RORI — Macro Development Plan v2.0

Research On Regulatory for Industry(s)

Version: 2.0 **Date:** 2026-02-23 **Authors:** The Frank-cicle & Candi **Status:** Draft — Pending Review **Supersedes:** v1.1 (2026-02-13)

What Changed from v1.1

v1.1 organized work into a flat phase model (Phase 0–3) with a parallel Collection swim lane. It named components and deliverables but lacked specifications for how those components actually work — particularly around ingestion adapters, content curation, and feedback loops. The Domain Discovery step was buried as a sub-item (`phase0/first-vertical-corpus-acquisition`) rather than being recognized as the foundational first activity that gates everything downstream.

v2.0 restructures the plan around a key insight: **before you can scrape, ingest, chunk, index, or retrieve anything, you need to systematically discover and map what exists in a regulatory domain.** Domain Discovery is now Phase 1 — the true starting point. Every subsequent phase has its own sub-plan and specification document, creating a chain of specs rather than a single monolithic plan with bullet-point deliverables.

Structural Changes

v1.1	v2.0	Rationale
Phase 0: Foundation & Research (flat)	Phase 0: Project Foundation (narrowed)	Separated research from scaffolding — research is now embedded in each phase as needed
`phase0/first-vertical-corpus-acquisition`	**Phase 1: Domain Discovery & Analysis** (promoted to full phase)	This is the true first step, not a sub-task of research
Collection swim lane (parallel)	Absorbed into Phase 2: Data Acquisition	Collection isn't parallel — it depends on Phase 1's manifest output
`phase1/ingestion-pipeline` + `phase1/curation-enrichment` (separate branches)	**Phase 3: Ingestion & Curation Engine** (unified phase with adapter architecture)	Ingestion and curation are inseparable — curation happens during ingestion, not after

v1.1	v2.0	Rationale
No feedback loop defined	**Phase 6: Feedback & Continuous Curation** (new phase)	Closed-loop accuracy improvement was entirely missing
Phases described with bullet-point deliverables	Each phase produces its own sub-plan and specification document	Specs before code — every phase is fully designed before implementation begins

---

Plan Structure

Each phase below is a summary. The actual work for each phase is governed by its own **Sub-Plan & Specification** document (`RORI-Phase-{N}-SPEC-v{X}.md`), which contains the full architecture, schemas, interface contracts, deliverables, acceptance criteria, and risk register. The macro plan defines the *what and why*; the sub-plan specs define the *how*.

Branch Convention

```
phase{N}/{component-name}
```

Examples: `phase1/domain-discovery-agent`, `phase2/web-scraping-infra`, `phase3/ingestion-adapters`

Phase 0: Project Foundation

Sub-Plan: `RORI-Phase-0-SPEC` | **Branch prefix:** `phase0/`

The scaffolding and governance layer. No application logic — just the skeleton that everything else builds on.

Components

`phase0/project-scaffold` — Monorepo structure, CI/CD pipeline (GitHub Actions), linting, formatting, branch protection rules, contributing guide. Folder convention, environment setup, `docker compose up` dev experience.

`phase0/evaluation-framework` — Define how RORI measures accuracy, completeness, consistency, and repeatability. Test harness scaffold, benchmark dataset definitions, metrics. This ships early because every subsequent phase validates against it.

Phase 0 Exit Criteria

- A contributor can clone the repo, run one command, and have a working dev environment
- The evaluation framework has defined metrics and a test harness skeleton
- Phase 1 sub-plan spec is written and approved

Phase 1: Domain Discovery & Analysis ⭐ FIRST BUILD

Sub-Plan: `RORI-Phase-1-SPEC` | **Branch prefix:** `phase1/`

This is where RORI starts. Before anything can be scraped, ingested, or indexed, the system must learn about the target regulatory domain — what regulatory bodies exist, what statutes and guides apply, how they're organized, where the source documents live, and how they relate to each other.

Domain Discovery is an **AI-agent-driven research process** that takes a domain description as input and produces a structured YAML manifest as output. That manifest is the contract between Phase 1 and Phase 2 — it tells the data acquisition layer exactly what to go get.

What the Domain Discovery Agent Does

Given a target domain (e.g., *"mortgage regulations impacting first-time homebuyers in the United States"*), the agent:

1. **Maps the regulatory landscape** — Identifies the federal agencies, state regulators, GSEs, and industry bodies that govern the domain. Builds a hierarchy: federal → state → local, primary legislation → implementing regulations → guidance → standards.
2. **Discovers source documents** — For each regulatory body, identifies the specific statutes, rules, guides, directives, and educational materials that apply. Captures URLs, document types, publication dates, update frequencies, and access methods.
3. **Classifies source characteristics** — Tags each source by type (statute, regulation, guidance, standard, educational), format (PDF, HTML, legal XML, API), jurisdiction (federal, state, municipal), and authority level (binding, advisory, informational).
4. **Maps relationships and dependencies** — Identifies supersession chains (what replaced what), cross-references between documents, and applicability hierarchies (which rules apply to which entities under which circumstances).
5. **Assesses coverage and gaps** — Evaluates whether the discovered sources provide complete coverage of the domain or if there are known regulatory areas with missing or inaccessible sources.
6. **Produces a structured YAML manifest** — The output artifact. Every discovered source becomes an entry in the manifest with all metadata needed for the data acquisition phase to fetch, validate, and stage it.

YAML Manifest Schema (Draft)

```
yaml
```

```
manifest:
  id: "rori-manifest-mortgage-fthb-001"
  domain: "Mortgage Regulations — First-Time Homebuyers"
  created: "2026-02-23T00:00:00Z"
  created_by: "domain-discovery-agent-v1"
  version: 1
  status: "pending_review"  # pending_review | approved | active | archived

  domain_map:
    regulatory_bodies:
      - id: "cfpb"
        name: "Consumer Financial Protection Bureau"
        jurisdiction: "federal"
        authority_type: "regulator"
        url: "https://www.consumerfinance.gov"
        governs:
          - "TILA/Regulation Z"
          - "RESPA/Regulation X"
          - "ECOA/Regulation B"
          - "HMDA/Regulation C"

      - id: "fannie-mae"
        name: "Fannie Mae"
        jurisdiction: "federal"
        authority_type: "gse"
        url: "https://www.fanniemae.com"
        governs:
          - "Selling Guide"
          - "Servicing Guide"

    jurisdiction_hierarchy:
      - level: "federal"
        sources_count: 0  # populated by agent
        children:
          - level: "state"
            sources_count: 0
            children:
              - level: "municipal"
                sources_count: 0

  sources:
    - id: "src-001"
      name: "TILA / Regulation Z — Truth in Lending"
```

```yaml
    regulatory_body: "cfpb"
    type: "regulation"          # statute | regulation | guidance | standard | educational | guide
    format: "html"              # html | pdf | legal_xml | api | structured_data
    authority: "binding"        # binding | advisory | informational
    jurisdiction: "federal"
    url: "https://www.consumerfinance.gov/rules-policy/regulations/1026/"
    access_method: "scrape"     # scrape | download | api | manual
    update_frequency: "as_amended" # annual | quarterly | as_amended | static | unknown
    last_known_update: "2025-11-15"
    estimated_size: "large"     # small (<50 pages) | medium (50-500) | large (500+)
    scraping_notes: "Multi-page HTML with section navigation. JS rendering required."
    relationships:
      supersedes: []
      superseded_by: []
      cross_references: ["src-002", "src-005"]
      implements: "15 USC 1601-1667f"
    classification_tags:
      - "lending-disclosure"
      - "consumer-protection"
      - "mortgage-origination"
    confidence: 0.95            # agent's confidence in accuracy of this entry
    needs_human_review: false
    review_notes: ""

  - id: "src-002"
    name: "RESPA / Regulation X — Real Estate Settlement Procedures"
    regulatory_body: "cfpb"
    type: "regulation"
    format: "html"
    authority: "binding"
    jurisdiction: "federal"
    url: "https://www.consumerfinance.gov/rules-policy/regulations/1024/"
    access_method: "scrape"
    update_frequency: "as_amended"
    last_known_update: "2025-08-20"
    estimated_size: "large"
    scraping_notes: "Similar structure to Reg Z."
    relationships:
      supersedes: []
      superseded_by: []
      cross_references: ["src-001"]
      implements: "12 USC 2601-2617"
    classification_tags:
      - "settlement-procedures"
      - "closing-disclosure"
      - "mortgage-servicing"
    confidence: 0.95
    needs_human_review: false
    review_notes: ""

coverage_assessment:
  total_sources: 0                 # populated by agent
  by_jurisdiction:
    federal: 0
    state: 0
    municipal: 0
  by_type:
    statute: 0
    regulation: 0
    guidance: 0
    standard: 0
    educational: 0
    guide: 0
  known_gaps:
    - description: ""
      severity: "high"           # high | medium | low
      mitigation: ""
  completeness_score: 0.0          # 0.0-1.0, agent's self-assessed coverage

review_history:
  - date: ""
    reviewer: ""
    action: ""                     # approved | revised | rejected
    notes: ""
```

Agent Architecture (High-Level)

The Domain Discovery Agent is the first RORI agent to be built. It establishes the agent patterns that subsequent phases will reuse.

```
┌─────────────────────────────────────────────────────────────┐
│          Domain Discovery Agent                  │
│                                    │
│ Input: Domain description (natural language)          │
│ Output: YAML manifest + coverage report              │
│                                    │
│ ┌──────────────┐ ┌──────────────┐ ┌──────────────┐ │
│ │ Landscape   │──▶│  Source    │──▶│ Relationship │ │
│ │ Mapper     │ │  Hunter    │ │  Mapper    │ │
│ │       │ │       │ │       │ │
│ │ Identifies   │ │ Finds URLs, │ │ Maps cross-  │ │
│ │ regulatory   │ │ doc types,  │ │ references,  │ │
│ │ bodies &    │ │ formats,   │ │ supersession │ │
│ │ hierarchy   │ │ access    │ │ chains,    │ │
│ └──────────────┘ │ methods    │ │ applicability│ │
│     │     └──────────────┘ └──────────────┘ │
│     │       │       │     │
│     ▼       ▼       ▼     │
│ ┌─────────────────────────────────────────────────────┐ │
│ │     Coverage Assessor                │ │
│ │ Evaluates completeness, flags gaps,           │ │
│ │ assigns confidence scores, marks items          │ │
│ │ needing human review                  │ │
│ └─────────────────────────────────────────────────────┘ │
│           │                   │
│           ▼                   │
│ ┌─────────────────────────────────────────────────────┐ │
│ │     Manifest Generator               │ │
│ │ Produces validated YAML manifest             │ │
│ │ conforming to schema + coverage report          │ │
│ └─────────────────────────────────────────────────────┘ │
│           │                   │
│           ▼                   │
│      YAML Manifest + Coverage Report           │
│      (input to Phase 2)                 │
└─────────────────────────────────────────────────────────────┘
```

Tools Available to the Agent

The Domain Discovery Agent uses a combination of web search, web fetch, and LLM reasoning. It does NOT scrape — it discovers. The actual content acquisition happens in Phase 2.

- **Web Search** — Find regulatory bodies, source documents, legal databases

- **Web Fetch** — Read agency homepages, regulation indexes, table of contents pages
- **LLM Reasoning** — Classify sources, map relationships, assess coverage
- **Schema Validator** — Validate the produced manifest against the YAML schema before output

Human-in-the-Loop

The agent produces a manifest with `status: "pending_review"`. Each source entry has a `confidence` score and a `needs_human_review` flag. Before the manifest moves to Phase 2, a human reviewer:

- Reviews flagged entries (low confidence, uncertain classification)
- Adds sources the agent missed
- Removes false positives
- Changes status to `approved`

The approved manifest is the input contract for Phase 2.

Phase 1 Components

`phase1/domain-discovery-agent` — The core agent: landscape mapping, source hunting, relationship mapping, coverage assessment.

`phase1/manifest-schema` — The YAML manifest schema definition, validator, and tooling. Includes a CLI for validating manifests and a diff tool for comparing manifest versions.

`phase1/manifest-review-ui` — Slim React interface for human review of agent-produced manifests. Shows sources on a dashboard, allows approve/reject/edit per entry, tracks review history. Does not need to be fancy — functional and clear.

`phase1/fthb-domain-run` — The first real execution: run the Domain Discovery Agent against the Mortgage/First-Time Homebuyer domain. Produces the FTHB manifest that feeds into Phase 2. This is validation of the agent, the schema, and the review workflow all at once.

Phase 1 Exit Criteria

- Domain Discovery Agent can accept a natural language domain description and produce a valid YAML manifest
- Manifest schema is defined, documented, and has a working validator
- Review UI allows human approval workflow
- FTHB manifest is produced, reviewed, and approved
- Phase 2 sub-plan spec is written and approved

Phase 2: Data Acquisition

Sub-Plan: `RORI-Phase-2-SPEC` | **Branch prefix:** `phase2/`

Phase 2 consumes the approved YAML manifest from Phase 1 and acquires the actual content. This is where the web scraping infrastructure (already spec'd in v1.0) lives, alongside file system ingestion and other acquisition methods.

The manifest's `access_method` field drives routing: `scrape` goes to the scraping engine, `download` goes to direct fetch, `api` goes to API adapters, `manual` flags for human acquisition.

Components

`phase2/acquisition-orchestrator` — Reads the approved manifest, routes each source to the appropriate acquisition adapter based on `access_method`, manages scheduling, retry logic, and progress tracking.

`phase2/web-scraping-engine` — The Firecrawl/Crawlee hybrid infrastructure (per v1.0 spec). Receives scrape jobs from the orchestrator, produces raw content in the standardized output envelope.

`phase2/direct-download-adapter` — Simple HTTP fetch for PDFs, documents, and files available via direct URL. Handles content-type detection, file validation, and staging.

`phase2/api-adapter` — For sources available via API (e.g., eCFR API, future Swagger/OpenAPI sources). Handles authentication, pagination, rate limiting, and response normalization.

`phase2/acquisition-monitor` — React dashboard for monitoring acquisition progress per manifest. Shows status per source (pending, in-progress, complete, failed, retrying), error logs, and summary statistics.

`phase2/raw-staging-layer` — Defines the staging format for acquired raw content. Each acquired document lands in a standardized envelope with full provenance metadata (source manifest entry, acquisition timestamp, method used, raw content hash, format).

Phase 2 Exit Criteria

- All acquisition adapters functional and tested against real sources
- FTHB manifest sources successfully acquired and staged
- Monitoring dashboard operational
- Raw staging layer populated with validated content
- Phase 3 sub-plan spec is written and approved

Phase 3: Ingestion & Curation Engine

Sub-Plan: `RORI-Phase-3-SPEC` | **Branch prefix:** `phase3/`

Phase 3 transforms raw acquired content into structured, enriched, queryable knowledge. This is where the gaps identified earlier (adapter contracts, curation workflow, quality gates) get fully specified.

Ingestion and curation are **unified** — curation happens during ingestion, not as a separate post-processing step. Every document that enters the repository passes through a defined pipeline of extraction, enrichment, validation, and approval.

Components

`phase3/ingestion-adapters` — Format-specific adapters that normalize raw content into a common internal document model. Each adapter handles one format family:

- `pdf-adapter` — PDF text extraction with structure preservation (headings, sections, tables, lists)
- `html-adapter` — HTML/web content normalization, boilerplate removal, structure extraction
- `legal-xml-adapter` — Legal XML parsing (USLM, Akoma Ntoso) with native structure mapping
- `guide-adapter` — Structured guides (GSE seller/servicer guides) with section hierarchy
- `plaintext-adapter` — Fallback for unstructured text

Each adapter implements a common interface contract: receives a raw staged document, produces a normalized `InternalDocument` with extracted text, preserved structure, and format-specific metadata.

`phase3/curation-pipeline` — The enrichment and validation workflow applied to every `InternalDocument`:

1. **Metadata extraction** — Jurisdiction tagging, effective dates, applicability scope, regulatory body attribution
2. **Relationship linking** — Cross-reference resolution, supersession chain validation (against manifest relationships)
3. **Deduplication** — Detect and resolve overlapping content across sources
4. **Quality gates** — Automated checks: completeness (no empty sections), consistency (metadata matches manifest entry), structural integrity (sections properly nested)
5. **Curation status** — Each document gets a curation status: `raw` → `enriched` → `validated` → `approved` → `indexed`

`phase3/semantic-chunking` — Section-aware, hierarchy-preserving, cross-reference-maintaining chunking optimized for regulatory text. Implements the strategy selected during research. Chunks maintain parent document lineage and section context.

`phase3/indexing-layer` — Hybrid index: dense vector embeddings + sparse/lexical index + structured metadata index. Supports graph-based regulatory relationships if the repository architecture research called for it.

Phase 3 Exit Criteria

- All ingestion adapters implemented and tested against real FTHB corpus documents

- Curation pipeline enriches and validates documents end-to-end
- Quality gates catch known failure modes
- Semantic chunking preserves regulatory text integrity
- Index supports hybrid retrieval queries
- FTHB corpus fully ingested, curated, and indexed
- Phase 4 sub-plan spec is written and approved

Phase 4: Retrieval & Agent Layer

Sub-Plan: `RORI-Phase-4-SPEC` | **Branch prefix:** `phase4/`

The core intelligence layer. Agent-based retrieval with tunable depth and completeness, citation threading, and cross-corpus analysis.

Components

`phase4/retrieval-engine` — Hybrid search (dense + sparse + metadata), re-ranking, confidence scoring, coverage estimation. Deterministic retrieval modes for audit repeatability.

`phase4/agent-core` — The orchestrating agent that plans retrieval strategy, executes queries, synthesizes results, and threads citations. Tunable response depth: quick applicability check vs. exhaustive regulatory audit.

`phase4/citation-provenance` — Every claim in an agent response traces back to a specific chunk, which traces to a specific document, which traces to a specific manifest source. The full chain is auditable.

`phase4/cross-corpus-analysis` — Gap analysis between two corpuses: compare a new directive against NIST standards, compare company policy against regulatory requirements, etc. Produces structured findings with specific citation pairs.

`phase4/developer-api` — REST/GraphQL API for programmatic access. Rate limiting, authentication, webhook notifications. The "context manifold" interface for downstream AI systems.

Phase 4 Exit Criteria

- Retrieval engine returns accurate results against FTHB corpus with measurable precision/recall
- Agent produces tunable-depth responses with full citation chains
- Cross-corpus analysis functional for document comparison use case
- Developer API serves retrieval and analysis endpoints
- Evaluation framework scores meet defined accuracy thresholds
- Phase 5 sub-plan spec is written and approved

Phase 5: Vertical Expansion & Packaging

Sub-Plan: `RORI-Phase-5-SPEC` | **Branch prefix:** `phase5/`

Onboard additional verticals using the proven pipeline, and package the agent with preconfigured front-ends for specific use cases.

Components

`phase5/insurance-vertical` — Run Domain Discovery for insurance regulation domain, acquire, ingest, and validate the corpus. Vertical-specific configuration: state-by-state jurisdiction mapping, coverage taxonomy, broker/agent applicability rules.

`phase5/medical-gig-vertical` — Same pipeline for medical regulations impacting gig platforms for clinicians.

`phase5/packaged-applications` — Preconfigured front-ends for specific use cases: FTHB regulatory navigator, insurance compliance checker, document comparison tool. These are thin UI layers over the Phase 4 API.

`phase5/vertical-onboarding-playbook` — Documented, repeatable process for onboarding new verticals. Everything from Domain Discovery through indexed corpus, templatized.

Phase 5 Exit Criteria

- At least two additional verticals onboarded end-to-end
- Packaged applications functional for defined use cases
- Vertical onboarding playbook documented and validated

Phase 6: Feedback & Continuous Curation

Sub-Plan: `RORI-Phase-6-SPEC` | **Branch prefix:** `phase6/`

The closed-loop system that was missing from v1.1. This phase ensures that RORI's knowledge stays accurate and improves over time.

Components

`phase6/response-feedback-capture` — Mechanism for users and developers to flag bad, outdated, or incomplete responses. Captures the response, the citation chain, and the feedback signal.

`phase6/feedback-to-source-tracer` — When a response is flagged, traces the citation chain back to the specific chunk → document → manifest source. Flags the source for re-evaluation.

`phase6/re-curation-queue` — Flagged sources enter a re-curation queue: re-scrape, re-ingest, re-validate, re-index. Can trigger re-running Domain Discovery for the affected domain area if the issue is a missing source.

`phase6/regulatory-change-monitoring` — Monitors known sources for changes (new amendments, updated guidance, superseded documents). When changes are detected, triggers re-acquisition and re-curation automatically.

`phase6/accuracy-dashboard` — Tracks feedback volume, resolution rate, accuracy trends over time, and curation health metrics per domain and per source.

Phase 6 Exit Criteria

- Feedback from retrieval propagates back to curation with no manual intervention
- Change monitoring detects real regulatory updates
- Re-curation pipeline handles flagged and changed sources end-to-end
- Accuracy trends are measurable and visible

Revised Critical Path

```
Phase 0: Project Foundation
  │
  └──▶ Phase 1: Domain Discovery & Analysis ← FIRST BUILD
      │
      ├──▶ FTHB manifest produced & approved
      │
      └──▶ Phase 2: Data Acquisition
          │
          ├──▶ FTHB corpus acquired & staged
          │
          └──▶ Phase 3: Ingestion & Curation Engine
              │
              ├──▶ FTHB corpus ingested, curated, indexed
              │
              └──▶ Phase 4: Retrieval & Agent Layer
                  │
                  ├──▶ Agent retrieval validated against FTHB
                  │
                  ├──▶ Phase 5: Vertical Expansion
                  │
                  └──▶ Phase 6: Feedback & Continuous Curation
```

Parallel Opportunities

- **Phase 0** components are parallelizable (scaffold + evaluation framework)

- **Phase 1** components can overlap: manifest schema can be built while the agent is being developed
- **Phase 2** acquisition adapters can be built in parallel once the manifest schema is stable
- **Phase 3** ingestion adapters can be built in parallel per format type
- **Phase 5** vertical runs can execute in parallel once the pipeline is proven
- **Phase 6** can begin development alongside Phase 4 — feedback capture doesn't require the full agent to be complete

---

Sub-Plan Specifications

Each phase produces its own specification document before implementation begins:

Phase	Spec Document	Status
Phase 0	`RORI-Phase-0-SPEC-v1.0.md`	Not started
Phase 1	`RORI-Phase-1-SPEC-v1.0.md`	**Next — Domain Discovery**
Phase 2	`RORI-Phase-2-SPEC-v1.0.md`	Partially covered by Web Scraping Spec v1.0
Phase 3	`RORI-Phase-3-SPEC-v1.0.md`	Not started
Phase 4	`RORI-Phase-4-SPEC-v1.0.md`	Not started
Phase 5	`RORI-Phase-5-SPEC-v1.0.md`	Not started
Phase 6	`RORI-Phase-6-SPEC-v1.0.md`	Not started

Existing Artifacts Carried Forward

Artifact	Disposition
Web Scraping Infra Spec v1.0	Rolls into Phase 2 spec (scraping engine component)
FTHB Seed Manifest (from v1.1)	Superseded — Domain Discovery Agent produces the manifest
Phase 0 Research Branches (from v1.1)	Research is now embedded within each phase rather than front-loaded. Retrieval algorithm research happens in Phase 4, chunking research in Phase 3, etc.

---

v2.0 — 2026-02-23 — The Frank-cicle & Candi