HAMR-FORGE

FORGE OS · PUBLIC TECHNICAL WHITEPAPER

FORGE OS Platform Overview

A single-document introduction to the FORGE OS platform for program, security, and network evaluators

PUBLIC RELEASE

VALIDATED BASELINE V0.9.173 · FIRST EDITION

PUBLISHED BY HAMR-FORGE, LLC · AUGUST 2026

Executive Abstract

FORGE OS is a self-hosted secure networking platform from HAMR-FORGE, LLC: a centralized **HUB** control point and autonomous field **SPOKE** appliances that bond multiple commodity WAN services — fiber, cellular, satellite, anything that routes IP — into one mutually authenticated, encrypted transport, with dynamic routing above it and fail-closed behavior beneath it. The customer owns and operates every component; no HAMR-FORGE cloud sits in the data path or the control path.

This overview is the single-document introduction for program evaluators, capture teams, and technical directors. It summarizes the whole platform in one read and points to the four-paper technical series wherever a claim deserves deeper scrutiny. Everything stated here as current is the behavior of FORGE OS v0.9.173, the validated pre-release baseline.

Intended audience. Program managers, acquisition evaluators, enterprise and security architects, and technical leadership making a first assessment.

> **Publication boundary: v0.9.173 capability is stated as current. Future work is labeled ROADMAP — NOT INCLUDED IN v0.9.173. The term *validated* refers to the project's own release gates and testing; it does not imply third-party certification, government approval, product-level FIPS validation, or general availability.**

Contents

1. The Operational Problem

Remote and austere sites rarely fail in a tidy way. Individual carriers degrade, disappear, return with different addresses, or stay technically reachable while delivering unusable latency. The central facility can itself restart or drop offline. Most solutions to this problem either accept a vendor cloud in the middle — unacceptable for many government and enterprise deployments — or leave the field site as a network-engineering exercise assembled from unrelated parts.

FORGE OS treats it as an appliance problem. The operator works with two explicit roles — HUB and SPOKE — while the platform coordinates transport, routing, isolation, enterprise integration, telemetry, and recovery behind one management interface. A field site that loses its overlay contains its protected traffic rather than leaking it to whatever default route remains; a site that loses one of three uplinks keeps its session moving on the others.

2. The Platform at a Glance

The HUB is the customer's fleet authority: it terminates spoke tunnels, runs the mesh certificate authority that spokes enroll against, coordinates BGP or OSPF routing, distributes service and QoS policy, collects audit evidence, and holds each site's recovery credentials in encrypted escrow. Every SPOKE remains locally operable when the management plane or any external service is interrupted — WAN health, isolation, routing, and reconnection all run on the appliance itself.

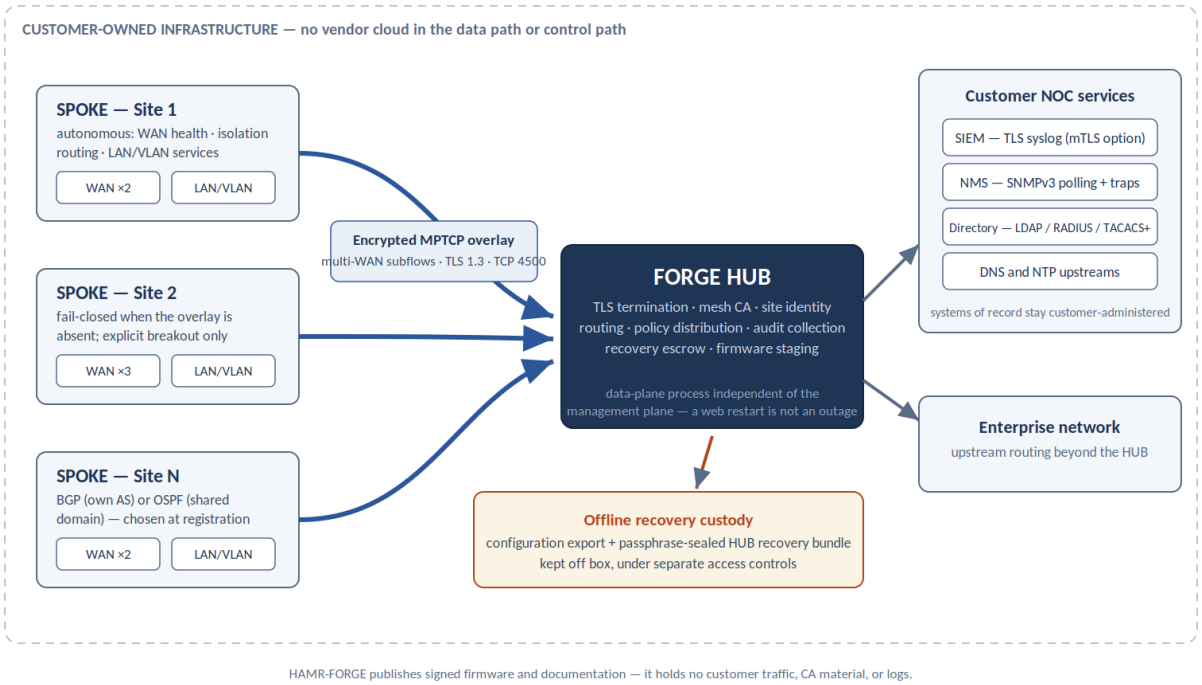

Figure 1 — platform architecture — customer NOC and enterprise services, HUB, spokes with bonded WAN paths, routed overlay

Responsibility boundaries are explicit. The customer owns placement, upstream routing, identity, DNS/NTP, SIEM/NMS, backups, and recovery custody. HAMR-FORGE publishes signed firmware, documented behavior, and validation guidance — and holds no customer traffic, no private CA material, and no operational logs.

3. How Traffic Moves

The platform keeps three planes distinct. The *underlay* is each WAN service as delivered, with its own addressing, NAT, and carrier policy. The *transport* is one authenticated MPTCP session per site over TCP 4500: the first reachable WAN creates it, every additional reachable WAN joins as a subflow, and a surviving subflow preserves the session while a failed path is repaired. The *overlay* is a routed point-to-point tunnel above that session, carrying only prefixes the operator has explicitly advertised, under BGP (independent site AS) or OSPF (shared routing domain).

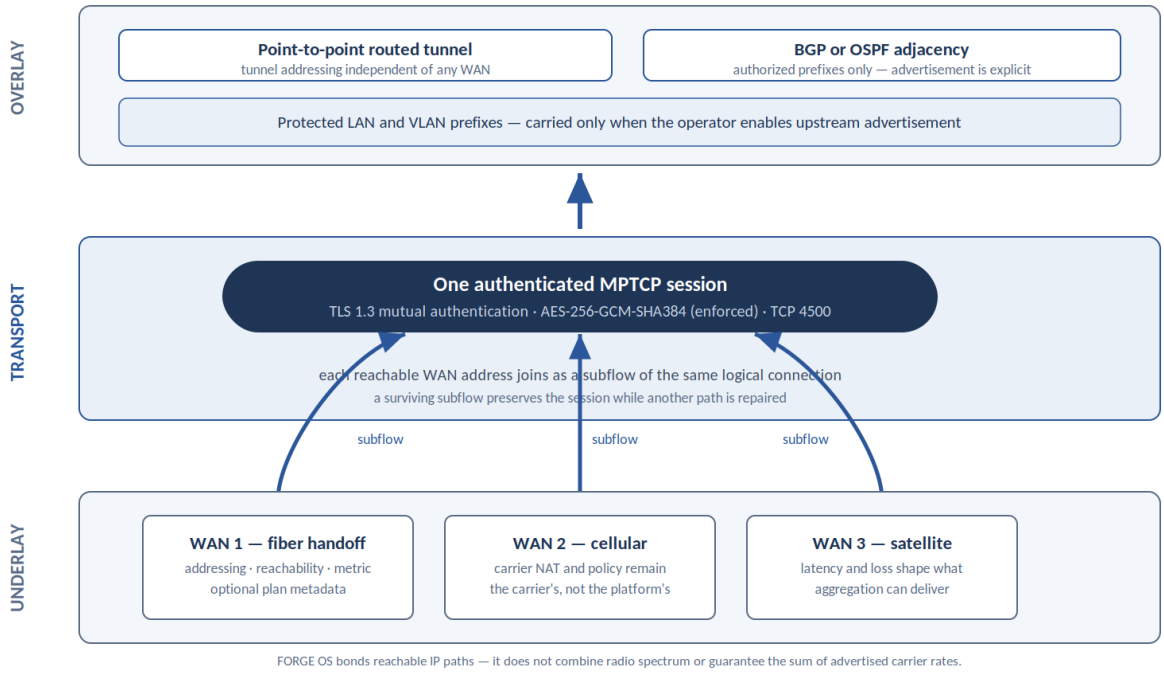

Figure 2 — layered model — WAN underlays feeding one authenticated MPTCP session, routed overlay above

Transport security is specific, not generic: TLS 1.3 mutual certificate authentication with ECDSA P-384 identity, and **AES-256-GCM-SHA384 as the required cipher suite** — selected for CNSA 1.0 alignment and enforced at client advertisement, server selection, and runtime verification, with no downgrade path. Cryptographic operations in the data path execute within a FIPS 140-3 validated cryptographic module (CMVP certificate #5247); FORGE OS itself is not separately certified.

The claims discipline matters as much as the claims: FORGE OS bonds reachable IP paths. It does not combine carrier radio spectrum, does not promise the arithmetic sum of advertised carrier rates, and does not eliminate congestion, loss, or policy imposed upstream. Multi-WAN availability, subflow aggregation, per-flow throughput, aggregate throughput, routing convergence, and application continuity are six different properties, and the technical series treats them separately.

4. Security Posture

Security is layered, and each layer has a stated limit — the Security Architecture paper's residual-risk table is part of the design, not an apology for it.

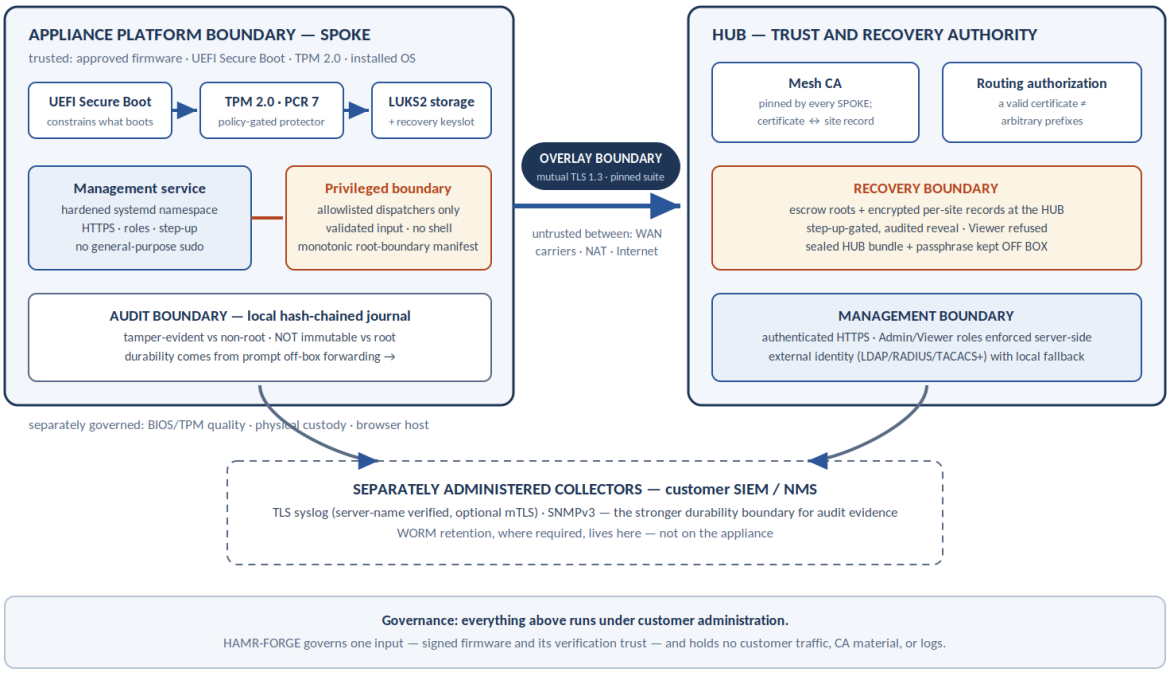

Figure 3 — trust boundaries — platform, management, overlay, recovery, and audit domains with the authority that governs each

- **Boot and storage.** UEFI Secure Boot constrains what boots; a TPM 2.0 protector bound to PCR 7 policy unlocks LUKS2 full-disk encryption; an independent recovery credential covers the day the TPM policy can't be satisfied. A removed drive is unreadable elsewhere.
- **Privileged execution.** The management service runs in a hardened systemd namespace without general-purpose sudo. Mutating operations cross allowlisted dispatchers that validate operation, arguments, and paths before any root action. A monotonic root-boundary manifest prevents a validly signed *older* package from restoring weaker privileged helpers during downgrade.
- **Access.** Local Admin and Viewer roles — Viewer read-only and server-enforced — plus RADIUS, verified LDAP/Active Directory, and TACACS+ integration, with local accounts retained for continuity. Sensitive recovery operations require Admin role and password step-up, and denials are themselves audited.
- **Audit.** Security-relevant events append to a hash-chained local journal, forwarded off box over TLS syslog to the customer's SIEM. The public wording is deliberate: *local integrity chain plus off-box durability* — not "locally tamper-proof," because a root attacker on the box is outside that boundary.
- **Firmware.** Ed25519-signed packages, verified before staging, installed into A/B slots with automatic health-checked rollback and the monotonic boundary described above.

5. Operating a Fleet

The management interface reports observed state, never configuration echoes: DNS answers a live lookup, NTP shows synchronization, syslog distinguishes connected from retrying from caught-up, routing shows protocol state, firmware shows both slots. *Configured*, *reachable*, *connected*, *established*, *synchronized*, *forwarding*, *degraded*, *unavailable* are distinct words with distinct evidence requirements.

At fleet scale, the HUB pairs an attention-oriented dashboard — site health with an explicit needs-attention count and live throughput — with an offline geographic **Fleet Map** served entirely from signed, customer-installed map packs. No external mapping service ever sees the fleet. Enterprise integration is deliberately conventional: TLS syslog to the SIEM, SNMPv3 to the NMS, directory-backed login, customer DNS and NTP upstream — the customer's existing systems of record stay authoritative.

Site lifecycle is a NOC workflow, not a field exercise: register the site name and routing role at the HUB, hand the field operator three values out-of-band (HUB address, CA fingerprint, site name), approve the enrollment when it arrives, and verify routing, path contribution, and escrow before calling it done. Revocation, re-enrollment control, and removal handle the other end of the lifecycle.

6. Resilience and Recovery

Failure handling is bounded and observable. One WAN lost: the session persists on surviving subflows while the missing path retries. All gateways absent at boot: the route guard withholds the HUB route until a gateway is resolvable, then reconnects without operator action. HUB restarted: spokes retry for as long as an endpoint is configured, and routing reconverges when the listener returns. Every scenario in the transport paper carries an operator-verifiable check, and the Secure Overlay paper closes with acceptance criteria a deployment team can run on day one.

Recovery is layered to match what failed. A field appliance's break-glass credential and LUKS recovery key escrow — encrypted — to the HUB after first authenticated connection, retrievable only by an Admin after audited step-up. The configuration export carries the fleet's operational configuration for routine backup. The passphrase-sealed HUB recovery bundle carries the region's cryptographic identity: restored together on replacement hardware during first-run setup, the two artifacts rebuild a HUB with the original CA and enrollment roster — no spoke re-enrolls, no field visit occurs.

7. Ownership and Deployment Model

FORGE OS ships as an appliance operating system for customer-qualified x86 hardware — the customer selects and qualifies platforms (UEFI, Secure Boot, TPM 2.0, storage and NIC behavior) rather than buying locked vendor boxes. Deployment guidance, an install guide, role-specific quick starts, and an administrator guide accompany the platform, and a release-gate and site-acceptance procedure exists for every firmware baseline.

The commercial posture mirrors the technical one: no subscription cloud, no phone-home license path (entitlement validation is offline), no vendor custody of anything the customer would need to recover, rotate, or audit. Signed firmware and its verification trust are the one input HAMR-FORGE governs.

8. Validation Status — Stated Plainly

v0.9.173 is a **validated pre-release baseline**: behavioral CI gates, virtual fleet exercises (HUB plus mixed BGP/OSPF spokes, failover, firmware forward/rollback, escrow continuity), physical-hardware validation of the Secure Boot/TPM/LUKS path, cold-boot and outage recovery testing, and a long-duration firmware-lock soak across live field spokes that is in progress at publication and informs the release-freeze decision.

What is deliberately *not* claimed: third-party certification, penetration-test completion, STIG/FIPS compliance mapping, general availability, or multi-HUB failover — those are roadmap, listed below, and the papers label them unambiguously. An evaluator who finds a claim in this series can expect to find the mechanism in the product and a way to verify it; the Security Architecture paper ends with assessment guidance written for exactly that purpose.

9. Roadmap — Not Included in v0.9.173

- Multi-HUB geographic resilience: multiple independently administered HUB destinations per SPOKE, deterministic failover, replicated readiness state, and operator-controlled failback.
- Independent third-party penetration test and remediation cycle.
- Customer-specific STIG/FIPS control mapping where contractually required.
- Optional external audit-head anchoring for deployments needing stronger local-at-rest tamper evidence.
- Additional guided remediation text and role-specific runbooks.

Roadmap items are architectural direction, not current capability, and must not be used as procurement or deployment claims.

10. Reading Guide — Where to Go Deeper

This overview compresses four technical papers, each written for a specific evaluator and each carrying its own boundaries, failure scenarios, and verification guidance.

If you are...	Read	It covers
A network or transport engineer	*FORGE OS Secure Overlay & Multi-WAN Transport*	Underlay/transport/overlay model, enrollment, MPTCP behavior, path recovery, QoS, failure scenarios, acceptance criteria
A security architect, assessor, or accreditation team	*FORGE OS Security Architecture, Trust Boundaries & Recovery*	Trust boundaries, boot/storage protection, privileged execution, audit integrity, recovery escrow, residual risk, assessment guidance, informational control-family cross-reference
NOC leadership or an operations engineer	*FORGE OS Fleet Operations & Enterprise Integration*	Site lifecycle, truthful health model, fleet monitoring, enterprise integrations, backup artifacts, firmware operations, incident workflow, readiness checklist
An architect or technical evaluator weighing the whole system	*FORGE OS Platform Architecture & Operational Resilience*	Roles and responsibility boundaries, reference architecture, transport and routing convergence, fail-closed behavior, firmware resilience, validation evidence

A complete operator documentation set — install guide, HUB and SPOKE quick starts, and an administrator guide, each verified against the shipping interface — accompanies the platform for deployment and operations staff.

11. Conclusion

FORGE OS's case to an evaluator is not a single feature; it is the way the pieces close over each other: bonded multi-path transport that survives carrier failure, routing that carries only what was authorized, sites that fail closed instead of leaking, firmware that can always fall back to a known-signed payload, recovery that assumes hardware will someday be lost, and telemetry that tells the NOC the truth. All of it runs under customer ownership, and all of it is written down with its limits attached — in this overview for the first read, and in the four-paper series for the deep one.

Document Control

Field	Value
Publisher	HAMR-FORGE, LLC
Platform	FORGE OS
Validated baseline	v0.9.173
Classification	Public release
Release date	August 2026
Edition	First edition — companion to the four-paper technical series
Claims boundary	No third-party certification or general-availability claim