Mother■Harness Role ACI Template & Registry Addendum (v1)

This addendum defines: (1) a reusable Role ACI template, (2) a machine■readable Role Registry, and (3) enforcement rules Mother uses to validate role compliance at runtime. Designed to be kept alongside the Role ACIs Addendum (v1) and the v3 Orchestration Addendum.

Why a Registry

- Makes adding new roles mechanical: define a contract once, register it, and Mother can enforce it automatically.
- Prevents “agent drift”: every role has bounded actions, required artifacts, and explicit phase exits.
- Enables capability■based routing: Mother chooses models/tools using declared needs + policy constraints.
- Supports replay and evaluation: each role declares what must be captured for reproducibility.

Role ACI Template

Use this template for every new role. The template is both human■readable and convertible to JSON/YAML.

Canonical JSON template

```
{ "role_id": "string (stable id, e.g., 'coder', 'research', 'update')", "role_name":
"string (display name)", "version": "string (e.g., 'aci:v1')", "purpose": "string",
"primary_phases": [ "plan", "execute", "review", "repair", "approve", "finalize",
"terminate" ], "allowed_actions": [ { "action_id": "string (tool or operation name)",
"category": "deterministic_tool | llm_call | workflow_dispatch | read_only | write",
"description": "string", "constraints": { "allowlist": [ "optional string list" ],
"rate_limit": "optional string", "max_bytes": "optional int", "requires_approval":
"optional bool" } } ], "required_artifacts": [ { "artifact_type": "string (e.g.,
'CodePatch', 'RetrievalReport')", "required_in_phases": [ "execute", "review" ],
"schema_ref": "string (e.g., '#/schemas/CodePatch')", "min_requirements": [ "string list" ]
} ], "phase_exit_criteria": [ { "phase": "string", "exit_when": [ "boolean expressions or
gate ids" ], "max_retries": 2, "on_exceed": "terminate | escalate | approval_required |
blocked" } ], "approval_triggers": [ { "trigger_id": "string", "condition": "policy
expression", "required_approver": "owner | admin | security", "expires_in": "PT2H" } ],
"replay_capture": { "capture_prompts": true, "capture_tool_calls": true,
"capture_retrieval": true, "capture_artifacts": true, "redaction_ruleset": "string id" },
"capability_needs": { "strict_json": true, "tool_calling": "low|medium|high", "vision":
false, "long_context": false, "code_quality": "low|medium|high" } }
```

Role Registry

The Role Registry is a single source of truth Mother uses to validate role behavior and drive routing. Store it in RedisJSON (recommended) and version it in Git.

Example registry document

```
{ "registry_version": "registry:v1", "updated_at": "2025-12-19T21:49:12.076155Z", "roles":
[ { "role_id": "mother", "aci_version": "aci:v1", "enabled": true, "entrypoint":
```

```
"orchestrator.mother" }, { "role_id": "librarian", "aci_version": "aci:v1", "enabled":
true, "entrypoint": "agents.librarian" }, { "role_id": "research", "aci_version": "aci:v1",
"enabled": true, "entrypoint": "agents.research" }, { "role_id": "coder", "aci_version":
"aci:v1", "enabled": true, "entrypoint": "agents.coder" }, { "role_id": "critic",
"aci_version": "aci:v1", "enabled": true, "entrypoint": "agents.critic" }, { "role_id":
"analyst", "aci_version": "aci:v1", "enabled": true, "entrypoint": "agents.analyst" }, {
"role_id": "vision", "aci_version": "aci:v1", "enabled": true, "entrypoint":
"agents.vision" }, { "role_id": "update", "aci_version": "aci:v1", "enabled": true,
"entrypoint": "agents.update" }, { "role_id": "toolsmith", "aci_version": "aci:v1",
"enabled": true, "entrypoint": "agents.toolsmith" } ], "policies": {
"default_max_retries_per_phase": 2, "default_phase_timeout_sec": 900,
"require_retrieval_report_on_rag": true, "require_test_results_on_code_patch": true } }
```


Mother Enforcement Rules

- **Contract check at dispatch:** Mother refuses to dispatch a role_id not present or disabled in the registry.
- **Action allowlist:** Any requested tool/action not declared in allowed_actions is rejected and logged as policy violation.
- **Artifact gate:** Phase transitions are blocked until required artifacts exist and validate to schema.
- **Retry accounting:** Each phase decrements remaining retries; on exceed, Mother applies on_exceed policy (terminate/escalate/approval/blocked).
- **RAG compliance:** Any response that uses retrieved context must attach a RetrievalReport + citations; otherwise it fails review.
- **Coder compliance:** Any code■changing task must include CodePatch + TestResults unless an explicit waiver exists and (if needed) approval recorded.
- **Replay capture:** Mother records the role's replay_capture settings; redaction is applied before persistence.

Runtime Data Model Hooks

Key/Stream	Purpose
registry:roles	Role Registry document (RedisJSON).
run:{run_id}	RunState snapshot; current phase, retries, timers, spawned agents.
artifact:{artifact_id}	Stored artifacts with schema refs and hashes.
activity:stream	Redis Stream for UI activity feed + audit trail.
replay:{run_id}	Minimal replay bundle: prompts, tool calls, retrieval hits, artifact hashes.

Adding a New Role (Procedure)

- 1) Copy Role ACI Template, fill fields, save as *aci:{role_id}:vN* in Git + RedisJSON.
- 2) Register role in registry:roles with entrypoint and enabled=true.
- 3) Implement entrypoint that produces ResultEnvelope and required artifacts.
- 4) Add schema validations for new artifact types (if any).

- 5) Add a small evaluation task (golden test) so regressions are measurable.
- 6) Deploy; Mother will enforce contract automatically at runtime.

Appendix: Minimal Agent Output Contract

```
{ "result_envelope": { "run_id": "uuid", "task_id": "uuid", "role_id": "string", "status":
"success|needs_repair|blocked|failed", "confidence": { "score": 0.0, "rationale": "string"
}, "artifacts": [ { "artifact_id": "uuid", "type": "string", "schema_ref": "string",
"hash": "sha256" } ], "next_actions": [ { "type":
"spawn_role|request_approval|escalate_model|terminate", "params": {} } ], "errors": [ {
"code": "string", "message": "string" } ] } }
```