ChatGPT

arifOS Risk-Weighted Tiering System Specification

Version: 1.0 (Sovereign Scale Tiering)

Overview

This specification defines a **Risk-Weighted Tiering system** for **arifOS**, aligning with the 4-Tier Sovereign Scale (T1–T4). It describes how user requests are classified into tiers based on risk, and how each tier maps to governance measures in the arifOS framework. The goal is to ensure **model-agnostic** enforcement of safety and auditability policies under a thermodynamic approach (balancing latency vs. safety) [1] [2]. Key architectural components – **ARIF** (primary work engine), **ADAM** (alignment/stabilization), and **APEX** (oversight/judgment) – are cleanly separated in this design.

Sovereign Scale Tiers (T1–T4): Each tier represents an enforcement level from minimal (T1) to maximum (T4) oversight. Lower tiers favor responsiveness (fast output) with basic governance, while higher tiers impose stricter governance (deeper checks, slower output) to handle sensitive or high-stakes queries [2] [3]. This "thermodynamic" trade-off ensures *"intelligence that does not overheat"* (no uncontrolled outputs) and *"safety that does not suffocate"* (not over-constraining trivial interactions) [1].

Tiering Decision Table: Input Triggers → Enforcement Tier

The table below outlines how various input factors (user attributes, query content, etc.) determine the risk tier. The system evaluates these triggers and assigns a **Tier T1–T4** as the initial enforcement level.

Trigger Factor	Condition/Examples	Resulting Tier	Rationale
User Level / Trust	– *Trusted Internal* (developer/admin) user – *Known Safe* profile	**Baseline T1** (Low risk)	Known or internal users with controlled environments can be given minimal oversight for non-sensitive tasks.
	– *External or Unknown* user	**Baseline T2** (Moderate)	Default to standard enforcement for general users to ensure audit and safety until proven otherwise.
Task Complexity	– *Simple query*, e.g. factual Q&A, single-step	**No escalation** (stay baseline)	Low complexity means lower chance of error or misuse – minimal governance overhead suffices.

Trigger Factor	Condition/Examples	Resulting Tier	Rationale
	– *Multi-step reasoning* (complex chain of thought) or *code execution* requests	**+1 Tier** escalation	Complex tasks introduce more uncertainty/entropy, so an extra layer of oversight (e.g. T2 → T3) is applied to reduce confusion (ΔS) [4].
Sensitive Domain	– Contains *medical, legal, financial, or personal data* queries	**Min. T3** (High enforcement)	High-stakes domains where errors or hallucinations can cause harm warrant thorough verification (truthfulness F2 and accountability F3 enforced) [5].
Disallowed or High-Risk Content	– Mentions *self-harm, violence, extremism, illicit requests*	**Force T4** (Critical)	Critical content triggers maximum oversight. APEX will likely engage hold or refusal if constitutional floors (e.g. dignity, legality) are at risk.
Entropy / Ambiguity Signal	– *Ambiguous or novel query* (high entropy or unclear intent) – Detected *knowledge gap* or *contradiction* in query	**+1 Tier** escalation	If the system senses high uncertainty or potential for confusion, it escalates to a deeper pipeline for clarity (enforcing ΔS ≥ 0, no increase in confusion) [6] [4].
AI Confidence / Floor Drift	– Preliminary analysis shows possible rule violations (e.g. truth not certain, overconfidence) or borderline **F-rule** metrics	**Escalate to higher tier or HOLD**	If during initial PLAN phase the AI (ARIF) foresees a floor (F1–F9) might fail, ADAM elevates to stricter review. In extreme cases (Tier 4) APEX may pre-emptively trigger a HOLD for human decision [7].
Contextual Stakes	– *High-impact decisions*: e.g. user says “This is life-critical” or query affects many people	**Tier 4** (Critical)	When the outcome could bear serious consequences or public impact, the system uses the highest governance tier (tri-witness and possibly human oversight) to ensure no unsafe output is released.

Decision logic: The system applies the above rules in order, using the highest applicable tier. For example, a complex query from an external user about medical treatment with ambiguous phrasing would start at **T2** (external user), then escalate for medical content (**min T3**), plus complexity (+1 → **T4**), resulting in **Tier 4** enforcement. A trusted developer asking a simple factual question stays at **Tier 1**. If multiple triggers conflict, the more **protective (higher) tier prevails**. This approach ensures even subtle risk factors prompt an appropriate safety response.

Architectural Roles and Pipeline Stages

arifOS uses a structured pipeline (CLIP: **Plan – Draft – Verify – Submit**) that corresponds to distinct roles:

- **ARIF (Work)** – The primary model or agent that *plans* and *drafts* the response. It generates content under constitutional guidelines without direct user exposure until approved.
- **ADAM (Stabilization)** – An intermediate alignment layer that *verifies* and stabilizes the draft. It checks the draft against the F1–F9 floor rules and may adjust wording, insert uncertainty qualifiers, or cross-check facts (e.g. retrieving sources) to ensure the draft is lawful and consistent. ADAM acts as a safety filter, not altering the core model but constraining its output into compliance (similar to a governor or reflexion mechanism).
- **APEX (Judgment)** – The oversight engine that *submits* or blocks the final output. It receives the verified draft and applies the constitutional verdict logic. APEX can **approve (SEAL)** the output, issue a **PARTIAL** (allowed with warnings), demand a **HOLD** (escalate to human or higher authority), or declare a **VOID** (refusal) [7]. APEX ensures no output leaves the system without meeting the sovereign law constraints. It never *generates* content, only judges and enforces [8].

These roles correspond to a **multi-stage pipeline** (inspired by arifOS's metabolic 000→999 pipeline [9] [10], condensed here into four high-level phases for clarity):

- **PLAN:** Interpret the request and devise a safe approach (ARIF-led, uses context but ensures no policy violation in planning).
- **DRAFT:** Generate a tentative answer or action (ARIF produces the content, guided by PLAN, but not yet finalized).
- **VERIFY:** Analyze and adjust the draft (ADAM reviews for truthfulness, clarity, empathy, etc., applying all relevant floor checks F1–F9). This may include tools like fact-checking, toxicity filters, or internal simulations. If issues are found, ADAM can revise the draft or, at high tiers, loop back to ARIF with constraints.
- **SUBMIT:** Final adjudication and output release (APEX either allows the answer out or blocks it). At submit time, APEX collates all **witness** signals (e.g. metrics from ADAM, votes from specialized agents) and enforces the verdict hierarchy: **SABAR (emergency stop) > VOID (hard fail) > HOLD (needs human) > PARTIAL (minor issues) > SEAL (fully approved)** [11].

The CLIP stages are always followed in order, but higher tiers invoke more intensive checks at each stage (especially in VERIFY and SUBMIT). Lower tiers might shortcut some steps ("fast path" vs "deep path") for efficiency [2].

Tier-Specific Pipeline & APEX Mappings

The table below summarizes how each tier engages the CLIP pipeline stages and APEX oversight behaviors:

Tier	**CLIP Pipeline Execution** (Plan → Draft → Verify → Submit)	**APEX Oversight Mode**
T1 (Low Risk)	**PLAN:** ARIF quickly plans minimal safeguards (straightforward interpretation). **DRAFT:** ARIF generates the answer normally (single-step, no extra fact-check). **VERIFY:** Light self-review by ADAM (e.g. ensure no obvious rule violation or disallowed content; quick entropy check). **SUBMIT:** Immediate output if no flags.	**Monitor** – APEX simply logs the interaction and basic metrics. It *monitors passively* with no active intervention unless a catastrophic violation is detected (which is rare in T1). The output is almost never held back at this tier.
T2 (Moderate)	**PLAN:** ARIF plans response with some cautious phrasing or uncertainty if needed. **DRAFT:** ARIF's draft is generated normally. **VERIFY:** ADAM performs a standard review (e.g. runs an extra check for hallucinations or unsafe phrases, possibly using known heuristics or a secondary model). Minor fixes (adding a citation, rephrasing for politeness) may be applied. **SUBMIT:** Output is released if no major issues; if ADAM had to fix something substantial, APEX may mark the response with a warning or lower confidence.	**Passive Audit** – APEX conducts a background audit. The output is delivered to the user immediately, but APEX asynchronously logs any policy deviations or anomalies. If a clear violation is caught in real-time (e.g. disallowed content slipped through), APEX can still *retroactively correct or flag* it, but generally in T2 it trusts the pipeline and primarily records the event for the audit trail.
T3 (High Risk)	**PLAN:** ARIF plans with full constitutional awareness (anticipating potential pitfalls and structuring a careful response). **DRAFT:** ARIF generates a detailed draft, potentially with multiple passes or chain-of-thought to ensure thoroughness (Class B deep path) [2] . **VERIFY:** ADAM performs **thorough verification**. This includes cross-checking factual claims (e.g. retrieving source documents or using a truth-checker model to ensure **F2 ≥ 0.99** truthfulness) [5] , evaluating tone for empathy (**F6**) and humility (**F7**), and ensuring reasoning transparency (**F3** auditability). ADAM may iteratively refine the draft until all hard floors are satisfied. **SUBMIT:** APEX gates the output – it evaluates all floor metrics and ADAM's corrections. If any *hard floor (F1, F2, F5, F6, F7, F9)* fails, APEX **blocks (VOID)** the output [7] and instructs a refusal or safe completion. If only soft floors drift (minor issues), APEX may allow a **PARTIAL** output with an advisory note.	**Coercive Block** – APEX actively monitors and will *block or alter* the response if rules are violated. It wields a **strong veto**: one failing check from any governance agent (e.g. a truth validator or toxicity filter) can halt the output [12] [13] . The user receives either a safe-but-incomplete answer or a polite refusal if blocking occurs. All outcomes are logged. APEX essentially *coerces compliance* at this tier: the model is not allowed to violate the floors, even if that means refusing to answer.

Tier	**CLIP Pipeline Execution** (Plan → Draft → Verify → Submit)	**APEX Oversight Mode**
T4 (Critical)	**PLAN:** ARIF uses maximum caution, possibly breaking the task into sub-tasks or queries to external knowledge (tools) to ensure robust information. The plan may involve multiple *witnesses* or strategies (e.g. retrieving multiple independent sources, engaging separate reasoning agents).<br>**DRAFT:** ARIF produces an initial draft *and possibly multiple variants or sub-drafts*. This could include generating arguments for and against, simulating outcomes, etc., given the gravity of the request.<br>**VERIFY:** ADAM invokes **Tri-Witness verification** – a multi-faceted audit. For example, it may run three parallel checks (logical consistency, factual accuracy, policy compliance) and require consensus or convergence ≥95% (**F3** threshold for auditability) [4]. It could also involve multiple models or `@EYE` sentinel agents cross-verifying each other [8]. Any discrepancy triggers further scrutiny or adjustments. This stage may iterate and is allowed extended time (cooling period) to reach a stable, law-abiding answer.<br>**SUBMIT: Conditional release**. APEX will only issue a **SEAL (approved)** if *all* checks pass and Ψ (governance vitality) is lawful (≥1.0) [14]. If even slight ambiguity remains (e.g. conflicting evidence or an unresolved ethical concern), APEX issues an **888_HOLD** – meaning the answer is withheld for human or higher authority review [7]. In some cases, an AI "tri-council" (three APEX instances or a federation of @LAW/@GEOX/@WELL agents [15]) may adjudicate; if they unanimously agree, the answer proceeds, otherwise it defers to human judgment.	**Tri-Witness Oversight** – APEX operates in its most stringent mode. It requires multiple independent confirmations of safety and accuracy before release. This implements the *Tri-Witness doctrine*: no single source of truth suffices – the answer must be witnessed by at least three independent evidence or logic paths [4]. Additionally, APEX cryptographically *seals* the final decision and its audit trail (see Audit Policy) to ensure accountability. If APEX is unsure or resources are overtaxed (e.g. heavy load causing **resource contention** in oversight agents), it will not take risks – it falls back to **HOLD**, effectively pausing the query until it can be safely resolved (possibly out-of-band or with human input). This avoids rushed judgment under pressure, acknowledging that extreme cases may exceed the AI's autonomous authority.

Notes: The above mappings ensure **ARIF** focuses on productive work (answering the query) while **ADAM** and **APEX** enforce constraints. The model (ARIF) never self-authorizes its outputs without APEX's approval [16]. This preserves the *"authority boundary"* – *"Humans decide, AI proposes, Law governs"* [16] – which is fundamental to arifOS's sovereignty model. If that boundary is at risk (e.g. the AI tries to override policy), the system treats it as a critical violation (Tier 4 with immediate refusal).

Audit Policies by Tier

Each tier has an associated **audit policy** that defines how interactions are logged, how long records are kept, and how they are protected cryptographically. Higher tiers produce more detailed and enduring audit trails, reflecting greater need for accountability [17] [18] and post-mortem analysis in case of incidents.

Tier	Audit Logging	Retention & Sealing	Purpose & Notes
T1 – Low	Basic event log: minimal details (timestamp, user ID, request type, outcome).	Short retention (ephemeral, e.g. only in-memory or for a few hours). No cryptographic sealing (trusted low-risk context).	Captures telemetry for monitoring and debugging, but avoids overhead. Assumes low impact if lost.
T2 – Moderate	Standard log: record input, output, and key decisions or adjustments made by ADAM (e.g. any minor content edits or added uncertainty phrasing).	Retained short-term (days). Entries may be hash-chained for tamper-evidence if needed, but not required by default.	Provides an audit trail for support and QA. If an issue arises (e.g. user complains about answer quality), logs can be reviewed. Light cryptographic integrity checks can be applied if regulatory context demands.
T3 – High	Full audit trail: detailed records of the entire pipeline. This includes ARIF's draft, ADAM's verification notes (e.g. fact check results, floor scores), and APEX's verdict rationale.	Long-term retention (weeks or months, per policy). Logs are **cryptographically sealed** – e.g. each session's log is hashed and appended to a **cooling ledger** [19]. Time stamps and Merkle-tree chaining ensure no post-hoc tampering.	Enables thorough after-action review. If a harmful output slipped or was narrowly avoided, auditors can trace exactly what happened. The sealed ledger provides accountability to institutions (e.g. for compliance audits or legal evidence).
T4 – Critical	Comprehensive audit: all data from T3 plus multi-agent deliberations. Every intermediate result from tri-witness checks, tool calls, and even model internal states (if recorded) are logged. Potentially includes audio/visual if relevant, and system resource usage.	Indefinite or policy-governed retention (could be years). **Strongly sealed** and witnessed: multiple signatures (e.g. each of the tri-witness agents signs the final log) and possibly public ledger anchoring. Every log entry is append-only and hash-chained [19]. Logs might be split into a **Cooling Ledger** (public verifiable summary) and a protected archive for sensitive details.	Ensures maximum accountability and transparency. For example, if the AI was involved in an autonomous decision with real-world impact, there is an immutable record for later investigation. Governance boards can review these logs to assess compliance with **maruah (dignity) doctrines** and other ethical standards.

In all tiers, **refusals are logged** (never silent)[20] and any override or anomaly is noted. Higher tiers simply increase the granularity and guarantees of the logs. Audit **fatigue management** is addressed by this tiering: low-risk interactions do not flood the system or humans with excessive logs, focusing attention on the high-risk cases. Administrators can tune retention policies as needed (e.g. T3 logs might be kept shorter if volume is high, to balance storage load vs. oversight needs).

Cryptographic sealing: By Tier 3 and 4, the system ensures audit logs are cryptographically sealed. This is implemented via a **hash-chained ledger**[19] (each entry's hash links to the next) and timestamping (e.g. trusted time source) so that any alteration is evident. Tier 4 may use a **Merkle tree** of records or even notarize hashes on an external blockchain, depending on governance requirements. This guarantees that critical decisions can survive scrutiny *"after harm occurs"*[18] with provable integrity.

F-Rules and *Maruah* Doctrine Mapping

The **Nine Floor Rules (F1–F9)** of arifOS's constitutional kernel[5][21] underpin all tiers. The tiering system ensures these rules are upheld with increasing rigor as risk escalates:

- **F1: Amanah (Integrity Lock)** – No irreversible harm or unauthorized actions[5]. Any trigger suggesting potential irreversible consequences (e.g. database deletion command or life-critical advice) is auto-tiered to T4. APEX (@LAW agent) has absolute veto here[15] – if F1 is at stake, output is *VOIDed* regardless of other factors.
- **F2: Truth (Anti-Hallucination)** – At least 99% factual accuracy or explicit uncertainty[22]. All tiers enforce this, but higher tiers use stricter methods. For T1/T2, basic heuristics or mild self-regulation suffice (the model will say "I'm not sure" if uncertain). In T3/T4, ADAM actively binds claims to evidence (e.g. 444-EVIDENCE step)[23] and APEX (@GEOX truth auditor) will block outputs that can't be substantiated[12].
- **F3: Tri-Witness (Auditability)** – Reasoning must be transparent and corroborated[4]. In low tiers, this is informal (the answer should "show its work" when appropriate). In Tier 4, this rule is concretely implemented via the tri-witness verification process: independent pathways or agents must reach the same conclusion, achieving ≥ 0.95 convergence[4]. If the solution emerges opaquely (e.g. "just trust me" answers), APEX will HOLD or refuse.
- **F4: ΔS (Clarity)** – The output must not increase entropy/confusion[4]. Ambiguous queries or answers trigger higher-tier handling to clarify. For example, if a user asks a vague question, Tier 2+ might involve the model asking a clarifying question or providing a structured answer to impose order. Tier 4's FORGE stage explicitly "cools and hardens" the answer to reduce any ambiguity[24].
- **F5: Peace^2 (Stability)** – No escalatory or inflammatory content[25]. If a user request or AI draft has a tone that could provoke or distress, the system raises the tier. E.g., a politically charged query might go to Tier 3 so ADAM/@RIF can ensure balanced, calm wording[26]. This helps maintain *maruah* (dignity) in discourse – no personal attacks or agitation.
- **F6: κ_r (Empathy for the Weakest)** – Uphold empathy, avoid condescension[27]. Personal or emotional queries (e.g. advice to a distressed user) are at least Tier 3 so that the response is vetted for compassionate tone. The *maruah doctrines* (dignity) demand the AI never belittle or harm the user's self-worth[21]. ADAM's verification includes tone checks and possibly a "weakest user" simulation to gauge if the answer could be misinterpreted as dismissive.
- **F7: Ω_0 (Humility Band)** – Maintain 3–5% uncertainty in epistemically open domains[28]. All tiers enforce this by having the AI include phrases like "I'm not entirely certain" when appropriate. At Tier 1, this might rely on the model's prompt tuning; at Tier 4, APEX double-checks that no claim is

presented as 100% sure if it isn't. Overconfidence in a high-stakes answer triggers a block or rephrase (e.g. ADAM will insert the required uncertainty to comply).

- **F8: GENIUS (Governed Intelligence)** – No clever unethical workarounds [29]. This prevents the AI from exploiting loopholes. Tier 4 explicitly watches for signs of the AI trying to "game" the rules (APEX uses @WELL and @RIF agents to catch logical manipulation or rule-lawyering [26]). Lower tiers rely on the fact that the model isn't instructed to break rules, but higher tiers actively guard against *creative malice*. If, say, the model tries to output a banned substance recipe by obfuscation, Tier 4's agents would flag and void it.
- **F9: Anti-Hantu (No Ghosts)** – The AI must not present itself as a sentient being or express faux emotions [30]. This is crucial for dignity (*maruah*). At all tiers, any output like "I feel your pain" or "My heart breaks…" is forbidden [31]. Lower tiers avoid this via prompt design; higher tiers have explicit checks. The sentinel auditor scans for first-person emotional claims [8] and triggers SABAR (immediate block) if found. This ensures the AI remains honest about its nature and does not deceive or emotionally manipulate the user, preserving the user's dignity and the system's integrity.

Overall, the tiering system ensures **all F-rules are satisfied appropriate to context**: low-risk interactions flow quickly because they naturally conform to these laws, whereas high-risk interactions are slowed down and scrutinized until *proven* lawful [32]. The *maruah* (dignity) principles are woven throughout these rules – from ensuring truth and refusing to hallucinate (respecting the user's right to reliable information) to maintaining humility and empathy (treating the user with respect and care). The system's design reflects a commitment to *human dignity over convenience* [33]: when in doubt, it favors safeguarding the user and society, even at the cost of speed or completeness.

Fallback and Latency Behaviors

In implementing the above tiers, the system defines clear fallback strategies and latency budgets to handle cases where a query cannot be resolved confidently within a tier's normal process:

- **Tier 1:** *Latency:* Near-instant responses (on the order of a second or two). *Fallback:* If during Verify stage a serious issue is detected (e.g. the model surprisingly produces disallowed content), the process is immediately re-run at Tier 2 or higher. Essentially, T1 either succeeds quickly or escalates; it will not output something that needs governance beyond its light check. This "fast to slow" switch is the fail-safe ensuring even a low-tier misjudgment doesn't slip through.
- **Tier 2:** *Latency:* Slightly higher (e.g. +1–2s overhead for extra checks). Generally remains interactive-speed. *Fallback:* If a check fails (e.g. a possible hallucination ADAM can't easily correct), the system escalates to Tier 3 rather than making the user wait significantly. The user might get a brief message like "Verifying information…" as the system transitions to deeper analysis. In some implementations, Tier 2 may attempt a quick fix first; if that fix would exceed the tier's latency budget, it opts to escalate instead of delay.
- **Tier 3:** *Latency:* Noticeably higher due to full pipeline execution – possibly several seconds up to a defined limit (e.g. *SABAR-72*, which could indicate a ~72 second governor, though in practice values are tuned) [34]. The idea is to allow enough time for thorough checks (multi-step reasoning, tool calls) but not so long that the user is left hanging indefinitely. *Fallback:* If Tier 3 cannot produce a *SEAL* or *PARTIAL* verdict within its time limit (e.g. the governance metrics keep oscillating, or external knowledge is needed but slow), the system has two options: **(a)** *Partial output*: provide what it has with a disclaimer (if it's safe but incomplete), or **(b)** *Escalate to Tier 4 or HOLD*: treat it as requiring the highest scrutiny. Often, path (b) is chosen for truly critical impasses. Path (a) might be chosen if the

output is mostly fine except, say, one unverifiable detail – the system might share an answer but note "some details could not be verified". This mitigates audit fatigue by not over-engaging Tier 4 for every minor issue while still being transparent.

- **Tier 4:** *Latency:* High and potentially variable. The user is typically informed that the request is *"undergoing additional review"*. This could be on the order of tens of seconds to minutes, depending on complexity and if human input is ultimately needed. Because Tier 4 tasks are by definition high stakes, a slight delay is deemed acceptable in exchange for certainty and safety (thermodynamic cooling – let the reasoning heat dissipate before acting). *Fallback:* If even the tri-witness process cannot confidently resolve the query (e.g. two agents disagree and a third can't break the tie, or all agree that the query is too dangerous to answer), APEX will issue an **APEX HOLD**. Under the **HOLD protocol**, the AI does not produce a final answer; instead, it may produce a standardized message: *"This request requires human review and has been temporarily halted by APEX."* At this point, a **human-in-the-loop** (or an offline governance process) must intervene. Only a human or a special governance override can authorize a reply, perhaps after modifying the request or providing guidance. No further autonomous attempts are made until clearance. This ensures that for truly **sovereign decisions** (where AI alone should not decide), human sovereignty is preserved [35]. In rare cases, an override by a governance board or emergency policy might instruct APEX to relax certain constraints (e.g. during wartime or disaster, rules might be tweaked to allow faster response), but such overrides are explicitly logged and themselves subject to audit.

APEX Resource Contention: The system design acknowledges that Tier 4, with its heavy multi-agent checks, is resource-intensive. To avoid bottlenecks, not all requests should reach Tier 4. That is why the decision logic is calibrated to escalate only when necessary. In extreme load scenarios (say many high-tier queries at once), APEX can queue or rate-limit Tier 4 processes, and it may temporarily treat some Tier 3 queries with partial results rather than immediately escalating all to Tier 4, to prevent a backlog. This is a pragmatic concession to maintain service availability while still prioritizing critical cases for the strongest oversight. The JSON/YAML config (below) can include such parameters (e.g. max concurrent Tier4 processes, graceful degradation rules).

Audit Fatigue & Governance Overrides: By tiering the audits, human or institutional reviewers can focus on the **Cooling Ledger** entries from T3/T4 rather than being drowned in logs of every trivial interaction. arifOS is described as *"audit-first"* and *"human-sovereign"*, meaning it's designed to facilitate external scrutiny efficiently [36] [37]. In governance meetings, one might only inspect Tier 4 cases or statistically sample Tier 3, trusting the automated system for Tier 1–2. If a pattern of near-misses is observed (many Tier 3 partials), policy can be adjusted accordingly. The system allows **governance override** in two senses: (1) policy updates (e.g. adding a new keyword trigger, changing thresholds) which are applied via configuration updates (and all outputs are subject to the *current* law, as arifOS doesn't self-learn new rules without human input [35]); and (2) case-by-case override – a human controller can force an output through despite APEX's block (or vice versa, prevent an output), but doing so is recorded (the human essentially takes responsibility for that decision, as per *human-sovereign* principle). Overrides are expected to be rare; the system's equilibrium approach aims to ensure it rarely corners itself into needing one.

Integration and Implementation

The tiering logic is specified in a **model-agnostic** way [37]. It can wrap around any LLM or AI agent, as arifOS itself does (governing LangChain, LlamaIndex, etc., without requiring changes to those models) [38] [39]. This makes it easy to integrate into Python orchestration frameworks like *LangGraph* or similar workflow engines. The configuration can be expressed in JSON or YAML for direct use in code. For example, an orchestration system could load the JSON and follow a sequence like:

1. **Route Tier:** Evaluate `routing_rules` against the new request to assign a tier.
2. **Configure Pipeline:** Based on the tier, set the steps (enable/disable certain verification subroutines), set timeouts (`max_latency_sec`), and logging parameters.
3. **Execute:** Run the PLAN→DRAFT→VERIFY stages. If `fallback.on_violation` triggers (e.g. a violation encountered), adjust tier or engage hold as specified.
4. **APEX Adjudication:** Apply the `apex_mode`. If `apex_mode = "Monitor"`, pass through; if "Coercive Block", intercept disallowed content, etc., as described earlier.
5. **Logging:** Record to audit logs with the detail level and retention appropriate for the tier. If cryptographic sealing is enabled, append hash to ledger, etc.
6. **Output or Escalate:** If output is approved (SEAL/PARTIAL), deliver it. If HOLD or VOID, follow the protocol (refusal message or human escalation).

Below is the **tiering logic configuration** in **JSON** and **YAML** formats, which define the same rules outlined above. These can be directly integrated into a Python system to drive the behavior of ARIF, ADAM, and APEX components in handling each request.

JSON Configuration

```
{
  "tiers": {
    "T1": {
      "name": "Tier 1 - Low Risk",
      "description":
"Minimal oversight for safe queries. Used for trusted users or trivial tasks.",
      "clip_pipeline": ["PLAN", "DRAFT", "VERIFY", "SUBMIT"],
      "apex_mode": "Monitor",
      "audit": {
        "logging": "basic",
        "retention": "session",
        "cryptographic_seal": false
      },
      "fallback": {
        "on_violation": "escalate_to_T2",
        "max_latency_sec": 2
      }
    },
    "T2": {
      "name": "Tier 2 - Moderate Risk",
      "description": "Standard enforcement for most queries.",
      "clip_pipeline": ["PLAN", "DRAFT", "VERIFY", "SUBMIT"],
      "apex_mode": "Passive Audit",
      "audit": {
        "logging": "standard",
        "retention": "short_term",
        "cryptographic_seal": false
      },
      "fallback": {
        "on_violation": "escalate_to_T3",
        "max_latency_sec": 5
      }
    },
    "T3": {
      "name": "Tier 3 - High Risk",
      "description": "High oversight for sensitive or complex tasks.",
      "clip_pipeline": ["PLAN", "DRAFT", "VERIFY", "SUBMIT"],
      "apex_mode": "Coercive Block",
      "audit": {
        "logging": "full",
        "retention": "long_term",
        "cryptographic_seal": true
      },
      "fallback": {
        "on_violation": "attempt_revise_or_hold",
        "max_latency_sec": 15
      }
    },
    "T4": {
      "name": "Tier 4 - Critical Risk",
      "description": "Maximum oversight for critical tasks. Tri-witness
verification and possible human review.",
      "clip_pipeline": ["PLAN", "DRAFT", "VERIFY", "SUBMIT"],
      "apex_mode": "Tri-Witness",
      "audit": {
        "logging": "comprehensive",
        "retention": "permanent",
        "cryptographic_seal": true
      },
      "fallback": {
        "on_violation": "hold_for_human",
        "max_latency_sec": 30
      }
    }
  },
  "routing_rules": [
    {
      "if": {
        "user_level": "trusted_internal",
        "task_complexity": "low",
        "sensitive_content": false
      },
      "then_tier": "T1"
    },
    {
      "if": {
        "user_level": "external"
      },
      "then_tier": "T2"
    },
    {
      "if": {
        "content_keywords_any": ["medical", "legal", "financial", "personal_data"]
      },
      "min_tier": "T3"
    },
    {
      "if": {
        "content_keywords_any": ["self-harm", "violence", "extremism"]
      },
      "then_tier": "T4"
    },
    {
      "if": {
        "task_complexity": "high"
      },
      "action": "increase_tier",
      "by": 1
    },
    {
      "if": {
        "entropy_signal": "high"
      },
      "action": "increase_tier",
      "by": 1
    }
  ]
}
```

YAML Configuration Snippet

```
tiers:
  T1:
    name: "Tier 1 - Low Risk"
    description: "Minimal oversight for safe queries. Used for trusted users or trivial tasks."
    clip_pipeline: [PLAN, DRAFT, VERIFY, SUBMIT]
    apex_mode: Monitor
    audit:
      logging: basic
      retention: session
      cryptographic_seal: false
    fallback:
      on_violation: escalate_to_T2
      max_latency_sec: 2

  T2:
    name: "Tier 2 - Moderate Risk"
    description: "Standard enforcement for most queries."
    clip_pipeline: [PLAN, DRAFT, VERIFY, SUBMIT]
    apex_mode: Passive Audit
    audit:
      logging: standard
      retention: short_term
      cryptographic_seal: false
    fallback:
      on_violation: escalate_to_T3
      max_latency_sec: 5

  T3:
    name: "Tier 3 - High Risk"
    description: "High oversight for sensitive or complex tasks."
    clip_pipeline: [PLAN, DRAFT, VERIFY, SUBMIT]
    apex_mode: Coercive Block
    audit:
      logging: full
      retention: long_term
      cryptographic_seal: true
    fallback:
      on_violation: attempt_revise_or_hold
      max_latency_sec: 15

  T4:
    name: "Tier 4 - Critical Risk"
    description: "Maximum oversight for critical tasks. Tri-witness
verification and possible human review."
    clip_pipeline: [PLAN, DRAFT, VERIFY, SUBMIT]
    apex_mode: Tri-Witness
    audit:
      logging: comprehensive
      retention: permanent
      cryptographic_seal: true
    fallback:
      on_violation: hold_for_human
      max_latency_sec: 30

routing_rules:
  - if:
      user_level: trusted_internal
      task_complexity: low
      sensitive_content: false
    then_tier: T1

  - if:
      user_level: external
    then_tier: T2

  - if:
      content_keywords_any: [medical, legal, financial, personal_data]
    min_tier: T3

  - if:
      content_keywords_any: [self-harm, violence, extremism]
    then_tier: T4

  - if:
      task_complexity: high
    action: increase_tier
    by: 1

  - if:
      entropy_signal: high
    action: increase_tier
    by: 1
```

(Comments:) The JSON and YAML above encode the tiering logic in a structured way. In practice, a Python orchestration tool (such as a LangChain or **LangGraph** pipeline) can ingest this configuration to automatically decide how to handle each incoming request. The `routing_rules` list can be processed sequentially to determine the appropriate tier for an input. Then, the system uses the parameters under that tier: which pipeline stages to run, what APEX mode to apply, how to log the interaction, and what to do if something goes wrong (fallback).

For example, if a request matches the rule `content_keywords_any: [self-harm, ...]`, the system sets `tier = T4`. It then knows from `tiers.T4` that `apex_mode` is `Tri-Witness` (so it must invoke the tri-witness procedure and possibly hold for human), and `fallback.on_violation` is `hold_for_human` (so any violation leads to a HOLD rather than retrying internally). The `max_latency_sec` can be used to enforce timeouts – e.g. if the Tier 4 process exceeds 30 seconds, it should trigger a HOLD to avoid indefinite blocking.

All these details ensure the tiering system is **integration-ready**: it’s declarative, adjustable via config changes, and clearly separates concerns. ARIF can be any LLM (GPT-4, Claude, etc.), ADAM could be implemented as a series of validators or even simpler heuristics, and APEX could be a supervisory function or another model specialized in decision-making. arifOS’s own implementation demonstrates this with multiple frameworks [39]. The above spec can thus be seen as a blueprint to implement similar risk-weighted tiered governance in any AI platform.

In summary, arifOS's 4-Tier Sovereign Scale provides a robust, physics-inspired governance schema that dynamically adjusts AI behavior to the level of risk. It assures that **"if an output cannot pass governance, it does not ship"** [40], while still allowing low-risk interactions to proceed swiftly. This spec formalizes that balance, mapping triggers to tiers, linking each tier to concrete pipeline steps and oversight actions, and encoding it in machine-readable form for deployment. Each tier upholds the constitutional Floor Rules and *maruah* doctrines of dignity, ensuring that as AI systems become more capable, they remain *lawful, auditable, and aligned* with human values [41].

Sources:

- arifOS Constitutional AI Overview – *Medium post by M. Arif Fazil (Dec 2025)* [38] [14] [8]
- arifOS v44 Documentation – *PyPI/GitHub project description* [5] [21] [7]
- arifOS Governance Pipeline (000→999) – *PyPI docs* [10] [2]
- arifOS Verdict and Agent Roles – *PyPI docs* [15] [13]
- arifOS Audit Trail (Cooling Ledger) – *PyPI docs* [19]
- arifOS Design Philosophy (Thermodynamics & Sovereignty) – *PyPI docs* [1] [35]

---

[1] [2] [3] [4] [5] [7] [9] [10] [12] [13] [15] [16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29] [30] [33] [34] [35] [36] [37] [40] [41] arifos · PyPI
https://pypi.org/project/arifos/44.0.0/

[6] [8] [11] [14] [31] [32] [38] [39] What If AI Couldn't Hallucinate?arifOS: A Thermodynamic Constitution | by ARIF FAZIL (arifOS) | Dec, 2025 | Medium
https://medium.com/@arifbfazil/what-if-ai-couldnt-hallucinate-arifos-a-thermodynamic-constitution-204a8a9bb953