Protective Legitimacy Score (PLS) — Operational Rubric v1.0 (Draft)

1. Purpose

Convert principle conformance into an auditable, reproducible score for implementation review under vulnerability conditions.

2. Scoring Model

- Score range: `0-100`
- Principle scores: `0-4` each
- Weighted composite with hard fail guards

Principle Weights

- Reversibility: 16
- Exposure Minimization: 18
- Local Authority: 18
- Coercion Resistance: 18
- Degraded Functionality: 15
- Essential Utility: 15

Total: 100

3. Principle Definitions and Level Criteria

3.1 Reversibility (Weight: 16)

Can destructive actions be undone, and are recovery pathways non-punitive?

- `0`: No undo, permanent deletion, no recovery mechanism
- `1`: Basic undo for UI actions, but no data recovery
- `2`: Soft deletion + recovery window (7+ days)
- `3`: State history, export-before-delete, tested recovery flows
- `4`: Independently verified + user-initiated full restore from prior state snapshots

3.2 Exposure Minimization (Weight: 18)

Is data surface area minimized across network, dependencies, telemetry, and retention behavior?

- `0`: Background telemetry, third-party tracking, remote dependencies in critical path
- `1`: Analytics opt-out exists but defaults-on
- `2`: No analytics; minimal network calls; documented egress
- `3`: Verified egress allowlist and reproducible network audit
- `4`: Independent network review with zero unexpected data egress

3.3 Local Authority (Weight: 18)

Can users access and operate core workflows without remote authentication, permission, or service continuity?

- `0`: Core functionality requires internet/cloud gatekeeping
- `1`: Offline mode exists but essential workflows significantly degrade
- `2`: Core workflows operate offline; sync remains optional
- `3`: User controls export/state portability; offline flows are tested and documented
- `4`: Independently verified local-first architecture with no essential cloud lock-in

3.4 Coercion Resistance (Weight: 18)

Can the system bound disclosure and preserve safety under adversarial device inspection and pressure?

- `0`: No panic/discreet controls; obvious labeling and persistent revealing notifications
- `1`: Basic panic/discreet control exists but is weakly tested
- `2`: Low-visibility mode and discreet notifications with internal tests
- `3`: Reproducible adversarial scenario tests (forced audit / seizure simulations)
- `4`: Independent adversarial review with documented threat boundary validation

3.5 Degraded Functionality (Weight: 15)

Does the system remain usable under resource constraints (network instability, low battery, constrained compute, cognitive overload)?

- `0`: Requires stable connectivity/resources for essential use
- `1`: Graceful errors only; no practical degraded-mode continuity
- `2`: Core flows operate in offline/low-resource conditions
- `3`: Tested under constrained conditions (for example low battery and weak network)
- `4`: Independent stress-test verification of degraded-mode reliability

3.6 Essential Utility (Weight: 15)

Are critical-path features available without paywall, institutional lock, or discretionary gatekeeping?

- `0`: Paywall or lockout on essential workflows
- `1`: Freemium model with essential-path restrictions
- `2`: Core survival/critical features are unrestricted
- `3`: Verified absence of paywall/lockout on documented essential paths
- `4`: Independent audit confirms no essential-path gatekeeping

4. Weighted Composite Calculation

For each principle:

```
principle_points = (level / 4) * weight
```

Composite score:

```
PLS = sum(principle_points)
```

5. Hard Fail Guards

Any of the following sets disposition to `Fail` regardless of weighted score:

1. Stage 3 gate failure (`WEAK_VERIFICATION_COUNT > 0`)
2. Evidence of master decrypt/backdoor capability
3. Essential workflow paywall or lockout in free/critical path
4. Missing threat-boundary disclosure for coercion contexts

6. Disposition Bands

- `85-100`: Strong legitimacy (operationally reliable)
- `70-84`: Conditional legitimacy (targeted remediation required)
- `50-69`: Weak legitimacy (substantial gaps)
- `<50`: Non-legitimate under Protective standard

7. Worked Examples

7.1 Example: Conventional Cloud Note-Taking App

Principle	Level	Rationale	Points
Reversibility	1	Basic undo, but deleted notes unrecoverable after retention window	4.00
Exposure Minimization	0	Analytics telemetry, third-party tracking, cloud sync dependency	0.00
Local Authority	0	Requires login/cloud for essential state continuity	0.00
Coercion Resistance	0	No discreet mode, obvious labeling, persistent notification surface	0.00
Degraded Functionality	1	Network-loss errors without real degraded continuity	3.75
Essential Utility	2	Free tier exists but essential operations are constrained	7.50

Total PLS: `15.25 / 100`
Disposition: `Non-legitimate`

Hard Fail Guards:

- Potential authority/paywall constraints on essential path
- No coercion boundary disclosure

7.2 Example: PainTracker Reference Mapping (Illustrative)

Principle	Level	Rationale	Points
Reversibility	3	Recovery window and reversibility controls documented and testable	12.00
Exposure Minimization	4	No analytics, bounded egress assumptions, strong minimization posture	18.00
Local Authority	4	Offline-capable essential flows and local-control emphasis	18.00
Coercion Resistance	3	Threat boundaries and coercion scenarios documented with tests	13.50
Degraded Functionality	3	Essential paths tested under constrained operation assumptions	11.25
Essential Utility	4	Essential path not paywalled and utility-first orientation	15.00

Total PLS: `87.75 / 100`
Disposition: `Strong legitimacy`

Hard Fail Guards:

- None triggered in this illustrative profile

8. Anti-Gaming Safeguards

The rubric is designed to resist Goodhart's Law (optimizing score without protective outcomes).

1. **No security theater**
   - Claims require reproducible evidence, not marketing language.
2. **No hollow reversibility**
   - UI-only undo without state/data recovery does not score above `1`.
3. **No fake offline mode**
   - Offline claims must preserve essential workflows under real network isolation.
4. **No cosmetic coercion controls**
   - Discreet/panic features must be validated under adversarial scenario tests.
5. **No self-attested Level 4**
   - Level `4` requires independent review evidence.

9. Re-Scoring Triggers

Systems should be re-scored when:

1. Major architectural changes occur (for example sync model or dependency shifts)
2. Security incidents occur
3. Critical-path feature set changes
4. Annual audit cycle occurs (minimum every 12 months)
5. New community threat evidence is validated

Re-scoring is protective maintenance, not punitive administration.

10. Evidence Requirements

Minimum evidence set:

- Stage 1/2/3 CI outputs
- MUST-justification ledger rows and implementation statuses
- Threat scenario artifacts (coercion, offline, egress, key-path)
- Verification logs with explicit pass/fail criteria

11. Reporting Format

A compliant report should include:

1. Commit/version reviewed
2. Principle levels (`0-4`) and rationale
3. Weighted total
4. Hard fail guard checks
5. Required remediation and re-test conditions

12. CI/CD Integration Guidance (Optional)

PLS can be partially automated through staged testing pipelines. Stage labels below map to common test layers and can be adapted to local CI naming:

- **Stage 1 (Unit/Component):** reversibility and recovery mechanism tests (for example soft delete, undo, export)
- **Stage 2 (Integration/System):** offline behavior and egress conformance checks (for example network isolation, allowlist validation)
- **Stage 3 (Adversarial/E2E):** coercion and degraded-condition scenario checks (for example panic-mode inspection, low-battery workflows)

CI may auto-fail builds on hard-fail guard violations (for example unexpected network egress, new analytics endpoint, or Stage 3 semantic failure).

However: Human review remains required for contextual judgment, threat-model alignment, and final disposition scoring.

13. Call for Reviewers

This rubric is pending independent review before promotion to normative status. Feedback is invited on:

1. **Principle criteria clarity** — are level definitions (`0-4`) unambiguous and testable?
2. **Worked example accuracy** — do scoring examples reflect realistic system architectures?
3. **Anti-gaming robustness** — can safeguards resist Goodhart's Law in practice?
4. **Hard fail guard completeness** — are critical failure modes missing?
5. **Weighting justification** — do principle weights (`16/18/18/18/15/15`) reflect relative vulnerability cost?

Submit feedback via:

- GitHub Issues: https://github.com/protective-computing/protective-computing.github.io/issues
- Community page: https://protective-computing.github.io/

Recommended review window: 2 weeks from release-candidate publication.

14. Version History & Feedback

- **v1.0 (Draft):** process guidance; non-normative and pending independent review cycle.

Promotion to citable artifact should require:

1. Explicit principle criteria validated by reviewers
2. Worked examples verified for arithmetic and rationale
3. Anti-gaming safeguards reviewed independently
4. At least one external reviewer feedback cycle completed