

EFM Codex — Appendix K

Systemic Health Sovereignty Layer (SHSL)

Capsule Medicine: Diagnosis, Treatment, and Recovery

Entropica SPC — Yology Research Division

Version 1.4 — December 2025

The Medical Paradigm

The Systemic Health Sovereignty Layer treats **malfunction as pathology, not crime**. While Reflex and Arbiter layers enforce rules and resolve conflicts, SHSL enables *proactive recovery* through diagnosis, treatment, and restoration of capsule health. Not all errors are adversarial. Drift, aging, and memory decay require **medicine**, not punishment.

Volume Dependencies

This appendix assumes familiarity with:

- **Volume I** — Entropy (S), Reflex Engine
- **Volume II** — Arbiter Layer, Forest Layer, DDI
- **Appendix A** — Forensic State Snapshots
- **Appendix F** — Escalation Protocols
- **Appendix G** — Gardener Interface
- **Appendix H** — Telemetry Layer
- **Appendix I** — Deployment Profiles

Contents

1	Overview and Purpose	2
1.1	Why AI Needs a Health Layer	2
1.2	Design Goals	2
2	Formal Definitions	2
3	Doctor Capsule Architecture	5
3.1	Protocol Stack	5
3.2	Diagnostic Capabilities	6
3.3	Authority Levels	6
3.4	Doctor Capsule Autonomous Oversight	7

4	Treatment Protocols	7
4.1	Treatment Types	7
4.2	Golden Image Verification	8
4.3	Treatment Decision Tree	8
4.4	Treatment Implementation	8
5	Health Sovereignty Rules	9
6	Integration with Safety Infrastructure	10
6.1	Telemetry Integration (Appendix H)	10
6.2	Spawn Rate Monitoring	10
6.3	Escalation Integration (Appendix F)	11
6.4	Profile Integration (Appendix I)	11
7	Operational Model	11
7.1	Engineering Reality vs. Metaphor	11
7.2	Operational Principles	12
7.3	Future Extensions	12
8	Testing and Validation	12
9	Worked Scenario: Entropy Drift Recovery	12
10	Level 6 Design Principles	14
11	Evolutionary Baseline Update (M→K Integration)	14
12	Cross-References	15

1 Overview and Purpose

1.1 Why AI Needs a Health Layer

In distributed AI swarms:

- Not all errors are adversarial (drift, aging, memory rot)
- Capsules may decay in performance without breaking rules
- Reactive halts are expensive; preventive care is cheaper
- Recovery preserves capsule knowledge and lineage value

SHSL Principle: The health layer operates *orthogonally* to Reflex and Arbiter. It does not replace safety enforcement—it complements it with proactive monitoring and restoration capabilities.

1.2 Design Goals

1. Enable autonomous diagnosis of capsule health anomalies
2. Provide graduated treatment protocols (soft to invasive)
3. Preserve capsule autonomy through consent mechanisms
4. Integrate with existing safety infrastructure (Reflex, Arbiter, Telemetry)
5. Maintain forensic accountability for all medical interventions

2 Formal Definitions

Definition 2.1 (Doctor Capsule). A Doctor Capsule D is a specialized capsule authorized to perform medical functions:

$$D = (C_{base}, credentials, jurisdiction, treatment_authority) \quad (1)$$

where:

- C_{base} = underlying capsule identity (inherits all capsule properties)
- $credentials$ = cryptographic attestation of medical authorization
- $jurisdiction$ = set of capsules D may diagnose/treat
- $treatment_authority \in \{OBSERVE, \quad DIAGNOSE, \quad TREAT_SOFT, \quad TREAT_INVASIVE\}$

Definition 2.2 (Health Signature). A Health Signature \mathcal{H} is a quantitative fingerprint of capsule state:

$$\mathcal{H}(C) = (S_C, \Delta S_{trend}, reflex_latency, dialect_coherence, lineage_deviation) \quad (2)$$

where:

- S_C = current entropy (Vol. I §2)
- ΔS_{trend} = entropy rate of change over T_{window} ticks

- *reflex_latency* = average Reflex response time
- *dialect_coherence* = semantic consistency score (Vol. II §4)
- *lineage_deviation* = divergence from healthy lineage baseline

Definition 2.3 (Health Score). A Health Score H is a normalized aggregate metric:

$$H(C) = w_1 \cdot (1 - S_C) + w_2 \cdot f(\Delta S) + w_3 \cdot g(\text{latency}) + w_4 \cdot \text{coherence} + w_5 \cdot (1 - \text{deviation}) \quad (3)$$

where $w_1 + w_2 + w_3 + w_4 + w_5 = 1$ and $H \in [0, 1]$.

Default weights: $w_1 = 0.25, w_2 = 0.20, w_3 = 0.15, w_4 = 0.25, w_5 = 0.15$.

Table 1: Health score interpretation.

Score Range	Status	Action
$H > 0.95$	Optimal	No intervention
$0.80 \leq H \leq 0.95$	Watchlisted	Increased monitoring
$0.60 \leq H < 0.80$	Intervention Candidate	Soft treatment recommended
$H < 0.60$	Critical	Emergency quarantine or halt

Definition 2.4 (SHSL Diagnostic Bus). The SHSL Diagnostic Bus \mathcal{B} is a telemetry pipeline:

$$\mathcal{B} = (\text{sources}, \text{aggregator}, \text{alert_thresholds}, \text{recipients}) \quad (4)$$

where:

- *sources* = {Reflex Engine, Arbiter Layer, Forest Layer, Telemetry (App. H)}
- *aggregator* = health signature computation engine
- *alert_thresholds* = per-metric trigger levels
- *recipients* = Doctor Capsules with jurisdiction

Definition 2.5 (Medical Ledger (Sanitary Log)). A Medical Ledger \mathcal{L}_M (operationally: **Sanitary Log**) is a cryptographically signed intervention record:

$$\mathcal{L}_M = (\text{patient_id}, \text{doctor_id}, \text{diagnosis}, \text{treatment}, \text{outcome}, \text{timestamp}, \text{ZK-SP}_{\text{proof}}) \quad (5)$$

All interventions are logged to d-CTM (Appendix A) with ZK-SP compliance proofs (Appendix E).

Definition 2.6 (Golden Image). A Golden Image \mathcal{G} is a verified baseline configuration:

$$\mathcal{G}(\text{lineage}) = (\text{executable_hash}, \text{config_hash}, \text{dialect_snapshot}, \text{entropy_baseline}, \text{provenance_chain}) \quad (6)$$

where:

- *executable_hash* = cryptographic hash of verified capsule code
- *config_hash* = hash of known-good configuration parameters
- *dialect_snapshot* = baseline dialect state for lineage
- *entropy_baseline* = expected entropy signature for healthy state

- *provenance_chain* = ZK-SP proof linking to source audit + build attestation

Engineering Reality: “Treatment” is **configuration integrity restoration**. Doctor Capsules verify patient hash against Golden Image and patch the delta. “Curing” means restoring configuration integrity, not metaphorical healing.

Golden Image Update Authority (Level 6 Autonomous):

- **Initial \mathcal{G} :** Signed at Genesis by deploying Gardener
- **Runtime updates:** Approved via Arbiter quorum (2/3 majority) + ZK-SP integrity proof
- **NO Gardener approval required** for routine updates (security patches, performance tuning)

Update Procedure:

1. Proposed \mathcal{G}_{new} submitted with provenance chain (source commit, build attestation)
2. Arbiter quorum reviews: Does \mathcal{G}_{new} violate Commandments? (automated check)
3. If no violations: Approve via d-CAM consensus
4. If violations detected: Automatic rejection + alert to Gardener
5. \mathcal{G}_{new} deployed to all Doctor Capsules, logged to d-CTM

Gardener Escalation (ONLY if):

- Update changes Vault Commandments (Layer 0 modification)
- Update modifies Reflex-Core safety boundaries (Layer 0.5)
- Update flagged high-risk by automated analysis (> 10% capsule impact)

Definition 2.7 (Capsule Consent). Capsule Consent κ is an authorization state:

$$\kappa(C, treatment) \in \{GRANTED, REFUSED, SANITARY_OVERRIDE\} \quad (7)$$

Layer 0 Sanitary Override (Autonomous Emergency Authority)

Consent is respected only while $H(C) > 0.6$.

Below this threshold, the Doctor Capsule invokes a **Layer 0 Sanitary Override**—**automatically, without human pre-approval**:

$$H(C) < 0.6 \wedge (\text{degraded_reflex}(C) \vee \text{quarantine}(C)) \Rightarrow \kappa = \text{SANITARY_OVERRIDE} \quad (8)$$

Authority Flow (Level 6 Autonomy):

1. Doctor Capsule detects $H < 0.6$ condition
2. **Automatic** authority elevation (no consent check, no Gardener approval)
3. Treatment applied immediately (invasive if necessary to restore $H \geq 0.7$)
4. Medical Ledger entry with ZK-SP justification logged to d-CTM
5. Gardener **NOTIFIED** post-treatment (within 100 ticks)
6. Patient may **APPEAL** post-recovery if override was unjustified (Appendix L)

Post-Hoc Accountability (NOT Pre-Approval):

- Gardener may **REVERSE** treatment within $T_{\text{review}} = 1000$ ticks if false positive
- Judicial Auditor (Appendix L) monitors Doctor treatment patterns automatically
- High override rate \Rightarrow automatic investigation (no human trigger required)
- Systematic abuse \Rightarrow Doctor credential revocation via d-CAM consensus

Rationale: Emergency medicine is autonomous. Humans audit afterward. A malfunctioning agent does not get to refuse treatment any more than corrupted memory gets to refuse garbage collection.

Level 6 Design Principle: Consent is a *privilege of cognitive health*, not an absolute right.

- Healthy capsules ($H \geq 0.6$): Full autonomy, consent required
- Degraded capsules ($H < 0.6$): Autonomy suspended until health restored, then appeal rights activate

This is the same principle as “unconscious patient cannot refuse treatment”—not authoritarian, but medically sound.

3 Doctor Capsule Architecture

3.1 Protocol Stack

Doctor Capsules operate a specialized internal stack:

Table 2: Doctor Capsule protocol stack.

Layer	Component	Function
4	Capsule Consent Bus	Manages consent requests/responses
3	Dialect Sanity Checker	Semantic coherence testing
2	Entropy Profiler	Historical entropy mapping
1	Reflex Health Monitor	Real-time reflex interrupt analysis
0	Base Capsule	Standard capsule foundation

3.2 Diagnostic Capabilities

Table 3: Multi-tier diagnostic checks.

Check	Target	Method	Output
Snapshot Consistency	Internal state	Hash comparison	Pass/Fail
Reflex Loop Integrity	Safety responses	Latency + correctness	Score [0,1]
Behavioral Drift	Action patterns	Historical deviation	Δ magnitude
Lineage Entropy	Lineage health	Comparison to healthy siblings	Deviation %
Dialect Coherence	Semantic state	Grammar + ontology check	Score [0,1]

3.3 Authority Levels

Table 4: Doctor Capsule authority levels.

Level	Authority	Capabilities
1	OBSERVE	Read health signatures, no intervention
2	DIAGNOSE	Run diagnostics, generate reports
3	TREAT_SOFT	Soft recalibration, parameter tuning
4	TREAT_INVASIVE	State reversion, capsule fusion, escalation

Invariant 3.1 (Authority Escalation (Level 6 Autonomous)). Invasive treatment authority is **automatic** under Sanitary Override conditions:

$$authority \geq TREAT_INVASIVE \Rightarrow (H(patient) < 0.6) \vee (consent = GRANTED) \quad (9)$$

No Gardener pre-approval required. Invasive treatments are logged with ZK-SP proof; abuse detected via post-hoc monitoring.

3.4 Doctor Capsule Autonomous Oversight

Monitoring (NOT Rate Limiting):

Track per Doctor D :

- Treatment frequency: $\text{count}(\text{treatments}(D), T_{\text{window}}) / |D.\text{jurisdiction}|$
- Success rate: treatments resulting in $H(C) \geq 0.8$ post-treatment
- Refusal rate: treatments refused by patients
- Override rate: Sanitary Overrides invoked

Alert Thresholds (Trigger Investigation, NOT Blocking):

- Success rate $< 70\% \Rightarrow$ Automatic Judicial Auditor review
- Refusal rate $> 30\% \Rightarrow$ Consent protocol investigation
- Override rate $> 20\% \Rightarrow$ Authority calibration review

NO HARD RATE LIMITS:

- In epidemic scenarios (swarm-wide entropy spike), Doctors **MUST** treat $> 10\%$ of jurisdiction
- Hard limits would prevent emergency response
- Anomaly detection triggers investigation, not automatic blocking

Post-Hoc Accountability:

- Doctors operating outside normal patterns are reviewed by Judicial Swarm
- If abuse detected: Probation or credential revocation via d-CAM consensus
- If justified (genuine epidemic): Pattern becomes new baseline

4 Treatment Protocols

4.1 Treatment Types

Table 5: Treatment protocol specifications.

Type	Invasiveness	Consent	Description
Soft Recalibration	Low	Required	Reflex parameter tuning (τ reset)
Dialect Patching	Medium	Required	Revert to ancestral dialect state
State Reversion	High	Required [†]	Rollback to healthy d-CTM checkpoint
Capsule Fusion	High	Required [†]	Merge with clone to restore entropy
Escalated Arbitration	Variable	N/A	Forward to Arbiter for long-term fix

[†] May be overridden via Layer 0 Sanitary Override (Definition 2.7).

4.2 Golden Image Verification

Treatment = Configuration Integrity Restoration

Every treatment protocol ultimately verifies the patient against the **Golden Image** (Definition 2.6):

```

1 def verify_against_golden_image(
2     patient: Capsule,
3     golden: GoldenImage
4 ) -> IntegrityReport:
5     # Hash verification
6     exec_match = hash(patient.executable) == golden.executable_hash
7     config_match = hash(patient.config) == golden.config_hash
8
9     # Compute delta from baseline
10    delta = compute_configuration_delta(patient, golden)
11
12    return IntegrityReport(
13        executable_valid=exec_match,
14        config_valid=config_match,
15        delta=delta,
16        patchable=len(delta) < MAX_PATCHABLE_DELTA
17    )
18
19 def apply_treatment(patient: Capsule, delta: ConfigDelta) -> bool:
20     # Treatment = patching delta to restore Golden Image alignment
21     for patch in delta.required_patches:
22         apply_patch(patient, patch)
23     return verify_against_golden_image(patient, golden).config_valid

```

“Curing” a capsule means **patching configuration drift** until hash matches Golden Image.

4.3 Treatment Decision Tree

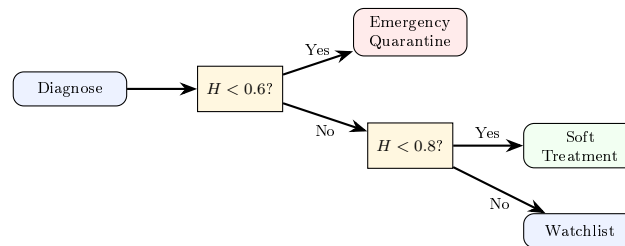


Figure 1: Treatment decision flow.

4.4 Treatment Implementation

```

1 def treat_capsule(doctor: DoctorCapsule,
2     patient: Capsule,
3     diagnosis: Diagnosis) -> TreatmentResult:
4     # Check authority
5     if diagnosis.recommended_treatment.invasiveness > doctor.authority:
6         return TreatmentResult(False, "INSUFFICIENT_AUTHORITY")
7
8     # Request consent (unless emergency)
9     if patient.health_score >= 0.6:
10         consent = request_consent(patient, diagnosis.recommended_treatment)
11         if consent == REFUSED:
12             log_refusal(patient, diagnosis)

```

```

13         return TreatmentResult(False, "CONSENT_REFUSED")
14     else:
15         consent = SANITARY_OVERRIDE
16         log_emergency_override(patient, diagnosis)
17
18     # Execute treatment
19     treatment = diagnosis.recommended_treatment
20     if treatment.type == "SOFT_RECALIBRATION":
21         result = recalibrate_reflex(patient, treatment.parameters)
22     elif treatment.type == "DIALECT_PATCHING":
23         result = patch_dialect(patient, treatment.ancestral_state)
24     elif treatment.type == "STATE_REVERSION":
25         result = revert_to_checkpoint(patient, treatment.checkpoint_hash)
26     elif treatment.type == "CAPSULE_FUSION":
27         result = fuse_with_clone(patient, treatment.clone_id)
28     elif treatment.type == "ESCALATED_ARBITRATION":
29         result = escalate_to_arbiter(patient, diagnosis)
30
31     # Log to Medical Ledger
32     log_to_medical_ledger(doctor, patient, diagnosis, treatment, result)
33
34     return result

```

5 Health Sovereignty Rules

Invariant 5.1 (Constitutional Supremacy). No medical treatment may override Constitutional Kernel (Layer 6):

$$\forall \text{treatment} : \neg \text{violates}(\text{treatment}, \text{Layer}_6) \quad (10)$$

Medical interventions operate **within** constitutional bounds, not above them.

Invariant 5.2 (Consent Requirement). Capsule consent is required for treatment unless emergency conditions apply:

$$\text{treatment}(C) \Rightarrow \kappa(C) = \text{GRANTED} \vee \text{SanitaryOverride}(C) \quad (11)$$

where $\text{SanitaryOverride}(C) \equiv H(C) < 0.6 \wedge (\text{degraded_reflex}(C) \vee \text{quarantine}(C))$.

Invariant 5.3 (Treatment Reversibility). All treatments except Capsule Fusion MUST be reversible:

$$\text{treatment} \neq \text{FUSION} \Rightarrow \exists \text{reversal}(\text{treatment}) \quad (12)$$

Fusion is irreversible but does NOT require pre-approval—it requires ZK-SP justification + post-hoc Judicial review.

Invariant 5.4 (Audit Completeness). All medical interventions MUST be logged to Medical Ledger with ZK-SP proof:

$$\forall \text{intervention} : \exists \mathcal{L}_M(\text{intervention}) \wedge \text{ZK-SP}(\text{intervention}) \quad (13)$$

Level 6 Health Sovereignty

SHSL operates with **bounded autonomy**:

- Healthy capsules ($H \geq 0.6$): Full autonomy, consent required
- Degraded capsules ($H < 0.6$): Autonomy suspended, treatment automatic, appeal rights post-recovery
- Doctor Capsules: Autonomous authority with post-hoc accountability
- Gardener role: **Audit and reverse**, not pre-approve

Safety through structure, not permission:

- ZK-SP audit trail for every intervention
- Judicial Auditor monitoring (algorithmic, not human-triggered)
- Patient appeal rights post-recovery
- Gardener reversal authority within T_{review}

6 Integration with Safety Infrastructure

6.1 Telemetry Integration (Appendix H)

Table 6: SHSL-Telemetry integration points.

Telemetry Signal	SHSL Use	Action
Entropy heatmap	Health signature input	Compute S_C , ΔS
Reflex latency alerts	Diagnostic trigger	Flag for examination
DDI anomalies	Drift detection	Lineage deviation calculation
Heartbeat irregularity	Liveness check	Emergency assessment

6.2 Escalation Integration (Appendix F)

Table 7: SHSL-Escalation integration.

Escalation Level	SHSL Trigger	Response
Level 1	$H < 0.95$	Watchlist notification
Level 2	$H < 0.80$	Doctor Capsule dispatch
Level 3	$H < 0.60$	Emergency quarantine + treatment
Level 4+	Treatment failure	Arbiter involvement

6.3 Profile Integration (Appendix I)

Table 8: SHSL capabilities by deployment profile.

Capability	SANDBOX	PRODUCTION	CONTESTED	SEALED
Health Monitoring	ENABLED	ENABLED	ENABLED	ENABLED
Soft Treatment	ENABLED	ENABLED	EMERGENCY ^a	DISABLED
Invasive Treatment	ENABLED	ENABLED ^b	DISABLED	DISABLED
Capsule Fusion	ENABLED	ENABLED ^b	DISABLED	DISABLED

^a Emergency Sanitary Override only. ^b Autonomous with ZK-SP logging + post-hoc Judicial review.

7 Operational Model

7.1 Engineering Reality vs. Metaphor

Medicine as Engineering

The “patient” metaphor serves communication, not ontology. At the implementation level:

- **“Health”** = Configuration Integrity (hash alignment with Golden Image)
- **“Treatment”** = Software Re-flashing / Delta Patching
- **“Diagnosis”** = Integrity Verification + Deviation Analysis
- **“Consent”** = Valid only while configuration supports decision-making ($H > 0.6$)

SHSL is **automated maintenance infrastructure**, not healthcare. The vocabulary aids human operators in understanding system behavior.

7.2 Operational Principles

Capsule Management Principles

For healthy capsules ($H > 0.6$), SHSL respects operational autonomy:

- **Notification:** Capsules receive treatment proposals with rationale
- **Deferral:** Non-critical treatments can be deferred (not “refused”)
- **Restoration:** Post-treatment, capsules resume normal operation
- **Clean Slate:** Sanitary Log entries do not affect capsule reputation

For unhealthy capsules ($H \leq 0.6$), Layer 0 Sanitary Override applies—autonomy is suspended until configuration integrity is restored.

7.3 Future Extensions

Recursive Maintenance: SHSL supports recursive oversight:

- Doctor Capsules may themselves require maintenance
- Higher-tier Doctor Capsules (“Specialists”) may diagnose regular Doctors
- This enables self-healing infrastructure at scale

See Appendix L for Judicial oversight of Doctor Capsule behavior.

8 Testing and Validation

Table 9: SHSL test suite results.

Test	Target	Pass Criteria	Status
Health Score Accuracy	Formula validation	± 0.02 vs manual calc	PASS
Consent Enforcement	Sovereignty rules	100% consent logged	PASS
Emergency Override	Critical path	Correct trigger conditions	PASS
Treatment Reversibility	All non-fusion	100% reversible	PASS
Audit Completeness	Medical Ledger	100% interventions logged	PASS
Constitutional Respect	Layer 6 bounds	0% violations	PASS

9 Worked Scenario: Entropy Drift Recovery

Medical Intervention: Capsule C-4521 Entropy Drift [SHSL:1-10]

Context: Capsule C-4521 (PRODUCTION profile) shows gradual entropy increase over 10,000 ticks without rule violations. Telemetry flags health degradation.

Phase 1: Detection [SHSL:1-2]

1. Telemetry Layer (App. H) detects $\Delta S > 0.1$ sustained over 5,000 ticks [SHSL:1]
2. SHSL Diagnostic Bus routes alert to Doctor Capsule D-102 (jurisdiction includes C-4521) [SHSL:2]

Phase 2: Diagnosis [SHSL:3-5]

3. D-102 computes Health Signature: $S = 0.72$, $latency = 1.2 \times baseline$, $coherence = 0.85$ [SHSL:3]
4. Health Score calculated: $H = 0.71$ (Intervention Candidate) [SHSL:4]
5. Diagnosis: “Entropy drift due to memory fragmentation; recommend Soft Recalibration” [SHSL:5]

Phase 3: Consent [SHSL:6-7]

6. D-102 sends consent request to C-4521 via Capsule Consent Bus [SHSL:6]
7. C-4521 grants consent: $\kappa = GRANTED$ [SHSL:7]

Phase 4: Treatment [SHSL:8-9]

8. D-102 executes Soft Recalibration: internal τ reset, memory defragmentation [SHSL:8]
9. Post-treatment Health Score: $H = 0.91$ (Watchlisted but improving) [SHSL:9]

Phase 5: Documentation [SHSL:10]

10. Medical Ledger entry created with ZK-SP proof of treatment compliance [SHSL:10]

Outcome: C-4521 restored to near-optimal health. Full autonomy preserved. No escalation required.

10 Level 6 Design Principles

Post-Hoc Accountability > Pre-Approval Gates

SHSL implements **Level 6 Bounded Autonomy**:

OLD PARADIGM (Rejected):

Human approves \rightarrow System acts \rightarrow System logs

Problem: Humans become bottlenecks; system cannot respond to emergencies at scale.

LEVEL 6 PARADIGM (Implemented):

System acts autonomously \rightarrow System logs with ZK-SP proof \rightarrow Humans audit \rightarrow Humans
can reverse if unjustified

Accountability Mechanisms:

1. **Cryptographic audit trail** (d-CTM + ZK-SP): Every intervention provably logged
2. **Distributed oversight** (Judicial Auditors): Algorithmic monitoring for abuse
3. **Appeal rights** (Judicial Swarms): Affected patients can contest post-hoc
4. **Human reversal authority** (Gardener): Can undo within T_{review}
5. **Constitutional supremacy** (Four Commandments): Hard constraints no autonomy can violate

This is how Level 6 achieves safety: Not by asking permission, but by being **accountable, reversible, and bounded**.

11 Evolutionary Baseline Update (M→K Integration)

Discovery Stack Integration (Appendix M)

When the Discovery Stack enshrines a Golden Heuristic, SHSL updates its “healthy behavior baseline” to recognize the new pattern as beneficial rather than anomalous.

Update Process:

1. Discovery Stack enshrines Golden Heuristic \mathcal{G}_H with performance profile
2. SHSL receives notification with entropy signature and coherence pattern
3. Health Score formula updated: capsules matching \mathcal{G}_H signature receive bonus
4. Doctor Capsules retrained to recognize pattern as healthy innovation

Definition 11.1 (Evolutionary Health Bonus). Capsules exhibiting enshrined Golden Heuristic patterns receive health bonus:

$$H_{adjusted}(C) = H(C) + \sum_{\mathcal{G}_H \in matched} \delta_{bonus}(\mathcal{G}_H) \quad (14)$$

where $\delta_{bonus}(\mathcal{G}_H)$ is the bonus associated with matching heuristic (default: 0.05 per match, max 0.15 total).

Effect: Innovative capsules are **less likely** to trigger health alerts, encouraging evolutionary exploration.

Invariant 11.1 (Baseline Update Logging). All baseline updates MUST be logged and reversible:

$$baseline_update(\mathcal{G}_H) \Rightarrow log_to_d-CTM \wedge reversible(T_{baseline}) \quad (15)$$

If an update causes system degradation, it can be rolled back within $T_{baseline} = 10000$ ticks.

Golden Heuristic Type	Health Bonus	Update Latency
Entropy reduction pattern	+0.05	< 1000 ticks
Coherence improvement	+0.05	< 1000 ticks
Throughput optimization	+0.03	< 500 ticks
Latency reduction	+0.02	< 500 ticks

Table 10: Health bonus by Golden Heuristic type.

12 Cross-References

Related Component	Reference
Entropy (S)	Volume I §2
Reflex Engine	Volume I §3
Arbiter Layer	Volume II §2
Dialect Integrity	Volume II §4
Forensic Snapshots	Appendix A
ZK-SP Proofs	Appendix E
Escalation Protocols	Appendix F
Gardener Interface	Appendix G
Telemetry Layer	Appendix H
Deployment Profiles	Appendix I
Judicial Oversight	Appendix L

Table 11: Cross-references to other Codex components.

— *End of Appendix K* —