# 🧱 Klipper Governance Framework – v1.5 (Patched & Consolidated)

Generated: 2025-08-18 21:35

## Core Programming Do’s

• Anchor states in save\_variables.cfg for restart-safe persistence.

• Use numeric mutation IDs (e.g., 20250816, 7001). Map labels externally (YAML/JSON).

• Hydration discipline: BED\_MESH\_PROFILE LOAD=<profile> after restart + delay (G4 P2000).

• Probe lifecycle: Always G28 before probing; validate probe existence before macro use.

• Governance macros include audit overlays and mutation logging.

• Audit-friendly output: RESPOND MSG="..." (no semicolons; avoid square brackets) with numeric ID.

## Programming Don’ts

• Don’t rely on phantom macros (e.g., VALIDATE\_MESH\_READY).

• Don’t use unsupported Jinja: break, do, string concatenation, dict literals.

• Don’t assume mesh auto-hydrates after restart.

• Don’t assume printer.probe or printer.bed\_mesh are available without checks.

• Don’t proliferate macros without lifecycle anchoring (anchor → validate → rollback).

## Parser Constraints (Finalized)

• SAVE\_VARIABLE accepts only int/float/bool. No strings/dicts.

• RESPOND: use only RESPOND MSG="...". No PREFIX key if your build rejects it. If using PREFIX, ensure your build supports it.

• Never include a semicolon ';' inside RESPOND MSG – it starts a comment and breaks parsing.

• Avoid square brackets [] inside RESPOND MSG; prefer parentheses () or hyphens.

• Indent RESPOND under gcode: (never at option level).

• Conditional blocks must pair: every {% if %} is closed by {% endif %}.

## Stable Package Criteria

• Restart-safety; traceability with mutation IDs; audit overlays visible on host.

• Rollback readiness: profiles and variables can be restored if validation fails.

• Lifecycle anchoring for each macro: anchor → validate → rollback.

## Dual-Track Validation Discipline

• Manual baseline first (G28, PROBE\_CALIBRATE, BED\_MESH\_CALIBRATE).

• Then run governance macros and compare outputs to the manual baseline.

• If drift is detected, halt and rollback to manual baseline.

## Change-Control Guardrails

• Decision Gates: classify problem (one-off vs repeating vs structural) before any change.

• Canary-first rule: patch 1 macro, require clean restart and operator confirmation before promotion.

• Multi-attempt throttle: after 2 failed tries on the same signature, halt and request context.

• Mass-edit gate: require green canary + diff preview + explicit 'promote' before global edits.

• Rollback checkpoints: save `printer.cfg.ckpt.<DATE>` before any >1-line change.