Workflow Artifacts Template

The Workflow Artifacts Template assembles the validated outputs of the GROW-S4 build chain — discovery protocol, task decomposition, decision logic, and role specification — into the five deployable artifacts that together constitute the complete workflow model: a process map, I/O contracts, a role matrix, a review gate register, and handoff rules. It is the C5 producer: for every executed workflow step, the running instance of this workflow emits one provenance record to s3-provenance-metadata-schema carrying {step_id, step_inputs_ref, step_outputs_ref, branch_taken, decision_origin}, with heavy payloads hash-and-ref'd rather than inlined, and with a shared correlation_id linking any HITL event to its C5 sibling record. This template is filled once per workflow model and versioned alongside the workflow.

Section 1 — Process Map

The process map is a directed model of the workflow's steps, decisions, branches, and handoffs. Every node and edge must resolve to a step or branch defined in e4-decision-logic-builder. No implicit steps or undocumented edges are permitted.

```
process_map:
  workflow_id: <kebab-case>
  workflow_name: <human-readable title>
  version: <semver>
  canvas_ref: <agent_id from s1-operating-context-canvas>
  last_updated: <ISO-8601 date>

  steps:
    - step_id: <kebab-case; must match unit_id in e4-task-decomposition-framework>
      name: <short title>
      type: <action | decision | gate | handoff | stop>
      owner: <role_id from e4-agent-role-specification OR "human">
      inputs:
        - source: <connector or prior step_id>
          field: <field name or schema pointer>
      outputs:
        - destination: <next step_id or external system>
          artifact: <artifact name or type>
      branches:
        - edge_id: <condition_id from e4-decision-logic-builder>
          label: <human-readable>
          branch_to: <step_id>
      hitl_gate: <true | false>
      hitl_handoff_ref: <handoff_id from e4-agent-role-specification or null>
      exception_handler_refs: [<exception_id from e4-decision-logic-builder>]
      provenance_emission: required   # C5 — see Section 5
```

Completeness check. Before the process map is considered final, verify:

- [ ] Every unit from e4-task-decomposition-framework appears as exactly one step_id
- [ ] Every branch edge from e4-decision-logic-builder appears as exactly one edge_id
- [ ] Every unsafe-to-automate unit has hitl_gate: true
- [ ] Every stop condition from e4-decision-logic-builder appears as a type: stop step
- [ ] No step appears in the map that is not in the step inventory from e4-workflow-discovery-protocol

Section 2 — I/O Contracts

I/O contracts define the exact interface between each step and its producers and consumers. Every step declares what it requires as input (with schema pointer and freshness constraint) and what it produces as output (with schema pointer and reversibility). This section is the basis for the test harness input fixtures in e4-operational-test-harness.

```
io_contracts:
  - step_id: <matches process_map step_id>
    inputs:
      - input_id: <kebab-case>
        source_system: <connector or step_id>
        schema_pointer: <URI or filename>
        confidence_band_required: <high | medium | any>
        freshness_budget_hours: <integer or null>
        nullable: <true | false>
        on_missing: <stop | fallback | escalate>
    outputs:
      - output_id: <kebab-case>
        destination: <step_id or external system>
        schema_pointer: <URI or filename>
        reversibility: <reversible | partially-reversible | irreversible>
        post_condition_check: <expression verifying the output is correct>
```

The on_missing policy for each input must be one of:

- stop — halt the step and emit a stop event (required when the input is load-bearing and no safe assumption can substitute)
- fallback — execute the declared fallback behavior (requires a named fallback in s1-fallback-architecture-blueprint)
- escalate — raise an HITL escalation (required when a human judgment can resolve the gap)

Section 3 — Role Matrix

The role matrix is the compact cross-reference of steps to roles. It is derived from e4-agent-role-specification and serves as the deployment-time authorization table.

```
role_matrix:
  - step_id: <matches process_map>
    primary_owner: <role_id>
    secondary_owner: <role_id or null>
    ownership_type: <agent-only | human-only | shared>
    tool_invocations_allowed: [<tool_id list>]
    tool_invocations_blocked: [<tool_id list>]
    can_produce_final_output: <true | false>
    hitl_required: <true | false | conditional>
    hitl_condition: <expression or null>
```

Section 4 — Review Gate Register

Review gates are the set of checkpoints at which the workflow must pause for explicit approval before proceeding. This register combines HITL gates (derived from e4-decision-logic-builder and e4-agent-role-specification) with any additional process review gates required by policy, compliance, or the irreversible_impact_boundary in

s1-operating-context-canvas.

```
review_gate_register:
  - gate_id: <kebab-case>
    step_ref: <step_id>
    gate_type: <hitl | policy-review | compliance-sign-off | final-approval>
    trigger_condition: <expression>
    reviewer_role: <role from s1-operating-context-canvas escalation_targets>
    sla_minutes: <integer>
    payload_required: [<fields the reviewer must receive>]
    on_approval: <step_id to resume>
    on_rejection: <step_id or terminal>
    on_sla_breach: <escalation action>
    blocks_irreversible_step: <true | false>
```

Every gate with blocks_irreversible_step: true must appear in the HITL gate inventory in s1-hitl-review-policy.

Section 5 — Provenance Emission (C5)

This section is mandatory for every deployed workflow instance. Per the C5 contract in interface-contracts, every step that produces an output or selects a branch emits exactly one provenance record to s3-provenance-metadata-schema. Drops are gate-failing.

Per-Step Emission Spec

```
c5_emission:
  step_id: <matches process_map step_id>
  correlation_id: <UUID v7; shared with any C2 HITL event fired at this step>
  record:
    step_id: <same as above>
    step_inputs_ref:
      ref: <pointer to the S2 audit record containing the full input payload>
      hash: <SHA-256 of the input payload at time of step execution>
      field: inputs
    step_outputs_ref:
      ref: <pointer to the S2 audit record containing the full output payload>
      hash: <SHA-256 of the output payload>
      field: outputs
    branch_taken: <edge_id of the branch selected, or "none" for linear steps>
    decision_origin: <"workflow-step" for autonomous steps; "human-override" | "escalation"
| "fallback" when a gate fired>
```

Emission Rules

1. Every step emits. A step that produces no output still emits a record with step_outputs_ref.hash of the empty payload.

2. Heavy payloads are never inlined. The C5 record carries only {ref, hash, field} pointers. The full payload lives in the S2 audit record. Inlining violates the C3 split-storage model.

3. HITL events share correlation_id. When an HITL gate fires at a step, the C2 event emitted by s1-hitl-review-policy and the C5 record for that step carry the same correlation_id. This is the linkage that allows provenance consumers to join the gate event to the workflow context.

4. decision_origin must be accurate. If the step executed autonomously, use workflow-step. If a human override occurred at the gate, use human-override. If a fallback fired, use fallback. If the HITL escalation decided the outcome, use escalation. Using workflow-step for a human-decided outcome is a provenance integrity violation.

5. Stop conditions emit. A step that terminates via a stop condition emits a record with the stop event as the output and branch_taken: stop-<stop_id>.

Emission Implementation Checklist

- [ ] Every step in the process map has a corresponding C5 emission record template
- [ ] All step_inputs_ref and step_outputs_ref point to the S2 audit record, not the source system
- [ ] Every HITL gate step has a correlation_id field populated and shared with the C2 event
- [ ] Stop-condition steps emit records (not silent termination)
- [ ] No C5 record contains an inline heavy payload (no raw document text, no full tool-call argument blobs)

Section 6 — Handoff Rules

Handoff rules define what is transferred, accepted, and resumed at every ownership transition in the workflow.

```
handoff_rules:
  - handoff_id: <kebab-case>
    from_owner: <role_id>
    to_owner: <role_id>
    trigger: <condition expression>
    payload_fields: [<field list; must match hitl_handoff payload_to_human in e4-agent-role-
specification>]
    acceptance_condition: <what the receiving owner must do>
    resume_step: <step_id>
    timeout_action: <escalate | stop | repeat-notification>
    provenance_emission: required   # C5 record with decision_origin=escalation or human-
override
```

Handoffs are bidirectional: agent-to-human, human-to-agent, and agent-to-agent are all handoffs. Every handoff must have a named payload and a declared acceptance condition. A handoff with payload_fields: [] is a spec defect.

Worked Example: South Walton Stormwater Permit Triage — Assembled Artifacts

The following illustrates assembled workflow artifacts for the permit triage workflow. All figures are illustrative.

Process map excerpt:

```
process_map:
  workflow_id: swc-permit-triage
  workflow_name: South Walton County Stormwater Permit Triage
  version: 0.1.0
  canvas_ref: stormwater-permit-triage
  last_updated: 2026-05-29
```

```
steps:
  - step_id: intake-receipt
    name: Application Receipt and Logging
    type: action
    owner: intake-processor
    inputs:
      - source: district-portal-submissions
        field: permit-application-v3.json
    outputs:
      - destination: completeness-check
        artifact: intake-log-entry
    branches: []
    hitl_gate: false
    hitl_handoff_ref: null
    exception_handler_refs: [portal-timeout-on-schema-fetch]
    provenance_emission: required

  - step_id: wetlands-proximity-check
    name: Wetlands Proximity Determination
    type: decision
    owner: wetlands-analyst
    inputs:
      - source: parcel-gis-layer
        field: parcel-feature-v1
      - source: fwc-wetlands-overlay
        field: wetlands-boundary-v2
    outputs:
      - destination: routing-decision
        artifact: wetlands-proximity-flag
    branches:
      - edge_id: clear
        label: Parcel outside regulated proximity
        branch_to: routing-decision
      - edge_id: within-proximity
        label: Parcel within regulated proximity
        branch_to: engineer-review-escalation
      - edge_id: stale-or-low-confidence
        label: Overlay data stale or low confidence
        branch_to: wetlands-data-quality-escalation
    hitl_gate: true
    hitl_handoff_ref: wetlands-data-quality-escalation
    exception_handler_refs: [gis-connector-stale]
    provenance_emission: required

  - step_id: notification-send-gate
    name: Applicant Notification HITL Gate
    type: gate
    owner: human
    inputs:
      - source: return-to-applicant-draft-store
        field: notification-draft-v1
    outputs:
      - destination: district-email-system
        artifact: applicant-notification
    branches:
      - edge_id: clerk-approved
        label: Clerk approved notification
        branch_to: send-notification-terminal
      - edge_id: clerk-rejected
        label: Clerk rejected or modified draft
        branch_to: notification-draft-revision
    hitl_gate: true
    hitl_handoff_ref: clerk-notification-approval
    exception_handler_refs: [unsafe-action-intercepted]
    provenance_emission: required
```

C5 emission record for wetlands-proximity-check (stale-data escalation path):

```
c5_emission:
  step_id: wetlands-proximity-check
  correlation_id: "01926b3f-4a12-7e00-8c4d-2f9a1b0c3e5d"
  record:
    step_id: wetlands-proximity-check
    step_inputs_ref:
      ref: "s2-audit://swc-permit-triage/run-20260529-0834/step-wetlands"
      hash: "sha256:a3f8c2e..."
      field: inputs
    step_outputs_ref:
      ref: "s2-audit://swc-permit-triage/run-20260529-0834/step-wetlands"
      hash: "sha256:9b1d4f7..."
      field: outputs
    branch_taken: stale-or-low-confidence
    decision_origin: escalation
```

The correlation_id above is shared with the C2 HITL event emitted by s1-hitl-review-policy when the district engineer escalation gate fires.

Review gate register excerpt:

```
review_gate_register:
  - gate_id: wetlands-data-quality-gate
    step_ref: wetlands-proximity-check
    gate_type: hitl
    trigger_condition: fwc_overlay.confidence_band in {low, unknown} OR fwc_overlay.
age_hours > 168
    reviewer_role: District Engineer
    sla_minutes: 240
    payload_required:
      - application_id
      - parcel_id
      - fwc_overlay.age_hours
      - fwc_overlay.confidence_band
      - parcel_to_wetlands_distance_ft
    on_approval: routing-decision
    on_rejection: return-to-applicant-draft
    on_sla_breach: escalate-to-commissioner-on-call
    blocks_irreversible_step: false

  - gate_id: clerk-notification-send-gate
    step_ref: notification-send-gate
    gate_type: hitl
    trigger_condition: always
    reviewer_role: district-intake-clerk
    sla_minutes: 480
    payload_required:
      - application_id
      - draft_notification_text
      - routing_rationale
      - cited_rule_or_deficiency
    on_approval: send-notification-terminal
    on_rejection: notification-draft-revision
    on_sla_breach: escalate-to-district-engineer
    blocks_irreversible_step: true
```

Usage Notes

Instantiate this template once per workflow model. Version it alongside the workflow; do not edit the signed-off artifact in place after first production deployment — create a new version. When the process map changes (new step

discovered at runtime, policy change, divergence correction), the full Course 4 build chain must be re-run for the affected steps: discovery delta, decomposition update, decision logic update, role spec update, and artifact re-assembly. A partial update that changes the process map without updating the role matrix or C5 emission spec is a provenance integrity gap.