

InquisitorNet Functional Modules & Refactor Report

Refactor Summary (What Changed & Why)

This report documents the directory refactor that removed the phase-based layout in favor of functional, capability-driven modules. The goal is to make the repository easier to navigate, align with common Python package conventions, and decouple directory names from implementation milestones.

Why the change

- **Clarity over chronology:** Phase names describe project progress, not what the code does. Functional folders make discovery and ownership clearer.
- **Scalability:** As features grow, functional modules make it easier to add new capabilities without creating new â phaseâ buckets.
- **Standard Python structure:** Grouping by capability aligns with common industry layouts (e.g., ‘ingestion’, ‘policy’, ‘metrics’).

What changed (high level)

- **Removed**: ‘phase1/’, ‘phase2/’, ‘phase3/’ directory structure.
- **Added**: ‘inquisitor/’ package with functional subpackages.
- **Updated**: CLI module paths, imports, and documentation references.
- **Regenerated**: This PDF to provide an auditable report of the refactor.

Old â New module mapping

Old path	New path	Rationale
---	---	---
‘phase1/cli.py’	‘inquisitor/ingestion/cli.py’	CLI belongs to ingestion pipeline.
‘phase1/config.py’	‘inquisitor/ingestion/config.py’	Configuration is ingestion-specific.
‘phase1/db.py’	‘inquisitor/ingestion/db.py’	DB helpers are used by ingestion.
‘phase1/scrapper.py’	‘inquisitor/ingestion/scrapper.py’	Scrapper is the ingestion entry point.
‘phase1/detector.py’	‘inquisitor/ingestion/detector.py’	Detector is part of ingestion pipeline.
‘phase2/gate.py’	‘inquisitor/policy/gate.py’	Gate is a policy capability.
‘phase2/gate_cli.py’	‘inquisitor/policy/gate_cli.py’	CLI aligns with policy capability.
‘phase2/label_cli.py’	‘inquisitor/labeling/label_cli.py’	Labeling is a standalone capability.
‘phase2/metrics_job.py’	‘inquisitor/metrics/metrics_job.py’	Metrics is a reporting capability.
‘phase3/inquisitor_cli.py’	‘inquisitor/operations/inquisitor_cli.py’	Action planning is an operations concern.
‘phase3/bots/base.py’	‘inquisitor/operations/bots/base.py’	Bots are part of operations.

Functional Module Guide

Ingestion ('inquisitor/ingestion/')

Purpose: Ingest data, store in SQLite, and run the rule-based detector.

- 'cli.py'
 - Entry point for the ingestion pipeline.
 - Loads settings from 'config/' via 'inquisitor.ingestion.config.Settings'.
 - Applies migrations ('migrations/001_init.sql', 'migrations/004_offline_fixtures.sql').
 - Runs scraper and detector, then prints a summary.
- 'config.py'
 - Loads YAML configuration ('subreddits.yml', 'scraper_rules.yml', 'detector_rules.yml').
 - Sets the database path (defaults to 'inquisitor_net_phase1.db').
- 'db.py'
 - SQLite connection and migration helpers.
- 'scraper.py'
 - Reads from fixtures, offline DB, or Reddit API.
 - Applies regex include/exclude rules and 'discard_if' filters.
 - Writes kept items to 'scrape_hits'.
- 'detector.py'
 - Compiles regex rules with weights and exculpatory patterns.
 - Inserts into 'detector_marks' or 'detector_acquittals' based on thresholds.

Policy ('inquisitor/policy/')

Purpose: Apply policy checks to outgoing drafts.

- 'gate.py'
 - Loads rules from 'config/policy_gate.yml'.
 - Returns 'allow', 'flag', or 'block' decisions with reasons.
- 'gate_cli.py'
 - Reads JSONL drafts and writes JSONL decisions.

Labeling ('inquisitor/labeling/')

Purpose: Human labeling of detector outcomes.

- 'label_cli.py'
 - Samples 'detector_marks' / 'detector_acquittals'.
 - Prompts for TP/FP/TN/FN labels and writes to 'labels'.

Metrics ('inquisitor/metrics/')

Purpose: Aggregated reporting for model quality.

- 'metrics_job.py'
 - Computes precision/recall/F1 from 'labels'.

- Writes CSV and Markdown reports to ‘reports/metrics/’.

Operations ('inquisitor/operations/')

Purpose: Plan actions based on detector outputs.

- ‘inquisitor_cli.py’

- Reads marks, chooses actions via bots, gates post drafts.

- Writes ‘planned_actions’ and ‘dossiers’ in SQLite.

- ‘bots/base.py’

- Defines ‘BaseBot’ decision logic and ‘InquisitorPersonality’.

Updated CLI Entry Points

- ‘python -m inquisitor.ingestion.cli’

- ‘python -m inquisitor.policy.gate_cli’

- ‘python -m inquisitor.labeling.label_cli’

- ‘python -m inquisitor.metrics.metrics_job’

- ‘python -m inquisitor.operations.inquisitor_cli’

Compatibility Notes

- Database schema remains unchanged; existing migrations still apply.

- Import paths have been updated to match new module locations.

- Phase naming is retained in data/models for historical continuity, but directory names are now functional.