RIFT-7 Secure Hardware Deployment Layer

OBINexus Computing - AEGIS Project Implementation

Version: 1.0.0-dev
Stage: Implementation Gate
Classification: Git-RAF Enforced
Author: AEGIS Core Engineering Team
Date: June 22, 2025

Executive Summary

The RIFT-7 stage represents the final governance enforcement layer, implementing hardware-level attestation and firmware integration for safety-critical deployment environments. This stage ensures cryptographic binding between compiled artifacts and target hardware platforms through TPM integration, BIOS signature validation, and AuraSeal chain verification.

Architecture Overview

Core Components

1. Hardware Attestation Module (HAM)

```
rift-7-ham.so.a → Trusted Platform Module Interface
├── TPM 2.0 Integration Layer
├── BIOS Signature Validation Engine
├── Hardware Security Module (HSM) Binding
└── Platform Configuration Register (PCR) Management
```

2. Firmware Governance Interface (FGI)

```
rift-7-fgi.so.a → BIOS/UEFI Integration Layer
├── Secure Boot Chain Validation
├── Firmware Policy Enforcement Engine
├── Hardware Root of Trust Integration
└── Platform Initialization Governance
```

3. AuraSeal Chain Validator (ACV)

```
rift-7-acv.so.a → Cryptographic Chain Management
├── Entropy Checksum Validation
├── Policy Tag Hash Verification
├── Cross-Stage Attestation Binding
└── Hardware-Bound Key Derivation
```

Git-RAF Integration Architecture

Commit Structure Enhancement

```
commit_hash: SHA-256(content + metadata + governance_vector)
governance_vector: {
    policy_tag_hash: BLAKE3(rift_policy_chain[0..6]),
    entropy_checksum: ChaCha20-Poly1305(hardware_entropy),
    aura_seal_t: Ed25519(hardware_private_key, commit_content),
    hardware_binding: TPM2_Quote(platform_state)
}
```

Repository Validation Chain

```
.git/raf/
├── governance_chain.sig    # Stage 0-6 policy inheritance
├── hardware_manifest.tpm  # TPM-bound platform attestation
├── entropy_pool.sealed    # Hardware-sealed randomness source
└── aura_chain.ed25519     # Cryptographic attestation chain
```

Hardware Integration Protocols

TPM 2.0 Integration Framework

c

```c
// RIFT-7 TPM Integration Interface
typedef struct {
    uint32_t pcr_selection[8];          // Platform Configuration Registers
    uint8_t  policy_digest[32];         # BLAKE3 policy chain hash
    uint8_t  entropy_seed[64];          # Hardware random source
    uint8_t  aura_signature[64];        # Ed25519 attestation signature
} rift7_tpm_context_t;

// Hardware attestation validation
int rift7_validate_hardware_context(
    const rift7_tmp_context_t* ctx,
    const uint8_t* compiled_artifact,
    size_t artifact_length
);
```

BIOS/UEFI Secure Boot Integration

c

```c
// Firmware governance hook interface
typedef struct {
    uint32_t firmware_version;          // BIOS/UEFI version identifier
    uint8_t  secure_boot_state;         # Secure boot validation status
    uint8_t  governance_policy[256];    # Stage 0-6 inherited policies
    uint8_t  hardware_fingerprint[32];  # Unique platform identifier
} rift7_firmware_context_t;

// Firmware policy enforcement
int rift7_enforce_firmware_governance(
    const rift7_firmware_context_t* fw_ctx,
    const char* deployment_target
);
```

Entropy Flow Architecture

Hardware Random Number Generation

```
Hardware Entropy Sources:
├── CPU RDRAND/RDSEED Instructions
├── TPM Random Number Generator
├── Hardware Security Module (HSM)
└── Platform-Specific Entropy Sources

Entropy Processing Pipeline:
RIFT-7-ENTROPY := ChaCha20-Poly1305(
    hardware_sources +
    stage_inheritance[0..6] +
    platform_measurement
)
```

Cryptographic Key Derivation

```
Platform-Bound Key Derivation:
hardware_master_key := TPM2_CreatePrimary(TPM_RH_OWNER, policy_template)
stage_derived_key := HKDF-BLAKE3(
    hardware_master_key,
    stage_context[0..6],
    platform_attestation
)
```

Deployment Validation Protocol

Pre-Deployment Validation Sequence

1. **Hardware Platform Verification**

   ```
   rift7_validate_platform() → TPM Quote + PCR Validation
   ```

2. **Firmware Governance Check**

   ```
   rift7_check_firmware_compliance() → Secure Boot + Policy Inheritance
   ```

3. **AuraSeal Chain Validation**

   ```
   rift7_validate_aura_chain() → Ed25519 Signature Chain Verification
   ```

4. **Cross-Stage Policy Inheritance**

   ```
   rift7_inherit_governance() → Stage 0-6 Policy Aggregation
   ```

Post-Deployment Monitoring

```
Continuous Validation:
├── Platform Configuration Register (PCR) Monitoring
├── Firmware Integrity Measurement
├── AuraSeal Chain Verification (Real-time)
└── Hardware Anomaly Detection
```

Error Handling and Attestation Failures

Hardware Attestation Failure Modes

```
RIFT7_ERR_TPM_UNAVAILABLE     → TPM 2.0 hardware not accessible
RIFT7_ERR_INVALID_PCR         → Platform configuration mismatch
RIFT7_ERR_FIRMWARE_UNTRUSTED  → BIOS/UEFI secure boot failure
RIFT7_ERR_AURA_BROKEN         → AuraSeal chain validation failure
RIFT7_ERR_ENTROPY_EXHAUSTED   → Hardware randomness source failure
```

Governance Escalation Protocol

```
Hardware Failure → Emergency Halt → Governance Assessment → Recovery Protocol
├── Automatic platform quarantine
├── Incident reporting to governance authority
├── Recovery validation requirements
└── Re-attestation procedures
```

Implementation Requirements

Development Dependencies

```
Required Hardware:
├── TPM 2.0 Compatible Platform
├── UEFI Secure Boot Capability
├── Hardware Security Module (Optional, Enhanced Security)
└── Platform with RDRAND/RDSEED Support

Required Software:
├── tpm2-tools (≥ 5.0)
├── OpenSSL (≥ 3.0) with Ed25519 support
├── BLAKE3 Cryptographic Library
└── ChaCha20-Poly1305 Implementation
```

Build Integration

makefile

```
# RIFT-7 Hardware Integration Build Target
rift-7-hardware: stage-0-6-complete
  @echo "Building RIFT-7 Hardware Deployment Layer..."
  $(CC) $(CFLAGS) -ltpm2 -lcrypto -lblake3 \
    src/rift7/hardware/*.c \
    -o build/rift-7-hardware.so.a
  rift7-validate-hardware-binding build/rift-7-hardware.so.a
  git-raf sign-hardware-attestation $@
```

Security Considerations

Threat Model Coverage

- **Hardware Tampering:** TPM-based platform attestation
- **Firmware Compromise:** Secure boot chain validation
- **Supply Chain Attacks:** AuraSeal cryptographic binding
- **Runtime Manipulation:** Continuous platform monitoring

Compliance Integration

- **FIPS 140-2 Level 3:** Hardware security module integration
- **Common Criteria EAL4+:** Platform evaluation requirements
- **NASA-STD-8739.8:** Safety-critical system compliance

Testing and Validation

Hardware Validation Test Suite

bash

```
# RIFT-7 Hardware Validation Protocol
rift7-test-suite:
├── tpm-integration-test.sh      # TPM 2.0 functionality validation
├── firmware-governance-test.sh  # BIOS/UEFI integration testing
├── aura-chain-test.sh          # Cryptographic chain validation
└── platform-attestation-test.sh # End-to-end hardware binding
```

Continuous Integration Requirements

- Hardware testing lab with TPM 2.0 platforms
- Automated firmware validation environments
- Cryptographic test vector validation

- Platform-specific attestation verification

Documentation and Maintenance

Operational Procedures

- Hardware platform onboarding procedures
- Firmware update governance protocols
- AuraSeal chain recovery procedures
- Emergency attestation bypass protocols (governance-approved only)

Next Implementation Target: rift-bridge.exe Governance Relay Interface

Stage Validation: ✅ RIFT-7 Architecture Documented
Git-RAF Status: Ready for hardware attestation binding
AEGIS Gate Status: Implementation Gate - Component Development Phase