USING WIREGUARD TO PROTECT INTRANET NETWORKS ON MOBILE DEVICES

Nikola Petrović[1]*,
[0009-0004-2325-2822]

Dejan Viduka[2],
[0000-0001-9147-8103]

Vojkan Nikolić[3]
[0000-0002-9230-7549]

[1]Ministry of Internal Affairs,
Belgrade, Serbia

[2]Alfa BK University,
Belgrade, Serbia

[3]University of Criminal Investigation
and Police Studies,
Belgrade, Serbia

Abstract:

Field teams often need secure access to intranet services while working over LTE/5G or public Wi-Fi. In that setting, the challenge is not only encryption, but limiting exactly what a mobile device can reach and keeping that behavior predictable when the network changes. This paper examines WireGuard as a practical solution for that problem. The proposed design uses a WireGuard gateway as the only external entry point to the intranet and combines peer-specific addresses, AllowedIPs, and firewall rules to keep access aligned with user roles. A prototype with 10 mobile clients was evaluated through scenarios that reflect real use: access to internal GIS services, Wi-Fi↔LTE handover, tunnel interruption, and leakage checks. The results show that the approach can protect internal services while remaining usable in day-to-day mobile work.

Keywords:

Wireguard, Mobile VPN, Intranet Access, RBAC, GIS.

INTRODUCTION

Mobile devices are now a routine part of field and operational work. Staff often need to reach internal web portals, APIs, map services, and other intranet resources while outside the organization, usually from phones or tablets.

The problem is that this access is commonly established over networks the organization does not control: carrier LTE/5G, public Wi-Fi, hotel networks, or home connections. In those conditions, the risk is wider than packet confidentiality alone. DNS leakage, routing mistakes, unstable handovers, and accidental overexposure of internal services all become practical security issues.

VPNs are the usual answer, but in mobile use their value depends on more than cryptography. A workable deployment has to recover during Wi-Fi↔LTE transitions, keep traffic inside the tunnel, and avoid giving the client a wider view of the intranet than necessary.

Correspondence:
Nikola Petrović

e-mail:
nikola.spetrovic@mup.gov.rs

DOI: 10.15308/Sinteza-2026-348-353

WireGuard is a good candidate here because it has a small configuration surface, a clear peer model, and straightforward Layer 3 behavior. Those properties make it easier to assign one identity per device, revoke access quickly, and reason about routing and exposure.

This paper therefore looks at a practical deployment question: how to organize WireGuard so that mobile devices can reach selected intranet services securely, without making those services public and without turning the VPN into broad internal access.

1.1. CONTRIBUTIONS OF THIS PAPER

- A single WireGuard gateway is used as the controlled entry point for mobile access to intranet services.
- Access control for 10 mobile clients is implemented through peer identity, dedicated VPN addresses, AllowedIPs, and firewall rules.
- A real-time GIS web application is evaluated during concurrent use and Wi-Fi↔LTE/5G transitions by observing RTT, jitter, reconnect events, and T_recover.
- An audit layer correlates peer activity with service access and blocked attempts.

2. WHY WIREGUARD INSTEAD OF TRADITIONAL MOBILE VPN SOLUTIONS

WireGuard was chosen because the goal was not only encrypted connectivity, but controlled and auditable access from mobile endpoints. Compared with heavier VPN stacks, its peer model makes device-level policies easier to define, review, and revoke. Its cryptokey routing model also fits deployments in which the VPN is the only external path into the intranet.

For this study, stability during network changes and fail-closed behavior were treated as part of the security design, not as secondary usability issues.

3. REVIEW OF RELATED WORK

Related work generally describes WireGuard as a lightweight VPN with simple key management and good performance characteristics. The protocol itself was introduced by Donenfeld [1]. Later work examined its use in 5G and NFV service-isolation settings [2], compared it with strongSwan and OpenVPN in controlled environments [3], reported real-world evaluation results [4], and analyzed performance in cloud and virtualized environments [5].

Broader comparative work on VPN protocols also helps place WireGuard in a wider operational context [6]. These results support its use here, but they do not answer the narrower question addressed in this paper: how to expose only a small, role-limited intranet surface to mobile devices under realistic field conditions.

4. THREAT MODEL AND SECURITY REQUIREMENTS

4.1. ENVIRONMENT AND SYSTEM BOUNDARIES

The assumed environment is an intranet with internal web applications, APIs, and related services that should not be directly reachable from the public Internet. Some mobile users need access to only a limited subset of those resources. The design goal is therefore to create a narrow and enforceable path for field work rather than to extend the full intranet outward.

The paper focuses on the tunnel, routing and filtering rules, and behavior during interruption or network switching. Vulnerabilities in the protected applications themselves are out of scope.

4.2. WHO CAN ATTACK, AND HOW

In this model, the attacker is usually somewhere between the mobile device and the organization, which is realistic when public wireless or carrier networks are used. The main risks are traffic observation, packet manipulation, DNS abuse, scanning of the VPN endpoint, and misuse of copied profiles or exposed keys.

- observe traffic and extract information (passive eavesdropping);
- modify or inject packets to redirect communication or cause errors;
- manipulate address resolution (DNS) to lead the user to the wrong server;
- attack the publicly reachable system point (VPN endpoint) via scanning and automated break-in attempts; and
- misuse obtained VPN profiles or keys if acquired, for example through copied configurations, a compromised device, or insecure backups.

The core concern is not a single instant takeover of the whole intranet, but exploitation of common mobile deployment weaknesses such as broad routing, weak DNS handling, exposed perimeter services, or poor credential handling.

4.3. TYPICAL SECURITY SITUATIONS THAT MUST BE COVERED

The design was checked against five practical situations: untrusted public networks, routing that exposes too much of the intranet, DNS or application traffic leaking outside the tunnel, short interruptions during network switching, and compromise of one client profile or key set.

4.4. SECURITY REQUIREMENTS

The resulting requirements are straightforward:

Z1 — The tunnel must provide confidentiality and integrity protection.
Z2 — Each device must have a distinct identity and access must be revocable quickly.
Z3 — Clients must receive only the minimum intranet visibility required for their role.
Z4 — Internal services must stay closed to the public Internet and remain reachable only through the VPN gateway.
Z5 — Internal DNS resolution must stay inside the tunnel.
Z6 — Compromise of one client should expose only a small part of the environment.
Z7 — Logs must be sufficient for troubleshooting and incident review.

4.5. OPERATIONAL REQUIREMENTS AND PERFORMANCE CRITERIA

Besides security, the deployment also has to remain usable in normal work. In practice this means fast connection establishment, quick recovery after Wi-Fi↔LTE changes, measurable tunnel behavior through RTT, jitter and reconnect counts, and a clear fail-closed response when the tunnel is down.

5. PRACTICAL PART: IMPLEMENTING A WIREGUARD SOLUTION FOR SECURE MOBILE INTRANET ACCESS

5.1. GOAL AND INITIAL ASSUMPTIONS

The prototype uses WireGuard as the only controlled entry point to intranet resources for 10 mobile clients operating over LTE/5G or public Wi-Fi. The protected workload is a real-time GIS web application, and the evaluation focuses on role-based access control, session stability during mobility, concurrent use, and the value of audit logging.

5.2. SOLUTION TOPOLOGY AND SYSTEM COMPONENTS

The topology is intentionally simple so that security policy and observed behavior can be tied together without too many moving parts.

Topology: mobile device → WireGuard tunnel → WireGuard gateway (public IP) → intranet segment → GIS web application.

Components:

- WireGuard gateway (WG-GW): Linux server at the perimeter with a public IPv4 address and only a UDP port open for WireGuard, for example 51820. The gateway terminates the tunnel, routes to the intranet, and applies per-peer firewall rules.
- GIS web application (intranet): internal HTTPS frontend with map view, a real-time channel such as WebSocket or SSE, and the accompanying API service.
- Tile or map service (intranet): serves map tiles over HTTPS.
- Mobile clients: Android and iOS devices using the WireGuard client; some devices operate in Always-On mode so the behavior is fail-closed.

Table 1. Example peer table

Peer	VPN IP	Role	Allowed services
M1	10.10.10.10	Patrol	GIS web + tiles + read API
M2	10.10.10.11	Patrol	GIS web + tiles + read API
M3	10.10.10.12	Patrol	GIS web + tiles + read API
M4	10.10.10.13	Patrol	GIS web + tiles + read API
M5	10.10.10.14	Patrol	GIS web + tiles + read API
M6	10.10.10.15	Patrol	GIS web + tiles + read API
M7	10.10.10.16	Patrol	GIS web + tiles + read API
M8	10.10.10.17	Patrol	GIS web + tiles + read API
M9	10.10.10.18	Dispatcher	GIS web + tiles + write API
M10	10.10.10.19	Administrator	GIS web + tiles + admin API + optional SSH

5.3. ADDRESSING AND CLIENT ASSIGNMENT (10 PEERS)

Each mobile device receives a unique VPN address tied to one operational role. That one-to-one mapping between device, tunnel address, and permitted services is central to the access-control model.

VPN subnet: 10.10.10.0/24
WG-GW (wg0): 10.10.10.1
Intranet subnet: 192.168.100.0/24
GIS frontend (HTTPS): 192.168.100.20:443
Tile service (HTTPS): 192.168.100.21:443
GIS API (HTTPS): 192.168.100.22:443
Optional internal DNS: 192.168.100.53:53

This mapping preserves least privilege at device level: patrol users receive only field-facing services, while dispatcher and administrator access is extended in a limited and auditable way.

5.4. WIREGUARD GATEWAY CONFIGURATION (SERVER SIDE)

At the gateway, IP forwarding is enabled so traffic from the VPN subnet can reach the intranet. The WireGuard interface is configured with 10.10.10.1/24, and each device is added as a separate peer with its own public key and /32 VPN address. The perimeter policy allows only UDP/51820 and blocks everything else.

That means neither the intranet services nor the management surface are directly exposed to the Internet; all further control happens behind the WireGuard boundary.

5.5. NETWORK-LEVEL RBAC: LEAST PRIVILEGE VIA ALLOWEDIPS AND FIREWALL

Role-based access is enforced by combining client-side route scoping through AllowedIPs with gateway-side firewall rules bound to source VPN addresses.

In the prototype, patrol clients are limited to the GIS web frontend, tile service, and read API; the dispatcher additionally receives write API access; and the administrator receives the same set plus the admin API and one optional management service.

- Patrol (M1-M8): allow 192.168.100.20:443 for GIS web, 192.168.100.21:443 for tiles, and 192.168.100.22:443 for read API access; deny admin endpoints, write operations, and other intranet subnets.
- Dispatcher (M9): same as patrol, plus write API access for actions such as event creation or validation.
- Administrator (M10): all of the above, plus the admin API and, optionally, one management service such as SSH to a single host.

This limits lateral movement: even after tunnel establishment, unauthorized ports and destinations are rejected at the gateway.

5.6. MOBILE CLIENT CONFIGURATION AND OPERATING MODE (FAIL-CLOSED)

Each phone is configured with its own key pair, a 10.10.10.x/32 tunnel address, the public gateway endpoint, and split-tunnel AllowedIPs that route only the needed intranet targets.

On Android devices that support it, Always-On VPN with "Block connections without VPN" is enabled so that tunnel failure stops intranet access instead of falling back to an uncontrolled path.

Internal GIS names are resolved through an internal DNS server only while the tunnel is active, which reduces leakage and inconsistent name resolution.

5.7. REAL-TIME GIS: VERIFYING SESSION STABILITY (WEBSOCKET/SSE)

Because the GIS application includes a real-time channel, evaluation has to check more than page availability. The important questions are whether the channel stays connected, how quickly it reconnects after a network switch, and whether message gaps remain acceptable.

- the real-time channel staying connected;
- quick recovery after a network switch; and
- acceptable reconnect counts and message loss.

The prototype therefore records reconnect counts, T_recover after interruptions, and the maximum stream gap for the real-time channel.

5.8. MONITORING AND AUDIT LAYER (OPERATIONAL CONTROL)

To make the setup useful beyond a lab demonstration, an audit layer correlates peer activity, firewall decisions, and GIS service logs so it is possible to see who accessed which service and when an access attempt was blocked.

- WireGuard status for peer activity and handshake time;
- firewall logs with ACCEPT and DROP per VPN IP and destination; and
- web server logs for the GIS frontend and API.

Correlation is based mainly on VPN source address and timestamps, which is sufficient for everyday troubleshooting and basic post-incident review.

5.9. TEST SCENARIOS AND MEASUREMENT METHODOLOGY

The evaluation uses six scenarios: unreachable without VPN, normal use with 10 concurrent clients, peak interaction during panning and zooming, a limited write-action incident scenario, Wi-Fi↔LTE mobility, and fail-closed verification during tunnel interruption.

- S1 — unreachable without VPN;
- S2 — normal functionality with 10 concurrent clients;
- S3 — peak interaction with concurrent panning and zooming;
- S4 — an incident scenario with write-oriented actions by dispatcher and administrator;
- S5 — mobility testing during Wi-Fi↔LTE switching; and
- S6 — fail-closed verification during tunnel interruption or simulated signal loss.

Measured metrics are RTT to the gateway and GIS server, jitter, T_recover during switching, reconnect counts for the real-time channel, HTTP response time for key endpoints, and gateway CPU/RAM load.

6. RESULTS

6.1. FUNCTIONALITY AND ACCESS POLICY

Without the tunnel, intranet resources are unreachable. Once the tunnel is established, all 10 clients can access the GIS application, while patrol devices remain blocked from administrative endpoints and the administrator can reach only explicitly allowed resources.

6.2. PERFORMANCE AND STABILITY (10 CLIENTS)

Under stable Wi-Fi conditions, the real-time channel stays connected without frequent interruptions. During peak interaction, tile-request response times increase, but not enough to prevent normal operational use.

6.3. MOBILITY (WI-FI↔LTE↔WI-FI)

During network transitions, the real-time channel disconnects briefly and then reconnects automatically. Recovery time varies with signal quality and carrier behavior, but in the tested setup the interruption remained short enough to preserve session continuity.

6.4. FAIL-CLOSED BEHAVIOR

On devices with Always-On mode, fail-closed behavior was confirmed: once the tunnel dropped, intranet endpoints became unreachable immediately and no meaningful intranet traffic was seen outside the VPN path. On devices without that mode, the application could retry, but access still did not resume until the tunnel was restored.

6.5. AUDIT AND MONITORING

The logs made it easy to see active peers, accessed internal services, and blocked attempts. In practice, this is useful for spotting misconfiguration early and for flagging device behavior that deserves closer inspection.

7. CONCLUSION

The results indicate that a WireGuard gateway with a tightly constrained firewall policy can provide a secure and workable model for mobile access to intranet resources. Internal services stay hidden from the public Internet and become reachable only through the tunnel, while peer-based addressing and policy enforcement support a meaningful least-privilege model.

Table 2. Example summary results for normal and peak use

Test	Avg RTT to GIS (ms)	Jitter (ms)	Avg HTTP response (ms)	Reconnect/client (10 min)
S2 normal	18-35	3-9	120-260	0-1
S3 peak	25-55	6-18	220-520	0-2

Table 3. Example mobility results

Transition	T_recover median (s)	T_recover max (s)	Reconnect events
Wi-Fi → LTE	2.5-5.0	8-12	1
LTE → Wi-Fi	2.0-4.5	7-10	1

The tests also show that the design remains usable under conditions that matter in practice: the tunnel recovers after ordinary Wi-Fi↔LTE transitions, the GIS workload remains stable for the tested scale, and fail-closed behavior can be enforced on supported clients. At the same time, the measured values should be read as deployment-specific and not as universal performance guarantees.

8. LIMITATIONS AND FUTURE WORK

This study still has several limits. The WireGuard gateway is a central control point and, without redundancy, also a single point of failure. Client lifecycle management, including onboarding, offboarding, key rotation, revocation, and secure storage on devices, also becomes more important as the number of clients grows.

In addition, the behavior of the real-time GIS application depends not only on the VPN tunnel, but also on radio conditions, server-side load, and reconnect logic in the application. Future work should therefore include high-availability gateway testing, automated provisioning and revocation, stronger audit integration, controlled degradation experiments with packet loss and limited bandwidth, and finer policy refinement toward micro-segmentation.

REFERENCES

[1] J. A. Donenfeld, "WireGuard: Next Generation Kernel Network Tunnel," NDSS Symposium, 2017.

[2] S. Haga, A. Esmaeily, K. Kralevska, and D. Gligoroski, "5G Network Slice Isolation with WireGuard and Open Source MANO: A VPNaaS Proof-of-Concept," Proc. 2020 IEEE Conference on Network Function Virtualization and Software Defined Networks, 2020, doi: 10.1109/NFV-SDN50289.2020.9289900.

[3] E. Dekker, "Performance comparison of VPN implementations WireGuard, strongSwan, and OpenVPN in a 1 Gbit/s environment," OS3, University of Amsterdam research report, 2019-2020.

[4] "Lightweight Security for Private Networks: Real-World Evaluation of WireGuard…," arXiv, 2025.

[5] "Empirical Performance Analysis of WireGuard vs. OpenVPN in Cloud and Virtualized Environments," Computers, 2025.

[6] J. Antoniuk and M. Plechawska-Wójcik, "Comparative analysis of VPN protocols," JCSI, 2023.