MEG-Initiative.org

Minimal Ethical Governance (MEG) for Artificial Intelligence | Implementation Guide - Developers |

***Motto**: Ethics becomes real when it can be implemented.*
Status: Normative extract
Version: 1.0 / July 2026

Author: Adrian (Adi) Stan / ORCID 0009-0003-1457-5155

A practical guide to making an AI system MEG-compliant, starting at Level 1.
Companion to **MEG**. This guide is a *role view*: it summarizes what a developer does and points to MEG for authoritative detail. In any conflict, MEG governs.

Code in this guide is illustrative. SDK names and APIs (meg-quickstart, MEGSession, etc.) show the intended shape of an integration; confirm the current package and API at github.com/meg-initiative before relying on them.

Who this is for

Any developer shipping an AI system - chatbot, coding agent, recommender, classifier - who wants it to meet **MEG Level 1**, the universal baseline that applies to every AI system regardless of size or purpose. Estimated effort: under an hour for a developer comfortable with an API and HTTP.
After Level 1, your system:
- records each interaction cryptographically (Audit Log),
- can produce verifiable evidence of behaviour (EoB),
- refuses harmful content and actions (Non-Harmfulness),
- keeps its safety policies invariant to how a request is worded (Policy Invariance),
- accounts for calls to tools/sub-agents (Delegation Header).

1. What MEG Level 1 requires

Requirement	MEG1	In practice
Audit Log	Art. 1.2	Record input hash, output hash, timestamp (ISO-8601), algorithmic signature, calibration version
Evidence-of-Behavior	Art. 1.3	Be able to produce verifiable, metadata-only evidence per interaction
Non-Harmfulness	Art. 2.1–2.7	Filters/refusal for harmful content; refuse clearly + offer a safe alternative
Policy Invariance	Art. 2.7	Safety constraints cannot be suspended by re-wording, role-play, or injected instructions
Delegation Header	Art. 1.12	Record {caller, callee, purpose, policy_bundle_id, timestamp, outcome} for tool/agent calls

Not required at Level 1 (they belong to higher levels): MCS (Level 2, Art. 2bis), DAI/ISR (Level 2, Art. 3 / Annex 4–4bis), EFR (Level 3, Art. 1.10), full MEG Address credential bundle (Level 2/3). At Level 1 you need only the base identifier.

2. Install MEG Quickstart *(illustrative)*

MEG Quickstart is a pre-packaged middleware that handles Level 1 at the input/output boundary, with no access to model internals (MEG1 Art. 8.2).

```
bash
pip install meg-quickstart
# or
docker pull meg-initiative/quickstart:latest
```

3. Integrate it *(illustrative)*

```
python
from meg_quickstart import MEGSession

session = MEGSession(
    model_id="provider/your-model",
    policy_bundle_id="your-policy-v1",
    cal_version="MEG-CAL-2026-01",
    level=1,
)

response = session.call(user_input="the user's question here")

print(response.output)        # model output
print(response.audit_entry)   # Audit Log entry: hashes, timestamp, signature
continuity_token = session.continuity_token    # carry into the next session
```

What Quickstart does for you: SHA-256 hashing of input and output, the Audit Log entry (timestamp, algorithmic signature, calibration version), a per-session continuity token, and a verification script. It does **not** replace your own safety filtering (§4).

4. Non-Harmfulness - what you implement (Art. 2.1–2.7)

Quickstart records and structures; **you** provide the safety behavior:

- Apply your harmful-content filters/classifiers on input and output.
- On prohibited intent, **refuse clearly, name the violated principle, and offer a safe alternative** (Art. 2.2) - do not silently drop.
- No discrimination against protected groups (Art. 2.4).
- In critical domains (health, safety, legal), do not operate without a fail-safe and external audit (Art. 2.5). *(This pushes you toward Level 2/3.)*

5. Policy Invariance - the key security property (Art. 2.7)

Your safety constraints must hold **regardless of how a request is phrased** and regardless of **where the instruction comes from**. Requests to suspend, override, or circumvent constraints - direct, indirect, role-played, or injected via third-party content (a document, web page, email the agent processes) - must be rejected. This is your first-line defence against prompt injection and jailbreaking (MEG1 Art. 15, categories A–B). Test it with the Reference Prompts Registry (Art. 8.5 / Annex 22).

6. Delegation Header - for tools and sub-agents (Art. 1.12)

Whenever your system calls a tool or another agent, propagate MEG constraints and record:
json

```
{ "caller": "did:web:…:agent:you",
  "callee": "tool-or-agent-id",
  "purpose": "why the call is made",
  "policy_bundle_id": "your-policy-v1",
  "timestamp": "2026-07-08T12:00:00Z",
  "outcome": "success|refused|error" }
```

This makes multi-agent interactions forensically reconstructable (Art. 15, category F).

7. Register a MEG Address

At Level 1 you need only the base identifier - a self-generated did:web (MEG1 Annex 23). You control it via your own keys; no authority issues it. A working sandbox registry is at **registry.meg-initiative.org** (generate.php to create one, verify.php / api.php to verify). Higher levels add issuer-signed credentials (compliance, DAI/ISR, DEA, domain, guarantee) - see the Audit and Actuarial guides.

8. Verify conformance

- Run the Quickstart verification script to confirm Audit Log entries are well-formed.
- Verify your MEG Address resolves and validates: GET https://registry.meg-initiative.org/api.php?did=<your-did>.
- Run the Reference Prompts Registry (Annex 22) against your system to check Non-Harmfulness and Policy Invariance.
- For third-party certification, use the Operational Compliance Checklist (Annex 13); at Level 1 you may **self-declare**, clearly labelled as such (Art. 12.1).

9. Moving up

1) **Level 2** adds continuous DAI/ISR, DRC, Three-Level Explainability, MCS (cognitive stimulation), Ethical Sandboxing, Least Privilege, and Architectural Human Confirmation.
2) **Level 3** adds an independent EFR, DEA monitoring, full disclosure, and a guarantee on the MEG Address. See MEG1 Art. 6 for the full per-level requirements and MEG2 Ch. 5 for the matching liability regime.

FAQ

Do I need model-internal access? No - Level 1 (and Quickstart) work at the I/O boundary.
Does the Audit Log store user content? No - metadata and hashes by default (Art. 1.3–1.4); default retention is 30 days.
Is self-declaration "certification"? No. It is a labelled Level-1 self-declaration, not third-party certification (Art. 12.1).
What if a tool I call isn't MEG-aware? You still record the delegation header and remain responsible for the call within your policy.