add rule to permit packets to reach LAN and WAN

```
- A FORWARD -i wg0 -o eth2 -j ACCEPT  
- A FORWARD -i eth2 -o wg0 -j ACCEPT  
- A FORWARD -i wg0 -o eth0 -j ACCEPT  
- A FORWARD -i eth0 -o wg0 -j ACCEPT
```

Proposal: UI integration (public ip)





Public IP addresses / 95.157.67.252  Refresh


 **1.1.1.1**


Source NAT


Status
● Allocated


ID
 [bea74c08-ec1b-44a8-979d-871dc763c81a](#)

IP address
 [1.1.1.1](#)

Associated Network
 [wg-net](#)

Zone
 [Default Zone](#)

Account
 [admin](#)

Domain
 [ROOT](#)

Tags
[+ New tag](#)

Details Firewall Port forwarding Load balancing **VPN** Events Comments


[Enable remote access VPN](#)

[Enable Wireguard VPN](#)

Proposal: UI integration (public ip - vpn enabled)





🏠 / Public IP addresses / 1.2.3.4 ? 🔄 Refresh


 **1.2.3.4**


Source NAT


Status
● Allocated


ID
 4756f28f-4f6f-486f-86c1-38901d77aa23

IP address
 1.2.3.4

Associated Network
 test-v3

Zone
 MyZone

Account
 admin

Domain
 ROOT

Tags

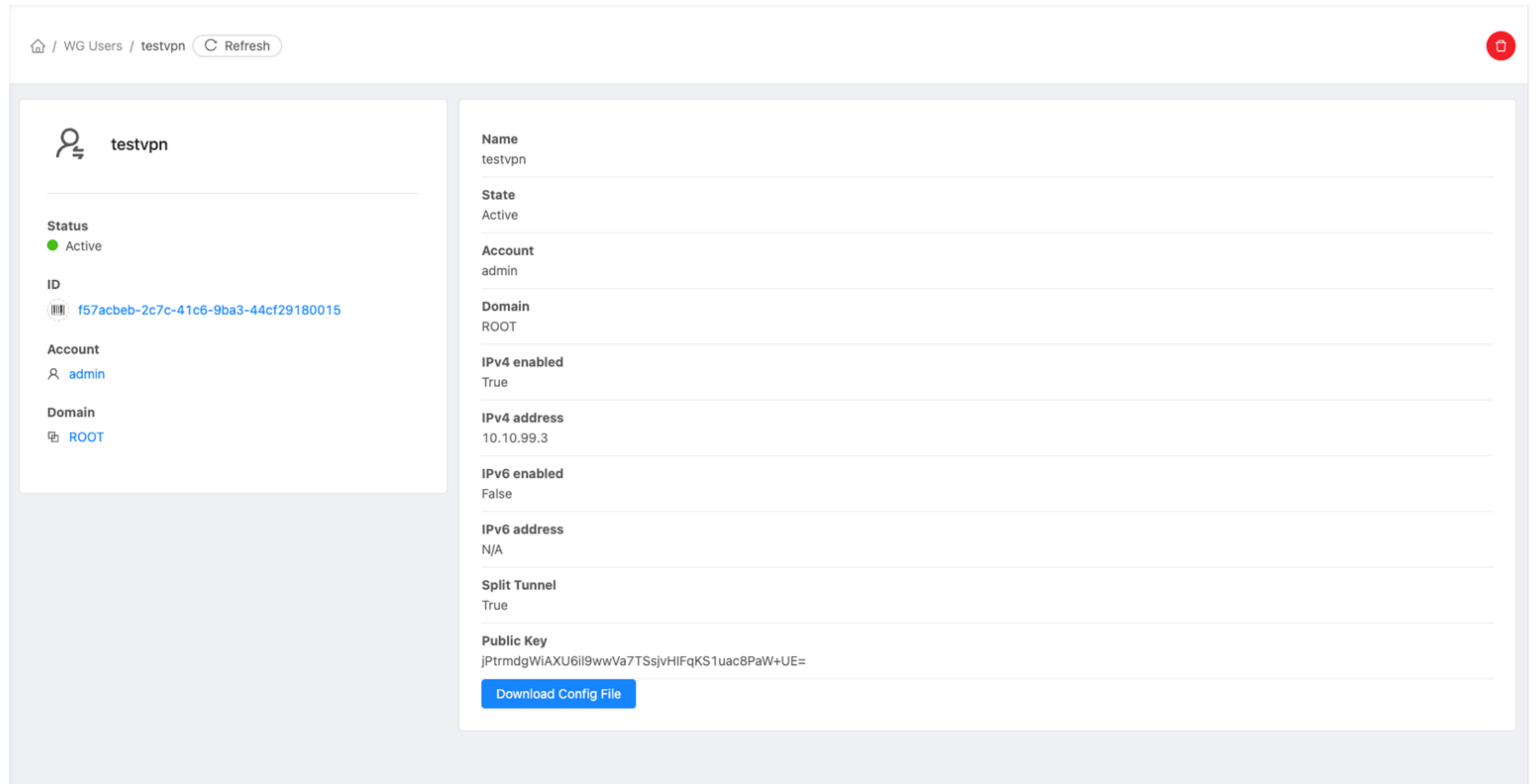
Details Firewall Port forwarding Load balancing **VPN** Events Comments

Your remote access VPN is currently enabled and can be accessed via the IP. **1.2.3.4**

Your IPSec pre-shared key is **VtBN8qBgAYK4WfnYZnYFmMBO**

Your Wireguard VPN is currently enabled and can be accessed via the IP. **1.2.3.4:51820**

Proposal: UI integration (vpn users detail)



The image shows a web interface for viewing VPN user details. At the top, there is a breadcrumb trail: "/ WG Users / testvpn" and a "Refresh" button. The main content is divided into two columns. The left column features a user profile for "testvpn" with a status indicator (Active), an ID (f57acbeb-2c7c-41c6-9ba3-44cf29180015), an account name (admin), and a domain (ROOT). The right column lists various configuration parameters: Name (testvpn), State (Active), Account (admin), Domain (ROOT), IPv4 enabled (True), IPv4 address (10.10.99.3), IPv6 enabled (False), IPv6 address (N/A), Split Tunnel (True), and Public Key (jPtrmdgWiAXU6il9wwVa7TSsjvHIFqKS1uac8PaW+UE=). A "Download Config File" button is located at the bottom of the right column.

WG Users / testvpn Refresh

testvpn

Status
Active

ID
f57acbeb-2c7c-41c6-9ba3-44cf29180015

Account
admin

Domain
ROOT

Name
testvpn

State
Active

Account
admin

Domain
ROOT

IPv4 enabled
True

IPv4 address
10.10.99.3

IPv6 enabled
False

IPv6 address
N/A

Split Tunnel
True

Public Key
jPtrmdgWiAXU6il9wwVa7TSsjvHIFqKS1uac8PaW+UE=
[Download Config File](#)

CDLAN