BIOS: Baseline Inference and Optimization Substrate

Author(s): Withheld for Confidentiality
Purpose: BIOS White Paper Technical Description
Type: Technical White Paper
Version: 2.1
Last updated: 2026-01-11

Abstract

BIOS (Baseline Inference and Optimization Substrate) is a distributed decision-coherence layer intended to reduce long-horizon instability (thrash, oscillation, fragmentation) in complex socio-technical environments. BIOS is non-predictive by design: it does not claim certainty about specific future events. Instead, BIOS evaluates classes of decision trajectories implied by structured artifacts, scoring them for coherence under uncertainty and repeated reassessment. A key protocol is time-offset robustness: forward-only rescoring under controlled perturbations and incremental updates. BIOS emits constraint vectors and manifests (rather than recommendations) that upstream systems interpret as feasibility masks, authorization thresholds, rate limits, sequencing requirements, and budget envelopes. Governance is implemented as a separate plane driven by a connected Governance AI that synthesizes and seals policy bundles; Principals govern membership and funding, but do not author operative rules directly.

Executive Summary

Large organizations fail through temporal incoherence: decisions made at time t are reversed at t+1 under modest updates; plans branch into incompatible subplans; local optimizations cascade into systemic volatility. These failures are amplified by fragmented tooling and incentives, where each subsystem can be “intelligent” while the organization is incoherent as a whole.

BIOS addresses this by acting as infrastructural firmware for decisions. It integrates beneath heterogeneous planning and execution stacks and conditions which option classes remain executable. Upstream systems submit structured decision artifacts. BIOS evaluates the induced trajectory distribution using a bounded state-space representation (the Trajectory Coherence Field, TCF), ensembles, causal hypotheses, and explicit uncertainty. BIOS then emits a constraint vector plus a constraint manifest binding constraints to rationale categories, evidence references, and governance bundle citations, including a procedural contestability pathway.

Governance is separated from optimization to prevent silent enforcement. The Governance Gate is the enforcement boundary: it validates admissibility, signs enforceable manifests, seals policy bundles, and attests federated deployments. Operative governance rules are synthesized by a connected Governance AI as versioned policy bundles and deployed through an explicit lifecycle (propose → replay-validate → stage → seal → attest → roll out). Principals control membership and financing through majority-gated admission and capital-weighted voting with time-decay due to operating costs. Founding Principals fund initial construction

and decide initial seeding; after activation, BIOS internally selects admissible problem domains subject to governance policy.

Key claims

- BIOS is implementable with contemporary distributed systems and applied ML; the hardest problems are integration position, data governance, and reliable operations.
- Time-offset robustness is forward-only and auditable; it does not require future data access or retrocausal signaling.
- Enforcement is bounded by signatures: hard-gated adapters reject unsigned manifests and unsigned override tokens.
- Governance is policy-as-code: a connected Governance AI produces sealed bundles; rule changes are staged and attestable across a federation.

Table of Contents

1. Background and Motivation

The baseline failure mode in high-consequence environments is not ignorance but instability. Decisions that appear justified at time t become indefensible at t+1 when the environment updates, causing reversals and loss of institutional trust. Over time this produces oscillation (thrash), fragmentation (branching into incompatible plans), and cascade risk (small changes triggering disproportionate downstream variance).

BIOS treats coherence as the primary objective. It avoids point-forecast claims and instead scores trajectory families for stability under repeated reassessment. This produces outputs that are measurable, testable in shadow mode, and auditable - properties required for real deployments in regulated or adversarial settings.

1.1 BIOS as substrate

BIOS is not a user-facing application. It is a substrate that integrates beneath other systems. Its outputs are structured constraints consumed by workflow gates, schedulers, risk engines, and execution pipelines. This integration position allows BIOS to shape outcomes without being the formal decision-maker.

2. Goals, Non-Goals, and Assumptions

2.1 Goals

- Reduce decision churn by penalizing update-fragile trajectories.
- Integrate via structured artifacts and constraint vectors (not open-ended text control).
- Maintain deterministic replayability through pinned model/policy bundles and evidence artifacts.
- Support federated deployments with local data boundaries and divergence detection.
- Enforce governance through signed artifacts and sealed policy bundles.

2.2 Non-Goals

- Predicting specific future events with certainty claims.
- Replacing institutional authority or legal accountability.
- Providing open public participation in problem selection or governance.

- Treating natural language as the binding enforcement surface.

2.3 Assumptions

- Decision proposals can be represented as structured artifacts with provenance binding.
- Upstream systems can incorporate constraint outputs without full architectural replacement.
- Adversaries may attempt input manipulation; provenance integrity and replay are mandatory.
- The system must degrade safely under partial failure (conservative default).

3. Definitions, Notation, and System Requirements

3.1 Core objects

- Decision Artifact: schema-validated proposal with parameters, assumptions, dependencies, and provenance.
- Trajectory: branching sequence of state transitions induced by an artifact under modeled dynamics.
- TCF: Trajectory Coherence Field; bounded admissible state-space representation used for scoring.
- Evaluator Ensemble: pinned set of evaluators producing score distributions and evidence artifacts.
- Constraint Vector: machine-readable feasibility/threshold/budget/sequencing modulation.
- Constraint Manifest: signed record binding constraints to evidence and governance bundle citations.
- Governance Gate: admissibility validator and signer; seals policy bundles and attests deployments.
- Principals: designated membership set that controls membership and funding of the BIOS White Paper.

3.2 Selected system requirements

ID	Requirement	Rationale
R1	Enforced constraints MUST have a signed manifest.	Auditable enforcement boundary.
R2	Policy bundles MUST be versioned, sealed artifacts.	Prevents silent rule drift; enables replay.
R3	Hard-gated adapters MUST reject unsigned manifests.	Prevents bypass of governance gate.
R4	Overrides MUST be represented as signed tokens.	Prevents informal bypass; logs accountability.
R5	Federated nodes MUST attest active bundle hashes.	Detects divergence/compromise.
R6	Shadow mode MUST precede hard enforcement.	Safe rollout and calibration.

3.3 Notation (informal)

Let A be an artifact, S_t a state snapshot, and T(A) the induced trajectory distribution. A perturbation policy Π defines forward-only stress updates. Evaluators compute score distributions f(A, S_t, Π). A policy template PT maps scores to constraints C. Governance seals a policy bundle GB_t containing (Π, PT, override logic, contestability workflows) and signs manifests binding (A, C, GB).

4. Architecture Overview and Trust Boundaries

BIOS separates optimization from governance. Optimization proposes constraints based on scoring. Governance authorizes and signs enforceable artifacts. Adapters enforce only signed manifests. This mirrors existing patterns in safety-critical and regulated systems (policy-as-code, approval gates, interlocks).

Component	Role	Boundary / enforcement
Artifact Ingestion	Schema validation, normalization, provenance binding.	Reject / quarantine invalid artifacts.
TCF Engine	Maintain bounded state-space and uncertainty.	Versioned state updates.
Evaluator Ensemble	Compute scores + evidence under Π.	Pinned evaluator set; drift monitored.
Constraint Composer	Map scores → constraint vectors via PT.	Uses sealed PT only.

Component	Role	Boundary / enforcement
Governance Gate	Admissibility, signing, sealing, attestation.	Keys + append-only governance log.
Adapters	Apply constraints to upstream systems.	Verify signatures; enforce gating.
Audit/Replay	Deterministic reconstruction.	Pinned hashes + evidence objects.

4.1 Reference flow

```
Artifact -> Ingest -> Evaluate (TCF+Ensemble+Π) -> Scores/Evidence -> Compose C via PT
    -> Governance Gate (admissibility + sign) -> Adapter enforce -> Audit/Replay
```

5. Interfaces and Data Model

BIOS is designed for integration. Interfaces are explicit and structured. Natural language may exist as annotation, but it is not the binding enforcement surface.

5.1 Core service interfaces (logical)

- SubmitArtifact(artifact): validate schema/provenance; return artifact_id + receipt.
- EvaluateArtifact(artifact_id, bundle_id): compute score distribution + evidence; propose constraints.
- AuthorizeConstraint(proposal): governance admissibility; return signed manifest or denial record.
- ApplyConstraint(manifest): adapter enforcement ack; reject if signatures invalid.
- AppealConstraint(manifest_id, appeal): create appeal record; route to adjudication workflow.
- GetReplayBundle(artifact_id, manifest_id): return pinned hashes + evidence pointers.

5.2 Canonical IDs and hashing

Artifacts, evaluators, templates, bundles, and manifests are immutably identified. Canonical serialization enables stable hashing for caching and replay. Hashes appear in logs and manifests; pinned hashes provide deterministic reconstruction.

5.3 Example schemas (simplified)

```
{
  "artifact_id": "DA-2026-01-11-1120",
  "domain": "Allocation/Risk",
  "proposed_actions": [{"type": "IncreaseExposure", "target": "Class:X", "delta": 0.15}],
  "assumptions": {"horizon_days": 180},
  "dependencies": [{"requires": "ApprovalTier:GovSigned"}],
  "risk_tags": ["highAmplification", "Irreversible"],
  "provenance": {"origin_system": "PlanningStack:A", "created_at": "2026-01-11T17:59Z",
                 "author_id": "UID-REDACTED", "signature": "SIG-ORIGIN-55AF"}
}

{
  "manifest_id": "CM-2026-01-11-8F3A",
  "artifact_id": "DA-2026-01-11-1120",
  "constraint_vector": {"class": "SOFT_GATE", "cooldown_h": 48, "approval": "GovSigned"},
  "rationale": {"categories": ["OscillationRisk", "Amplification"], "evidence_refs": ["EV-77C1"]},
  "governance": {"bundle_id": "GB-2026-01-11-rc2", "signatures": ["SIG-GATE-9B12"]},
  "contestability": {"appeal_channel": "Standard", "sla_h": 72, "reversal_guaranteed": false}
}
```

6. TCF Modeling: Structure, Causality, Uncertainty

The TCF is a bounded representation of admissible future state space used to score trajectory families. BIOS does not require a single monolithic model. Instead it composes structural dependencies, causal hypotheses, and uncertainty ensembles to produce consistent scoring under repeated reassessment.

6.1 Structural layer

Structural dependencies are represented using incrementally updateable probabilistic structures (e.g., factorized graphical representations). This supports partial updates, decomposition by domain, and federated operation.

6.2 Causal layer

Where feasible, causal hypotheses support intervention reasoning and counterfactual evaluation. Causal artifacts are versioned and governed to prevent silent shifts in what counts as an intervention.

6.3 Uncertainty and ensembles

Ensembles represent epistemic uncertainty. Disagreement among evaluators is treated as risk. High disagreement increases stability penalties for high-amplification and high-irreversibility trajectories, encouraging conservative constraint templates.

6.4 Metric suite (example)

Metric	Operational meaning	Typical constraint mapping
Update invariance	Rank stability under incremental updates	Increase approval tier; add cooldown
Oscillation index	Sensitivity to small parameter drift	Throttle reversals; raise evidence floor
Branch proliferation	Growth of incompatible subplans	Require staging; prerequisites
Amplification potential	Cascade sensitivity	Apply budgets; restrict leverage-like actions
Rollback feasibility	One-way-door risk proxy	Require override token; hard gate class

7. Evaluation: Time-Offset Robustness and Evidence Artifacts

BIOS evaluates induced trajectories under repeated reassessment. Time-offset robustness rescans artifacts after controlled forward-only updates and bounded perturbations, penalizing decisions that succeed only at a single snapshot.

7.1 Perturbation policy Π

Perturbation policy defines what counts as stress: evidence update rules, drift assumptions, and adversarial bounds. Π is embedded in the governance bundle and recorded in every evaluation and manifest for replay.

7.2 Evidence artifacts

Evaluations produce evidence artifacts: perturbation triggers, ranking shifts, and invariants. Evidence artifacts enable deterministic replay and contestability review.

7.3 Pseudocode (illustrative)

```
scores = []
evidence = []
for eval in EvaluatorEnsemble(bundle_id):
    for step in range(0, K):
        S = ApplyPerturbation(CurrentState(), Pi=bundle.Pi, step=step)
        s, ev = eval.Score(artifact, S)
        scores.append(s); evidence.append(ev)

agg = Aggregate(scores, evidence)
proposal = ComposeConstraints(agg, PT=bundle.PT)
manifest = GovernanceAuthorizeAndSign(proposal, bundle_id)
```

8. Constraint Layer: Vectors, Manifests, Shadow Trajectories, Overrides

Constraints are BIOS’s executable outputs. They shape feasibility, thresholds, and sequencing in upstream systems without BIOS issuing commands.

8.1 Constraint taxonomy

- Soft gating: cooldowns, throttles, confidence floors, escalation tiers.
- Hard gating: explicit non-executability unless a signed override token is presented.
- Budgets: risk/exposure/resource envelopes as enforceable bounds.
- Sequencing: prerequisites, staged rollout, and kill-switch triggers.
- Visibility shaping: disclosure/redaction settings controlled by governance bundle.

8.2 Manifests and shadow trajectories

Every constraint emission produces a manifest. Suppressed options remain as shadow trajectories to preserve reconstructability and procedural contestability.

8.3 Overrides

Overrides are issued as signed tokens according to governance bundle logic. Hard-gated adapters verify override signatures. Override issuance is logged and may be rate-limited.

8.4 Contestability

Contestability is procedural: affected parties can see constraints (via manifests) and submit appeals as structured artifacts. Governance adjudication produces signed outcomes. Reversal is not guaranteed; process integrity and audit linkage are.

9. Deployment and Federation: Modes, Reliability, Operations

BIOS is deployed as a federation of nodes. Federation enables compartmentalization and local data boundaries while sharing methodology via sealed bundles and attestation.

9.1 Deployment modes

- Shadow mode (compute only; constraints not enforced)
- Soft enforcement (throttles / elevated approvals)
- Hard enforcement (signed non-executability)

9.2 Reliability targets (illustrative)

Category	Target	Notes
Availability	>= 99.9% (evaluation)	Degrade conservatively on failure
Constraint emit p95	<= 250 ms	Evidence artifacts may be async
Audit integrity	100% signed manifests	Adapters enforce signature boundary
Rollback	<= 15 minutes	Pinned bundles; last-known-good

9.3 Attestation and divergence detection

Federated nodes periodically attest their active model hashes and governance bundle hashes. Divergence is treated as an incident and triggers isolation or rollback.

10. Governance Plane: Governance AI, Policy Bundles, Sealing, Attestation

Operative governance rules are implemented as policy-as-code bundles synthesized by a connected Governance AI. The governance plane’s primary job is to make rule changes explicit, versioned, replay-validated, and enforceable only through signatures.

10.1 Governance authority model

Principals do not directly author rules. Principals control membership and funding and may set high-level objective coefficients (risk appetite, stability thresholds, operational budget ceilings). The Governance AI converts those coefficients plus observed system behavior into operative rules encoded in policy bundles.

10.2 Governance bundle contents

- Perturbation policy Π (stress model and bounds)
- Policy templates PT (score → constraint mappings)
- Override issuance logic and authorization tiers

- Contestability workflow definitions (channels, SLAs, record formats)
- Monitoring thresholds, drift triggers, and rollback gates

10.3 Bundle lifecycle (closed loop)

- Propose: Governance AI generates candidate bundle deltas.
- Replay-validate: offline evaluation on historical artifacts and adversarial test suites.
- Stage: shadow mode and canary nodes in the federation.
- Seal: Governance Gate signs the bundle and stores it in an append-only registry.
- Attest: nodes prove pinned bundle hashes; divergence alarms trigger incident response.
- Roll out: gradual deployment with explicit rollback strategy.

10.4 Why this is plausible

This structure mirrors real deployments of policy-as-code, regulated change control, and safety interlocks. The novelty is not signatures or templates; it is the use of a connected AI to synthesize operative rules continuously under an explicit artifact lifecycle.

11. BIOS White Paper: Principals, **Membership, Voting Weight, Seeding**

BIOS is operated by the BIOS White Paper. The BIOS White Paper recognizes a limited set of Principals. An entity may be an individual or a standard legal entity (LLC, corporation, government, etc.). Membership is invitation-only in practice due to majority-gated admission.

11.1 Majority-gated admission

A new Principal may be admitted only by a simple majority of existing Principals computed over current voting weight (not headcount).

11.2 Capital-weighted voting with time-decay

Voting weight is determined by sustained contribution to development and ongoing operations. Because BIOS has continuing costs, voting rights decay for lesser payers over time.

Illustrative weight function

```
w_i(t) = Normalize( α · C_i(total) + β · C_i(recent, Δt) + γ · U_i(underwrite) )
       · exp(-λ · delinquency_i)
```

Parameters (α, β, γ, λ) are encoded in governance bundle meta-coefficients and are themselves versioned artifacts. This produces a dynamic voting share aligned to ongoing continuity burden.

11.3 Founding seeding and scope transition

Founding Principals fund initial construction and decide initial seeding (initial domains and baseline posture). After activation, BIOS selects admissible problem domains internally based on tractability and stability-impact scoring, subject to governance bundle policy. There is no external problem submission interface.

12. Integration with Agentic and AGI-Class Systems

BIOS is compatible with agentic systems, including AGI-class planners, because its control surface is structured. Agents propose candidate plans as decision artifacts; BIOS constrains the executable plan space via signed manifests before execution.

12.1 BIOS in the agent compute loop

A planner generates a distribution of candidate artifacts. BIOS evaluates them under time-offset robustness and returns constraint vectors defining feasibility, escalation requirements, and hard gates. The agent regenerates plans within admissible regions. This implements alignment as feasibility shaping rather than instruction following.

12.2 Governance interaction

Because constraints and bundles are sealed and attestable, an agent cannot silently bypass them. Overrides require signed tokens; appeals exist as structured process artifacts.

13. Security and Threat Model

BIOS operates in adversarial environments. Security emphasizes provenance integrity, separation of duties, signed enforcement boundaries, and replayability.

Threat	Impact	Mitigation (high level)
Artifact forgery / schema drift	Manipulated constraints	Signed provenance, strict schemas, quarantine
Data poisoning / backdoors	Evaluator corruption	Offline validation, ensembles, rollback

Threat	Impact	Mitigation (high level)
Governance key compromise	Unauthorized hard gates	Key management, multi-party signing, audit escalation
Override abuse	Bypass constraints	Signed tokens, rate limits, override audits
Federation divergence	Inconsistent behavior	Attestation, pinned bundles, isolation

14. Validation, Monitoring, and Failure Modes

Validation is staged. Shadow mode replay establishes baseline metrics and false suppression rates. Soft enforcement introduces constraints while measuring override pressure. Hard enforcement is deployed only after stability and false suppression bounds are met.

14.1 Monitoring signals

- Constraint rate/severity by domain and bundle version
- Override and appeal rates; adjudication latency
- Decision churn reduction (oscillation index)
- Model drift and ensemble disagreement
- Audit integrity (signature verification, log replication health)

14.2 Failure modes

- Over-regularization (beneficial novelty suppressed)
- Proxy capture (metrics become targets)
- Bundle drift (governance changes too rapidly)
- Federation divergence (inconsistent policy adoption)

15. Implementation Roadmap and Cost Drivers

The primary cost drivers are integration, data governance, reliability engineering, and ongoing operations. Model development is necessary but not usually the limiting factor.

15.1 Phased rollout

- Phase I: Prototype + shadow mode (schemas, ingestion, evaluator ensemble, manifests, logging, replay).
- Phase II: Soft enforcement (governance signing + adapter verification; monitoring and rollback).
- Phase III: Hard enforcement (selected option classes) + federation expansion + attestation.
- Phase IV: Governance AI bundle synthesis loop (continuous propose → replay → stage → seal).

15.2 Scaling considerations

- Constraint output standardization enables many upstream integrations.
- Federation reduces single-point failure and supports jurisdictional compartmentalization.
- Change control discipline (bundle lifecycle + rollback) is required for operability.

16. External Interfaces and Communications Posture

Public surfaces are intentionally minimal. The white paper is the primary technical artifact. Inbound correspondence (if enabled) is accepted via a web form protected by CAPTCHA, with an explicit consent-to-contact checkbox. Submissions require email verification prior to review. Media inquiries are not handled through the inbound form.

16.1 Contact form verification workflow

A single validation email confirms address control and consent. Unconfirmed submissions are not reviewed. This reduces automated misuse and provides a durable consent record prior to any outreach.

17. References

- J. Pearl, Causality: Models, Reasoning, and Inference, 2nd ed., 2009.
- R. Sutton and A. Barto, Reinforcement Learning: An Introduction, 2nd ed., 2018.
- D. Koller and N. Friedman, Probabilistic Graphical Models: Principles and Techniques, 2009.
- S. Boyd and L. Vandenberghe, Convex Optimization, 2004.
- NIST, AI Risk Management Framework (AI RMF 1.0), 2023.

Appendix A: Example Policy Bundle Sketch (Simplified)

```
{
  "bundle_id": "GB-2026-01-11-rc2",
  "Pi": {"drift_bounds": "bounded", "adversarial_tests": ["AT-01","AT-02"]},
  "PT": {"maps": [{"metric": "Amplification", "threshold": 0.75, "constraint": "SOFT_GATE"}]},
  "override": {"requires_token": true, "rate_limit_per_day": 50},
  "appeals": {"channel": "Standard", "sla_h": 72, "reversal_guaranteed": false}
}
```

Appendix B: Example Audit Log Record

```
2026-01-11T18:03:22Z EVENT=CONSTRAINT_EMIT
artifact_id=DA-2026-01-11-1120 manifest_id=CM-2026-01-11-8F3A
model_bundle=MB-2026-01-10-rc3 bundle_id=GB-2026-01-11-rc2
governance_signature=SIG-GATE-9B12 outcome=APPROVED
notes=SOFT_GATE; appeal_path=yes; reversal_guaranteed=no
```