RHUMB WORKFLOW PROTOCOL™ PAPER

Rhumb Workflow Protocol™ Executive Brief

A neutral protocol for durable AI-assisted delivery records

RWP™ 0.31.0 Executives, product leaders, technical leads

Contents

POSITION

Rhumb™ is the protocol layer, not another workflow app

Rhumb Workflow Protocol™ standardizes the durable record around AI-assisted work: what was requested, what was planned, what state execution is in, what evidence was produced, and what context was handed to the next person, agent, or tool.

The protocol is intentionally plain-file first. Markdown stays readable by humans. YAML carries structured state and metadata. JSON Schemas, templates, sequence grammar, and conformance checks make those files inspectable by tools without turning Rhumb into one vendor product.

PROBLEM

AI delivery becomes fragile when the record lives inside chat

A productive AI session can still leave a weak operational trail. Chat transcripts do not reliably define phase boundaries, recovery points, ownership, test evidence, or handoff context. Proprietary project formats create the same problem in another form: the workflow record becomes hard to move, audit, or replay outside the tool that created it.

Rhumb™ addresses that failure mode by making the workflow record portable. The protocol does not require one runtime, one AI provider, one editor, or one product UI.

- Intake artifacts preserve the request, constraints, and success criteria before work begins.
- Plan artifacts define phases, deliverables, dependencies, risks, and verification.
- State artifacts make progress and recovery explicit instead of implicit in a conversation.
- Handoffs preserve operational context when people, agents, tools, or sessions change.
- Manifests register produced artifacts so evidence can be reviewed later.

ADOPTION

Adoption should increase with risk

Rhumb™ does not need to start as a heavy governance system. A team can begin with the small loop: intake, plan, state, and handoff. That gives enough structure for continuity across sessions. Higher-risk workflows add manifests, dependencies, audits, lifecycle metadata, UUIDs, and validator-backed conformance.

This matters commercially because the same protocol can support a lightweight solo workflow, a consulting team handing work between agents, or an enterprise environment that needs audit evidence without changing the underlying artifact model.

- Minimal: plan and state continuity for short work.
- Standard: intake, manifest, handoff, dependencies, and sub-phase support for shared work.
- Full: audits, lifecycle checks, UUIDs, schema validation, and governance for high-rigor environments.

REFERENCE IMPLEMENTATION

YAKKL®Meridian™ proves the protocol but must not own it

Meridian™ is the reference implementation. That is valuable because Rhumb™ is not theoretical; Meridian™ uses the artifact set, lifecycle ideas, templates, and validation paths in a real product direction.

The boundary is equally important: Meridian-specific runtime mechanics, SQLite indexes, private workspace state, desktop UI, and CLI operations are implementation concerns. Rhumb should document a Meridian Reference Profile, not force every adopter to copy Meridian™.

CURRENT-STATE HONESTY

Protocol drift is being resolved in the source, not hidden by the site

RWP™ 0.31.0 moves the protocol-side decisions into source-controlled schemas, templates, examples, and validator fixtures instead of leaving them as website notes. The current contract uses `MP-NNNN-short-name` plan IDs, current-only status vocabulary, and RWP™ -named templates.

Meridian remains the reference implementation, but Rhumb™ should stay vendor-neutral. Meridian-specific runtime mechanics belong in the Meridian Reference Profile or in namespaced extensions.

EXECUTIVE TAKEAWAY

The value is portability, evidence, and optional productization

Rhumb creates a durable contract that can sit underneath multiple interfaces: a technical CLI, a polished desktop app, a non-technical project assistant, a consulting workflow, or a management progress view. Those can be packaged differently while sharing the same core artifact semantics.

The wrong move is to present shallow website copy as a specification. The right move is to make the protocol legible first, then let Meridian and other products mask or expose complexity according to audience.