



ECHO
Ops and Governance

Version v0.1-draft
Commit: HEAD
December 4, 2025

James Ross & Echo Contributors

Copyright

Copyright (c) James Ross and FLYING ROBOTS; Echo contributors. Licensed under Apache-2.0 OR MIND-UCAL-1.0.

Trademarks

Echo and associated marks may be trademarks of their respective owners.

Warranty

Provided “as is”, without warranties or conditions of any kind.

Source

This booklet is generated from the Echo repository documentation.

Foreword

Echo is a deterministic, multiverse-aware engine. This booklet walks you in with progressive layers: orient yourself, learn the core building blocks, then dive into math and operations. Each shelf can stand alone; together they form the full Echo field guide.

If you are new, start with the onboarding roadmap and glossary. If you build or extend Echo, keep the determinism contract and scheduler flow in view. Future work will deepen each part and add more diagrams as Echo evolves.

Contents

I Ops and Governance	1
1 Ops and Governance (Starter)	3
1.1 Working Agreements	3
1.2 CI / Docs Guard	3
1.3 Next Additions	3

Part I

Ops and Governance

Chapter 1

Ops and Governance (Starter)

Echo favors merge-only history, deterministic builds, and documented intent.

1.1 Working Agreements

- No rebases or force pushes; use the branch merge playbook.
- Keep `execution-plan.md` and `decision-log.md` in sync with any runtime change.
- Run `cargo clippy -all-targets` and `cargo test` before PRs; Docs Guard expects updated metadata.

1.2 CI / Docs Guard

- Hooks run `fmt`, `clippy`, `docs guard`, and `SPDX` checks.
- GitHub Actions enforce `SPDX` headers and `LaTeX` docs builds.

1.3 Next Additions

- Determinism incident runbook.
- Branch merge decision tree (with TikZ diagram).
- Telemetry/diagnostics quickstart.

License and Legal Notice

This project is made available under an open source, dual-licensing model.

Code

All *code* in this repository—including Rust source files, scripts, build tooling, and any compiled binaries—is licensed under the **Apache License, Version 2.0**.

- Canonical text: LICENSE-APACHE
- SPDX identifier: Apache-2.0

Users may use, modify, and redistribute the code under the terms of the Apache License, Version 2.0.

Theory, Mathematics, and Documentation

The *theory*, *mathematics*, and *documentation* corpus associated with this project—for example LaTeX sources, notes, and expository materials—is dual-licensed under:

1. the Apache License, Version 2.0 (Apache-2.0), *or*
2. the MIND-UCAL License, Version 1.0 (MIND-UCAL-1.0),

at the user’s option.

If you do not wish to use MIND-UCAL, you may freely use all theory, mathematics, and documentation under the Apache License, Version 2.0 alone. No part of this project requires adopting MIND-UCAL in order to be usable.

SPDX Headers

To make licensing machine-readable and unambiguous, the project uses SPDX license identifiers in file headers. Typical examples include:

- Code files (Rust, scripts, etc.):

```
// SPDX-License-Identifier: Apache-2.0
```

- Documentation and theory files (Markdown, LaTeX, etc.):

```
% SPDX-License-Identifier: Apache-2.0 OR MIND-UCAL-1.0
```

These identifiers correspond directly to the licenses described above.

Disclaimer

Unless required by applicable law or agreed to in writing, the material in this project is provided on an “AS IS” basis, without warranties or conditions of any kind, either express or implied. For the full terms, see LICENSE-APACHE and LICENSE-MIND-UCAL.