HAMR-FORGE

FORGE OS · PUBLIC TECHNICAL WHITEPAPER

Security Architecture, Trust Boundaries & Recovery

Secure Boot, TPM-bound storage, authenticated transport, privileged boundaries, audit integrity, and disaster recovery

PUBLIC RELEASE

VALIDATED BASELINE V0.9.173 · SUPERSEDES V0.9.118 EDITION

PUBLISHED BY HAMR-FORGE, LLC · AUGUST 2026

Executive Abstract

FORGE OS uses layered controls rather than a single security claim: UEFI Secure Boot, TPM-bound LUKS2 storage, TLS 1.3 mutual authentication, explicit routing authorization, least-privilege helper dispatch, signed A/B firmware, a monotonic rollback boundary, local audit-chain integrity, off-box forwarding, and passphrase-sealed disaster recovery. This paper defines those controls and their limits for v0.9.173.

Intended audience. Security architects, assessors, incident responders, system owners, and accreditation teams.

> **Publication boundary: v0.9.173 capability is stated as current. Future work is labeled ROADMAP — NOT INCLUDED IN v0.9.173.**

Contents

1. Security Objectives

This paper describes the publicly releasable behavior of FORGE OS v0.9.173, the current validated pre-release baseline. Validation evidence cited here comes from the project's automated release gates, virtual-appliance exercises, physical hardware testing, and a long-duration firmware-lock soak across live field spokes. The term *validated* does not imply a third-party certification, government approval, product-level FIPS validation, or general-availability designation.

- Protect customer traffic between authorized appliances.
- Keep a removed storage device unreadable without a valid LUKS protector.
- Prevent routine management processes from receiving unrestricted root execution.
- Make security-relevant operations attributable and forwardable off box.
- Support recovery without requiring HAMR-FORGE custody of customer secrets.
- Fail closed when the trusted overlay is unavailable.

2. Trust Boundary Model

Boundary	Trusted elements	Untrusted or separately governed elements
Appliance platform	Approved firmware, UEFI Secure Boot, TPM 2.0, installed OS, protected root helpers	Unqualified BIOS/TPM implementations, malicious physical peripherals
Management	Authenticated HTTPS session, role checks, step-up controls	Browser host, local network, external identity availability
Overlay	Pinned HUB CA, approved SPOKE certificate, TLS 1.3 session, authorized prefixes	WAN carriers, NAT, public Internet
Recovery	HUB escrow roots, encrypted records, sealed recovery bundle, offline custody	Unprotected downloads, shared passwords, unavailable backups
Audit	Permission-restricted local chain and TLS forwarding	A root attacker on the same box; external collector administration

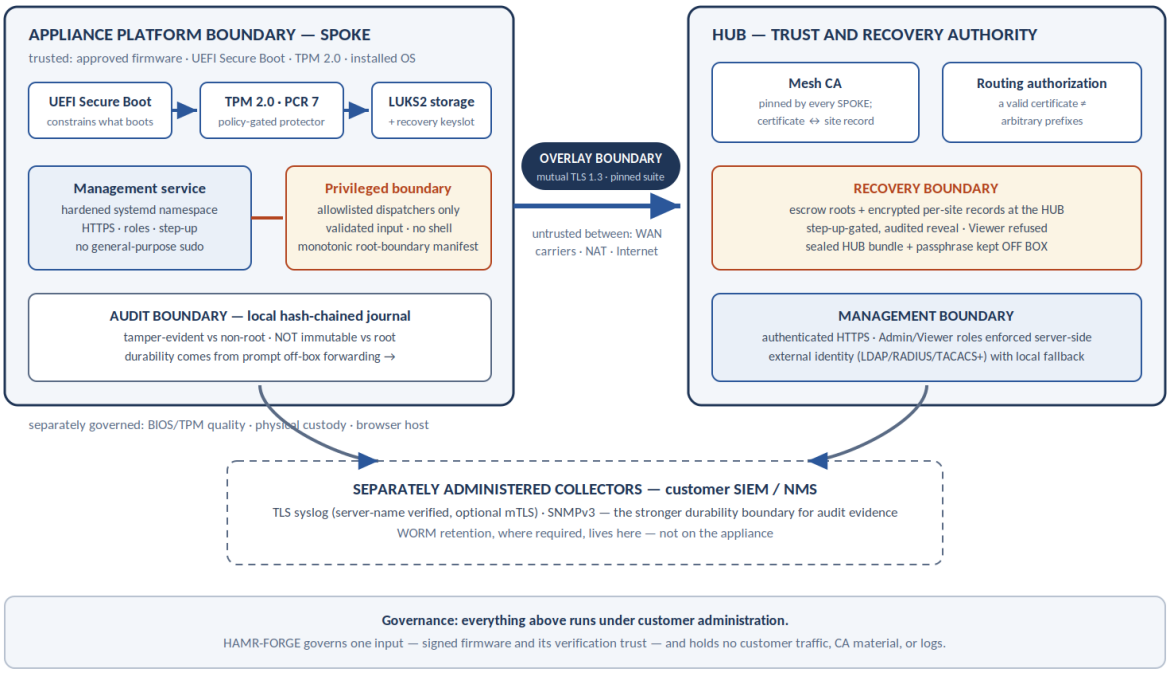

Figure 1 — trust boundaries — platform, management, overlay, recovery, and audit domains with the authority that governs each

3. Boot and Storage Protection

Four mechanisms interlock here, each covering the previous one's limit: UEFI Secure Boot constrains what may boot, the TPM releases a storage protector only when the platform's measured policy state matches, LUKS2 encrypts the disk that protector unlocks, and an independent recovery credential covers the case where the TPM policy can no longer be satisfied.

The encrypted-install path requires UEFI boot, Secure Boot, a TPM 2.0 resource-manager device, and an active SHA-256 PCR (Platform Configuration Register) bank whose values the installer can read — confirming, before any storage change begins, that the registers it will bind policy to are present and readable. The installer creates LUKS2 root storage, enrolls a transitional TPM protector, and converges to the installed-system PCR 7 policy after boot.

A separate random recovery credential protects an independent LUKS keyslot. Setup replaces the installer bootstrap credential before presenting the final recovery value. Validation confirms active recovery and TPM protectors, LUKS2 format, PCR 7 binding, initramfs unlock support, and absence of FORGE-format recovery credentials from installer artifacts.

> **TPM auto-unlock releases a protector only when policy conditions match. It is not the disk's raw volume key and does not make an extracted drive decryptable on another system.**

4. Physical-Access Boundary

Full-disk encryption protects data at rest when the appliance is powered off and the expected platform policy is unavailable. Once a trusted platform boots and unlocks storage, OS authorization and boot-loader protections become separate concerns. An attacker with physical control, firmware access, or an already-unlocked running system presents a different threat.

- Approved hardware must support controllable UEFI, Secure Boot, TPM 2.0, SHA-256 PCRs, and stable storage discovery.
- A GRUB password and broader STIG hardening may be appropriate for regulated deployments but are not claimed as certification in v0.9.173.
- Physical tamper protection, chassis locks, custody, and secure disposal remain deployment controls.

5. Transport Identity and Authorization

The SPOKE verifies the customer-controlled HUB trust anchor and establishes TLS 1.3 with mutual certificate authentication. The HUB maps the certificate identity to an expected site record. This protects the transport from unauthenticated peers while leaving routing authorization as an explicit, separate layer.

The tunnel cipher suite is pinned: AES-256-GCM-SHA384, selected for CNSA 1.0 alignment and enforced at three layers — client advertisement, server selection, and a runtime verification of the negotiated suite. A peer unable to negotiate it fails the handshake; no downgrade path exists. 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.

- Enrollment requires out-of-band verification of the HUB CA fingerprint.
- Certificate issuance is tied to the registered site identity.
- LAN advertisement is explicit; a valid certificate does not authorize arbitrary prefixes.
- Rejected or unknown connection attempts are observable at the HUB.

6. Privileged Execution

The web/API service runs inside a hardened systemd namespace and does not receive general-purpose sudo. Mutating operations cross narrow wrapper boundaries. Dispatchers validate operation names, argument shapes, path scope, ownership, and input grammar before invoking specific root functions — including all rendering and configuration input destined for privileged writes.

The privileged layer is protected by a *monotonic root-boundary manifest*: a recorded, verified inventory of the privileged dispatchers, wrappers, rollback logic, and firmware verification key that may advance with new releases but can never silently regress. Because the manifest is verified separately from the replaceable application slots, an older, still-validly-signed application cannot restore weaker privileged helpers during a downgrade.

Control	Purpose
Allowlisted dispatcher operations	No caller-selected executable or shell
Validated rendering and	Reject unsafe or ambiguous values before privileged write

Control	Purpose
configuration input	
Systemd sandbox	Read-only system with enumerated writable paths
Monotonic root-boundary manifest	Preserve hardened verifier, wrappers, and signing trust across downgrade

7. Authentication and Authorization

FORGE OS provides local Admin and Viewer roles and can integrate with RADIUS, verified LDAP/Active Directory, or TACACS+. Local accounts remain available when the external directory or HUB relay is unavailable. Sensitive recovery operations require Admin authorization and password step-up.

- Viewer is read-only and cannot reveal disaster-recovery material.
- Wrong step-up credentials produce a denied audit event.
- Successful reveal/export records actor, operation, result, and non-secret metadata.
- Secrets are write-only or masked in normal configuration views.

8. Audit Integrity and Off-Box Durability

Security-relevant events are appended to a JSON-lines journal with sequence number, previous hash, event hash, chain identity, boot identity, actor, result, target, and bounded non-secret details. A root-owned state file anchors the expected head. The local verifier checks ownership, mode, sequence, and hash continuity.

This is tamper-evident against non-root modification and useful for continuity validation. It is not immutable against an attacker who controls root and can rewrite both journal and head. TLS syslog forwarding to a separately administered collector supplies the stronger off-box durability boundary. WORM (write once, read many) retention, if required, belongs in that external logging architecture.

> **Accurate public wording: local integrity chain plus off-box durability. FORGE OS v0.9.173 does not claim locally immutable or WORM audit storage.**

9. Recovery Escrow

Three recovery artifacts exist, and they answer different failures. *Per-site recovery material* — a SPOKE's break-glass credential and LUKS recovery key — escrows to the HUB for the day a field appliance needs rescue. The *configuration export* carries the fleet's operational configuration, including each site's escrow record in encrypted form, so routine backups stay complete as sites are added — but it deliberately contains no plaintext secrets and cannot re-create the HUB's cryptographic identity. The *HUB recovery bundle* is the identity artifact: a passphrase-sealed package of the CA and enrollment roster that lets a replacement HUB adopt the region's identity during first-run setup. Routine backup is not disaster recovery; the export and the bundle are both required to rebuild a HUB without re-enrolling every spoke.

The HUB holds its escrow encryption identity and master key in permission-restricted persistent storage. A newly authenticated SPOKE transmits its recovery record through the authenticated control channel; the record is encrypted at rest on the HUB and associated with the site identity and generation. Authorized

administrators can reveal per-site values after password step-up. The identity-restore import that consumes the recovery bundle exists only in first-run setup; it is unavailable once initial setup is complete.

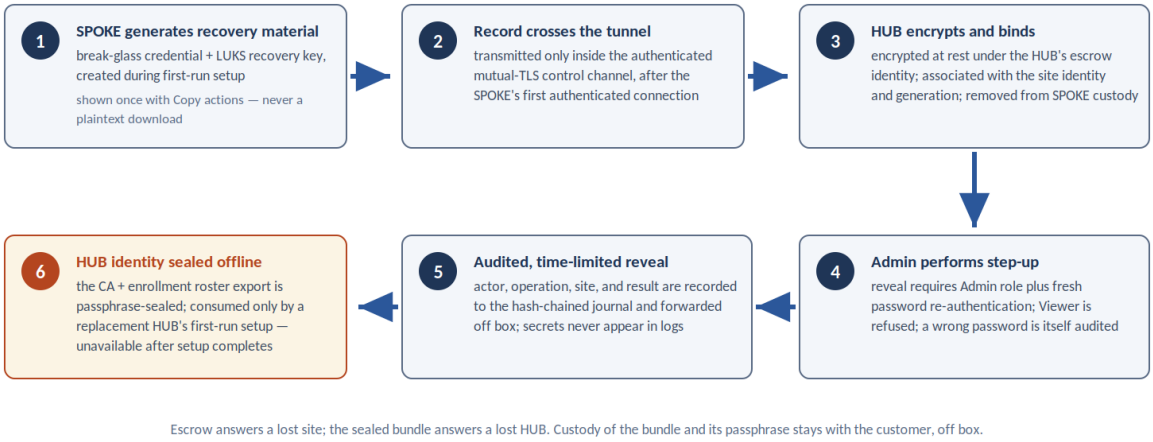

Figure 2 — recovery chain — from SPOKE-generated credentials through authenticated escrow to audited, step-up-gated reveal, with HUB identity sealed offline

- Recovery reveal is time limited and audited.
- No plaintext recovery download is offered at SPOKE setup conclusion; the UI provides copy actions.
- The customer must keep the HUB recovery bundle and passphrase off box and separate.

10. Firmware Supply and Rollback

Firmware packages are signed and verified before staging. A/B application slots provide a recoverable prior payload. The installer preserves the live data plane when its binary is unchanged and verifies execution of the new binary when it changes.

The signing key is a build-infrastructure crown jewel. Encryption, offline custody, controlled release ceremony, and issuer separation are operational controls outside the appliance codebase and must be completed before production shipment.

11. Residual Risk

Risk	Mitigation	Residual responsibility
Root compromise	Least privilege, systemd isolation, local chain, off-box logs	Rebuild/restore and external forensic correlation
Build-key compromise	Offline encrypted custody and controlled signing	HAMR-FORGE release operations
Physical unlocked system	OS access controls and operational custody	Customer physical security
External service outage	Local accounts and appliance autonomy	Customer DNS/NTP/identity/SIEM continuity
Unqualified hardware	Approved-hardware validation	Platform selection and lifecycle

Risk	Mitigation	Residual responsibility
		control

12. Roadmap — Not Included in v0.9.173

Roadmap items are architectural direction, not v0.9.173 capability. They are visually labeled and must not be used as procurement or deployment claims.

- Independent third-party penetration test and remediation cycle
- Customer-specific STIG/FIPS mapping where contractually required
- Multi-HUB replicated security and recovery state
- Potential external audit-head anchoring for deployments requiring stronger local-at-rest tamper evidence

13. Conclusion

FORGE OS reduces security risk through layered, independently verifiable controls rather than claiming the elimination of threats. Secure Boot, TPM-bound storage, mutual TLS, privileged wrappers, role-based access, audit chains, firmware signing, and recovery escrow each address a different threat inside a different trust boundary — and each has a stated limit: encryption protects powered-off storage, audit chains resist non-root tampering, escrow depends on custody, and signing depends on build-infrastructure discipline. The residual-risk table above is part of the architecture, not an apology for it. Assessors are invited to test these claims directly using the guidance that follows.

14. Assessment Guidance

- Verify Secure Boot and TPM state independently of the web UI.
- Inspect LUKS2 slots and the Clevis token-to-keyslot mapping (Clevis is the automated-decryption framework that binds a LUKS keyslot to the TPM policy) without exporting secrets.
- Exercise wrong-role, wrong-password, and expired-step-up cases.
- Disconnect the collector, generate events, restore it, and confirm queued delivery.
- Attempt a signed application downgrade and confirm the root boundary remains verified.
- Restore a HUB recovery bundle on a disposable replacement system and confirm fleet identity continuity.

Appendix A. Control-Family Cross-Reference (Informational)

The mapping below identifies where FORGE OS mechanisms are *conceptually relevant* to common control families. It is informational only: it is not a certification, a compliance claim, an assessment result, or a statement that any control is satisfied. Formal mappings are produced per customer under the roadmap's STIG/FIPS mapping work.

Control family	Conceptually relevant FORGE OS mechanisms
Identification & authentication	Local roles, RADIUS/LDAP/TACACS+ integration, step-up authentication for recovery operations (Section 7)
Access control	Admin/Viewer separation, server-enforced read-only Viewer, allowlisted privileged dispatch (Sections 6–7)
Audit & accountability	Hash-chained local journal, attributable events, TLS syslog forwarding, denied-action auditing (Section 8)
System & communications protection	Mutual TLS 1.3 with pinned AES-256-GCM-SHA384, certificate-scoped authorization, fail-closed isolation (Section 5)
Configuration management	Signed A/B firmware, monotonic root-boundary manifest, validated configuration input (Sections 6, 10)
Contingency planning	Escrowed site recovery, configuration export, sealed HUB recovery bundle, documented restore path (Section 9)
Supply-chain risk	Signed packages, verification before staging, signing-key custody controls (Section 10)

Document Control

Field	Value
Publisher	HAMR-FORGE, LLC
Platform	FORGE OS
Validated baseline	v0.9.173
Classification	Public release
Release date	August 2026
Supersedes	v0.9.118 edition (July 2026)
Claims boundary	No third-party certification or general-availability claim