Shelby PQC Gateway Series

Post-Quantum Cryptography for Existing Enterprise Infrastructure

Architecture, Engineering Evolution, and Validated Performance

TECHNICAL WHITE PAPER

Elgens Co., Ltd. | www.elgens.com.tw

August 2026

EXECUTIVE SUMMARY

This white paper documents the engineering evolution, final architecture, and field-validated performance of the Shelby PQC Gateway Series — an edge appliance that brings NIST-standardized post-quantum cryptography (PQC) to existing enterprise infrastructure without server-side modification. It is written for technology decision-makers evaluating how to close the gap between browser-side quantum-safe readiness and server-side migration timelines.

- Zero server-side modification: in the finalized bridge-mode architecture, protected servers require no software installation, no configuration change, no certificate re-issuance, and no network-layer change — the server's IP address, subnet, and default gateway remain completely untouched (Section 4.1).
- Hybrid ML-KEM-768 key exchange adds only **~2.3% (~2 ms)** handshake latency versus classical elliptic-curve TLS, at a 100% success rate across 1,000 connection trials (Section 5.1).
- On the finalized bridge-mode architecture, internal engineering testing on dedicated physical hardware sustained **100,000+ concurrent connections with 0% failure**. Degradation began above 110,000 connections and became severe between 140,000 and 160,000 connections (Section 5.2).
- The final architecture arrived through three engineering stages, not chosen upfront; Section 2 documents why each stage was superseded and what

specifically made bridge mode with hardware bypass the only combination compatible with the product's physical fail-safe design.

- TPM 2.0-sealed private keys, dual-plane hardware-bypass networking, and Active/Passive high-availability clustering combine physical- and logical-layer fail-safes into a single mission-critical resilience architecture (Section 4).

Deployment boundaries, stated upfront: the hardware-bypass fail-safe and its source-IP-preservation mechanism both require the Gateway and the protected server to sit on the same Layer-2 broadcast domain and IP subnet; non-HTTPS clinical protocols (HL7/FHIR, DICOM/PACS) are not yet covered; and performance figures in this paper were measured in internal engineering tests on dedicated physical hardware, not by an independent third-party laboratory. Each boundary is discussed in context in the relevant section.

1. The Quantum Threat and the Design Objective

1.1 Harvest Now, Decrypt Later

The security community has debated the arrival of cryptographically relevant quantum computers for over a decade. For enterprise decision-makers, the more urgent question is not when such a machine arrives, but what happens to data encrypted today. "Harvest Now, Decrypt Later" (HNDL) describes an attack pattern already underway: adversaries capture encrypted traffic today — patient records, financial transactions, government communications — and store it, betting that a future quantum computer will break the classical key-exchange algorithms (RSA, ECDH) protecting it now. For any data whose confidentiality must outlast the timeline to a cryptographically relevant quantum computer, the threat is realized now of capture, not at the moment of decryption.

1.2 Client Readiness Has Already Arrived — Server Readiness Has Not

This concern is already reflected in the deployment decisions of major platform providers. Google Chrome has defaulted to enabling ML-KEM — the post-quantum key-encapsulation mechanism standardized as NIST FIPS 203 — for all TLS 1.3 connections since Chrome 131 (November 2024), across Windows, macOS, Linux, Android, and ChromeOS. This was independently confirmed by forcing a connection to a major search engine's public endpoint with openssl's_client -groups X25519MLKEM768: the handshake completed and reported Negotiated TLS1.3 group: X25519MLKEM768 — not an optional capability, but the default negotiated outcome. Google Cloud Load Balancing, Cloudflare, and Akamai have followed with production PQC-TLS support at the CDN and load-balancer layer.

The same test applied to a typical production server tells a different story. A direct TLS handshake test against a regional hospital's online outpatient registration system returned Protocol: TLSv1.2, Cipher: ECDHE-RSA-AES128-GCM-SHA256. Forcing the same server to negotiate X25519MLKEM768 was refused outright, and the connection fell back to classical key exchange — precisely the configuration HNDL targets. The gap is not a lack of client-side readiness; it is the practical difficulty of migrating server-side TLS stacks embedded inside production systems that were never designed to be touched. The same assessment also surfaced that the registration system's certificate omitted the standard Subject Key Identifier extension — an omission that does not itself break TLS but may reduce interoperability with stricter certificate-management and

validation workflows, and a useful reminder that a site can look entirely normal to end users while resting on cryptographic hygiene weaker than assumed.

1.3 Why "Rip and Replace" Doesn't Work

Three structural constraints make a direct TLS-stack migration impractical for most regulated, infrastructure-heavy organizations — hospitals foremost among them.

- Interconnected, fragile system landscapes — a single hospital's environment routinely spans dozens of interdependent systems (outpatient registration, HIS, LIS, PACS, payment gateways), often built by different vendors on different release cycles. These systems are tightly coupled: a TLS-configuration change to one can silently break authentication or integration behavior in another.
- IT departments already at capacity — hospital and enterprise IT teams are typically staffed for day-to-day operations, not a coordinated cryptographic-migration program spanning every externally facing system, each requiring vendor coordination, regression testing, and a maintenance window.
- The practical answer: move the upgrade to the edge — insert a single purpose-built appliance at the network edge that terminates the post-quantum TLS handshake on behalf of every server behind it, without requiring those servers to change at all. This is the problem the Shelby PQC Gateway Series is engineered to solve.

2. Architectural Evolution — Three Stages to a Deployable Design

The final architecture was the result of three successive engineering stages, not a design chosen upfront. This section documents that evolution — what each stage got right, what it exposed, and why bridge mode combined with hardware bypass is, specifically, the only combination compatible with the product's physical fail-safe design.

Stage	Deployment Mode	Key Protection	Source-IP Method	Main Limitation	Status
Stage 1	Remote signing	Customer server	Application agent	External signing dependency	Superseded
Stage 2	Routed mode	Local TPM sealing	IP-layer transparent proxy	Requires server default-gateway change	Field validated
Stage 3	Bridge mode	Local TPM sealing	PROXY protocol / X-Forwarded-For	Requires same Layer-2 broadcast domain for bypass	Final architecture

2.1 Stage 1 — "Keyless SSL" (Concept Stage, Superseded Before Field Validation)

The earliest design concept kept the private key on the customer's own original server and had the Gateway make a remote signing-service call over mutual TLS for every handshake, so the key itself never left customer-controlled infrastructure. The concept was sound cryptographically, but it reintroduced exactly the burden the product exists to remove it required a signing-service agent running on the customer's own protected server (Section 1.3), and it added a live network dependency and a per-handshake round-trip to an external signer, directly working against the latency objective validated in Section 5.1. This design was superseded before field validation began, in favor of a Gateway that holds its own key.

2.2 Stage 2 — Local TPM-Sealed Key, Layer-3 Routed Mode

The second stage kept the private key local to the Gateway — TPM-sealed, detailed in Section 4.6 — and deployed the appliance in routed mode: traffic was steered to the Gateway by an IP-swap or DNS-redirection method (Section 6.2 discusses both), and the client's real source IP was preserved on the server-facing side using an IP-layer transparent-proxy technique (Nginx's proxy_bind $ remote_addr transparent,

combined with policy-based routing). This required the protected server's default gateway to be pointed at the Gateway's internal-segment address — a one-time, network-layer parameter change, not a modification to the server application itself, but a genuine change, nonetheless. This is the architecture validated in the field engagement documented in Section 7.

Stage 2 worked and worked well — but exposed two structural limits. First, IP-layer source-address spoofing is a routed-mode-only technique: it depends on the Gateway and the protected server sitting on different subnets so that a spoofed source address does not collide with a real device on the same broadcast domain; it does not extend across cloud-hosted or cross-subnet backends. Second, and more consequentially for the product's resilience design: a routed deployment necessarily places the Gateway between two different IP subnets. That single fact turned out to be fundamentally incompatible with the strongest physical fail-safe layer the High Availability design depends on — detailed next.

2.3 Stage 3 — Bridge Mode with Hardware Bypass (Final Architecture)

Hardware bypass (Section 4.7) is a mechanical relay on a purpose-built network interface card: on total power loss or a missed software watchdog check-in, the relay physically shorts the Gateway's two network ports together, collapsing the appliance into a transparent wire so that traffic keeps flowing even if the unit is completely dead. This provides a fail-safe mechanism that remains functional during a complete loss of software and power, since the relay itself requires no software, no power, and no CPU cycles to operate.

A relay-shorted wire, however, is a pure Layer-2 splice — it has no awareness of IP addressing at all. If the two sides of that wire sit on different IP subnets, as they do in a routed deployment, a physical short between them does not gracefully degrade the network: it collapses two distinct subnets into one, and traffic that expects to be routed instead loops or blackholes at the point of the short — the opposite of "fail open." Hardware bypass is only a safe fail-safe when both sides of the Gateway already sit on the same Layer-2 broadcast domain and IP subnet — that is, when the Gateway itself is deployed as a transparent bridge.

This is the specific reason for the deployment model being changed. To use the fail-safe layer that survives even total power loss, the Gateway had to be re-deployed in bridge mode: a single Layer-2 segment (br0) spanning both the client-facing and server-facing ports, with the protected server's real IP address, subnet, and default gateway

completely unchanged. This is a deployment-topology decision, not a compromise to the cryptographic architecture — but it did require re-solving two problems that the routed-mode design in Stage 2 had already solved:

- Selective interception on a bridge. A bridge forwards ordinary frames by destination MAC address, not by IP header, so the IP-layer redirection technique used in Stage 2 has no effect on bridged traffic. Only traffic addressed to the protected TLS port needed to be identified and redirected for PQC termination; everything else needed to keep flowing as an ordinary, fully transparent bridge.
- Source-IP preservation once client and server share one segment. The moment the client and the protected server sit on the same Layer-2 broadcast domain, the IP-spoofing technique from Stage 2 breaks: the server's reply to a spoofed source address is delivered directly to whichever device owns that address, via ordinary ARP resolution, bypassing the Gateway entirely.

Both problems were resolved at a conceptual level by changing which layer does the work, without reintroducing implementation detail that this paper intentionally omits in-scope traffic is identified at the bridge layer itself rather than by IP-layer redirection, and the real client IP is carried forward at the application layer — via the PROXY protocol and X-Forwarded-For — rather than through network-layer address spoofing. An application-layer signal works regardless of how many subnets sit on either side of the bridge, trading away a technique that only worked inside one subnet for one that generalizes. The result is the architecture described in Section 4 and validated in Section 5 of this paper — the architecture now targeted for production units.

Protected servers require zero application code, TLS-stack, or network-layer changes for PQC protection itself. Where real client-IP visibility is required for upstream audit trails, rate limiting, or abuse detection, the protected application or web-server layer must be configured to read and trust the PROXY protocol or X-Forwarded-For header — the same standard configuration step required by any reverse proxy or load balancer, PQC or otherwise. A backend that does not enable this will simply see the Gateway's own address for logging purposes, with no effect on the TLS protection itself; a backend that enables it without also restricting which upstream is trusted to set that header exposes itself to source-IP spoofing, exactly as with any other reverse-proxy deployment.

3. Engineering Lessons from the Transition

Consistent with this paper's focus on technology selection and evolution rather than implementation detail, the lessons below are stated at the level of "what changed and why it mattered," not as configuration instructions.

- IP-layer tools stop working the moment the deployment becomes a bridge. Any redirection technique built on IP-header inspection needs a bridge-layer equivalent; this is not a settings change but a different mechanism entirely.
- Source-IP spoofing is a routed-mode-only technique, not a general one. It relies on the Gateway and the protected server not sharing a broadcast domain. Once bridge mode puts them on the same segment, real-client-IP delivery must move to the application layer.
- A reverse-proxy engine chosen for routed mode may not carry over. Fully transparent listening sockets are not a feature every reverse-proxy engine supports; validating this assumption early avoids discovering the gap mid-migration.
- Conservative firewall defaults deserve re-verification on a topology change, not removal. A default-deny "is this really local traffic?" protection correctly rejected an early version of the redirected bridge-mode traffic. The correct fix was a narrowly scoped exception, not disabling the protection — a reminder that hardening decisions made for one deployment mode need re-checking, not discarding, when the deployment mode changes.
- Legacy TLS backend compatibility is a build decision, not a configuration one. On current-generation Linux distributions, mainstream OpenSSL builds exclude TLS 1.0/1.1 entirely at compile time. A protected server that still requires a legacy-TLS backend connection cannot be accommodated by a proxy configuration change alone; it would require an isolated, purpose-built TLS build — a scoped decision to make deliberately if a future deployment ever requires it.

4. Final Architecture

4.1 Deployment Model

In its final form, the Gateway is inserted as a transparent Layer-2 bridge between the client-facing network segment and the existing server infrastructure. Ordinarily, non-intercepted traffic is forwarded transparently through the bridge. Only traffic addressed to the protected server's TLS port is selectively identified and redirected: the Gateway terminates a hybrid post-quantum TLS handshake on the client-facing side, then forwards the request to the protected server over a standard, unmodified TLS 1.3 connection. The protected server's IP address, certificate, application code, and configuration all remain completely untouched, with no network-layer change required either — the refinement over Stage 2 (Section 2.2) that removes the one remaining server-side touchpoint from the routed-mode era.

4.2 Hardware Platform

Item	Specification
Form Factor	1U, 19-inch rack-mount, 431.8 (W) x 321.6 (D) x 44.0 (H) mm
Processor	Intel(R) Core(TM) 12th / 13th / 14th Gen (up to 24 cores), LGA1700
System Memory	2x DDR4 3200 MHz SO-DIMM, up to 64 GB
Power	300 W 1+1 hot-swappable redundant power supply, 100-240 V AC
Trusted Security	Onboard TPM 2.0, Secure Boot, LUKS full-disk encryption
Expansion	PCIe Gen4 x16 via 1U riser (10 GbE / 25 GbE, hardware-bypass configurations)
Certification	FCC / CE Class A; cryptographic module implements NIST FIPS 203 / 204 / 205
Operating Environment	0C to 45C operating, -20C to 60C storage, 10-90% RH non-condensing

4.3 Model Lineup

Model	CPU	Network Interface	Memory	Storage
Shelby-G5135	Intel(R) Core(TM) i5-13500TE	10 GbE	16 GB DDR4	128 GB SATA SSD
Shelby -G7137	Intel(R) Core(TM) i7-13700TE	25 GbE	16 GB DDR4	128 GB SATA SSD

Both models share an identical software and security architecture; the G7137's higher core count and 25 GbE interface target higher-throughput insertion points such as core-network segments or multi-tenant hosting environments, while the G5135 is sized for typical single-institution edge deployment.

4.4 Software Stack

The Gateway runs on Ubuntu Linux, with a core engine built on liboqs and oqs-provider — the Open Quantum Safe project's OpenSSL 3.0 provider — fronted by an Nginx reverse-proxy layer. This combination gives the Gateway a standards-based, independently auditable post-quantum TLS implementation rather than a proprietary cryptographic stack.

4.5 Hybrid Post-Quantum TLS

At the center of the Gateway's client-facing TLS termination is a hybrid key-exchange configuration: ssl_conf_command Groups X25519MLKEM768:X25519:secp256r1. This instructs the Gateway to prefer ML-KEM-768 combined with classical X25519 elliptic-curve Diffie-Hellman — a hybrid construction that remains secure even if either the classical or the post-quantum component is later broken — while falling back gracefully to classical X25519 or secp256r1 for clients that do not yet support PQC. TLS 1.3 is the primary protocol, with TLS 1.2 retained for compatibility; TLS 1.1 and below are intentionally not supported, closing off a class of legacy downgrade attacks independent of the PQC question.

Because the hybrid handshake terminates entirely at the Gateway, the connection between the Gateway and the protected server behind it uses ordinary TLS 1.3 — the server does not need oqs-provider, a PQC-aware certificate, or any part of the post-quantum trust boundary. The Gateway uses a separate client-facing certificate, which may be self-signed or CA-issued depending on deployment requirements (Section 7

documents a self-signed pilot configuration); the protected server's existing certificate remains unchanged throughout deployment.

Certified Algorithms, Honestly Scoped

The cryptographic module integrated into the Shelby PQC Gateway Series implements algorithms specified in NIST FIPS 203 (ML-KEM), FIPS 204 (ML-DSA), and FIPS 205 (SLH-DSA) — the three algorithms standardized under NIST's post-quantum cryptography program. The production configuration validated in this white paper activates ML-KEM for hybrid key exchange, the layer most directly exposed to Harvest Now, Decrypt Later risk. ML-DSA and SLH-DSA — post-quantum digital-signature algorithms — are present within the same module and are reserved for a future migration once dependent infrastructure (PQC-capable TPM signing and certificate-authority tooling) reaches production maturity; see Section 9. This paper describes the module's algorithm implementation; it does not assert a specific CAVP or CMVP validation certificate number, and readers requiring formal module-validation evidence should request current certification status directly from Elgens.

4.6 Hardware Root of Trust — TPM 2.0 Private Key Sealing

For environments — healthcare foremost among them — that require the Gateway's TLS private key to never exist in plaintext on disk, while also requiring the appliance to boot completely unattended, the Gateway implements TPM-sealed key protection:

- The plaintext private key and certificate are supplied on a customer-controlled USB drive and read directly by the provisioning process; the private key is never written to the Gateway's disk in plaintext at any point.
- The private key is encrypted with a randomly generated password (AES-256-CBC, PKCS#8).
- That password — not the private key itself — is sealed into the onboard TPM 2.0 chip using a persistent handle.
- At boot, a systemd oneshot service (ordered before Nginx starts) unseals the password from the TPM into tmpfs (memory-backed, never persisted to disk); Nginx decrypts the key into memory at startup.

This design has been validated through repeated full power-cycle testing: the Nginx service reaches active (running) fully unattended, the hybrid ML-KEM handshake is unaffected, and an automated post-boot scan confirms no plaintext private-key material remains on disk.

Security Boundary — Stated Honestly

What TPM sealing protects against: physical theft of the disk or the entire appliance (the sealed secret cannot be unsealed on a different TPM without the required authorization policy and hardware context it was sealed to) and copying files to a different machine (TPM sealing binds to the physical chip, not the data).

What it does not protect against: an attacker who obtains root/sudo access to a running system. Because full unattended boot is a hard requirement, this TPM seal is not bound to PCR boot-state measurements and carries no additional authorization password — a root-level process can read the unsealed password from tmpfs, invoke the TPM directly, or extract the key from Nginx's own process memory. This is the shared ceiling of every software-mediated key-protection scheme, not a gap unique to this design. Where a stricter threat model is required, the architecture would need to shift to a key that never leaves the TPM at all, performing the signing operation on-chip — a model that trades away performance (materially higher per-handshake latency) and is not yet supported for post-quantum signature operations on current mainstream TPM 2.0 silicon.

4.7 High Availability, Hardware Bypass, and Network Resilience

Gateway appliances that sit in line with production traffic raise a single question every network architect must answer before deployment: what happens to traffic when the Gateway itself fails? The Shelby PQC Gateway Series addresses this with two independent, complementary fail-safe layers, plus a physically separated management plane.

4.7.1 Management-Plane / Data-Plane Separation

Production units are provisioned with three physically separate network interfaces — External (client-facing), Internal (server-facing), and a dedicated Management interface used exclusively for SSH and the Web management console — matching the bridge-mode implementation validated in Section 2.3, where the client- and server-facing ports form a single Layer-2 bridge and the management interface remains entirely outside that bridge. In the standard hardware configuration (Appendix A), the onboard 2.5 GbE port is dedicated to the out-of-band Management interface, while the dual-port PCIe expansion card (10 GbE or 25 GbE, hardware-bypass capable) provides the External and Internal bridge data-plane ports. The onboard firewall (UFW) restricts SSH and the dashboard port strictly to the Management interface, so an administrator

troubleshooting or reconfiguring the appliance can never accidentally interrupt live production traffic.

4.7.2 Two Fail-Safe Layers, Not One

Failure Domain	Protection Mechanism	Nature	Gap If Used Alone
Software crash, CPU lockup, individual interface failure	Active/Passive HA clustering (state-synchronized dual-Gateway pair)	Logical failover	Cannot help if both units lose power at once
Total power loss, kernel panic, catastrophic hardware failure	Hardware Bypass (mechanical relay)	Physical fail-safe	Will not trigger on its own for a unit that is powered but "hung," unless paired with a watchdog

Neither mechanism alone is sufficient for a mission-critical deployment. The Gateway's High Availability architecture pairs two units in bridge (transparent) mode, each equipped with a hardware-bypass network interface, coordinated through Active/Passive state synchronization. Logical HA recovers instantly from software failures; physical bypass keeps the wire alive even in the worst case where both paired units lose power simultaneously — degrading gracefully from "protected" to "unprotected but connected" rather than failing to a full network outage.

4.7.3 Why an Ordinary Network Card Is Not Enough

- Link state can lie. Residual voltage or PoE-adjacent coupling on a standard network interface can leave a peer switch reporting "link up" even after the device behind it has lost power.
- The "zombie" failure mode. A kernel panic or a fully locked CPU leaves the appliance electrically powered, its network interface still showing green, while the operating system silently stops forwarding or inspecting any traffic — in practice the most common real-world failure mode for an inline appliance. A hardware-bypass NIC solves it with an onboard watchdog timer: a missed software check-in trips the mechanical relay in under a second.
- A manual last resort. Even if both paired Gateway units go down simultaneously, a hardware-bypass-equipped unit can mechanically bridge its own ports, turning

what would otherwise be a complete network outage into a temporary, unprotected-but-connected pass-through.

4.7.4 Active/Passive Only — A Hard Constraint

Hardware Bypass is only safe alongside Active/Passive clustering; it must never be combined with an Active/Active (load-balanced) topology. In an AA deployment, a bypassed unit's traffic and its surviving partner's takeover traffic can both remain "open" at the same instant, producing asymmetric routing and, worse, traffic that completes a round trip through the bypass wire entirely uninspected. For this reason, the Shelby PQC Gateway HA cluster operates strictly Active/Passive whenever Hardware Bypass is engaged, so that at any given instant exactly one path — either the bypass wire or the surviving Active unit — is open, never both simultaneously.

4.7.5 Configurable Fail-Open / Fail-Closed Behavior

Operators select, per deployment risk tolerance, how a failed unit should behave: Fail-Closed — the failed unit forces its own ports link-down, so the standby unit cleanly and immediately takes over 100% of traffic with full PQC-TLS inspection intact (the default posture for most deployments); or Fail-Open — the hardware relay mechanically bridges the wire so raw, uninspected traffic keeps flowing rather than the network dropping entirely (reserved for segments where any interruption is judged worse than a temporary loss of inspection). This selectable behavior is a capability of the Gateway's programmable bypass NIC, and the operational trade-off of each mode is documented explicitly in the deployment guide rather than left for the customer to discover mid-incident.

4.7.6 Dedicated Heartbeat Interconnect

Paired Gateway units require a direct, dedicated heartbeat connection. A redundant dual-link is recommended — carrying both liveness detection and session-state synchronization, independent of the production switching fabric. Relying on the shared production switch to also carry heartbeat traffic risks a split-brain condition, where both units independently and incorrectly conclude their partner is down and simultaneously asserting themselves as Active.

4.7.7 Syslog Integration for Coordinated Network-Wide Redundancy

The Gateway's Syslog-compatible connection- and access-log export lets the customer's own monitoring stack — a SIEM, a NOC dashboard, or an upstream firewall cluster running its own redundant-protocol control plane — factor Gateway health

directly into broader network failover decisions, rather than treating the Gateway as an isolated appliance the rest of the network cannot see into.

4.8 Management Console

Operational visibility for an inline security appliance carries an inherent trade-off: a fully configurable remote management interface improves day-to-day operability but also expands the appliance's own attack surface. The Gateway's management console — internally designated Shelby-GD — reflects a deliberate choice on that trade-off: it began as a read-only monitoring dashboard and has since grown into a fuller management console while retaining conservative security defaults throughout its expansion.

- Bilingual (English / Traditional Chinese) Web UI, with language preference bound to the user account rather than the browser.
- Live operational visibility: service status, TLS configuration, certificate expiry countdown, disk-encryption status, and system resource usage.
- Certificate lifecycle management scoped to the TPM boundary described in Section 4.6 — renewal and CSR-based issuance, never raw private-key handling.
- Mutual SSH / console recovery paths, so neither access path becomes a single point of lockout.
- Structured log export (JSON / CSV) built on the system's native systemd journal, and a fixed-English audit trail for every administrative action regardless of the operator's interface language.
- CLI access for scripted operations alongside the Web UI.

5. Measured Performance and Operating Limits

All figures in this section were measured in internal engineering tests conducted on dedicated physical hardware — not simulated, not extrapolated, and not an independent third-party validation — using a Gateway and client host connected over real network segments.

5.1 Handshake Latency Overhead

A controlled comparison ran 1,000 connections through each of two key-exchange configurations — the hybrid ML-KEM group and classical X25519 — at a concurrency of 20, measuring complete end-to-end latency (process start, TCP connection, TLS handshake, HTTP response).

Metric	ML-KEM Hybrid (X25519MLKEM768)	Classical (X25519)	Difference
Success rate	100.00%	100.00%	-
Average	85.53 ms	83.60 ms	+1.93 ms (+2.3%)
P50 (median)	85 ms	83 ms	+2 ms
P90	91 ms	89 ms	+2 ms
P95	93 ms	91 ms	+2 ms
P99	96 ms	97 ms	-1 ms (within margin of error)

A separate browser-based measurement, isolating only the TLS handshake phase, recorded a single-sample overhead of approximately 4.13 ms. Across both methodologies, the conclusion holds: post-quantum protection adds roughly 2 ms — about 2.3% — an overhead that is small relative to the end-to-end latency observed in the test environment.

5.2 Concurrency Validation — Bridge-Mode Production Architecture

In early August 2026, Elgens presented the Shelby PQC Gateway Series at an industry seminar attended by IT leadership from several of Taiwan's leading medical centers, including National Cheng Kung University Hospital, Veterans General Hospital, and Chi Mei Medical Center. During the Q&A session, the Information Systems Director of Chi Mei Medical Center's Intelligent Healthcare Center posed a specific, data-backed

question: based on the hospital's own operational data, peak-hour traffic could reach as many as 20,000 concurrent connections — could the Gateway sustain that load reliably? That figure became the original validation target for the load-testing program, rather than an arbitrary round number chosen for a benchmark report.

That target was set, and first validated, during the Stage 2 routed-mode phase (Section 2.2). Once the architecture moved to the finalized bridge-mode design (Section 2.3), the concurrency test was repeated end-to-end on the new architecture — and validated well beyond the original target. Test conditions: multi-source-IP client generation (distinct source addresses per test process, needed once a single source address's available ephemeral ports become the limiting factor rather than the Gateway itself), a clean reboot of the Gateway before each concurrency tier (to eliminate residual connection-table state from contaminating the next tier's result), and failure defined as a refused, reset, or timed-out connection. Under these conditions, the bridge-mode architecture sustained the following:

Concurrent Connections	Result
50,000 - 110,000	0.00% failure
120,000	0.66% failure
130,000	3.10% failure
140,000	6.74% failure
150,000	26.00% failure
160,000	84.61% failure

The system sustained the hospital-reported operational peak more than five times without observed connection failures under the stated test conditions. Beyond 110,000 concurrent connections, the system degrades gradually and then sharply rather than failing abruptly at a single cliff: the true collapse band falls between 140,000 and 160,000 concurrent connections. Reporting this full curve, rather than only the clean result, reflects this paper's consistent approach of stating operating boundaries explicitly rather than only publishing favorable numbers.

5.3 Sustained-Churn Endurance Test

A separate methodology validated a different property: whether the ML-KEM handshake mechanism remains stable under sustained, high-rate connection churn — every request establishing a brand-new connection rather than reusing an existing one, simulating a continuous stream of new users rather than a fixed set of persistent sessions. A five-minute run targeting over one million fresh handshakes completed 1,557,584 fresh ML-KEM handshakes (55.8% above the 1,000,000 target), at a sustained throughput of 5,180.18 req/s and an observed error rate at or below 0.05%.

6. Deployment Methodology

Beyond the appliance itself, the Gateway's deployment methodology is built around a consistent principle established in Section 1.3: minimize what has to change on the protected side, and make the decision points explicit rather than assumed.

6.1 Choosing the Insertion Point

Candidate Location	Traffic Profile	Assessment
Inter-core-switch trunk (Layer 2, multi-VLAN)	Mixed, multi-tenant	Not suitable - wrong technical layer; would become a single point of failure for the entire segment
Firewall passthrough segment (Layer 2 bridge, single purpose)	Single-purpose, externally facing	Typically, the best-fit insertion point
General firewall LAN interface	Mixed internal/external	Not suitable - traffic not yet insulated to the target destination
Firewall WAN interface fronting a third-party CDN	Single-purpose, external dependency	Conceptually workable, contingent on the CDN itself supporting PQC-TLS

6.2 Traffic Redirection Method

The routed-mode phase of the architecture (Section 2.2) supported three traffic-redirection methods, summarized here for completeness; the finalized bridge-mode architecture (Section 2.3) requires none of them, since it is inserted as a physically transparent bridge rather than redirected to at the IP layer.

Method	Server-Side Change	Upstream Change Required
DNS redirection	None	DNS record
Firewall NAT/VIP redirection	None	NAT rule (may require cross-team coordination)
IP swap (Gateway assumes the server's original IP)	One-time IP change	None - upstream/DNS entirely untouched

6.3 Rapid Multi-Site Replication

Production rollout is built on a two-part model: a hardware-independent "golden image" (Ubuntu, the compiled liboqs/oqs-provider stack, Nginx, and supporting services) restored onto each new unit's disk, followed by a first-boot provisioning script that performs only the machine-specific steps a golden image cannot pre-bake — detecting each physical interface's role via cable-pull testing, regenerating the machine's unique identity and SSH host keys, collecting site-specific IP addressing, and, on every individual unit, generating and TPM-sealing that unit's own private key locally, a step that by design can never be part of a shared image, since it is bound to each unit's own physical TPM chip. This lets a verified configuration be replicated across many sites quickly, while keeping every unit's cryptographic identity unique to its own hardware.

7. Case Study: Regional Hospital Outpatient Registration System

The following describes an actual field engagement; the specific institution, IP addressing, and internal network identifiers have been anonymized at the customer's request. This engagement was conducted during the Stage 2, Local TPM + Routed Mode phase of the architecture described in Section 2.2 — it is presented here as the field validation of that stage, and as the basis for the transition to the finalized bridge-mode architecture described in Section 2.3.

7.1 Topology Analysis

Engagement began with the customer's own hand-drawn network topology diagram, covering core switches, aggregation-layer switches, a load balancer, and a dual-firewall pair. Rather than treating the diagram as authoritative, the analysis proceeded methodically: each device's role and connections were confirmed one layer at a time; ambiguous cabling was verbally confirmed with the customer's own network staff rather than assumed; and DNS resolution and live TLS state for externally facing services were independently queried to cross-check the diagram against what the network was actually doing.

7.2 Insertion Point Selection

Of the candidate locations evaluated using the general framework in Section 6.1, the firewall passthrough segment — a Layer-2 bridge carrying only the single, externally facing outpatient-registration traffic flow — was selected: it matched the Gateway's technical layer, carried homogeneous single-purpose traffic, and required coordination between only the two parties directly involved.

7.3 High-Availability Design — An Engineering Correction Worth Documenting

The initial HA design assumed a single Gateway with four physical network ports — two connected to each of the customer's paired core switches, using LACP link aggregation in an active-active configuration. This design is valid only if the two core switches are themselves configured as a virtualized stacking pair, which was confirmed to be the case here, so the design was sound on paper. Physical verification of the actual deployed hardware, however, showed this unit had two physical network ports, not four — a configuration variant distinct from the three-interface standard reference design in Section 4.7.1, and one that made a single-unit four-port LACP configuration impossible.

The design was revised to a two-Gateway Active/Passive pair instead: each unit's two ports connect to a single core switch (no cross-connection), so a core-switch failure cleanly takes only the Gateway attached to it offline, leaving the partner unit completely unaffected. The two units then ran VRRP/keepalived for Active/Passive failover, with both the external- and internal-segment virtual IPs bound together so they always fail over as a pair. The lesson generalizes beyond this engagement: a design that is logically sound can still be invalidated by a specific hardware constraint, and the correction was straightforward once the mismatch was caught during physical verification rather than after deployment.

7.4 Fallback, Rollback, and Honest Disclosure

Two operational fallback options were prepared for the initial rollout period, reflecting a deliberate trade-off between recovery speed and infrastructure dependency: a manual-rollback plan (a single DNS record pointed at the Gateway, TTL shortened to 60-300 seconds during the trial period, with a documented manual procedure for the hospital's own IT staff to revert DNS if needed), and a health-check-driven automatic-failover plan, the latter requiring the protected server's real IP to remain reachable from the internet and DNS infrastructure capable of health-check-driven switching — a capability the hospital's own self-hosted DNS did not support without additional engineering work.

Two further points from this engagement are worth stating plainly rather than glossing over: extending source-IP-preserving traffic forwarding required the protected server's default gateway to be pointed at the Gateway's internal-segment virtual IP — a one-time, network-layer parameter change, not a modification to the server application itself, but a genuine change nonetheless (the finalized bridge-mode architecture in Section 4.1 removes this requirement entirely); and the engagement used a self-signed certificate and an internal test network segment, in a hospital field pilot and engineering validation configuration, not yet a CA-issued production certificate or full production cutover — both are the next milestones as this specific engagement transitions toward general availability on the finalized architecture.

8. Target Applications

The Gateway's non-intrusive deployment model and hardware-backed security architecture generalize beyond the healthcare case study above to any sector where existing server infrastructure is difficult to touch directly but faces the same HNDL exposure.

Sector	Why the Non-Intrusive Model Fits	Representative Scenario
Healthcare	Interconnected clinical systems (HIS/LIS/PACS) with long data-confidentiality lifetimes and limited IT capacity for direct migration	Outpatient registration and appointment portals
Finance	Regulatory data-retention requirements combine with transaction data whose confidentiality must outlast typical migration timelines	Transaction and payment-authorization API gateways
Telecommunications	Core-network infrastructure where downtime tolerance is near zero and PCIe-expandable 10/25 GbE throughput matters	Subscriber-facing portals and core-network management interfaces
Defense / Government	Long-lifetime sensitive communications are a textbook HNDL target; TPM-backed key protection and hardware bypass align with existing mission-critical resilience requirements	Long-lifetime classified or sensitive communication links
Critical Infrastructure	Legacy operational systems that cannot be modified directly, but whose perimeter can be upgraded independently	SCADA / OT remote-maintenance access points

9. Known Limitations and Roadmap

- Same-segment requirement is architectural, not temporary. The bridge-mode deployment and its hardware-bypass fail-safe (Section 2.3) require the Gateway and the protected server to share the same Layer-2 broadcast domain and IP subnet. This is a direct consequence of the resilience model chosen, not an unfinished feature; extending PQC protection to cloud-hosted or cross-subnet backends would require a distinct deployment mode with a different resilience strategy, evaluated on its own merits rather than as a simple extension of bridge mode.
- Non-HTTPS protocol coverage. Clinical protocols outside HTTPS (HL7/FHIR, DICOM/PACS) are not yet covered by the current reverse-proxy engine.
- Post-quantum signature migration. The onboard cryptographic module already implements FIPS 204 (ML-DSA) and FIPS 205 (SLH-DSA), but the Gateway's certificate signature currently remains classical (ECDSA), because mainstream TPM 2.0 silicon does not yet natively support post-quantum signing operations. Migrating the signature layer is planned as TPM/HSM ecosystem support matures.
- Production-certificate and full production cutover. The engagement in Section 7 remains in a pilot configuration; CA-issued certificate deployment and full production routing validation on the finalized bridge-mode architecture are the next milestones before general availability at that site.

Consistent with the engineering approach documented throughout this paper, Elgens considers stating these boundaries explicitly — rather than only publishing favorable results — part of the product's credibility, not a weakness of it.

10. Conclusion

The Shelby PQC Gateway Series demonstrates, through three engineering stages and an active field engagement, that post-quantum protection can be added to existing infrastructure without the cost, risk, and multi-year timeline of a direct server-side migration. Reaching the final bridge-mode-with-hardware-bypass architecture required abandoning a remote-signing concept, then re-engineering a working routed-mode design once its incompatibility with the strongest available physical fail-safe became clear — each transition driven by a specific, documented engineering constraint rather than a preference. Hybrid ML-KEM key exchange adds roughly 2 ms of latency; the finalized architecture sustains well beyond the concurrency figures reported directly by hospital IT leadership at industry scale; and TPM-sealed keys, hardware bypass, and Active/Passive clustering combine physical and logical fail-safes into a single resilience architecture — all while leaving the protected server, its certificate, and its URL completely unchanged.

11. Working With Elgens — Product Adoption in Three Steps

Organizations evaluating post-quantum readiness rarely need to choose between a multi-year migration program and doing nothing. The Shelby PQC Gateway Series is adopted through a structured, low-risk three-step process:

Step 1 — Discovery and Topology Assessment

Elgens engineers review the customer's existing network topology — insertion point, traffic-redirection method, and high-availability configuration — and provide a tailored deployment recommendation before any commitment is made, following the same evaluation framework documented in Section 6.

Step 2 — Proof of Concept

Scope and schedule are negotiated for a Proof-of-Concept deployment that validates the recommended configuration against the customer's own traffic patterns, integration requirements, and operational constraints — the same rigor applied to the engagement documented in Section 7.

Step 3 — Production Rollout

Once the PoC confirms fit, mass-production scale is confirmed and the deployment goes live, supported by the golden-image and first-boot provisioning model (Section 6.3) for rapid, consistent replication across multiple sites.

NEXT STEPS

Organizations interested in evaluating the Shelby PQC Gateway Series for their own infrastructure are invited to get in touch with Elgens for further information.

- Visit www.elgens.com.tw and leave a message to request a topology assessment or further technical information.

Appendix A: Technical Specifications

System

Category	Specification
CPU	Intel(R) Core(TM) 14th / 13th / 12th Gen Processors (LGA1700, 35 W TDP recommended)
System Memory	2x 260-pin DDR4 3200 MHz SO-DIMM, max. 64 GB
PQC Optimization	Designed for high-performance post-quantum cryptography gateway applications
External I/O	1x HDMI 2.0b, 2x DisplayPort 1.4a, 1x line-out, 1x mic-in, 2x 2.5 GbE RJ45 LAN, 4x USB 3.2 Gen2 (rear), 2x USB 2.0 (optional front)
Expansion	1x PCIe Gen4 x16 slot via 1U riser card, supporting 10 GbE / 25 GbE network interfaces and hardware-bypass configurations (customer infrastructure must support SNMP/IP management for bypass deployment)
Storage & Expansion	1x M.2 Key M (2242/2260/2280, PCIe Gen3 x4 NVMe SSD); 1x M.2 Key M (2242, PCIe Gen4 x4 / SATA3 SSD); 4x SATA3 (6 Gb/s, RAID 0/1/5/10)
Operating System	Ubuntu Linux
Security & Management	TPM 2.0 + Secure Boot + LUKS disk encryption; PQC-enabled hybrid TLS with TLS 1.3 / TLS 1.2 compatibility; security audit logging & Syslog SIEM export; High Availability deployment support

Environment

Category	Specification
Power Input	300 W 1+1 hot-swappable redundant power supply (100-240 VAC)
Dimensions	431.8 (W) x 321.6 (D) x 44.0 (H) mm
Certification	FCC / CE Class A certified
Operating Temperature	0C to 45C
Storage Temperature	-20C to 60C
Humidity	10% to 90% (non-condensing)

Software

Category	Specification
Core Engine	Reverse proxy, PQC TLS module, legacy TLS support, ML-KEM (FIPS 203)
Hybrid Handshake	Automatically negotiates PQC-enabled TLS with compatible endpoints and maintains legacy TLS compatibility when PQC support is unavailable
Certificate Management	Independent Gateway certificate for handshake identity verification; the protected server certificate remains unchanged
Audit Logging	Connection log, access log, Syslog-compatible forwarding, CSV / JSON export
Management Interface	Web UI, CLI, High Availability (HA) support

Ordering Information

Model	CPU	Network Interface	Memory	Storage
Shelby-G5135	Core(TM) i5-13500TE	10 GbE	16 GB DDR4	128 GB SATA SSD
Shelby-G7137	Core(TM) i7-13700TE	25 GbE	16 GB DDR4	128 GB SATA SSD

Appendix B: Key Figures Quick Reference

Figure	Meaning
+2.3% (+1.93 ms)	ML-KEM handshake overhead versus classical key exchange
100%	Success rate across 1,000 ML-KEM connection trials
100,000+	Concurrent connections sustained with 0% failure (finalized bridge-mode architecture, Section 5.2)
140,000 - 160,000	Concurrency band where degradation becomes severe (collapse band, Section 5.2)
1,557,584	Fresh ML-KEM handshakes completed in a 5-minute endurance test
≤ 0.05%	Observed error rate in the million-scale endurance test
5,180 req/s	Sustained fresh-connection establishment throughput
FIPS 203 / 204 / 205	NIST post-quantum standards implemented in the onboard cryptographic module
AES-256-CBC	Private-key encryption algorithm used prior to TPM sealing
X25519MLKEM768	Hybrid ML-KEM key-exchange group in use (IANA code 0x11EC)