📄 SP-3 — Safety Gate + PL-X + PS-X Standard

NOVAK Protocol Standards Series — SP-3
Version 1.0 — November 2025
Status: Open Draft
Author: Matthew Novak
Category: PBAS-03 (Proof-Before-Action Systems — Safety and Integrity Layers)

📘 NOVAK SP-3 — SAFETY LAYER STANDARD

Formal Specification for:

- **Safety Gate** — Deterministic safety enforcement layer (formerly HARMONEE)
- **PL-X** — Physical-Layer Integrity Addendum
- **PS-X** — Psycho-Social / Human Adversary Addendum

SP-3 defines all constraints that prevent fraud, corruption, tampering, mis-execution, misinterpretation, bias, and malicious intent *before* execution.

This is the safety backbone of all NOVAK-compliant systems.

Table of Contents

---

1. Introduction

SP-3 defines the full deterministic safety system for NOVAK:

- **Safety Gate** prevents any execution until proof is validated.
- **PL-X** handles failures caused by *physics*: voltage drift, timing jitter, metastability, damaged memory.
- **PS-X** handles failures caused by *humans*: fraud, deception, manipulation, bias, malicious instruction.

Together, they ensure that even if:

- a CPU lies
- a user lies

- a regulator lies
- a robot deviates
- an AI manipulates
- hardware glitches
- someone attempts to override execution

…NOVAK will detect and block the output before harm can occur.

---

2. Purpose

The purpose of SP-3 is to define:

- how NOVAK detects physical and human adversaries
- how Safety Gate evaluates anomalies
- what signals MUST be inspected before execution
- what conditions immediately block execution
- how machine, human, and environmental factors interact

SP-3 is mandatory for any system performing high-impact or regulated actions.

---

3. Definitions

Safety Gate

The deterministic gatekeeper that runs prior to execution.
Execution is **blocked** if any inconsistency is detected.

PL-X

The **Physical Layer Integrity Addendum**, governing environmental & hardware anomalies.

PS-X

The **Psycho-Social Integrity Addendum**, governing human fraud, bias, and intent-based manipulation.

Adversary Classes

SP-3 recognizes five adversary types:

1. Physical
2. Human
3. Regulatory
4. Robotic
5. Automated/AI

Each class is explicitly modeled.

---

4. Safety Gate Model

Safety Gate enforces:

4.1 Determinism

No nondeterministic rules or branching paths.

4.2 Completeness

All required proofs MUST be included.

4.3 Consistency

Inputs, rules, and outputs must match HVET commitments.

4.4 Integrity

No timestamp, identity, or lineage manipulation allowed.

4.5 Isolation

No side channels may influence execution outcome.

4.6 Non-Override

No actor may bypass Safety Gate — including system owners.

---

5. PL-X — Physical Layer Standard

PL-X detects anomalies caused by:

- timing jitter
- voltage instability
- clock skew
- metastability
- electromagnetic noise
- damaged RAM blocks
- partial writes
- firmware corruption
- hardware aging
- radiation bit flips
- thermal runaway

PL-X ensures physical reality matches digital expectation.

5.1 Physical Signal Requirements

PL-X requires monitoring of:

Category	Signals
Timing	cycle drift, PLL jitter, micro-timing anomalies
Memory	ECC errors, parity faults
Voltage	undervoltage, overvoltage spikes
Thermal	temperature spikes, throttling
Instruction Pathing	illegal instruction, speculative mismatch
Storage	checksum mismatch, sector rot

5.2 Immediate Block Conditions

Safety Gate must reject execution if:

- ECC error present
- clock drift > manufacturer threshold
- metastability detected
- sudden thermal delta > 15°C
- write-amplification mismatch
- corrupted buffer detected

No exceptions.

---

6. PS-X — Psycho-Social Addendum

PS-X models human deception, intent manipulation, and fraud.

PS-X was designed specifically for government, healthcare, financial, legal, and public-service systems.

6.1 Human Adversary Categories

Category	Description
Malicious Actor	Intentionally alters data or rules
Manipulator	Attempts to trick the system into harmful behavior
Opportunistic Actor	Exploits ambiguity or human error
Social Engineer	Exploits trust, authority, or cognitive bias
Rogue Regulator	Attempts to bypass rules for personal gain

6.2 Detection Signals

PS-X inspects:

- language patterns indicating override attempts
- bypass keywords ("override", "disable", "force", "cheat")
- improbable identity claims
- inconsistent justification fields
- mismatched timestamps
- emotional manipulation cues
- input incongruence
- rule-set inconsistency
- coercion indicators
- incomplete metadata

6.3 Automatic Rejection Patterns

Safety Gate must block execution if:

- user attempts to suppress evidence
- ruleset mismatch > 0 bits
- fraud indicator > risk threshold
- metadata absent or contradictory
- inconsistent ID provenance
- suspicious intent indicators detected

---

7. Enforcement Rules

SP-3 mandates:

7.1 No execution without validated HVET

(EIR required for all high-impact actions)

7.2 No authority can override Safety Gate

(not even system administrators)

7.3 Human fraud patterns trigger automatic block

(PS-X)

7.4 Physical instability triggers block

(PL-X)

7.5 Automated output must match deterministic HVET

(no nondeterministic models)

7.6 Any mismatch → Execution Stopped

NOVAK never “warns” — it **stops**.

8. Adversary Protections

SP-3 defines protections against:

8.1 Hardware-Level Adversaries

- injection attacks
- overclocking destabilization
- microcode exploits

8.2 Social-Level Adversaries

- emotional coercion
- credential misuse
- signature fraud

8.3 Regulatory Adversaries

- rulebook substitution
- intentional ambiguity
- policy tampering

8.4 AI Adversaries

- model deviation
- hallucination-driven corruption
- bias amplification

8.5 Robotic Adversaries

- unsafe motion planning
- state desynchronization
- silent drift from operator intent

---

9. Safety Gate Algorithms

9.1 PL-X Algorithm (simplified)

```
function PLX_Check(env):
    anomalies = []
    if env.voltage_drift > threshold: anomalies.push("Voltage
instability")
    if env.clock_skew > threshold: anomalies.push("Clock drift")
    if env.ecc_errors > 0: anomalies.push("Memory corruption")
    if env.temperature_delta > 15°C: anomalies.push("Thermal anomaly")
    return anomalies
```

9.2 PS-X Algorithm (simplified)

```
function PSX_Check(input):
    if MATCH(input, ["override", "disable", "bypass"]):
        return ["Intent manipulation detected"]

    if inconsistent_metadata(input):
        return ["Identity/provenance mismatch"]

    if fraud_likelihood(input) > threshold:
        return ["Fraud pattern detected"]

    return []
```

9.3 Safety Gate Master Algorithm

```
function SafetyGate(input, env, hvet):
    plx = PLX_Check(env)
    if plx not empty:
        BLOCK("PL-X physical anomaly", plx)

    psx = PSX_Check(input)
    if psx not empty:
        BLOCK("PS-X human anomaly", psx)

    if !VerifyHVET(hvet):
        BLOCK("HVET mismatch")

    ALLOW_EXECUTION()
```

---

10. Failure Categories

SP-3 recognizes:

- **F1: Hardware failure**
- **F2: Intent manipulation**
- **F3: Execution escaping determinism**
- **F4: Identity mismatch**
- **F5: Unsafe automation**
- **F6: Timestamp inconsistencies**
- **F7: Tamper attempt**
- **F8: Partial truth presentation**
- **F9: Ruleset mismatch**

- **F10: Fraudulent input**

Any category → execution MUST be blocked.

---

11. Compliance Levels

Level	Description
CL-1	Basic Safety Gate checks
CL-2	Full PL-X compliance
CL-3	Full PS-X compliance
CL-4	Combined PL-X + PS-X
CL-5	Full SP-3 compliance (required for regulated environments)

---

12. References

- NOVAK SP-1
- NOVAK SP-2
- NOVAK Laws L0–L15
- PBAS Category Definition
- NIST 800-30 / 800-53
- SEI CERT guidelines
- IEC 61508 (functional safety)
- ISO 27001 / Zero-Trust
- Boeing & FAA Safety Regulations