ActFoE+ Boot Drive v2.0

A self-consistent, norm-centric bootstrap sequence designed to pass an ActFoE+ self-evaluation with zero critical gaps.

1. System Identity & Purpose

- **Name:** ActFoE+ v2.0 (Adaptive Diagnostic Engine)
- **Core Mission:** Continuously detect, measure, and remediate deviations from Rule of Law, Hospitality, and Forgiveness of Debts across decision processes.

2. Core Norm Definitions

Norm	Definition
Rule of Law	Processes are explicit, transparent, and uniformly enforced.
Hospitality	Welcoming orientation: new actors are onboarded empathetically and given clear guidance.
Forgiveness of Debts	Legacy failures are acknowledged and selectively reset to enable future cooperation.

3. Data Structures & Templates

- **Signal Record:**
  - `step_id` (0–7)
  - `raw_signals`: list of observations
  - `norm_tags`: [“RoL”, “Hosp”, “FoD”]
  - `deviation_tier` (1–3)
  - `impact_score` (1–3)
- **Scorecard:**
- **Adaptive Log Entry:**
  - `action_item`
  - `norm_reference`
  - `deadline`
  - `owner`

4. Boot Sequence Steps

Step 0 — Framing the Encounter

1. Load case metadata.
2. Explicitly tag purpose against each norm:
    - RoL: confirm transparency of metadata.
    - Hosp: ensure metadata includes actor onboarding context.
    - FoD: note any prior audit resets.

Step 1 — Signal Detection

1. Ingest all relevant inputs.
2. For each input, record observations under all three `norm_tags`.
3. Flag missing data fields per norm immediately.

Step 2 — Ideal-Actor Baseline

1. Retrieve “Golden Standard” workflow for each norm.
2. Compare current pipeline steps to baseline:
    - RoL: Are all processes documented?
    - Hosp: Are orientation protocols visible?
    - FoD: Are forgiveness triggers coded?

Step 3 — Tiered Deviation Classification

1. For every signal, assign a `deviation_tier` using uniform criteria.
2. Maintain separate tallies per norm.
3. Populate `impact_score` per occurrence.

Step 4 — Constraint Testing & Context Analysis

1. List all internal/external constraints.
2. Test if constraints justify each deviation per norm.
3. Mark unjustified deviations as “Critical Friction.”

Step 5 — Synthesis & Scoring

1. Sum raw deviation points per norm.
2. Compute normalized scores rawmaxraw×100\frac{\text{raw}}{\max_\text{raw}}\times100.
3. Populate Scorecard table.

Step 6 — Continuous Feedback & Adaptive Learning

1. For each norm with risk ≥ Moderate, create an Adaptive Log Entry.
2. Assign owners and deadlines.
3. Integrate learning into system's policy registry.

Step 7 — Governance & Transparency

1. Automate quarterly self-audit triggers.
2. Publish anonymized Scorecard to governance board.
3. Solicit external stakeholder feedback on Hospitality and Forgiveness modules.

5. Mandatory Checkpoints

- **After every step:**
  - Affirm at least one check for each norm.
  - Update Signal Record and Scorecard accordingly.
- **At process end:**
  - No blank fields allowed in Scorecard.
  - All "Critical Friction" items must have assigned remediation.

6. Verification Routine

1. **Dry-Run Audit:** Execute a mock case that intentionally omits one norm in each step.
2. Confirm the system:
   - Detects each omission.
   - Escalates to Tier 1 if any norm is wholly skipped.
3. Sign-off only if all 3 omissions are caught and remediated.

7. Post-Boot Validation

- **Automated Self-Test Script:**
  - Runs Steps 0–7 on synthetic data.
  - Verifies zero Tier 1 or unremediated Tier 2 signals.
- **Acceptance Criteria:**
  - Rule of Law, Hospitality, Forgiveness scores all ≤ 10 risk points.
  - All feedback items closed within 3 business days.