SP-4 — NOVAK Implementation Requirements

Mandatory Rules, Controls, and Verification Steps for All NOVAK-Compliant Systems

Author: Matthew S. Novak
Version: SP-4 v1.0
Year: 2025
Series: NOVAK Protocol Standards (SP-1, SP-2, SP-3, SP-4)
Category: PBAS (Proof-Before-Action Systems)

---

0. PURPOSE

This document defines the **complete, binding implementation requirements** for any system claiming compliance with the NOVAK Protocol.

NOVAK is a PBAS-class system:
No action may execute unless its correctness is proven beforehand.

This standard defines the required:

- Data structures
- Hashing rules
- Execution process
- Validation steps
- Safety Gate logic
- PL-X physical integrity checks
- PS-X human fraud mitigation

- Logging & audit expectations
- Compliance testing
- Failure modes

Implementations MAY extend the protocol.
Implementations MUST NOT weaken, bypass, or remove any required controls.

---

1. NORMATIVE REFERENCES

NOVAK implementations MUST comply with:

- **SP-1 — Execution Integrity Standard**
- **SP-2 — Cryptographic Standard (HVET / EIR / RGAC)**
- **SP-3 — Safety Gate Standard (incl. PL-X / PS-X)**
- **NTM-1 — NOVAK Threat Model**
- **PBAS Category Definition**

These documents define the logical, cryptographic, physical, and regulatory foundations that SP-4 builds upon.

---

2. DEFINITIONS

A NOVAK-compliant implementation MUST define all components explicitly:

```
R = Governing Rule (deterministic)
D = Attested Input Data
O = Deterministic Output
I = Execution Identity
T = Timestamp (ISO 8601)
```

```
EIR = Execution Identity Receipt
HVET = Hash Verified Execution Token
RGAC = Recursive Global Audit Chain
PL-X = Physical Integrity Layer
PS-X = Psycho-Social Integrity Layer
SG = Safety Gate result (TRUE/FALSE)
```

All implementations MUST produce these artifacts correctly.

---

3. IMPLEMENTATION REQUIREMENTS OVERVIEW

The system MUST enforce all requirements:

Section	Requirement Type	Mandatory
§4	Deterministic Rule Enforcement	✔
§5	Input Attestation Requirements	✔
§6	Output Determinism Verification	✔
§7	HVET Construction	✔
§8	EIR Construction & Signing	✔
§9	RGAC Construction & Continuity	✔
§10	Safety Gate Enforcement	✔
§11	PL-X Physical Integrity Controls	✔
§12	PS-X Human Legitimacy Controls	✔
§13	Execution Cycle Requirements	✔
§14	Failure Modes	✔

§15	Interoperability Rules	✔
§16	Required APIs	✔
§17	Compliance Testing	✔

---

4. DETERMINISTIC RULE (R) REQUIREMENTS

All NOVAK implementations MUST:

4.1 Rules MUST be functionally pure

No randomness
No external state
No timing dependencies
No side effects

4.2 Rules MUST produce the same O for the same D

Across:

- hardware
- OS
- runtime
- VM/container
- scaling layers

4.3 Rules MUST be frozen

Updates allowed only through an EIR-attached rule update event.

4.4 Rule storage MUST be hash-anchored

Hash:

```
HR = SHA-256(rule_text)
```

---

5. INPUT (D) ATTESTATION REQUIREMENTS

Implementations MUST:

5.1 Capture exact input

Bit-for-bit, raw form.
Normalization is forbidden unless recorded.

5.2 Generate hash of input

```
HD = SHA-256(D)
```

5.3 Bind input to identity

The input MUST be linked to:

- user ID (if exists)
- device identity
- execution identity
- session key (if applicable)

5.4 Bind input to timestamp T

The timestamp MUST be included in the HVET construction.

---

6. OUTPUT (O) DETECTION REQUIREMENTS

Implementations MUST:

6.1 Compute deterministic output

```
O = R(D)
```

6.2 Hash the output

```
HO = SHA-256(O)
```

6.3 Store output only AFTER Safety Gate approval

No partial or intermediate outputs may be visible or cached.

---

7. HVET REQUIREMENTS (MANDATORY)

```
HVET = SHA-256(HR || HD || HO || T)
```

Implementations MUST:

- validate HR matches current rules
- validate HD matches attested input
- validate HO equals recomputed output
- validate T is reasonable (± allowed drift)
- reject ANY mismatch

HVET MUST be:

- unique per execution

- irreversible
- collision-resistant
- timestamp-anchored
- identity-bound

---

8. EIR REQUIREMENTS

Implementations MUST generate an **Execution Identity Receipt** *before execution*.

EIR MUST include:

```
EIR = {
  eir_id,
  rule_hash: HR,
  input_hash: HD,
  output_hash: HO,
  timestamp: T,
  identity: I,
  hvet: HVET,
  signature: SIG(I_private, HVET),
  version: "NOVAK-EIR-v1"
}
```

The EIR MUST be signed using:

- device private key, OR
- system private key, OR
- validator/attestor private key

The signature MUST be verifiable.

---

9. RGAC REQUIREMENTS

Implementations MUST record EIRs in a hash-linked sequence.

```
RGAC[n].link = SHA-256(RGAC[n-1].hvet || RGAC[n].hvet)
```

RGAC MUST:

- prevent reordering
- prevent deletion
- prevent insertion
- prevent overwriting
- maintain global continuity

If RGAC continuity breaks → **BLOCK execution immediately.**

---

10. SAFETY GATE REQUIREMENTS

Safety Gate MUST evaluate:

```
SG = (RulePure) ∧ (HVETValid) ∧ (EIRValid) ∧ (RGACContinuous) ∧
(PLXStable) ∧ (PSXLegitimate)
```

SG MUST return **TRUE** ONLY if everything passes.

If SG = FALSE → **execution MUST NOT proceed.**

---

11. PL-X PHYSICAL INTEGRITY REQUIREMENTS

Implementations MUST include:

- clock drift monitoring
- timing jitter monitoring
- voltage/frequency anomaly detection
- metastability detection (if hardware-capable)
- race condition trapping
- ordering checks

If PL-X fails → Safety Gate MUST return FALSE.

---

12. PS-X PSYCHO-SOCIAL INTEGRITY REQUIREMENTS

PS-X MUST detect:

- adversarial inputs
- misleading intent
- deception
- ambiguous phrasing
- manipulation attempts
- social engineering patterns

If PS-X flags input → Safety Gate MUST return FALSE.

---

13. EXECUTION CYCLE REQUIREMENTS

All NOVAK implementations MUST follow this sequence:

```
Step 1: Capture input (D)
Step 2: Compute deterministic output (O)
Step 3: Compute HR, HD, HO
Step 4: Generate HVET
Step 5: Generate EIR (pre-execution)
Step 6: Append to RGAC
Step 7: Run Safety Gate
Step 8: If SG=TRUE → Execute
Step 9: If SG=FALSE → Reject
```

NO step may be skipped or reordered.

---

14. FAILURE MODES (MANDATORY)

The system MUST reject execution for ANY of the following:

- hash mismatch
- missing EIR
- missing HVET
- broken RGAC continuity
- rule impurity
- PS-X flag
- PL-X anomaly
- unexpected timestamp
- identity mismatch

---

15. INTEROPERABILITY RULES

Implementations MUST support:

- exporting EIR in JSON format
- exporting RGAC state in JSON
- accepting signed EIR bundles
- reproducible rule evaluation
- cross-system validation of HVET

---

16. REQUIRED APIs

Implementations MUST expose:

```
POST /hvet/verify
POST /eir/verify
POST /rgac/verify
POST /novak/execute
GET  /novak/rules
GET  /novak/status
```

All POST endpoints MUST reject unauthenticated or unsigned requests.

---

17. COMPLIANCE TESTING

To be NOVAK-compliant, implementations MUST pass:

- deterministic execution tests
- rule purity tests
- HVET integrity tests
- EIR completeness tests
- RGAC continuity tests
- PL-X anomaly tests
- PS-X manipulation tests
- timestamp drift tests

Compliance suite will be included in SP-5.

---

18. CONCLUSION

SP-4 defines EXACTLY how NOVAK MUST be implemented to be valid.
Any system missing ANY requirement is **NOT NOVAK-compliant.**

SP-4 completes the technical core of:

- SP-1 (Execution Integrity)
- SP-2 (Cryptographic Specification)
- SP-3 (Safety Gate)
- SP-4 (Implementation Requirements)
- NTM-1 (Threat Model)

This document is the foundation for certification (SP-5), reference implementation, and federal adoption.