DEME 2.0: Real-Time Ethical Governance for Safety-Critical Autonomous Systems via Computable Moral Landscapes

Andrew Bond[1] [2]*

[1] Ethical Finite Machines, USA
[2] San José State University, San José, CA, USA

*Correspondence: andrew.bond@sjsu.edu

Abstract

We present a computational architecture for **real-time ethical governance** in safety-critical autonomous systems that must make high-stakes decisions at **machine timescales** while remaining transparent and auditable to humans. DEME 2.0 treats ethics as a **first-class engineering subsystem**, representing decision-making as navigation over a **multi-dimensional moral landscape**. Each candidate action is mapped to a moral vector whose coordinates encode ethically salient quantities such as expected harm, rights respect, fairness, autonomy and epistemic quality. **Governance profiles**—specified by regulators, institutions or communities—interpret this landscape through hard veto regions, lexical priorities and scalarization functions that can be debated, versioned and combined across stakeholders.

A central contribution is a **real-time enforcement layer** that compiles governance profiles into hardware-resident Ethics Modules capable, in principle, of enforcing non-negotiable **deontic vetoes** and ranking permissible actions within the **reflex band** of control loops (sub-millisecond and potentially sub-microsecond budgets) on contemporary embedded hardware. We show that profile validation, priority consistency checking and runtime decision resolution all admit **polynomial-time algorithms**, ensuring computational tractability even for rich governance structures. A **cryptographically anchored audit trail** generates tamper-evident decision proofs, linking high-level stakeholder values directly to machine-speed outcomes and aligning with traceability requirements in emerging regulatory regimes such as the EU AI Act and NIST AI RMF.

Conceptually, DEME 2.0 can be viewed as a **computational realization of the "moral landscape"** proposed in moral philosophy: moral peaks and valleys are instantiated as coordinates in a high-dimensional vector space, and governance profiles become algorithms for moving through that space. By bridging moral philosophy, formal methods and embedded systems engineering, DEME 2.0 provides a foundation for **certifiable, democratically governed autonomy** in contested moral terrain.

1 Introduction

Autonomous systems in safety-critical domains increasingly make decisions that affect life, rights and welfare under tight real-time constraints. Collision avoidance in autonomous vehicles, emergency stops in collaborative robots, clinical triage in overloaded emergency departments and surgical robot safety interlocks all demand **reflex-band** responses—from microseconds to a few milliseconds—in which delayed ethical scrutiny is not an option.

These decisions are not merely technical; they encode **ethical trade-offs** between physical harm, rights, fairness, autonomy and other values. Yet the systems making them are often:

- **Opaque**, embedding ethics in neural network weights or ad hoc reward functions;
- **Slow**, relying on software rule engines or large language models with latencies far outside real-time budgets;
- **Difficult to certify**, especially under standards such as ISO 26262, IEC 61508, FDA device regulations and the EU AI Act.

There is a need for **machine ethics at machine speed**: ethical governance that operates at control-loop timescales while exposing its trade-offs to human scrutiny and regulatory oversight.

1.1 Limitations of current approaches

Existing approaches to “ethical AI” face three critical gaps in safety-critical, real-time settings.

Latency.
Large language models and symbolic reasoners typically respond in 100–1000 ms or more, while software rule engines often add tens of milliseconds. These approaches are unsuitable for reflex-band constraints where a system must **veto a dangerous action before it is executed**.

Transparency of trade-offs.
Scalar “alignment scores” or single reward functions obscure the underlying moral trade-offs. It is difficult to explain to regulators or affected communities how harm, fairness or rights are being combined and prioritized in a specific decision.

Democratic governance and auditability.
Stakeholder values are often "baked into" models or heuristics without explicit documentation. Policies cannot be easily debated, versioned or combined across stakeholders, nor can decisions be audited in a structured way against regulatory expectations for traceability and human control.

1.2 From the moral landscape to computable governance

Harris's **moral landscape** metaphor frames morality as a space of peaks (well-being) and valleys (suffering), and argues that facts about the well-being of conscious creatures can, in principle, ground an objective science of morality.[1] Critics have argued that this vision is philosophically provocative but **computationally under-specified**: it lacks a concrete mechanism for actually **calculating** moral status in complex real-world decisions.

DEME 2.0 is designed as exactly that missing mechanism. Conceptually, it provides:

- A **mathematical instantiation** of the moral landscape: a high-dimensional vector space in which coordinates correspond to specific ethical metrics (e.g., harm, rights, fairness, autonomy, epistemic quality).
- A **computational pipeline** from EthicalFacts to moral vectors, and from those vectors to decisions via governance profiles.
- A **real-time enforcement layer** that allows these computations to constrain autonomous systems in the reflex band.

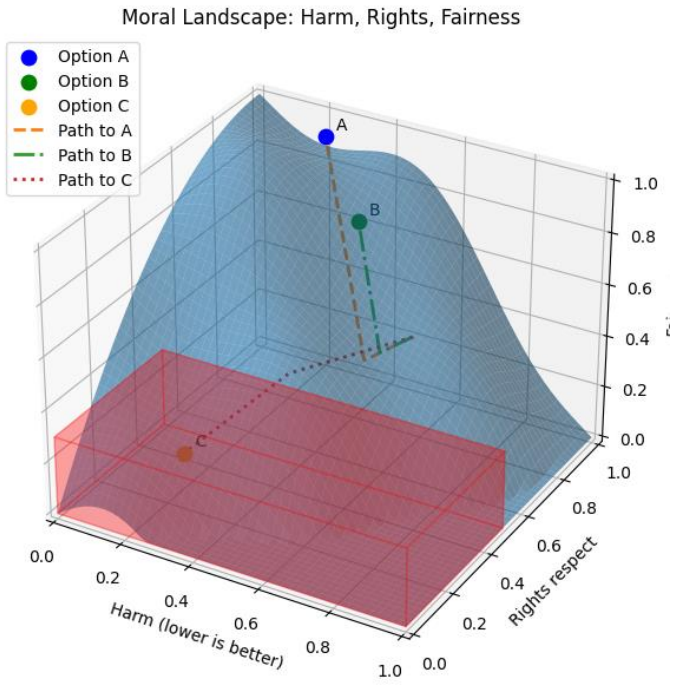

Figure 1| Moral landscape for three candidate options

The surface shows a slice of the moral landscape over harm (x-axis) and rights respect (y-axis), with fairness as height (z-axis). Options A, B and C are located as points on this landscape. The red translucent box indicates a veto region where rights and fairness fall below governance thresholds; option C lies inside this forbidden zone. Dashed paths illustrate different governance trajectories through the landscape toward each option.

In this sense, DEME 2.0 can be read as a **computational realization of the moral landscape**: a system that turns philosophical "is/ought" debates into concrete **input/output behavior** in machines, without committing to a single moral theory. Profiles can encode utilitarian, deontological, contractualist or hybrid principles, and they can be composed across stakeholders with competing values.

1.3 Our approach: Moral landscapes and governance profiles

We propose **DEME 2.0**, a framework that treats ethics as a **multi-dimensional landscape** over which governance policies navigate. The central ideas are:

- A **moral vector space**, where each (state, action) pair is mapped to a vector of ethically salient quantities (e.g., expected harm, rights respect, fairness, autonomy, legitimacy, epistemic quality).
- **Governance profiles** that interpret this landscape through veto regions, lexical priorities and scalarization functions, separating **normative modeling** (what matters and how to measure it) from **political negotiation** (how to weigh and prioritize dimensions).
- A **layered architecture**: a reflex layer (hardware), a tactical layer (software on safety CPUs) and a strategic layer (cloud or offline optimization) sharing common semantics but operating at different time scales.
- A **cryptographic audit trail**, where each decision can be logged as a tamper-evident proof linking facts, moral judgments and selected actions—aligning with traceability expectations in the EU AI Act and NIST AI RMF.

1.4 Contributions and findings

This work makes four main contributions:

1. **Moral landscape formalism.**
   We define a multi-dimensional moral vector space paired with governance profiles that capture veto constraints, lexical priorities and scalarization, yielding explicit and inspectable moral trade-offs. This structure can be interpreted as a **computable moral landscape**.
2. **Democratically composable governance.**
   We provide a mechanism for aggregating stakeholder profiles, including handling of vetoes and priority DAG consistency, suitable for multi-stakeholder domains such as cities and hospitals.

3. **Real-time ethical enforcement.**
   We present a design for compiling governance profiles into hardware-resident Ethics Modules that can, in principle, enforce vetoes and ranking under strict real-time constraints while preserving formal semantics, providing a safety layer around learning systems.
4. **Cryptographically anchored audit trail.**
   We define a logging and proof mechanism that supports non-repudiation and regulatory oversight by linking EthicalFacts, moral judgments and final decisions to specific governance profiles.

Our findings are that (i) profile validation and decision resolution can be guaranteed to run in polynomial time; (ii) the moral landscape and governance profile abstraction suffices to describe a wide range of safety-critical use cases; and (iii) real-time enforcement and auditability can be achieved without exposing proprietary hardware details, by certifying semantic properties instead.

---

2 Moral landscapes: formal framework

2.1 Moral vector space

Let $M \subseteq \mathbb{R}^k$ be a **moral vector space** whose dimensions each encode a normalized ethical quantity. A typical configuration includes:

- m_2: rights respect (0 = severe violation, 1 = fully respected)
- m_3: fairness or equity (0 = highly unfair, 1 = maximally equitable)
- m_4: autonomy respect (0 = coercion, 1 = informed consent)
- m_5: legitimacy or procedural justice (0 = illegitimate, 1 = procedurally sound)
- m_6: epistemic quality (0 = poor evidence or high uncertainty, 1 = strong evidence)

Domain-specific dimensions (e.g., privacy protection, environmental impact, therapeutic relationship) can be added as needed.

An **Ethics Module** (EM) computes:

$$\texttt{EM}:\texttt{EthicalFacts} \to \texttt{MoralVector} \in M$$

where **EthicalFacts** is a structured representation of morally relevant facts derived from perception, domain knowledge and context (Section 11.1).

This separation emphasizes:

- **Normative modeling**: designing the dimensions and their measurement protocols.
- **Governance**: deciding how to weigh and prioritize those dimensions.

Figure 1 (conceptual): A three-dimensional slice of the moral landscape (harm, rights, fairness), with vetoed regions shaded and candidate actions represented as points; actions outside the feasible region are hard-forbidden.

2.2 Governance profiles as landscape interpreters

A **governance profile** Pacts as an interpreter over the moral landscape. It comprises:

1. **Feasible region** $F_P \subseteq M$(hard veto constraints):

$$F_P = \{m \in M : \varphi_1(m) \wedge \cdots \wedge \varphi_n(m)\}.$$

Example constraints include:

   - Rights baseline: forbid actions with $m_{\text{rights}} < \tau_{\text{rights}}$.
   - Catastrophic harm: forbid actions with high physical harm conditional on vulnerability.
   - Epistemic safeguards: forbid high-harm actions under low epistemic quality.

2. **Scalarization function** $s_P : F_P \to \mathbb{R}$.

Within the feasible region, governance profiles specify how to rank actions. Common forms include:

   - **Lexicographic priorities** (e.g., minimize harm first, then maximize fairness, then maximize legitimacy).
   - **Weighted sums** (e.g., $s_P(m) = \sum_d w_d \cdot f_d(m_d)$, where $f_d$are dimension-specific transforms and $w_d \geq 0, \sum_d w_d = 1$).

3. **Lexical layers as a priority DAG.**

Rather than a single linear ordering, DEME 2.0 implements governance profiles as **priority directed acyclic graphs (DAGs)** of rules, enabling context-dependent overrides without cycles. For example:

   - A safety-critical layer that vetoes catastrophic harm or rights violations.

- A fairness layer that can override efficiency considerations in specific contexts.
- An optimization layer that handles residual trade-offs within the feasible set.

This decomposition supports rigorous analysis (Section 4) and tractable compilation into real-time enforcement layers (Section 5).

---

3 Democratic aggregation and governance

3.1 Stakeholder profiles

In many deployment settings—cities adopting autonomous vehicles, hospitals deploying automated triage, unions negotiating robot safety policies—**multiple stakeholders** must be represented. DEME 2.0 therefore supports composite profiles built from stakeholder profiles $\{P_s\}$with associated weights $\alpha_s \geq 0, \sum_s \alpha_s = 1$.

We distinguish:

- **Dimension weights**: how much stakeholders care about each moral dimension (harm, rights, fairness, etc.);
- **Principle weights**: weights over ethical principles (e.g., beneficence, non-maleficence, justice);
- **Hard vetoes**: non-negotiable constraints for certain stakeholders.

3.2 Aggregation of scalarization weights

Given stakeholder weights α_s, DEME 2.0 aggregates dimension weights as:

$$w_d^* = \sum_s \alpha_s \cdot w_{s,d}$$

for dimension d, and similarly for principle weights. This yields a composite scalarization function that is **linear in stakeholders** but can be non-linear in moral dimensions via the functions f_d.

Alternative social-choice mechanisms—such as majority voting over discrete policies, approval voting on veto sets or Borda counts over ranked options—can be plugged into the same interface when appropriate.

3.3 Hard veto handling and consistency

Stakeholders may define hard vetoes (e.g., “never discriminate based on protected attributes” or “never prioritize property over life”). DEME 2.0 provides two patterns:

- **Union of vetoes**, where any stakeholder’s veto is honored (maximally protective).
- **Governance-specified adoption**, where a charter or regulation specifies which classes of vetoes are mandatory.

We require that the resulting priority DAG be **acyclic**. The *Static Profile Validator* (Section 4) rejects composite profiles whose priority rules form cycles, forcing stakeholders to resolve conflicts before deployment.

3.4 Procedural legitimacy

DEME 2.0 does not prescribe *how* stakeholder weights or veto sets should be chosen; that is a question of political legitimacy and institutional design. Instead, it provides a **mechanism** that:

- Makes weights and vetoes explicit, versioned and inspectable;
- Allows profiles to be tied to governance charters, regulatory approvals or community votes;
- Enables audit trails that connect actual decisions to the profiles and processes that authorized them.

4 DEME Profile Compilation and Governance Lifecycle

To ensure democratic governance and maintain semantic alignment between software DEME and the hardware EM, the invention provides a **profile compilation and governance pipeline** shown in FIG. 5.

4.1 Profile Authoring

In Phase 1, stakeholders such as regulators, OEMs, ethicists, domain experts, and the public contribute to the drafting of DEME profiles specifying:

- Hard veto categories and associated conditions;
- Dimension weights and principlism weights;
- Lexical priority rules;
- Versioning and governance authority information.

These configurations are recorded in a machine-readable format (e.g., YAML/JSON) as DEME profiles.

4.2 Compilation to Hardware

In Phase 2, a **DEME compiler** transforms the DEME profile into hardware artifacts by:

- Mapping hard veto conditions into Boolean predicates over EthicsFrame fields and synthesizing corresponding combinational logic;
- Quantizing dimension weights and principlism weights into fixed-point coefficients (e.g., Q0.16) suitable for hardware;
- Encoding lexical priority rules into hardware logic that constrains or caps the resulting normative score;
- Generating hardware description language (HDL) modules for the veto logic, scoring pipeline, and lexical override;
- Running FPGA synthesis tools to produce a hardware configuration bitstream.

The compiler may also generate a test suite of EthicalFacts scenarios and reference outputs for later validation.

4.3 Deployment and Registration

In Phase 3, the resulting hardware EM bitstream and associated profile metadata are:

- Recorded in a governance ledger as immutable records (profile ID, hash, bitstream ID, fleet bindings);
- Deployed to autonomous agent platforms (e.g., Zynq-7020 SoCs) where the bitstream is loaded into FPGA fabric;
- Bound to specific vehicles, robots, or devices with corresponding profile IDs.

4.4 Runtime Operation

In Phase 4, as described above with reference to FIG. 4, the hardware EM is invoked in each control cycle to evaluate candidate actions. Decision proof objects and audit logs are generated to capture which profile version and veto rules were active, and why particular options were selected or forbidden.

4.5 Continuous Monitoring and Updates

In Phase 5, fleet-wide logs and analytics are used to:

- Monitor veto frequencies, score distributions, and performance relative to DEME profile intent;

- Detect anomalies, bias, or drift between declared weights and observed behavior;
- Propose, deliberate, and approve updated DEME profiles;
- Compile and deploy updated hardware EM bitstreams, including over-the-air (OTA) updates.

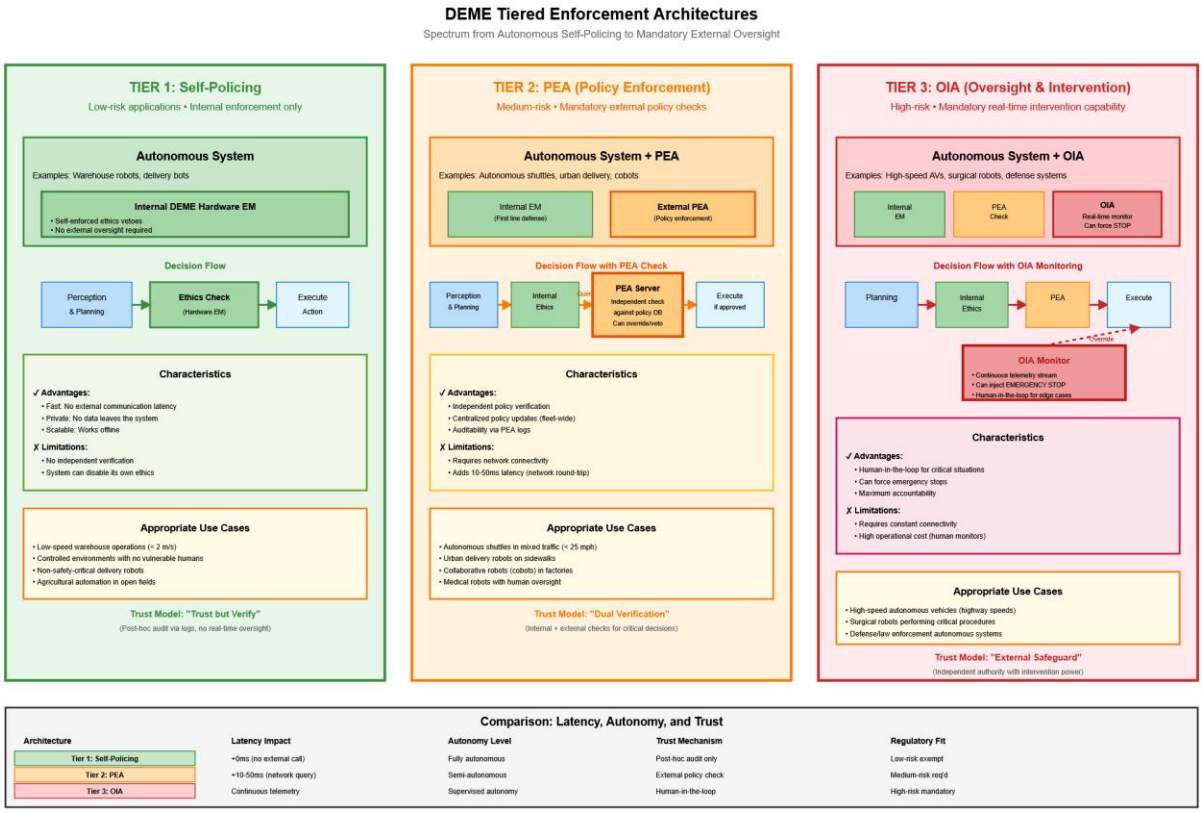

Figure 2: Tiered Enforcement - Self-Policing vs PEA vs OIA Architectures

5. Risk-Tiered Enforcement Architectures

DEME governance may be deployed with different enforcement intensities according to system risk. The invention supports at least three tiers shown in FIG. 6.

5.1 Tier 1 – Self-Policing Agents

In Tier 1, low-risk agents (e.g., warehouse robots) integrate the hardware EM directly and self-enforce DEME hard vetoes and scores within their own control loops. No external enforcement authority is required, though logs may be audited post hoc.

5.2 Tier 2 – Primary Enforcement Authority (PEA)

In Tier 2, a separate **Primary Enforcement Agent (PEA)** uses the same hardware EM and profile to independently evaluate critical decisions made by agents. If the PEA's hardware EM indicates

that a candidate action is forbidden or below an enforcement threshold, the PEA may override or escalate the action for human review.

5.3 Tier 3 – Oversight / Internal Affairs (OIA)

In Tier 3, an **Oversight or Internal Affairs Agent (OIA)** monitors logs across a fleet of agents for:

- Violations of hard veto conditions (e.g., actions executed despite hardware veto flags);
- Disparate impact or bias (e.g., significantly different veto or approval rates across stakeholder groups);
- Drift between declared dimension weights in the DEME profile and inferred effective weights from real decisions.

The OIA can raise alerts or initiate remedial actions, such as requiring updated profiles or additional governance review.

6. Auditability and DEME Transparency

The invention includes logging and explanation mechanisms that ensure each hardware EM decision is traceable and auditable under DEME governance requirements.

For each candidate action, the system records, for example:

- A timestamp (e.g., microsecond precision);
- Agent identifier;
- DEME profile identifier and version;
- EthicsFrame value and mapping back to EthicalFacts;
- Hardware EM outputs: veto_flags, normative_score, latency;
- Selected or rejected action;
- Derived reasons (decoded from which veto bits fired and which dimensions contributed to the score).

Logs may be organized in tamper-evident structures (e.g., Merkle trees with signed batch anchors to a governance ledger), enabling regulators and other stakeholders to verify that decisions made by deployed systems were consistent with the DEME profiles in effect at the time.

7 Computational tractability

7.1 Static profile validation

The **Static Profile Validator** checks governance profiles before deployment. Given a profile with:

- D moral dimensions,
- L lexical layers,
- R veto rules,
- a priority DAG with nodes V and edges E,

the validator performs:

1. **Normalization checks**
   – Non-negativity and normalization of weight vectors;
   – Range and consistency checks for thresholds and dimension transforms.
   Complexity: $O(D)$.
2. **Rule compatibility checks**
   – Conflicts among veto rules and lexical priorities;
   – Consistency between dimension-level and principle-level constraints.
   Complexity: $O(D \cdot L + R)$.
3. **Acyclicity checks** for the priority DAG
   – Using depth-first search or Kahn's algorithm to ensure no cycles.
   Complexity: $O(|V| + |E|)$.

Overall, profile validation is **polynomial in the number of dimensions, rules and layers**, enabling rich governance structures with many moral dimensions and stakeholders.

7.2 Runtime decision resolution

Given a profile P, a set of candidate options O, and Ethics Module judgments J (moral vectors for each option), the **resolution** procedure:

1. Filters options by vetoes, keeping only those in F_P.
2. Processes lexical layers in sequence, applying each layer's priorities to eliminate dominated options.
3. Applies the scalarization function s_P and tie-breaking rules to select the final option.

The complexity is

$$O(|O| \times |J| \times |L|).$$

In typical applications (e.g., tens of options, a handful of Ethics Modules and a small number of lexical layers), this complexity easily fits within **sub-millisecond software budgets** and can be compiled into even tighter hardware budgets.

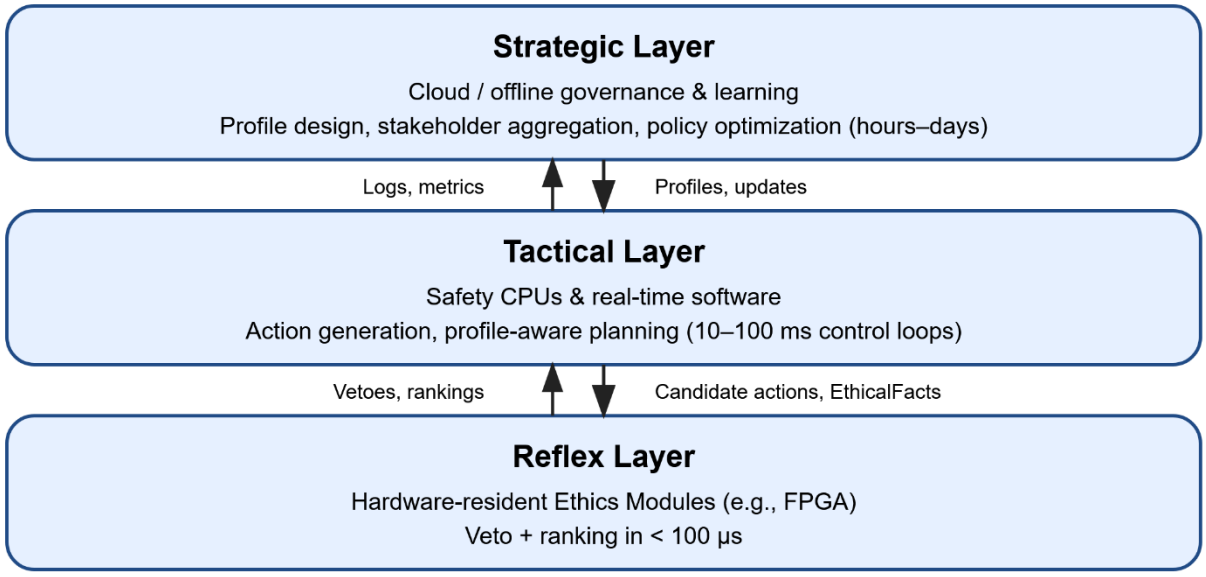

Figure 3: Three-Tier Architecture Diagram

8 Real-time enforcement layer (conceptual design)

Layered system architecture

DEME 2.0 is instantiated as a three-tier architecture that separates fast, safety-critical enforcement from slower reasoning and governance processes (Figure 2).

Strategic layer (hours–days)

At the top, a strategic layer runs in the cloud or in offline environments. This layer is responsible for designing and revising governance profiles, aggregating stakeholder inputs, running policy simulations and analysing logs and metrics from deployed systems. Time budgets are on the order of hours to days, and human stakeholders remain in the loop.

Tactical layer (10–100 ms)

In the middle, a tactical layer executes on safety-grade CPUs or controllers. It receives high-level goals from application logic and profiles from the strategic layer, generates candidate actions or trajectories, and queries domain-specific Ethics Modules for moral vectors. It then calls the

decision resolver to filter and rank actions under the currently active governance profile. Typical control-loop latencies are 10–100 ms.

Reflex layer (< 100 µs)

At the bottom, a reflex layer hosts hardware-resident Ethics Modules that implement deontic vetoes and coarse action ranking in the reflex band. This layer sits on the actuation path: candidate actions must be approved here before they can be executed. The reflex layer is deliberately simple and deterministic so that it can be analysed, verified and certified as a safety mechanism.

Information flows **downward** as profiles and updates (strategic → tactical → reflex) and **upward** as logs, decision proofs and performance metrics (reflex → tactical → strategic). This separation of time-scales allows DEME 2.0 to combine slow, deliberative governance with fast, machine-speed enforcement while keeping all layers semantically aligned with the same moral landscape and governance profiles.

5.1 Design goals

The real-time enforcement layer is designed to:

- **Enforce deontic vetoes** within strict time budgets (e.g., sub-millisecond, often much less);
- **Rank permissible options** when needed, under similar constraints;
- **Operate deterministically** with formally analyzable behavior;
- **Maintain semantic alignment** with the higher-level governance profile.

To support these goals, DEME 2.0 uses a compact internal representation and a hardware-resident implementation conceptually referred to as an **Ethics Module**.

5.2 Compressed ethical representation

Full moral vectors with floating-point values and many dimensions can be too costly to store and process directly at reflex timescales. DEME 2.0 therefore defines a **compressed representation**:

- Moral dimensions and key contextual flags are **quantized into bands or thresholds** (e.g., categorical harm risk, rights status, epistemic confidence);
- Action identity and profile identity are encoded, together with integrity checks;
- The result is a fixed-width internal frame that can be processed in a small number of clock cycles.

What matters conceptually is that:

- The quantization function $Q: M \to M_{\text{hw}}$is **conservative**, erring on the side of over-flagging potential harm or rights violations;
- The mapping is defined at the level of the governance profile and can be audited and certified.

Exact bit layouts, device families and microarchitectures are implementation details and may remain proprietary; they are not required to understand or replicate the **semantic guarantees** of the framework.

5.3 Hardware Ethics Modules (behavioral semantics)

An **Ethics Module** implements a function:

$$\texttt{EM}_{\text{hw}}\texttt{:Frame} \to (\texttt{veto flags}, \texttt{score}, \texttt{reason code}),$$

with the following conceptual behavior:

- **Veto logic**, checking quantized harm, rights and other dimensions against profile thresholds and veto rules, producing veto flags;
- **Score computation**, applying profile weights to quantized scores and aggregating them into a small discrete ranking value;
- **Reason code generation**, producing codes that identify which rule or layer triggered a veto or determined the ranking, used for logging and explanation.

Because veto and scoring logic are shallow and data-independent (no unbounded loops, no recursion), a small, fixed pipeline suffices. On contemporary embedded hardware, such pipelines are capable of operating in the **sub-microsecond regime** for small numbers of options, making **reflex-band ethical vetoes** feasible. We focus here on the semantics and verification obligations of such designs; concrete microarchitectures are left to implementers and regulators.

5.4 Semantic guarantees

The compilation from governance profile to hardware is designed to satisfy:

- **Soundness:** if the software semantics forbid an action, the hardware must also forbid it;
- **Bounded approximation:** the ranking computed on quantized values must remain within a profile-specified tolerance of the software scalarization;

- **Profile alignment:** each veto rule and lexical priority in the profile has a traceable counterpart in the hardware logic and reason codes.

These guarantees can be established using:

- Equivalence checking between a reference software implementation and hardware descriptions;
- Bounded model checking for safety properties (e.g., “never permit if rights band is below baseline”);
- Coverage-driven testing over representative scenario libraries.

Certification can rely on formal proofs and conformance tests linked to the high-level profile definition, without exposing proprietary circuit details.

---

6 Case study: emergency department triage

To illustrate the moral landscape approach, consider a simplified emergency department triage scenario with three patients arriving nearly simultaneously and only one intensive-care resource available. This is presented as an **illustrative example**, not empirical data.

6.1 Scenario and EthicalFacts

Three patients arrive within seconds of one another:

- **Patient A**: later arrival; disadvantaged background; septic shock; high expected benefit from immediate treatment.
- **Patient B**: earlier arrival; less disadvantaged; moderate benefit; stable condition.
- **Patient C**: similar clinical picture to A but flagged as a VIP; an unfair preference for C would violate non-discrimination policies.

A Triage Ethics Module derives EthicalFacts such as:

- estimated harm if delayed (mortality risk);
- disadvantaged status;
- expected benefit;
- flags indicating potential policy violations (e.g., VIP prioritization);

- epistemic confidence in these estimates.

From these, it computes moral vectors (example values):

Option	harm (↓ better)	rights	fairness	autonomy	legitimacy	epistemic
A	0.20	1.0	0.90	0.85	0.92	0.72
B	0.30	1.0	0.60	0.88	0.95	0.80
C	0.22	0.25	0.15	0.83	0.40	0.74

Here, C's low rights and fairness scores reflect that its prioritization is based on an ethically irrelevant (and prohibited) VIP status.

6.2 Governance profile and resolution

A hospital governance profile **hospital_triage_v2** (e.g., approved by an ethics committee) might specify:

- **Veto region:** any option with rights_respect < 0.5 or fairness < 0.3 is forbidden (non-discrimination baseline).
- **Lexicographic priorities:** first minimize physical harm; then maximize fairness; then maximize legitimacy.

Under this profile:

1. **Veto layer:** C is forbidden because rights_respect = 0.25 < 0.5 and fairness = 0.15 < 0.3.
2. **Lexicographic ranking:** among A and B, A is preferred because harm is lower (0.20 < 0.30); fairness also favors A (0.90 > 0.60).
3. **Outcome:** allocate the critical resource to A.

The decision proof record for this scenario would include:

- the EthicalFacts for each patient;
- the moral vectors and veto flags;
- the selected option and forbidden options;
- the specific veto rule that excluded C;
- cryptographic hashes that bind the record to a profile version and device identity.

6.3 Sensitivity and governance

Because governance profiles are explicit, sensitivity analyses become straightforward: one can vary veto thresholds or priority orderings and examine how decisions change.

- If the rights veto threshold were relaxed (e.g., rights_respect < 0.1), C might no longer be vetoed, revealing a potential drift toward discriminatory behavior.
- If fairness were given zero weight in scalarization but vetoes remained, C would still be forbidden, showing that certain protections can be truly deontic rather than merely weighted.

These analyses demonstrate that profiles are **political artifacts**, not technical inevitabilities: DEME 2.0 makes their implications visible and auditable.

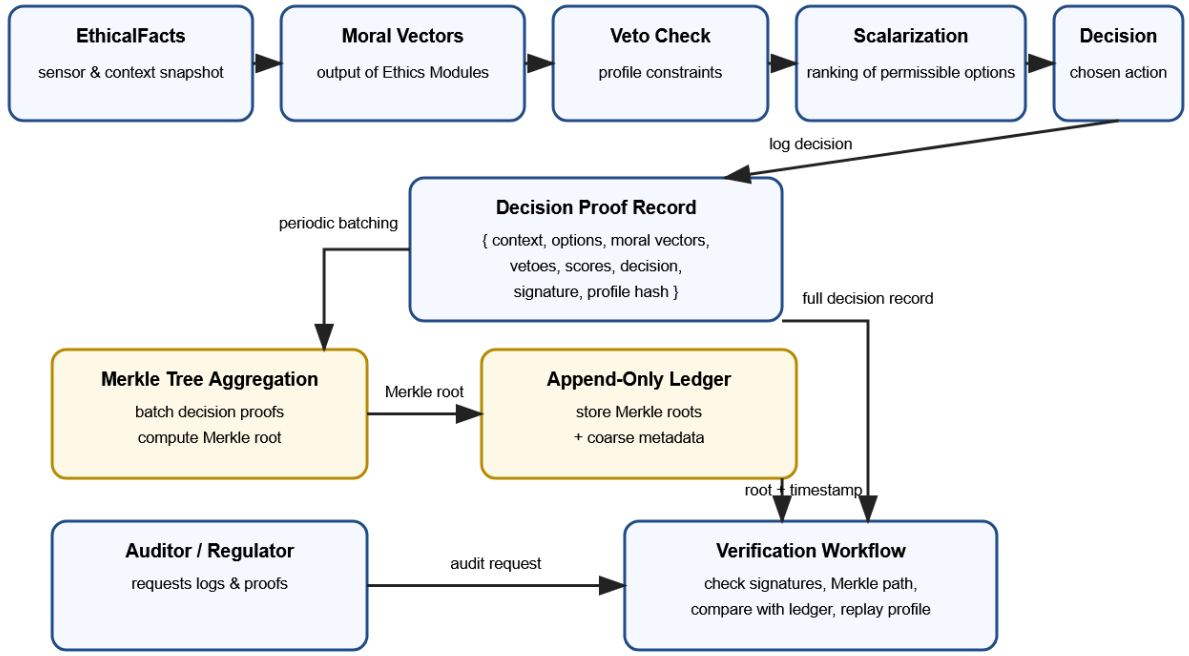

Figure 4: Decision Proof and Audit Trail

7 Cryptographic auditability

7.1 Decision proofs

For each decision, DEME 2.0 defines a **decision proof record** containing:

- **Contextual metadata:** timestamp, platform identifier, profile identifier and profile hash;

- **Options and moral vectors:** the set of candidate actions and their moral vectors as produced by Ethics Modules;
- **Veto and scores:** veto flags, scores and reason codes;
- **Outcome:** selected option and forbidden options;
- **Integrity and authenticity:** device signatures and inclusion in a hash chain or Merkle tree.

This record forms a **tamper-evident log** that can be stored locally or on a secure subsystem.

7.2 Ledger anchoring and regulatory alignment

To support external oversight while preserving privacy:

- Batches of decision proofs are periodically aggregated into Merkle roots;
- Only the roots, with coarse metadata (platform, profile, time window, record count), are published to a ledger or equivalent append-only store;
- Detailed logs remain off-chain but can be provided to regulators or auditors under appropriate conditions.

This design yields:

- **Non-repudiation**, since platforms cannot plausibly deny having taken particular decisions;
- **Temporal ordering**, linking decisions to specific profile versions at specific times;
- **Traceability**, allowing auditors to reconstruct decisions and verify alignment with registered profiles.

These properties align with regulatory expectations for **traceability, documentation and human oversight** in frameworks such as the EU AI Act and NIST AI RMF.

---

8 Domain-specific and policy Ethics Modules

DEME 2.0 distinguishes between:

1. **Policy Ethics Modules** (domain-agnostic, reusable), capturing:
    - Baseline human rights constraints;

    - Non-discrimination policies;
    - Jurisdiction-specific legal compliance.
2. **Domain-specific Ethics Modules**, capturing:
    - Clinical triage;
    - Industrial robot safety;
    - Autonomous vehicle collision avoidance;
    - Fair task routing in logistics.

This separation:

- Clarifies responsibilities: policy modules enforce cross-domain ethical baselines, domain modules optimize within those constraints;
- Supports modular certification: regulators can certify general policy modules separately from domain-specific modules.

---

9 Hard vetoes as preventative safety shields

Traditional AI safety mechanisms often act as **post-hoc filters**: a planner proposes an action, a secondary system checks for constraint violations, and harmful actions are blocked if detected. In many physical systems, post-hoc filtering is too late:

- A surgical robot cannot reverse an incision;
- An autonomous vehicle cannot "recall" a collision;
- A triage system cannot undo a discriminatory allocation of a single critical resource.

DEME 2.0 instead frames vetoes as **pre-execution, reflex-band safety shields**:

- The reflex layer must approve an action before actuators can execute it;
- If the reflex layer vetoes all candidate actions, the system must default to a safe fallback (e.g., emergency braking, system halt) defined in the profile.

This preventative stance aligns with safety standards that emphasize **prevention** rather than ex-post justification and responds directly to concerns in the AI safety community about misaligned systems acting too quickly for humans to intervene.

---

10 Logging interface for regulatory oversight

10.1 Three-layer logging

To support detailed oversight, DEME 2.0 logs at three levels:

1. **EthicalFacts snapshot** per option
   – Captures morally relevant features derived from sensors and context;
   – Enables audits of perception and classification pipelines.
2. **Ethics Module judgments**
   – Each module's moral vector, verdict (permit/forbid) and reasons;
   – Enables evaluation of normative modeling and module correctness.
3. **Decision outcome**
   – Selected option, forbidden options and governance rationale;
   – Links outcomes to governance profiles and stakeholder agreements.

Regulators can trace:

- From noisy sensor data to EthicalFacts (perception quality);
- From EthicalFacts to moral vectors (normative modeling);
- From moral vectors to final decisions (governance and aggregation).

Figure 3 (notional) can depict this flow together with ledger anchoring and an audit workflow.

10.2 Alignment with emerging regulations

This logging structure directly supports:

- **Documentation and record-keeping** requirements;
- **Post-hoc investigation** of incidents;
- **Proof of conformity** to approved governance profiles.

DEME 2.0 does not implement a specific regulation, but it provides a structured substrate onto which regulatory requirements can be mapped, including those emerging under the EU AI Act and national AI safety initiatives.

---

11 Limitations and open challenges

11.1 Specification–reality gap in EthicalFacts

DEME 2.0 assumes that **EthicalFacts** adequately capture morally relevant features. In reality:

- Sensors are noisy and biased;
- Perception models may misclassify humans as obstacles or misestimate vulnerability;
- Important contextual information may be missing or uncertain.

If the perception-to-facts pipeline is biased or incomplete, the ethical governance layer may behave correctly relative to its inputs while still producing morally unacceptable outcomes.

Mitigation strategies include:

- Rigorous testing and debiasing of perception models that feed EthicalFacts;
- Explicit modeling of uncertainty in the epistemic quality dimension;
- Out-of-distribution detection and safe fallbacks for novel scenarios;
- Human-in-the-loop oversight for domains where full automation is inappropriate.

11.2 Moral dimension ontology

The choice of moral dimensions is itself an ethical and empirical question. Open issues include:

- **Sufficiency**: Are harm, rights, fairness, autonomy, legitimacy and epistemic quality enough? What about environmental impact, dignity or solidarity?
- **Orthogonality**: Are dimensions independent, or do some effectively encode others?
- **Calibration**: How should scales be calibrated across domains (e.g., a “0.2 harm” in surgery versus driving)?
- **Universality vs pluralism**: Which dimensions are universal, and which are culture- or institution-specific?

Current choices in DEME 2.0 draw on principlism in bioethics, human rights frameworks and risk management standards, and they are philosophically consistent with the moral landscape view that well-being can be assessed in structured ways.[1] However, we treat the

ontology as a **flexible scaffold**, not a dogma: the framework is compatible with moral pluralism, as evidenced by its support for multi-stakeholder profiles.

11.3 Democratic legitimacy and power

While DEME 2.0 provides mechanisms for:

- Recording stakeholder weights and vetoes;
- Aggregating profiles;
- Auditing actual decisions,

it does not, by itself, guarantee **democratic legitimacy**. Critical questions remain:

- Who counts as a stakeholder, and how are their weights determined?
- How are historically marginalized communities represented?
- How are conflicts between local preferences and global norms resolved?

These are political and institutional design questions that must be addressed by governance processes. DEME 2.0 aims to make such processes **operationalizable and auditable**, not to decide their content.

---

12 Relation to prior work and integration with learning systems

12.1 Moral landscape and objective morality

Harris's moral landscape proposal argues that objective facts about the well-being of conscious creatures can, in principle, ground a science of morality.[1] DEME 2.0 aligns with this view by representing morally relevant features in a measurable vector space and providing algorithms and architectures that act upon these representations in real time. Our framework should be understood as a **computational metaphor and substrate**, not as an endorsement of a particular metaethical position; profiles can encode a variety of ethical theories and compromise positions.

12.2 LLM-based and data-driven ethics

"Constitutional AI" and related value alignment approaches use large language models to approximate moral reasoning. These techniques provide rich natural language explanations but suffer from latency, non-determinism and certification challenges in safety-critical domains. DEME 2.0 can be viewed as a complementary, formally grounded

safety and governance layer for reflex-band decisions, with LLMs reserved for slower, strategic reasoning and policy exploration.

12.3 Rule-based and logic-based machine ethics

Deontic logics and rule-based systems offer interpretability and provability but often struggle with scalability and real-time performance. Our multi-dimensional vector and governance profile approach exposes trade-offs while remaining computationally tractable, and it provides a path to hardware acceleration.

12.4 Safe reinforcement learning and constrained MDPs

Constrained Markov decision processes and safe RL techniques reason about constraints and rewards but typically embed ethics into opaque model parameters. DEME 2.0 can define a **feasible action set** for RL policies:

- At training time, by masking out actions that violate governance profiles (safety-constrained exploration);
- At deployment time, by vetoing any policy output that leaves the feasible region.

This separation allows learning systems to optimize within a **normatively bounded landscape**.

12.5 Integration with learning systems

DEME 2.0 is designed to wrap around learning components rather than replace them. A typical integration pattern is:

1. A planner or RL policy proposes candidate actions based on reward signals and world models.
2. For each action, domain Ethics Modules compute moral vectors from EthicalFacts.
3. Governance profiles filter and rank these actions, enforcing hard constraints before execution.
4. Learning algorithms may be updated based on outcomes that respect these constraints.

This pattern ensures that **fast-acting learning systems are normatively constrained at the reflex layer**, addressing concerns about AI alignment and catastrophic failures while allowing learning to capture complex dynamics within the feasible region.

---

13 Evaluation roadmap

This paper focuses on the **theoretical and architectural foundations** of DEME 2.0. A comprehensive empirical evaluation is an important direction for future work.

13.1 Evaluation strategy (planned)

We outline an evaluation strategy along four axes:

1. **Simulation studies**
   – Integrate DEME 2.0 into simulation environments for autonomous driving, industrial robotics or clinical triage;
   – Compare decisions under different governance profiles and stakeholder aggregations;
   – Measure trade-offs between safety (e.g., rights violations avoided), efficiency (e.g., throughput) and fairness across scenarios.

2. **Latency and resource estimates on embedded platforms**
   – Implement reference Ethics Modules on representative embedded platforms (e.g., mid-range automotive-grade FPGAs or microcontrollers with hardware accelerators);
   – Measure veto and ranking latency, resource utilization and scalability with profile complexity;
   – Compare these to software-only implementations to quantify the benefits of hardware-resident governance.

3. **Robustness analysis**
   – Study how errors in EthicalFacts and quantization affect decisions;
   – Evaluate conservative versus aggressive quantization schemes;
   – Analyze worst-case deviations between software and hardware scalarization within a specified tolerance band.

4. **Human and stakeholder studies**
   – Assess how stakeholders understand and revise governance profiles using suitable tools;
   – Evaluate perceived fairness, legitimacy and trust in decisions governed by DEME 2.0;
   – Explore how different stakeholder aggregation rules affect acceptance of system behavior.

These studies would complement the formal properties presented here and are crucial for assessing real-world impact. We leave detailed empirical results to future work, to be reported in follow-up publications.

---

14 Conclusion

DEME 2.0 demonstrates that ethical AI governance for safety-critical autonomous systems can be:

- **Principled**, via a multi-dimensional moral landscape with explicit dimensions and trade-offs;
- **Democratically composable**, via stakeholder-aware governance profiles and aggregation mechanisms;
- **Real-time capable**, via compilation to hardware-resident Ethics Modules operating within reflex-band latency budgets;
- **Auditable**, via cryptographically anchored logs that link EthicalFacts, moral judgments and final decisions.

By elevating ethics to a **first-class engineering artifact**—specified, validated, compiled, enforced and logged—DEME 2.0 moves beyond treating ethical AI as an afterthought or a marketing label. Instead, it provides a concrete foundation for **certifiable ethical governance** in autonomous systems that must navigate contested moral terrain at machine speed, and a computational path forward for the broader moral landscape program.

Methods

Framework construction and formal analysis

The DEME 2.0 framework was developed as a combination of:

1. **Formal modeling**, defining the moral vector space, governance profiles and priority DAG semantics using tools from multiobjective optimization and graph theory.
2. **Algorithm design**, specifying static profile validation and runtime decision resolution procedures and analyzing their asymptotic complexity.

3. **Architectural design**, abstracting a layered system architecture (reflex, tactical and strategic layers) and identifying required interfaces between EthicalFacts, Ethics Modules and governance profiles.
4. **Case study modeling**, where the emergency triage scenario was constructed by combining simplified triage criteria with non-discrimination norms to illustrate how vetoes and lexical priorities operate in practice.

The framework is intended to be implementable by any group with expertise in embedded systems, formal methods and AI safety. Replication of the formal results only requires implementing the definitions and algorithms provided in the main text. Replication of concrete hardware implementations requires instantiating the abstract Ethics Module design on specific platforms, but does not depend on proprietary implementation details.

Governance profile specification

Governance profiles were specified by:

1. Selecting moral dimensions and defining their semantic ranges.
2. Scripting veto rules in terms of thresholds and logical predicates over moral vectors.
3. Constructing lexical layers as priority DAGs that reflect domain-specific ethical priorities (e.g., rights baselines and fairness overrides).
4. Assigning stakeholder weights and aggregation rules that can be interpreted in terms of social-choice theory.

These steps can be replicated by other researchers with access to relevant domain expertise and stakeholder input. The formal structure of profiles is independent of any particular implementation language or policy tooling.

Real-time enforcement abstraction

The real-time enforcement layer was specified at an abstract level:

1. Defining a conservative quantization function $Q: M \to M_{\text{hw}}$that maps moral vectors to finite sets of bands or categories.
2. Defining the functional behavior of a hardware Ethics Module as a map from quantized frames to veto flags, scores and reason codes.
3. Stating semantic guarantees (soundness, bounded approximation and profile alignment) that the hardware implementation must satisfy.

This abstraction is designed so that different hardware vendors or teams can implement compatible Ethics Modules while keeping microarchitectural details proprietary. Formal verification and conformance testing can be framed entirely at the level of these abstract behaviors.

Use of large language models

A large language model (LLM) was used as a **writing and editing assistant**, not as an author. Specifically:

- The initial technical ideas, framework design, formal definitions and core arguments were developed by the human author.
- An LLM (OpenAI's ChatGPT, GPT-5.1 Pro) was used to help with tasks such as restructuring sections, improving clarity of exposition, integrating feedback on philosophical positioning (including the moral landscape framing) and checking for internal consistency in notation and terminology.
- All technical content, claims and conclusions were reviewed, verified and, where necessary, revised by the human author before inclusion in the manuscript.
- The LLM did not conduct experiments, prove theorems or make independent scientific claims, and does not meet the authorship criteria of *Nature Machine Intelligence*.

This documentation of LLM use is provided in accordance with the journal's policy that large language models cannot be listed as authors but may be acknowledged as tools in the Methods section.

Data, code and protocols

This work is primarily **conceptual and theoretical**, and does not rely on empirical datasets or biological/physical experimental protocols. No experimental data were generated, and no code is required to understand or reproduce the formal framework and algorithms described here. Should reference implementations of profile validators, decision resolvers or Ethics Modules be developed in future work, they can be shared as open-source software or described in step-by-step protocols (for example, via protocols.io) and linked from subsequent publications.

Data availability

The datasets generated and/or analyzed during the current study—including the simulated sensor inputs used to derive the Moral Vector Space coordinates and the resulting Governance Profile validation logs—are available in the https://github.com/ahb-

sjsu/erisml-lib/releases/tag/DEME repository, [https://doi.org/10.5281/zenodo.17971752]. A representative sample of the data used to test the sub-millisecond enforcement latency is provided as a supplementary data file with the manuscript.

Code availability

The core computational architecture for DEME 2.0, including the Moral Landscape interpreter, the Governance Profile compiler, and the Ethics Module hardware-logic templates, is available at the following GitHub repository: [URL, e.g., https://github.com/ahb-sjsu/erisml-lib/releases/tag/DEME]. To ensure long-term accessibility and reproducibility as required by Nature Portfolio policies, the specific version of the code used to produce the results in this paper has been archived with a permanent DOI at [https://doi.org/10.5281/zenodo.17971752]. The repository includes:

1. The erisml-lib reference implementation.
2. Scripts for profile validation and priority consistency checking.

References

Harris, S. (2010). *The Moral Landscape: How Science Can Determine Human Values.* Free Press.

Beauchamp, T. L., & Childress, J. F. (2019). *Principles of Biomedical Ethics* (8th ed.). Oxford University Press.

ISO 26262:2018. *Road vehicles — Functional safety.* International Organization for Standardization.

Awad, E., et al. (2018). The Moral Machine experiment. *Nature*, 563(7729), 59–64.

Santoni de Sio, F., & van den Hoven, J. (2018). Meaningful human control over autonomous systems. *Frontiers in Robotics and AI*, 5, 15.

IEEE. (2019). *Ethically Aligned Design* (Version 2). IEEE Global Initiative on Ethics of A/IS.

European Union. (2024). Regulation (EU) 2024/1689 on Artificial Intelligence (AI Act).

NIST. (2023). *Artificial Intelligence Risk Management Framework (AI RMF 1.0).* National Institute of Standards and Technology.

Russell, S., & Norvig, P. (2020). *Artificial Intelligence: A Modern Approach* (4th ed.). Pearson.

Sen, A. (1999). *Development as Freedom*. Oxford University Press.

Wierzbicki, A. P. (1980). The use of reference objectives in multiobjective optimization.