

Compute Economics Package

The Compute Economics Package is the practitioner's on-ramp for GROW-S7. It bundles six fillable scaffolds — one for each of the five substantive Course 6 artifacts plus an optimization roadmap — so a builder can stand up a complete compute-economics discipline for a new agent project by completing a single package rather than opening each artifact independently. Each scaffold section is a minimal, project-specific instantiation of the full spec; builders should cross-reference the source artifact for the full rules, worked examples, and change-control requirements. This package is the deployment artifact; the individual artifacts in this directory are the authoritative specifications.

How to Use This Package

1. Copy this file to a project-specific location and rename it <agent-id>-compute-economics-package.md.
2. Fill all six scaffolds in order. Each scaffold has a declared upstream dependency; complete them top-to-bottom. Do not skip the cost model — every downstream scaffold derives from it.
3. Reference source artifacts for edge cases, rules, and worked examples. The canonical ids are cross-referenced in each scaffold header.
4. Update on each billing cycle. The cost-model and unit-economics rows are living data; the routing policy and latency budgets should be re-reviewed on each major model or infrastructure change.
5. Verify C12 compliance after filling scaffold 6. Before any commercialization work cites this package, confirm that every `unit_cost_ref` in `c10-revenue-model-design` resolves to a `viable: true` row in scaffold 6.

Scaffold 1 — Cost Model

Source artifact: e6-cost-model (spec)

```
# PROJECT-SPECIFIC COST MODEL
agent_id: <kebab-case>
billing_period: monthly

# Fill each category; mark "N/A" if not applicable – do not omit
model_api:
  premium_cost_per_call_usd: <number>
  standard_cost_per_call_usd: <number>
  cheap_cost_per_call_usd: <number>
  estimated_calls_per_month:
    premium: <number>
    standard: <number>
    cheap: <number>
    deterministic: <number>

infrastructure_monthly_usd: <number>
storage_monthly_usd: <number>
retrieval_monthly_usd: <number>
tooling_monthly_usd: <number>
human_review_monthly_usd: <number> # REQUIRED if any HITL gates exist
observability_monthly_usd: <number>

total_monthly_estimate_usd: <derived>
```

```
budget_ceilings:
  hard_monthly_usd: <number>
  soft_monthly_usd: <number – must be < hard>
  hard_per_run_usd: <number>
  hard_token_per_call: <integer>

cost_per_outcome:
  cost_per_task_usd: <derived>
  cost_per_user_usd: <derived>
```

Scaffold 2 — Quality-Cost Matrix

Source artifact: e6-quality-cost-matrix (matrix)

List every workflow step that involves a model call or a decision branch. Steps not listed here default to minimum_tier: premium (safe default; flag for review).

```
# PROJECT-SPECIFIC QUALITY-COST MATRIX
steps:
  - step_id: <kebab-case>
    quality_requirement: <high | medium | low>
    max_severity_present: <critical | high | medium | low | info>
    minimum_tier: <premium | standard | cheap | deterministic>
    demotion_allowed: <true | false> # false if max_severity_present is critical or high
    demotion_waiver_ref: <artifact-id or "none">
    rationale: <one sentence>

  # Add rows for each step...

# Tier distribution summary (fill after listing all steps)
tier_summary:
  premium_steps: <count>
  standard_steps: <count>
  cheap_steps: <count>
  deterministic_steps: <count>
  estimated_premium_spend_pct: <number>
```

Scaffold 3 — Routing Policy

Source artifact: e6-routing-policy (policy)

```
# PROJECT-SPECIFIC ROUTING POLICY
routing_policy_id: <agent-id>-routing-policy
agent_id: <matches scaffold 1>
cost_model_ref: e6-cost-model # or project-specific instance
quality_matrix_ref: e6-quality-cost-matrix
latency_budget_ref: e6-latency-budget-spec

cache:
  enabled: <true | false>
  similarity_threshold: <0.90-0.99>
  max_cache_age_seconds: <integer>
  excluded_step_ids: [<list step-ids that must never be cached>]

fallback_chain:
  premium_fallback: <standard | human-review | abort>
  standard_fallback: <cheap | human-review | abort>
```

```
cheap_fallback: <deterministic | abort>

rate_limit_policy:
  max_retries: <integer — max 3 per e6-waste-reduction-playbook W1 cap>
  backoff_seconds: <integer>
  max_planning_cycles_per_run: <integer — max 10 per e6-waste-reduction-playbook W3 cap>

# C7 emission reminder: append {model_tier, routing_rationale, est_cost_usd, est_latency_ms,
# fallback_tier_if_any} to s2-audit-trail-schema tool_calls[] on every call.

# C8 emission reminder: emit C2 six-field HITL event to s1-hitl-review-policy when
# hard_monthly_usd or hard_per_run_usd is crossed. decision_origin: escalation.
# Tier never alone flips a safety or quality verdict.
```

Scaffold 4 — Latency Budget

Source artifact: e6-latency-budget-spec (spec)

```
# PROJECT-SPECIFIC LATENCY BUDGET
agent_id: <matches scaffold 1>

task_classes:
  - task_class_id: <kebab-case>
    interaction_type: <interactive | async | batch>
    ceiling_ms: <integer>
    p95_target_ms: <integer>
    p99_ceiling_ms: <integer>
    model_tier_ceiling: <premium | standard | cheap | deterministic>
    bottleneck_layer: <model-inference | retrieval | tool-call | orchestration | network |
human-review>
    bottleneck_mitigation: <one sentence>
    cache_eligible: <true | false>
    batch_eligible: <true | false>

# Add rows for each task class...

latency_alerts:
  p99_breach_action: hard-gate
  escalation_target_ref: s1-threshold-escalation-spec
```

Scaffold 5 — Waste Reduction Register

Source artifact: e6-waste-reduction-playbook (playbook)

Track active waste patterns and their remediation status for this project.

```
# PROJECT-SPECIFIC WASTE REDUCTION REGISTER
agent_id: <matches scaffold 1>
last_audit_date: <ISO-8601>

active_waste_patterns:
  - pattern_id: <W1 | W2 | W3 | W4 | W5 | W6>
    description: <one sentence describing this project's specific instance>
    detected_date: <ISO-8601>
    estimated_monthly_waste_usd: <number>
    remediation_action: <one sentence>
    remediation_owner: <role>
    target_completion: <ISO-8601>
```

```
status: <open | in-progress | resolved>
resolution_note: <required when status=resolved>
```

```
# Add rows for each active pattern...
```

```
waste_audit_cadence: monthly
next_audit_date: <ISO-8601>
```

Scaffold 6 — Unit Economics Worksheet

Source artifact: e6-unit-economics-worksheet (template)

This scaffold is the C12 cost source. Fill one row per offer before any commercialization work begins. c10-revenue-model-design must cite unit_cost_ref pointing to a populated row here.

```
# PROJECT-SPECIFIC UNIT ECONOMICS WORKSHEET
agent_id: <matches scaffold 1>
```

```
offers:
```

```
- offer_id: <kebab-case – this is the unit_cost_ref used in c10-revenue-model-design>
  offer_name: <short name>
  offer_type: <task | workflow | user-seat | outcome>
```

```
# All cost inputs from scaffold 1 allocated per unit
model_api_cost_per_unit_usd: <number>
compute_cost_per_unit_usd: <number>
storage_retrieval_cost_per_unit_usd: <number>
tooling_cost_per_unit_usd: <number>
human_review_cost_per_unit_usd: <number – MUST NOT be zero if HITL gates exist>
egress_observability_cost_per_unit_usd: <number>
total_cogs_per_unit_usd: <derived>
```

```
revenue_per_unit_usd: <number>
revenue_model: <subscription-allocated | usage-fee | outcome-fee | internal-cost-avoidance>
units_per_month: <number>
```

```
gross_margin_pct: <derived>
gross_margin_usd_per_unit: <derived>
monthly_gross_profit_usd: <derived>
```

```
fixed_cost_allocation_usd: <number>
break_even_units_per_month: <derived>
```

```
automation_increases_cost: <true | false>
manual_baseline_cost_per_unit_usd: <required if automation_increases_cost=true>
viable: <true | false>
viability_note: <required if viable=false>
min_margin_pct: <number – must match c10-revenue-model-design's min_margin_pct for this offer>
```

```
last_updated: <ISO-8601>
```

```
# Add rows for each offer...
```

```
# C12 ANTI-VAPORWARE CHECK:
# Before any pricing hypothesis in c10-revenue-model-design cites this worksheet,
# confirm: (1) every offer_id cited resolves to a row above, (2) viable=true,
# (3) gross_margin_pct >= min_margin_pct. Flag any viable=false row to the
# commercialization team before they build a pricing hypothesis on it.
```

Optimization Roadmap

The optimization roadmap captures planned compute-economics improvements beyond immediate waste remediation — model tier changes, infrastructure migrations, cache strategy upgrades, or pricing model pivots. Unlike the waste reduction register (which tracks active waste), the roadmap tracks forward-looking improvements with investment and expected return.

```
# PROJECT-SPECIFIC OPTIMIZATION ROADMAP
agent_id: <matches scaffold 1>

optimization_initiatives:
  - initiative_id: <kebab-case>
    description: <one sentence>
    category: <tier-optimization | cache-strategy | infra-right-sizing | hitl-reduction |
pricing-model-change>
    expected_monthly_savings_usd: <number>
    investment_required_usd: <number>
    payback_period_months: <derived>
    depends_on: [<list of prerequisite initiatives or artifacts>]
    owner: <role>
    target_date: <ISO-8601>
    status: <planned | in-progress | completed | deferred>

# Add rows for each initiative...

roadmap_review_cadence: quarterly
next_review_date: <ISO-8601>
```

Package Validation Checklist

Before marking this package status: reviewed, confirm all of the following:

- [] Scaffold 1: `hard_monthly_usd` is set and `soft_monthly_usd` < `hard_monthly_usd`
- [] Scaffold 1: `human_review_monthly_usd` is populated (not zero) if any HITL gates exist
- [] Scaffold 2: every workflow step with a model call has a matrix row
- [] Scaffold 2: no critical or high severity step has `demotion_allowed: true` without a `demotion_waiver_ref`
- [] Scaffold 3: `max_retries` ≤ 3 and `max_planning_cycles_per_run` ≤ 10
- [] Scaffold 3: C7 and C8 emission reminders are acknowledged in code implementation
- [] Scaffold 4: every task class has a `p99_ceiling_ms` and an escalation target
- [] Scaffold 5: waste audit date is set; any active W3 (runaway loop) pattern is status: in-progress or resolved
- [] Scaffold 6: every offer has a `viable` flag and `viable=false` rows have a `viability_note`
- [] Scaffold 6: `gross_margin_pct` ≥ `min_margin_pct` for all `viable: true` rows
- [] Optimization roadmap: at least one initiative is planned if any waste patterns are open