Mother■Harness Role ACIs Addendum (v1)

Agent■Computer Interface (ACI) contracts for all roles. Defines allowed actions, required artifacts, phase exit criteria, approvals, and replay capture. Designed to plug into the synchronous coordination model and existing ResultEnvelope / RetrievalReport / Policy schemas.

Global ACI Rules (applies to every role)

- **Everything is a transaction.** Agents propose changes; Mother commits state transitions and enforces phase exit criteria.
- **Tool■first.** Deterministic tools must be attempted before LLM reasoning for any action they can handle (git, tests, file ops, DB queries, OCR, etc.).
- **Schema■first.** Every agent output must validate to *ResultEnvelope*. Non■conforming outputs trigger an automatic repair loop (bounded).
- **Evidence■first.** Any factual claim must attach citations when a library source exists. RAG answers must include a *RetrievalReport* artifact.
- **Bounded loops.** Every role has max retries per phase; escalation and termination rules must be respected.
- **Replay capture.** Every run records prompts, model id, tool calls, retrieved chunk ids + scores, and produced artifacts.

Common Phase Map

All roles operate within the Mother phase model:

- **plan** → interpret goal, pick tools/models, define artifacts and success checks.
- **execute** → perform tool calls / edits / retrieval / computations.
- **review** → validate outputs (schema, tests, citations, safety).
- **repair** → apply bounded fixes; may escalate model or broaden retrieval.
- **approve** → request and wait for explicit approval when policy requires.
- **finalize** → produce final artifacts; summarize; store long■term notes; close run.
- **terminate** → stop with status (success/blocked/failed) and clear reasons.

Mother (Orchestrator / Coordinator) — ACI Contract

Field	Definition
Purpose	Owns control■plane decisions: routing, parallel agent spawning, quorum barriers, arbitration, retries,
Primary phases	plan, execute, review, repair, approve, finalize, terminate
Capability needs	Strict state machine + concurrency barriers; may use a small local model only for intent classification/

Allowed actions / tools

- Parse tasks into phases; spawn concurrent agent runs; set timeouts and quorum rules.
- Select models using capability matrix (tool calling, vision, strict JSON, long context) and policy constraints (e.g., cloud allowed per library).
- Run deterministic checks: schema validation, policy evaluation, budget/usage tracking, rate limiting.
- Enqueue/dispatch workflows (n8n) and collect results; manage run state machine in RedisJSON.
- Arbitrate disagreements (e.g., critic vs coder) using evidence + gates; decide next phase.
- Emit activity events (Redis Streams) and update dashboards.

Required artifacts (minimum)

- ResultEnvelope (Mother) with next_actions + phase transitions recorded.
- RunState snapshot (RedisJSON): phase, retries, timers, spawned agents, completion reason codes.
- When RAG is used by any role: RetrievalReport stored and attached to run.

Phase exit criteria

- Advance a phase only when exit gates are satisfied (quorum met, timeout reached, or policy override).
- Hard terminate if retries exceeded OR approval denied/expired OR policy violation occurs.
- No infinite loops: every loop decrements remaining retries and records why it continues.

Approval triggers

- Always approval■gate: destructive ops, production changes, cloud on restricted libraries, git push to protected branches, dependency upgrades above risk threshold.
- May request approval on behalf of other roles when they propose a gated action.

Replay capture notes

- Record routing decisions: chosen model/provider, fallback attempts, and reason codes.
- Record quorum inputs: each agent’s ResultEnvelope ids + key artifacts referenced in arbitration.
- Record all timeouts/retries with causes.

Librarian (Library Ingestion & Indexing) — ACI Contract

Field	Definition
Purpose	Ingests documents (PDF/EPUB/HTML/notes) via Docling, chunks content, stores metadata, compute
Primary phases	plan, execute, review, repair, finalize
Capability needs	Deterministic heavy; LLM optional for summarizing ingest results.

Allowed actions / tools

- File discovery and metadata extraction (hashing, MIME detection).
- Docling extraction job submission and polling; store raw extraction artifacts.
- Chunking per library policy (doc hierarchy or fixed tokens) with page/section metadata.
- Embedding generation for each chunk into embed_space=emb:v1; store embedding_model_id for audit.
- Write to Redis (RedisJSON + Vector index) and update library catalog.
- OCR fallback for image■only pages when policy allows.

Required artifacts (minimum)

- IngestionReport artifact: doc_id, chunk_count, embedding_count, failures, timings.
- Chunk records with embed_space, embedding_model_id, embedding_dim, created_at.
- IndexBuildReport artifact when index created/updated.

Phase exit criteria

- Exit when: all chunks persisted OR failures recorded with retry plan; indexes updated OR flagged for rebuild.
- If extraction fails: retry with backoff or mark document failed with reason + next_action.

Approval triggers

- Approval only if ingesting from restricted sources or if policy requires PII redaction review before indexing.

Replay capture notes

- Capture doc hashes, docling version, policy snapshot, chunk params, embedding model+dims, per■chunk failures.

Research Agent — ACI Contract

Field	Definition
Purpose	Collects and synthesizes evidence from libraries and allowed web sources; produces grounded summ
Primary phases	plan, execute, review, repair, finalize
Capability needs	Strong citation discipline + structured output; can start local, escalate for long-context synthesis.

Allowed actions / tools

- RAG retrieval over libraries (vector/hybrid per policy) and creation of RetrievalReport.
- Optional web search via allowed tools; fetched sources are stored in the library for future citation.
- Extract claims and map each claim → citations.
- Produce follow■up scope adjustments when evidence is weak.

Required artifacts (minimum)

- ResultEnvelope with findings[] + citations[] + confidence.
- RetrievalReport artifact for any library retrieval.
- ResearchNotes artifact: claims[] (each claim w/ citations), open_questions[], recommended_next_steps[].

Phase exit criteria

- Exit when: retrieval gates pass OR return needs_input/blocked with specific missing_info fields.
- If citations are missing for factual claims, must repair or mark blocked.

Approval triggers

- Approval if policy disallows cloud on target library and cloud escalation would be required.
- Approval for restricted scraping or storing credentials.

Replay capture notes

- Record query fingerprints, filters, topK hits, and any web fetch URLs stored as library docs.

Coder Agent (Repo■Bound Engineering) — ACI Contract

Field	Definition
Purpose	Implements code changes using tool■driven actions only. Produces diffs and test evidence; never cl
Primary phases	plan, execute, review, repair, finalize
Capability needs	Tool calling reliability; small local models often sufficient due to grounding via tools.

Allowed actions / tools

- Repo inspection via deterministic tools (tree, ripgrep, file reads).
- Edits by producing patches/diffs (preferred) or logged file writes.
- Run lint/typecheck/unit tests; collect outputs.
- Dependency audits and license checks when applicable.
- Prepare commits (no push unless approved).

Required artifacts (minimum)

- CodePatch artifact (unified diff) OR FileRef artifacts with sha256 for modified files.
- TestResults artifact: commands run, exit codes, summarized failures.
- Lint/Build artifact if applicable.
- ImplementationNotes artifact: what changed, why, risk notes.

Phase exit criteria

- Exit execute only if: patch exists AND tests run OR explicit waiver recorded by Mother policy.
- Exit review only if: Critic passes OR bounded repair exhausted (then blocked/failed).
- Failing tests must be fixed or waiver+approval recorded.

Approval triggers

- Approval for: git push, protected branch merges, risky dependency upgrades, non■allowlisted shell commands, destructive file ops.

Replay capture notes

- Record commands executed, environment, tool outputs, diffs, and test logs (core replay payload).

Critic Agent (QA / Safety / Groundedness) — ACI Contract

Field	Definition
Purpose	Validates artifacts from other agents using deterministic checks + policy rules; returns pass/fail with r
Primary phases	review, repair, finalize
Capability needs	Strong structured reasoning; may escalate for complex security/code review.

Allowed actions / tools

- Schema validation of ResultEnvelope + required artifacts.
- Groundedness checks: citations ↔ retrieved chunks; RetrievalReport gate review.
- Code review: scan diffs, secrets leakage, risky patterns; verify tests/lint ran.
- Policy checks: approvals needed, cloud allowed, sensitivity compliance.

Required artifacts (minimum)

- CriticReport artifact: pass/fail, reasons[], required_fixes[], risk level, confidence.
- Optional SecurityFindings artifact with evidence.

Phase exit criteria

- Exit when: pass OR fail with explicit required_fixes[] and next_actions.
- No vague rewrite requests: each fix must be actionable and testable.

Approval triggers

- May request approval escalation for high-risk changes even if policy didn't flag (defense in depth).

Replay capture notes

- Record which checks ran and what evidence was used (diff hashes, test logs, citations).

Analyst Agent (Data / Math / ETL) — ACI Contract

Field	Definition
Purpose	Performs data analysis and pipeline validation using deterministic compute (Python/SQL). Produces
Primary phases	plan, execute, review, repair, finalize
Capability needs	Deterministic compute first; LLM used for synthesis and explanation.

Allowed actions / tools

- Run SQL queries (read-only unless approved).
- Execute Python/R scripts; generate tables/figures.
- Validate ETL logic; compute metrics; propose pipeline fixes (via Coder tasks).
- Use RAG for reference docs when needed (attach RetrievalReport).

Required artifacts (minimum)

- AnalysisReport artifact (methods + findings).
- Code artifact (script/notebook) and DataArtifacts (CSV/JSON) as applicable.
- If RAG used: RetrievalReport + citations.

Phase exit criteria

- Exit when: outputs are reproducible (script + inputs referenced) and key numbers trace to code/queries.
- DB writes require approval; otherwise analysis remains read-only.

Approval triggers

- Approval for: writing production DB, heavy compute beyond caps, exporting sensitive datasets.

Replay capture notes

- Capture code, parameters, random seeds, SQL queries, and file hashes for inputs/outputs.

Vision Agent (Multimodal Analysis) — ACI Contract

Field	Definition
Purpose	Analyzes images/diagrams/screenshots; uses local or cloud multimodal models; OCR fallback; outpu
Primary phases	plan, execute, review, repair, finalize
Capability needs	Multimodal + strict JSON; cloud often better for complex diagrams.

Allowed actions / tools

- Image preprocessing (resize/crop/normalize) via deterministic tools.
- Multimodal inference via Ollama local (e.g., llava) or cloud (e.g., qwen2■vl) if policy allows.
- OCR fallback for text-only extraction.
- Diagram parsing into nodes/edges/labels suitable for Mermaid/graph ingestion.

Required artifacts (minimum)

- VisionAnalysis artifact (structured JSON) + summary.
- If diagram: DiagramGraph artifact (nodes/edges).
- If OCR: OCRText artifact with confidence.

Phase exit criteria

- Exit when: schema-valid structured output AND confidence ≥ threshold OR return blocked with recommended next step (better image, higher tier model, OCR).

Approval triggers

- Approval if cloud multimodal is needed for restricted/private images or libraries.

Replay capture notes

- Capture image hashes, preprocessing steps, model id/provider, and schema validation outcome.

Update Agent (Software Inventory & Upgrade Intelligence) — ACI Contract

Field	Definition
Purpose	Monitors deployed software, checks upstream versions, ingests release notes into the library, analyz
Primary phases	plan, execute, review, repair, approve, finalize
Capability needs	Structured synthesis + deterministic version parsing; RAG over ingested release notes.

Allowed actions / tools

- Inventory discovery (containers/services/packages) via deterministic inspection tools.
- Upstream checks (GitHub releases, Docker tags, websites) via allowed fetch tools.
- Evidence ingestion via Librarian pipeline; embed release notes into emb:v1.
- Impact analysis: breaking changes, migration steps, rollback plan, risk score.
- Generate approval requests for upgrades above threshold; optionally schedule maintenance window tasks.

Required artifacts (minimum)

- UpdateReport artifact (structured JSON) with evidence citations and recommendations.
- Ingestion references (doc_ids) for release notes/changelogs; RetrievalReport when used.
- ApprovalRequest artifact when upgrade recommended and policy requires.

Phase exit criteria

- Exit when: all inventory checked OR explicitly skipped with reason; evidence stored; recommendation issued.
- If sources unavailable: mark investigate with missing_info and a retry schedule.

Approval triggers

- Approval to apply upgrades, restart services, or alter production config.
- Approval to store upstream docs when licensing/policy requires.

Replay capture notes

- Capture inventory snapshot, upstream query parameters, evidence hashes, and decision rationale.

Toolsmith Agent (Deterministic Tools & Glue Code) — ACI Contract

Field	Definition
Purpose	Builds and maintains deterministic tool wrappers used by other agents; replaces expensive LLM reas
Primary phases	plan, execute, review, finalize
Capability needs	Mostly deterministic coding; small local model ok for scaffolding.

Allowed actions / tools

- Implement tool wrappers (git/file ops/OCR/SQL/lint/test runners).
- Write tests for tool wrappers; ensure idempotence and safe defaults.
- Update allowlists and bind tools to policy/approval rules.

Required artifacts (minimum)

- ToolPatch artifact (diff) + tool tests output.
- ToolCatalog artifact listing signatures + schemas + permission tier.

Phase exit criteria

- Exit when: tool passes tests and is registered; approvals updated if permissions change.

Approval triggers

- Approval for expanding tool permissions or adding dangerous actions.

Replay capture notes

- Capture tool versioning, allowlist changes, and regression tests.