OCC — Operational Consistency Compiler

Canonical release document (formal defense + operational standard)

Version: 1.0.0 Release ID: OCC-2026-02-15 Date: 2026-02-15
Author: Marco A. Isaac Alcuria

Scope and intended placement in the research workflow

OCC is designed as a late-stage filter for candidate mathematical models and “claims”: it is not intended to restrict early-stage creative model-building. The recommended workflow is: Creative phase (free exploration): propose candidate structures and mechanisms. Compilation phase (OCC): declare domain Ω_I, projection, observables, datasets and resources; run judges/locks; obtain PASS/FAIL/NO-EVAL with witnesses. Final judge (Universe): compare predictions with independent data and experiments. OCC therefore operates as an operational compiler: it enforces evaluability, traceability, and non-negotiable consistency constraints before a claim is presented as physics.

Foundation and update rule: concept first, equations second

OCC’s core non-negotiable foundation is ISAAC: an operational closure implied by the conjunction of (i) relativistic causal signalling limits, (ii) quantum energy–momentum carried by probes, and (iii) gravitational backreaction. ISAAC is conceptually stable as long as SR, QM and GR remain valid in their tested regimes. At the same time, any concrete numeric expression used to operationalize ISAAC (and other judges) is allowed to evolve with improved empirical knowledge. The key rule is: Equations depend on the concept. Numeric thresholds are an empirical layer. The concept does not depend on any single equation. Changing a fit or a bound does not change the operational closure itself.

Reproducibility quickstart (one-liner)

The canonical MRD suite is shipped inside this release. A minimal end-to-end run is:

```
# From the release root
python -m venv .venv
source .venv/bin/activate
pip install -r CANONICAL/ILSC_MRD_suite_15_modulos_CANON/requirements.txt

python CANONICAL/ILSC_MRD_suite_15_modulos_CANON/RUN_ALL.py \
--root CANONICAL/ILSC_MRD_suite_15_modulos_CANON \
--summary CANONICAL/ILSC_MRD_suite_15_modulos_CANON/verification_summary.json
```

Outputs:

verification_summary.json: PASS/FAIL/NO-EVAL per module and per input case.
certs/: audit certificates (hash-checked, deterministic where declared).

Data policy (external sources, licensing, provenance)

This release ships small, hash-checked data files. External sources and licensing are summarized in DATASETS.md. For each data artifact, a companion *.meta.json records provenance (source URL/DOI, access date) and a sha256 hash. If a host does not state an explicit license for a compilation, OCC records this explicitly and treats the file as a reproducibility mirror with attribution; users may delete the mirrored copy and re-fetch directly from the source.

Nomenclature (ILSC ↔ OCC) and archival policy

Earlier drafts used “ILSC” as a working name. This canonical edition adopts: OCC — Operational Consistency Compiler (official name of the standard and reference implementation). ILSC — appears in some archived filenames and historical text; it should be read as an alias of OCC. This release keeps an ARCHIVE/ folder to preserve older artifacts for auditability. Canonical materials are under CANONICAL/.