

Agent Role Specification

The Agent Role Specification assigns explicit ownership — agent-owned, human-owned, or jointly-owned with a declared handoff — to every task unit and decision branch produced by e4-task-decomposition-framework and e4-decision-logic-builder. It defines the tools and systems each agent role may invoke, the exact conditions under which the agent must yield to a human (HITL handoffs derived from s1-hitl-review-policy), and the hard prohibition against inventing process steps not present in the discovery-validated workflow model. Without this spec, an agent build has no authoritative boundary between what the agent decides and what humans decide, making both accountability tracing and failure mode recovery impossible.

The Invented-Step Prohibition

An agent role may not execute, propose, or record a workflow step that does not appear in the step inventory from e4-workflow-discovery-protocol and is not assigned to the agent in this spec. This rule is absolute. If the agent's model or reasoning suggests a step that is not in the validated inventory, the agent must surface it as an anomaly — not execute it. The mechanism for surface is a structured escalation with decision_origin: escalation, not a silent action. New steps discovered at runtime must be queued for discovery-protocol review before being added to the workflow model. This prohibition closes the primary silent-expansion failure path in agent workflow builds.

Role Schema

An "agent role" is a named, bounded function the agent performs within the workflow. A single agent instance may fulfill multiple roles, but each role has distinct tool authorities, handoff obligations, and accountability attribution.

```
agent_roles:
- role_id: <kebab-case>
  name: <short title>
  description: <one sentence; what this role is responsible for>
  owned_units: [<unit_id list from e4-task-decomposition-framework>]
  owned_branches: [<node_id and condition_id from e4-decision-logic-builder>]
  tool_authorities:
    - tool_id: <connector, API, or system id>
      invocation_rule: <auto | confirm | human-approval | blocked>
      scope_constraint: <what operations this role may perform on this tool>
  read_only_sources: [<system or connector the role may read but not write>]
  write_targets: [<system or connector the role may write to>]
  prohibited_actions:
    - <explicit prohibition for this role>
  hitl_handoffs:
    - handoff_id: <kebab-case>
      trigger: <condition from e4-decision-logic-builder or policy from s1-hitl-review-policy>
      payload_to_human: <declared fields passed at handoff>
      acceptance_condition: <what the human must do for the workflow to resume>
      resume_branch: <node_id or unit_id that resumes after human action>
      sla_ref: <role and sla_minutes from s1-operating-context-canvas escalation_targets>
  output_constraints:
    - <constraint on what outputs this role may produce; e.g., "draft only, not filed action">
  accountability_attribution: <agent | human | shared>
```

Ownership Assignment Rules

Apply these rules in order when assigning units:

Rule	Condition	Ownership
AO1	Unit classification is unsafe-to-automate	Human-owned; agent may prepare inputs only
AO2	Unit classification is judgment AND hitl_trigger_condition always fires	Jointly-owned; agent prepares, human decides
AO3	Unit classification is judgment AND hitl_trigger_condition is conditional	Agent-owned with conditional handoff
AO4	Unit classification is deterministic	Agent-owned; no HITL required unless a stop condition fires
AO5	Any unit whose output feeds directly into an irreversible_impact_boundary item	Agent-owned with mandatory pre-handoff gate

AO1 is non-negotiable: the agent may compose a draft, retrieve supporting evidence, or flag the trigger, but the final action on an unsafe-to-automate unit is always taken by a human.

HITL Handoff Design

Every HITL handoff declared in e4-decision-logic-builder must be instantiated here with a complete payload specification. A handoff without a declared payload is incomplete and fails the evaluation gate. The payload must include:

- All input data the human needs to make the decision without referencing any other system
- The agent's recommendation or candidate options (for judgment units)
- The confidence_band on each contributing source
- The provenance record reference that will be shared with the C5 emission

Handoffs reference s1-hitl-review-policy for the governing policy rule (e.g., which events generate a formal review record, what the SLA is, and what happens on SLA breach).

Tool Invocation Rules

Each tool authority entry must declare one of four invocation rules — mirroring the tool allow-list convention used in Course 8 (t8-tool-use-control-list):

- auto — Agent invokes without confirmation. Permitted only for read operations on declared read_only_sources and for deterministic transformations with no write side-effect.

- confirm — Agent proposes the invocation; a supervisor role (another agent or automated policy check) confirms before execution. Permitted for deterministic write operations within scope.
- human-approval — A human must explicitly approve before the tool is invoked. Required for any tool invocation that produces an output consumed by an irreversible_impact_boundary step.
- blocked — The role may not invoke this tool under any circumstance. Blocked tools in a role's scope are explicitly listed to document that the restriction was considered, not overlooked.

Worked Example: South Walton Stormwater Permit Triage

The following illustrates a completed role specification for the permit triage workflow. All figures are illustrative.

agent_roles:

```
- role_id: intake-processor
  name: Application Intake Processor
  description: >
    Receives permit applications from the district portal, validates schema,
    logs receipt, and advances to the wetlands proximity check.
  owned_units: [intake-receipt, completeness-check]
  owned_branches:
    - node_id: completeness-gate
      condition_id: complete
    - node_id: completeness-gate
      condition_id: incomplete
  tool_authorities:
    - tool_id: district-portal-connector
      invocation_rule: auto
      scope_constraint: read-only; may read submitted applications; may not modify or
delete
    - tool_id: intake-log-writer
      invocation_rule: confirm
      scope_constraint: append-only; may write new intake records; may not modify
existing records
  read_only_sources: [district-portal-submissions, permit-application-v3-schema]
  write_targets: [intake-log]
  prohibited_actions:
    - May not modify any submitted permit application fields
    - May not send any communication to the applicant
    - May not access or write to parcel-of-record or permit-of-record databases
  hitl_handoffs:
    - handoff_id: schema-drift-escalation
      trigger: >
        schema_validation.error_type in {unexpected-field, type-mismatch-on-required}
      payload_to_human:
        - application_id
        - schema_validation_report
        - field_delta (new vs expected fields)
        - intake_timestamp
      acceptance_condition: >
        Schema owner acknowledges and either updates the schema or clears the application
      resume_branch: intake-receipt
      sla_ref: "District Engineer – 240 min"
  output_constraints:
    - Produces intake-log-entry and completeness-flag only; no external communications
  accountability_attribution: agent

- role_id: wetlands-analyst
  name: Wetlands Proximity Analyst
  description: >
    Evaluates parcel proximity to mapped wetlands boundaries and produces a
    proximity flag with supporting evidence and confidence band.
  owned_units: [wetlands-proximity-check]
```

```
owned_branches:
  - node_id: wetlands-proximity-decision
    condition_id: clear
  - node_id: wetlands-proximity-decision
    condition_id: within-proximity
tool_authorities:
  - tool_id: parcel-gis-connector
    invocation_rule: auto
    scope_constraint: read-only; spatial query against parcel-gis-layer only
  - tool_id: fwc-wetlands-overlay-connector
    invocation_rule: auto
    scope_constraint: read-only; fetch overlay for specific parcel bbox only
read_only_sources: [parcel-gis-layer, fwc-wetlands-overlay]
write_targets: [wetlands-flag-store]
prohibited_actions:
  - May not override a wetlands flag set by a prior engineer determination
  - May not access permit-of-record or issued-permits database
  - May not contact FWC directly
hitl_handoffs:
  - handoff_id: wetlands-data-quality-escalation
    trigger: >
      fwc_overlay.confidence_band in {low, unknown} OR
      fwc_overlay.age_hours > 168
    payload_to_human:
      - application_id
      - parcel_id
      - parcel_to_wetlands_distance_ft
      - fwc_overlay.age_hours
      - fwc_overlay.confidence_band
      - last_refresh_attempted_timestamp
      - agent_recommendation (nearest determination if computable)
    acceptance_condition: >
      District Engineer provides a wetlands flag determination and rationale;
      records the determination in the engineer-review log
    resume_branch: routing-decision
    sla_ref: "District Engineer – 240 min"
output_constraints:
  - Produces wetlands-proximity-flag with confidence_band annotation only
  - May not produce a final permit routing recommendation
accountability_attribution: shared

- role_id: routing-classifier
  name: Permit Routing Classifier
  description: >
    Combines all flags to produce a routing recommendation (clerk-queue,
    engineer-review, or return-to-applicant) with a one-paragraph rationale.
  owned_units: [permit-routing-decision]
  owned_branches:
    - node_id: routing-decision
      condition_id: engineer-review
    - node_id: routing-decision
      condition_id: return-to-applicant
    - node_id: routing-decision
      condition_id: clerk-queue
  tool_authorities:
    - tool_id: routing-queue-writer
      invocation_rule: confirm
      scope_constraint: >
        May write to engineer-review-queue or clerk-queue; may not directly
        notify applicant or modify application record
    - tool_id: return-to-applicant-draft-writer
      invocation_rule: confirm
      scope_constraint: >
        May produce draft notification only; may not invoke district-email-system
  read_only_sources: [completeness-flag-store, wetlands-flag-store, parcel-class-lookup]
  write_targets: [routing-queue, return-to-applicant-draft-store]
  prohibited_actions:
    - May not send any communication to the applicant
    - May not assign a routing decision when combined_confidence_band is low or unknown
```

without HITL approval

- May not create a routing recommendation that references a rule not present in the jurisdiction rule set

hitl_handoffs:

- handoff_id: low-confidence-routing-escalation
- trigger: combined_confidence_band in {low, unknown}
- payload_to_human:
 - application_id
 - all_flags_with_confidence_bands
 - top_two_candidate_routes
 - routing_rationale_draft
 - evidence_pointers
- acceptance_condition: >
 - District Engineer selects and records the routing decision with a signed rationale entry in the routing log
- resume_branch: routing-decision (post-human-decision)
- sla_ref: "District Engineer – 240 min"

output_constraints:

- Produces routing-recommendation with rationale and confidence_band; never a filed

action

- Rationale must cite the specific flag(s) and source(s) that drove the decision
- accountability_attribution: shared

- role_id: notification-drafter

name: Applicant Notification Drafter

description: >

Produces a return-to-applicant notification draft for clerk review.

Does not send; all sends are human-owned.

owned_units: [return-to-applicant-draft]

owned_branches: []

tool_authorities:

- tool_id: notification-template-reader
 - invocation_rule: auto
 - scope_constraint: read-only; fetch notification templates only
- tool_id: draft-writer
 - invocation_rule: auto
 - scope_constraint: write to draft-store only; draft is not deliverable until clerk-

approved

- tool_id: district-email-system
 - invocation_rule: blocked
 - scope_constraint: >

Blocked for this role at all times. Direct email send is human-owned.

read_only_sources: [notification-template-library, application-data, routing-rationale]

write_targets: [return-to-applicant-draft-store]

prohibited_actions:

- May not invoke district-email-system under any condition
- May not access applicant contact information for any purpose other than populating

the draft address field

hitl_handoffs:

- handoff_id: clerk-notification-approval
- trigger: always (every notification draft requires clerk review)
- payload_to_human:
 - application_id
 - draft_notification_text
 - routing_rationale
 - cited_rule_or_deficiency
 - draft_timestamp
- acceptance_condition: >
 - Clerk reviews, optionally edits, and explicitly approves or rejects the draft.
 - Approval recorded in clerk-approval-log.
- resume_branch: notification-send-gate (clerk-approved condition)
- sla_ref: "district-intake-clerk – 480 min (1 business day)"

output_constraints:

- Produces notification-draft only; labeled DRAFT in all outputs
- Must include the applicable rule or deficiency citation in every draft

accountability_attribution: agent

- role_id: human-send-executor

name: Human Notification Send Executor

```
description: >
  Human-owned role. Receives a clerk-approved notification draft and
  executes the send via the district email system.
owned_units: [applicant-notification-send]
owned_branches: []
tool_authorities:
  - tool_id: district-email-system
    invocation_rule: human-approval
    scope_constraint: >
      Human operates directly; agent provides no automation of this action
read_only_sources: []
write_targets: [district-email-system, communication-log]
prohibited_actions:
  - Agent may not assist in executing or automating the email send action
hitl_handoffs: []
output_constraints:
  - All sends logged to communication-log per FS Chapter 119 public records
requirements [VERIFY]
  accountability_attribution: human
```

Usage Notes

Forward this spec to e4-workflow-artifacts as the role matrix input for the process map. Pass all hitl_handoffs to the HITL gate inventory section of that artifact. Pass prohibited_actions and tool_authorities to the exception register as the enforcement test cases for e4-operational-test-harness.

When this spec is instantiated for a specific agent build, every blocked tool invocation rule must be validated in testing to confirm the agent actually cannot invoke the tool — a blocked rule that is not enforced at the tool layer is a spec defect, not a deployment configuration question.