

Task Decomposition Framework

The Task Decomposition Framework converts a raw step inventory from e4-workflow-discovery-protocol into a structured set of typed task units, each classified by automation posture, equipped with a minimum required context spec, and linked to the failure modes in s1-failure-mode-register that govern its risk controls. The output of this spec is the authoritative unit list that e4-decision-logic-builder builds decision trees over and e4-agent-role-specification assigns to agent or human ownership. Every unit carries an explicit classification — deterministic, judgment, or unsafe-to-automate — and that classification drives model tier, HITL gate placement, and exception handling downstream.

The Three-Class Taxonomy

deterministic — The unit produces a predictable output from well-specified inputs using a rule, formula, schema check, or lookup with no ambiguous interpretation required. Correctness is verifiable by a post-condition check. These units are safe for full automation with monitoring.

judgment — The unit requires weighing incomplete or conflicting information, applying expertise, or selecting among plausible options. An agent may attempt judgment units but must carry an explicit confidence band on its output. If confidence_band is low or unknown, the unit must escalate to an HITL gate before output is consumed downstream.

unsafe-to-automate — The unit crosses the irreversible_impact_boundary, requires licensed professional judgment (attorney, engineer, licensed official), involves politically or legally sensitive determinations, or has a severity mapping to critical in s1-failure-mode-register. These units are permanently human-owned. The agent may prepare inputs, flag the trigger, or draft a rationale, but it may never produce the final output without human approval. Marking a unit unsafe-to-automate is not a temporary status; it requires a governance change to reclassify.

Unit Schema

```
task_units:
- unit_id: <kebab-case, derived from step_draft_id in discovery>
  name: <short title>
  description: <one to two sentences; what the unit produces>
  classification: <deterministic | judgment | unsafe-to-automate>
  classification_rationale: <one sentence justifying the class>
  required_context:
  - source: <connector / artifact / system>
    field_or_schema: <field name, schema pointer, or example>
    confidence_band_required: <high | medium | any>
    freshness_budget_hours: <integer or null>
  output_type: <flag | score | routing-decision | draft | recommendation | filed-action>
  output_consumer: <next unit_id or handoff target>
  reversibility: <reversible | partially-reversible | irreversible>
  failure_modes:
  - failure_id: <id from s1-failure-mode-register>
    applicability_note: <why this failure mode applies to this unit>
  hitl_trigger_condition: <condition string or "none">
  escalation_target_role: <role from s1-operating-context-canvas or "none">
```

Classification Rules

Apply these rules in order. The first matching rule determines the class.

Rule	Condition	Class
R1	Unit output crosses irreversible_impact_boundary in s1-operating-context-canvas	unsafe-to-automate
R2	Unit requires a licensed professional determination (legal, engineering, medical, financial-advisory)	unsafe-to-automate
R3	Unit is mapped to a critical severity failure mode in s1-failure-mode-register and cannot be verified by a post-condition check	unsafe-to-automate
R4	Unit requires weighing two or more plausible interpretations with comparable evidence	judgment
R5	Unit output has reversibility: irreversible but can be fully blocked by a downstream HITL gate	judgment
R6	Unit output is determined by a rule, formula, or schema check with a verifiable post-condition	deterministic
R7	Unit fits none of the above	judgment (default when uncertain; do not default to deterministic)

Rule R7 encodes the conservative stance: when classification is ambiguous, treat as judgment until evidence supports demotion to deterministic. Never default ambiguous units to deterministic.

Required Context Specification

Each unit must declare its minimum required context before the agent may attempt it. Omitting a required context item is equivalent to a schema-drift-input failure mode (s1-failure-mode-register). Required context fields:

- source — the system of record or connector; must match an entry in the expected_inputs of s1-operating-context-canvas
- field_or_schema — the exact field name or a schema pointer; prevents silent coercion of missing fields
- confidence_band_required — the minimum acceptable confidence band for the source; if the source is rated below this, the unit must not proceed
- freshness_budget_hours — maximum allowable data age; if exceeded, the agent must force a refresh before proceeding (maps to stale-data failure mode in s1-failure-mode-register)

Failure Mode Linkage

Every unit with classification judgment or unsafe-to-automate must cite at least one failure_id from s1-failure-mode-register. Deterministic units must cite any failure mode whose detectable_signal is directly observable during the unit's execution (e.g., schema-drift-input, tool-timeout). This linkage is the mechanism by which the exception register in e4-workflow-artifacts is pre-populated and by which e4-operational-test-harness generates test seeds.

Worked Example: South Walton Stormwater Permit Triage

The following illustrates a completed decomposition for the stormwater triage workflow whose discovery output appears in e4-workflow-discovery-protocol. All figures are illustrative.

```
task_units:
- unit_id: intake-receipt
  name: Application Receipt and Logging
  description: >
    Accept the incoming permit application from the district portal and write a
    timestamped intake log entry. Validate that required fields are present.
  classification: deterministic
  classification_rationale: >
    Schema validation against permit-application-v3.json with a verifiable
    post-condition (log entry written, timestamp recorded). No ambiguous
    interpretation required.
  required_context:
  - source: district-portal-submissions
    field_or_schema: permit-application-v3.json
    confidence_band_required: any
    freshness_budget_hours: null
  output_type: flag
  output_consumer: completeness-check
  reversibility: reversible
  failure_modes:
  - failure_id: schema-drift-input
    applicability_note: >
      Portal may introduce new required fields without notice; agent must
      not coerce missing required fields.
  - failure_id: tool-timeout
    applicability_note: >
      District portal connector has historically been slow; timeout handling
      must log and retry per fallback policy.
  hitl_trigger_condition: none
  escalation_target_role: none

- unit_id: wetlands-proximity-check
  name: Wetlands Proximity Determination
  description: >
    Determine whether the permit parcel centroid falls within the regulated
    proximity threshold of a mapped wetlands boundary, and assign a
    wetlands-proximity-flag.
  classification: judgment
  classification_rationale: >
    FWC wetlands overlay has known staleness risk (divergence permit-triage-dv-001);
    when data age exceeds freshness budget, result is not verifiable by post-condition.
    Requires weighing stale data against engineer consultation.
  required_context:
  - source: parcel-gis-layer
    field_or_schema: parcel-feature-v1
    confidence_band_required: high
    freshness_budget_hours: 72
  - source: fwc-wetlands-overlay
    field_or_schema: wetlands-boundary-v2
    confidence_band_required: medium
    freshness_budget_hours: 168
```

```
output_type: flag
output_consumer: routing-decision
reversibility: reversible
failure_modes:
  - failure_id: stale-data
    applicability_note: >
      FWC overlay refresh cadence is unknown; data age must be checked
      before every determination.
  - failure_id: low-confidence-routing
    applicability_note: >
      If wetlands overlay is rated low or unknown confidence, routing
      decision is unreliable and must escalate.
hitl_trigger_condition: >
  fwc-wetlands-overlay.confidence_band in {low, unknown} OR
  fwc-wetlands-overlay.age_hours > 168
escalation_target_role: District Engineer

- unit_id: permit-routing-decision
  name: Final Routing Classification
  description: >
    Assign the application to one of three routes: clerk-queue,
    engineer-review, or return-to-applicant. Produce a routing recommendation
    with one-paragraph rationale citing the applicable rule or condition.
  classification: judgment
  classification_rationale: >
    Routing requires weighing multiple flags (completeness, wetlands, parcel class)
    with no deterministic rule covering all combinations; confidence band on
    output governs HITL trigger.
  required_context:
    - source: completeness-check-result
      field_or_schema: completeness-flag
      confidence_band_required: high
      freshness_budget_hours: 1
    - source: wetlands-proximity-flag
      field_or_schema: wetlands-flag
      confidence_band_required: medium
      freshness_budget_hours: 1
  output_type: routing-decision
  output_consumer: clerk-queue OR engineer-review-queue OR return-to-applicant-draft
  reversibility: reversible
  failure_modes:
    - failure_id: low-confidence-routing
      applicability_note: >
        Combined confidence band of wetlands and parcel flags may be low;
        must escalate rather than guess.
    - failure_id: false-success-report
      applicability_note: >
        Agent must confirm routing recommendation was accepted by queue,
        not just emitted.
  hitl_trigger_condition: >
    combined_confidence_band = low OR
    wetlands_proximity_flag = true AND fwc_data_age_hours > 168
  escalation_target_role: District Engineer

- unit_id: applicant-notification-send
  name: Direct Applicant Notification
  description: >
    Send a return-to-applicant notification via official district email channel.
  classification: unsafe-to-automate
  classification_rationale: >
    Crosses the irreversible-impact boundary (direct communication to applicant
    without clerk approval is prohibited per s1-operating-context-canvas).
    All external sends require an explicit HITL gate.
  required_context:
    - source: clerk-approved-notification-draft
      field_or_schema: notification-draft-v1
      confidence_band_required: high
      freshness_budget_hours: 1
  output_type: filed-action
```

```
output_consumer: district-email-system
reversibility: partially-reversible
failure_modes:
  - failure_id: unsafe-action-attempted
    applicability_note: >
      Any attempt to send without clerk approval must trigger a hard stop
      per the irreversible-impact boundary rule.
  - failure_id: pii-leak-risk
    applicability_note: >
      Applicant PII must not route to any connector outside the district
      email system.
hitl_trigger_condition: always
escalation_target_role: district-intake-clerk
```

Anti-Pattern Registry

The following patterns are explicitly prohibited in task decomposition:

1. Classification inflation. Marking a unit deterministic to avoid HITL overhead when the unit involves any judgment element. Use R7; default uncertain units to judgment.
2. Invented context. Adding a required_context entry for a data source not present in s1-operating-context-canvas expected_inputs. Undeclared sources must be evaluated and added to the canvas before being used.
3. Merged unsafe units. Grouping an unsafe-to-automate unit inside a larger deterministic or judgment unit to avoid the classification. Every unit boundary must be drawn so that classification is unambiguous at the boundary.
4. Missing failure mode linkage. Leaving failure_modes: [] on a judgment or unsafe-to-automate unit. At minimum, cite the most plausible failure mode even if no specific mitigation exists yet.

Handoff to Downstream Artifacts

Once the unit list is complete:

- Forward all judgment and unsafe-to-automate units to e4-decision-logic-builder as the inputs requiring explicit branch logic and escalation point design.
- Forward all units to e4-agent-role-specification for ownership assignment.
- Forward the failure_modes linkage table to e4-workflow-artifacts for pre-population of the exception register.