Engineer

VPN

a logically separate network over shared infrastructure

What it is

A VPN, short for virtual private network, is a set of tunnels or labelled paths that lets a closed group of machines reach one another by their own addresses and routes, whatever networks the packets cross on the way. The IETF's terminology treats it as a matter of separation rather than secrecy: the group is kept apart from other users, and encryption is mentioned as a possible extra that lies outside the working groups' scope. In daily use two shapes dominate. A site-to-site VPN joins whole networks, such as an office and a cloud project, through a gateway at each end. A remote-access VPN joins one laptop or phone to a network it is away from.

Very different technologies share the name. IPsec protects IP packets inside the network stack of a host or a security gateway and commonly uses IKEv2 to negotiate its keys; it is what most firewalls, routers and managed cloud gateways speak. WireGuard is a newer tunnel that appears as an ordinary network interface, travels over UDP, identifies each peer by a public key and deliberately offers one fixed set of ciphers instead of a menu to negotiate. OpenVPN runs as a user-space process and sets up its keys with TLS. Carrier VPNs built on BGP and MPLS keep customers apart inside a provider's backbone and, by their own specification, provide no cryptographic privacy at all. When a contract, a diagram or a firewall vendor says VPN, it pays to ask which of these is meant.

Use the concept in a decision

Put "What is a VPN?" to work in a real decision. We can help relate the concept to the system you are building or buying. Wi-Fi, firewalls and remote access set up so staff can work from anywhere and only the right people get in.

Explore our services Get in touch →

Test the assumptions behind the term

Use the explanation and linked work to frame questions for your project. We can examine the assumptions with you before a tool is chosen. Managed 30 Ubuntu Linux VPS instances — 30 Linux servers managed, with recovery planned for

Managed 30 Ubuntu Linux VPS instances Get in touch →

Why it matters

The decision a VPN usually settles is how two places that need to talk can do so without putting the service between them on the open internet. A database that only a pipeline in another cloud reads, an internal admin panel, a backup target: each can listen publicly behind a login, or listen on a private address that only the tunnel makes reachable. The second option shrinks what a scanner can find to one gateway port, and for links between sites it remains the plain, well-understood answer.

WireGuard or IPsec is the next question, and the far end usually decides it. If that end is a firewall appliance, a cloud provider's managed gateway or a partner's network team, IPsec is often the only thing on offer, and interoperability outweighs elegance. If both ends are Linux machines under your own control, WireGuard is quicker to set up and easier to reason about: one short configuration file per side, one public key per peer, and a list of the addresses each peer may send from. Its design paper puts the Linux implementation at under 4,000 lines of code, small enough to audit, while an IPsec setup on Linux pairs the kernel's packet transform with a separate key-exchange daemon and a longer list of options to agree on.

The running cost is rarely the software, which ships with Linux and with most firewalls. The cost is attention. Tunnels drop at night, rekeying breaks hours after somebody changed a setting on the far side, and the only symptom is silence: packets go in and nothing comes back. A tunnel deserves the same treatment as any other production service, with its own health check, its own alert and a written note of who owns each end.

A VPN is also the wrong tool in some places. Handing every remote employee a full tunnel into the office network grants reach to a whole subnet when the person needed one application, and once connected, everything on that subnet tends to trust them. Per-application access, where each request is authenticated on its own and usually sold as zero trust, fits better when the users are people and the targets are web applications. The consumer apps that advertise a VPN

for privacy are a third thing again: they move trust from the internet provider to the VPN company and can hide the original source IP address from destination sites, but do not by themselves guarantee anonymity.

What it looks like in practice

Between 2022 and 2024, secure traffic between Google Cloud and on-premises Linux systems ran over eight WireGuard and IPsec tunnels, managed and troubleshot as one set. They had been fragile and slow to diagnose, and a dead tunnel could cut communication between environments. Their configuration was standardised so that no tunnel remained a special case, their health was monitored, and the recurring routing and key-exchange problems were fixed at the root rather than cleared with a restart. All eight then ran reliably, and the recurring connectivity incidents stopped.

In the same years an organisation needed data from a remote warehouse that sat behind an IPsec VPN. For six ETL/ELT pipelines feeding Google BigQuery, the tunnel was the first link in the chain: secure connections to the protected server were set up so that retrieval could run unattended. Scheduled jobs then downloaded compressed archives of Parquet, CSV and .bak files, restored and staged them in MSSQL and PostgreSQL, and loaded the result into BigQuery. Over a tunnel that no longer waited for a person, the full run took under 20 minutes against at least three hours of manual work, and the data refreshed daily rather than weekly. The VPN is the least visible part of that result and the part everything else rests on: a pipeline that cannot reach its source stops at step one, however good the SQL behind it is.

Where it goes wrong

Big packets vanish while small ones get through. Every tunnel adds headers, so the largest packet that fits inside is smaller than the path outside allows. A WireGuard data message adds 32 bytes of fixed overhead before the outer UDP and IP headers, about 60 bytes in all over IPv4 and 80 over IPv6, in addition to any packet padding. When the tunnel's MTU is set too high and the path drops the ICMP messages that would report it, ping and an SSH login work while file transfers and TLS handshakes hang. Test with a full-size packet that forbids fragmentation, then lower the interface MTU or clamp the TCP segment size to fit.

```
# When each WireGuard peer last completed a handshake, as Unix time
wg show wg0 latest-handshakes
# A 1,420-byte packet that may not be fragmented on the way
ping -M do -s 1392 -c 3 10.0.0.2
# Which interface and route the kernel picks for a remote host
ip route get 10.20.0.5
```

Two sites picked the same addresses. Private ranges such as 10.0.0.0/8 and 192.168.0.0/16 are reused everywhere, and joining two networks that both use 10.0.0.0/24 leaves a router with no way to tell which side a packet is meant for. Plan the ranges before the first tunnel, or accept address translation at one end, which complicates every log line and firewall rule after it.

The tunnel is up and the route is wrong. A peer can show a fresh handshake while traffic still leaves through the default route, because the route for the remote subnet is missing or because WireGuard's list of allowed addresses for that peer leaves it out. In IPsec the counterpart is traffic selectors that differ between the two ends. IKEv2 either narrows them to the overlap or refuses them with a TS_UNACCEPTABLE notification, so a tunnel can come up carrying less than either side intended.

Rekeying fails long after the change. Both protocols replace active session keys over time, so a mismatch introduced at one end, such as an edited proposal or a rotated key, may show only when the current keys are replaced. WireGuard starts a new session about every two minutes while traffic flows. IPsec lifetimes are local policy at each end and need not match. For WireGuard, interpret the age of the last successful handshake alongside expected traffic or health probes: an idle peer's old handshake alone does not establish failure. Enable IKEv2's liveness check, often called dead peer detection, so a gateway notices a silent peer instead of sending into a black hole.

Connected becomes trusted. On many remote-access setups a laptop on the VPN is treated as if it sat in the office, so one stolen credential reaches everything on that subnet. Filter what each VPN client can reach at the gateway, keep authentication on the services behind it, and decide deliberately whether name lookups and internet traffic go through the tunnel or around it.

Further reading

- RFC 4026, section 3.10 — the IETF's generic definition of a VPN, and the note that encryption lies outside the working groups' charters.
- WireGuard: Next Generation Kernel Network Tunnel — the design paper: cryptokey routing, the handshake and the timers behind rekeying.
- RFC 4301 — the security architecture for IP that every IPsec implementation follows.
- RFC 7296 — IKEv2, including traffic selector narrowing, lifetimes as local policy and the liveness check.
- RFC 4364 — BGP/MPLS IP VPNs, the carrier kind, with overlapping address spaces and no cryptographic privacy.

Glossary

Choose for your actual constraints

Bring the deployment path, the operational risks and the people on call. We can scope a change with checks, a recovery route and clear ownership.

See what it costs Get in touch →

Make the terminology practical

Ask how "What is a VPN?" applies to your organisation. A concrete example from your current system is enough to start the discussion.

Explore our services

welcome@engineer.company +45 50 10 61 84 Copenhagen, Denmark

Take this page with you

Contact card

Carefully Crafted With Love by D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Copyright © 2025 – present · Engineer ApS · All rights reserved.

Printed from https://engineer.company/glossary/vpn/