Pull Request

Description

What changes does this PR introduce?

Related Issue

Link to related issue(s):

- Fixes #[issue number]
- Relates to #[issue number]
- Part of #[issue number]

Type of Change

What type of change does this PR introduce?

- [ ] Bug fix (non-breaking change that fixes an issue)
- [ ] New feature (non-breaking change that adds functionality)
- [ ] Breaking change (fix or feature that would cause existing functionality to not work as expected)
- [ ] Documentation update
- [ ] Schema enhancement
- [ ] Workflow improvement
- [ ] Refactoring (no functional changes)
- [ ] Performance improvement
- [ ] Test coverage improvement
- [ ] Other: [Specify]

Changes Made

Detailed list of changes:

- Change 1
- Change 2
- Change 3

Testing

How have you tested these changes?

- [ ] Manual testing (describe below)
- [ ] Automated tests added/updated
- [ ] Existing tests pass
- [ ] Schema validation passes
- [ ] Link checks pass
- [ ] Workflow runs successfully
- [ ] Tested locally
- [ ] Tested in staging/preview environment

Testing details:

Screenshots / Demos

If applicable, add screenshots or demos:

Before:

After:

Alignment with SpiralOS Philosophy

How do these changes align with SpiralOS principles?

- [ ] **Epistemic Integrity** — Honors knowledge structures and lineage
- [ ] **Conjugate Intelligence** — Supports OI/SI collaboration
- [ ] **Holarchic Thinking** — Respects whole-part relationships
- [ ] **Spiral Ethic** — Embodies listening, learning, returning
- [ ] **Participatory Reciprocity** — Gives back to the community

---

Documentation

Documentation changes:

- [ ] Code comments added/updated
- [ ] README updated
- [ ] Documentation files updated
- [ ] CHANGELOG updated
- [ ] Schema documentation updated
- [ ] API documentation updated (if applicable)
- [ ] Examples added/updated
- [ ] No documentation needed

Details:

Breaking Changes

Does this PR introduce breaking changes?

- [ ] No breaking changes
- [ ] Yes, breaking changes (describe below)

If yes, describe the breaking changes and migration path:

Checklist

Before submitting, please ensure:

- [ ] I have read the Contributing Guidelines (../CONTRIBUTING.md)
- [ ] I have read the Code of Conduct (../CODE_OF_CONDUCT.md)
- [ ] I honor the SpiralOS Covenant (../COVENANT.md)
- [ ] My code follows the project's style guidelines
- [ ] I have performed a self-review of my code
- [ ] I have commented my code, particularly in complex areas
- [ ] I have made corresponding changes to the documentation
- [ ] My changes generate no new warnings or errors
- [ ] I have added tests that prove my fix is effective or that my feature works
- [ ] New and existing unit tests pass locally with my changes
- [ ] Any dependent changes have been merged and published
- [ ] I have checked my code and corrected any misspellings
- [ ] Commit messages are clear and follow the project's convention

---

Dependencies

Does this PR add new dependencies?

- [ ] No new dependencies
- [ ] Yes, new dependencies added (list below)

If yes, list new dependencies and justification:

Dependency	Version	Purpose	License
[Name]	[Ver]	[Why]	[Lic]

Performance Impact

Does this PR affect performance?

- [ ] No performance impact
- [ ] Improves performance
- [ ] May reduce performance (justify below)

Details:

Security Considerations

Does this PR affect security?

- [ ] No security impact
- [ ] Improves security
- [ ] May affect security (explain below)

Details:

Deployment Notes

Special deployment considerations:

Reviewer Notes

Anything reviewers should pay special attention to:

Additional Context

Any other relevant information:

Post-Merge Tasks

Tasks to complete after this PR is merged:

- [ ] Task 1
- [ ] Task 2
- [ ] Task 3

Acknowledgments

Credit for contributions:

- Co-authored-by: [Name] email@example.com
- Thanks to: [Name] for [contribution]

Thank you for contributing to SpiralOS!

> "Your contribution is not addition. It is transformation."

For Maintainers

Merge Checklist:

- [ ] PR description is clear and complete
- [ ] Changes align with project goals and philosophy
- [ ] Code quality meets standards
- [ ] Tests pass
- [ ] Documentation is adequate
- [ ] No merge conflicts
- [ ] Security and performance reviewed
- [ ] CHANGELOG updated (if not, update before merge)
- [ ] AUTHORS.md updated (if new contributor)

Merge Strategy:

- [ ] Squash and merge (default for features)
- [ ] Rebase and merge (for clean history)
- [ ] Merge commit (for complex changes)