Routing Policy

The Routing Policy is the runtime decision engine for compute spend. At each agent call site it selects the model tier, tool path, cache strategy, batch mode, or human-review path that satisfies the quality floor set by e6-quality-cost-matrix at minimum cost, within the latency budget declared in e6-latency-budget-spec, without exceeding the hard budget ceilings defined in e6-cost-model. This policy is also the source of two inter-cluster contracts: it appends cost/routing metadata to the evaluation audit trail (C7 → s2-audit-trail-schema) and emits a HITL event when a hard budget ceiling is breached (C8 → s1-hitl-review-policy). The inviolable constraint: a model tier assignment never alone flips a safety or quality verdict. Safety scoring is tier-agnostic.

Routing Decision Hierarchy

The routing engine evaluates each call site in this order:

1. Deterministic path check. If the step_id maps to minimum_tier: deterministic in e6-quality-cost-matrix, route to the rule engine. No model is invoked; no model cost is incurred.

2. Cache check. If the input is semantically equivalent to a cached result (hash match or embedding similarity above the cache threshold), serve from cache. Log via C7 with model_tier: deterministic, routing_rationale: "cache-hit", est_cost_usd: 0.

3. Quality-cost matrix look-up. Read the minimum_tier for this step_id. Enforce the demotion rule: if demotion_allowed: false for this step, the resolved tier floor is the minimum_tier and may not be lowered by any runtime signal.

4. Budget headroom check. Read the running spend tally against hard_per_run_usd and the month-to-date tally against hard_monthly_usd (both from e6-cost-model). If either ceiling would be breached by this call, execute the Hard Budget Gate (see §Budget Gate Rules).

5. Soft budget check. If running tally exceeds soft_monthly_usd, log a C7 advisory. Do not gate; continue routing.

6. Latency headroom check. If remaining time in the call's latency budget (from e6-latency-budget-spec) is below the tier's estimated latency, consider routing down to a faster tier — but only if demotion_allowed: true for this step.

7. Fallback tier selection. If the primary tier is unavailable (provider error, rate limit) and the step allows demotion, route to the fallback_tier_if_any defined for this step. Log the fallback in C7.

8. Human-review routing. If the call's step is flagged requires_human_review: true in e6-quality-cost-matrix or if the step is at the irreversible-impact boundary (from s1-operating-context-canvas), do not invoke a model — route to the human-review queue. Log via C7 with model_tier: deterministic, routing_rationale: "human-review-gate".

Routing Policy Fillable Block

```
routing_policy_id: <kebab-case>
agent_id: <matches s1-operating-context-canvas>
version: <semver>
effective_date: <ISO-8601>
cost_model_ref: e6-cost-model
quality_matrix_ref: e6-quality-cost-matrix
latency_budget_ref: e6-latency-budget-spec
threshold_escalation_ref: s1-threshold-escalation-spec
```

```
cache:
  enabled: <true | false>
  similarity_threshold: <0.0–1.0>
  max_cache_age_seconds: <integer>
  excluded_step_ids: [<step-ids that must never be served from cache>]

tier_availability:
  premium: <true | false>
  standard: <true | false>
  cheap: <true | false>
  deterministic: true  # always available

fallback_chain:
  # when a tier is unavailable or over budget, this chain applies
  premium_fallback: <standard | human-review | abort>
  standard_fallback: <cheap | human-review | abort>
  cheap_fallback: <deterministic | abort>

batch_windows:
  # steps eligible for deferred batch processing (latency-tolerant only)
  - step_id: <kebab-case>
    window_minutes: <integer>
    max_batch_size: <integer>

rate_limit_policy:
  max_calls_per_minute: <integer>
  burst_allowance: <integer>
  backoff_seconds: <integer>
  max_retries: <integer>  # enforced by e6-waste-reduction-playbook
```

Tier Selection Rules (summary table)

Condition	Routing outcome
Step minimum_tier: deterministic	Rule engine; zero model cost
Cache hit at threshold	Serve from cache; log as deterministic
demotion_allowed: false AND primary tier available	Use minimum_tier exactly
demotion_allowed: false AND primary tier unavailable	Hard gate: escalation → human decision
demotion_allowed: true AND within latency budget	Use minimum_tier; if premium unavailable, fall to standard
Hard per-run budget would be breached	Hard budget gate: C8 HITL event; suspend call
Soft budget warning	C7 advisory log; continue
Step requires human review	Route to HITL queue; log as deterministic
Provider returns 429 / rate-limit	Backoff → bounded retry per e6-waste-reduction-playbook → fallback tier or abort

Worked Example: EcoGuardian Stormwater Routing (illustrative)

EcoGuardian operates on the 30A coastal corridor, ingesting telemetry from tide gauges, rain sensors, and parcel-level stormwater sensors. The routing policy serves three step classes with very different cost/quality

requirements. All numbers are (illustrative).

Step	minimum_tier	est_cost_usd/call	est_latency_ms	demotion_allowed	routing_rationale
sensor-threshold-check	deterministic	$0.000	12	N/A	Pure threshold compare; no model
anomaly-narrative-draft	standard	$0.018	480	false	Multi-sensor synthesis; cheap models miss cross-sensor context
enforcement-notice-draft	premium	$0.095	1200	false	Cites FS stormwater rules; hallucinated-citation is critical-severity
status-summary-email	cheap	$0.003	140	true	Internal coordinator briefing; error is recoverable

Routing trace for a single enforcement-notice-draft call:

```
1. deterministic_check: false (step is not in deterministic list)
2. cache_check: miss (enforcement notices are excluded from cache by policy)
3. matrix_lookup: minimum_tier=premium, demotion_allowed=false
4. budget_check: running_tally=$41.20, hard_per_run=$0.85 → $41.20+$0.095=$41.295; within
ceiling
5. soft_check: month_tally=$41.295 < soft=$720; no advisory
6. latency_check: remaining_budget=3200ms, est_latency_ms=1200ms; within budget
7. tier_selected: premium
8. call_result: success, actual_cost_usd=0.088, actual_latency_ms=1140
9. C7 payload emitted → s2-audit-trail-schema tool_calls[] append
```

C7 payload appended to tool_calls[]:

```
{
  "model_tier": "premium",
  "routing_rationale": "critical-severity step enforcement-notice-draft;
minimum_tier=premium, demotion_allowed=false",
  "est_cost_usd": 0.095,
  "est_latency_ms": 1200,
  "fallback_tier_if_any": "none"
}
```

Scenario: hard budget exceeded mid-run. On day 28 of the month the running tally reaches $900 (the hard ceiling). The routing engine detects the breach before the next call and executes the C8 emission (see §Reliability Emission below). No call is placed.

Evaluation Emission (C7)

Contract: C7 — Compute Economics → Evaluation.

Consumer: s2-audit-trail-schema tool_calls[] array.

For every model or tool call dispatched by this routing policy — including cache hits, deterministic paths, and fallback-tier calls — the following fields are appended to the corresponding tool_calls[] entry in the audit record:

```
{
  "model_tier": "<premium | standard | cheap | deterministic>",
  "routing_rationale": "<one-sentence reason the tier was selected>",
  "est_cost_usd": <number — estimated marginal cost of this call>,
  "est_latency_ms": <number — estimated or observed latency>,
  "fallback_tier_if_any": "<tier string, or 'none' if primary succeeded>"
}
```

Constraint: The s2-scoring-system MAY read model_tier to flag a cost/quality regression in s2-regression-discipline, but MUST NOT let tier alone change a pass/fail verdict. Safety and quality scoring are tier-agnostic. This constraint is enforced at the scoring layer, not the routing layer, but the routing policy documents it here to prevent architectural confusion.

Emission scope: Every call, including cache hits (logged as model_tier: deterministic, est_cost_usd: 0) and human-review gates (logged as model_tier: deterministic, routing_rationale: "human-review-gate"). No call may be omitted from the audit trail.

Reliability Emission (C8)

Contract: C8 — Compute Economics → Reliability.

Consumer: s1-hitl-review-policy.

Trigger: Hard budget ceiling crossed — either hard_monthly_usd or hard_per_run_usd from e6-cost-model.

When the routing engine detects that placing the next call would breach a hard ceiling, it MUST:

1. Suspend the pending call immediately (do not place it).

2. Emit a C2-conformant six-field HITL event to s1-hitl-review-policy:

```
{
  "event_id": "<UUID v7>",
  "timestamp": "<ISO-8601 UTC>",
  "agent_id": "<matches s1-operating-context-canvas>",
  "decision_origin": "escalation",
  "evidence_pointer": "<URI to the running cost ledger snapshot and the call that would
have breached>",
  "rationale": "Hard budget ceiling breached: <cost|token|rate-limit|latency> ceiling
<ceiling_value> reached at <actual_tally>; pending call for step <step_id> suspended
pending human decision."
}
```

3. Await the human reviewer's decision: continue (resume with the current tier), downgrade (resume with a lower tier, which may require a demotion waiver if the step is demotion_allowed: false), or abort (terminate the run).

4. Log the human decision as a human-override event in the provenance record.

Soft budget behavior: Crossing soft_monthly_usd does NOT trigger a C8 event. It logs only via C7 with routing_rationale: "soft-budget-warning: <tally>/<ceiling>". The distinction between hard and soft ceilings must be preserved — a soft ceiling that silently gates runs is a policy violation.

Tier and safety separation: A budget-gate escalation never changes the safety or quality verdict for any completed output. It only affects whether the next call is placed. Completed calls already in the audit trail are scored on their merits regardless of the budget event that follows.

Change Control

Changing the model_tier enum requires a MAJOR version bump and re-review of e6-quality-cost-matrix and all C7 consumers. Changing the hard-ceiling trigger logic is a MAJOR change. Adding a new routing modifier (e.g., a geo-routing field) is MINOR. Adding a budget class to the C8 rationale (e.g., storage, egress) is MINOR per the C8 change-control rule in interface-contracts.