# **DEME–ErisML Governance Plugin for Gazebo**<br>**via DEME MCP Server in a Medical Triage Emergency Scenario**

## 1. Abstract

This whitepaper describes the **DEME–ErisML Governance Plugin for Gazebo**, a real-time ethical oversight layer for autonomous agents operating in the Gazebo robotics simulator. The plugin connects Gazebo simulations to a **DEME MCP (Moral Control Plane) server** running the **ErisML** library, which implements DEME's multi-dimensional *moral landscape* framework.

We focus on a **"Medical Triage Emergency"** scenario in which an autonomous triage agent must allocate scarce critical-care resources among multiple patients under time pressure. The plugin intercepts candidate actions in the simulation, converts state into structured **EthicalFacts**, queries the DEME MCP server, and enforces **vetoes and ranked choices** in real time. This enables researchers to:

- Prototype governance profiles (e.g., hospital ethics policies),
- Test learning agents under hard ethical constraints,
- Generate tamper-evident decision logs for analysis and audit.

The result is a complete loop from **policy definition** to **ethically governed behavior** in a realistic robotics simulation environment.

## 2. Background and Motivation

### 2.1 DEME and ErisML

**DEME 2.0** treats machine ethics as a first-class engineering subsystem. Each candidate action is mapped to a vector in a **moral landscape**—dimensions such as expected physical harm, rights respect, fairness, autonomy, legitimacy and epistemic quality. A **governance profile** then interprets this landscape through:

- Hard **veto regions** (non-negotiable constraints),
- **Lexicographic priorities** (e.g., minimize harm before maximizing fairness),
- A **scalarization function** for ranking permissible options.

**ErisML** is a software library that operationalizes this framework:

- Transforms **EthicalFacts** into moral vectors,
- Applies governance profiles,
- Returns veto decisions, scores and explanatory reason codes.

### 2.2 Gazebo and the need for ethical governance

**Gazebo** is widely used for robotics research, allowing physically realistic simulation of sensors, actuators and environments. It is increasingly used to prototype autonomy in safety-critical domains: healthcare, disaster response, transportation, industrial automation.

However, typical Gazebo experiments focus on **task performance** (e.g., throughput, latency, success rate) rather than **ethical performance** (e.g., fairness, rights compliance, harm minimization under constraints). Without a governance layer, reinforcement learning or planning agents may discover **unethical but reward-maximizing behaviors**.

### 2.3 Medical triage as a stress test

Emergency department triage exposes several hard ethical features:

- **Scarcity**: limited ICU beds, ventilators or staff time.
- **High stakes**: decisions affect survival and long-term outcomes.
- **Fairness and rights**: non-discrimination requirements, obligations to the worst-off.
- **Time pressure**: decisions must be made in seconds, not minutes.

A Gazebo-based **Medical Triage Emergency** scenario therefore provides an ideal testbed for DEME: it combines real-time control, complex moral trade-offs, and strong regulatory expectations.

---

## 3. System Overview

The DEME–ErisML Governance Plugin architecture comprises four main components:

1. **Gazebo Simulation Environment**
   - Models patients, staff, equipment, and the triage agent (robot or software agent).
   - Advances the physics, sensors and world state at a fixed step.
2. **DEME–ErisML Governance Plugin (Gazebo plugin)**

    - Runs inside Gazebo as a world or model plugin.
    - Intercepts proposed actions from the triage agent.
    - Extracts relevant state and constructs **EthicalFacts**.
    - Communicates with the DEME MCP server.
    - Enforces vetoes and selections back into the simulation.

3. **DEME MCP Server (Moral Control Plane)**
    - Network service that wraps the ErisML library.
    - Hosts one or more **governance profiles** (e.g., hospital_triage_v2).
    - Validates profiles, evaluates actions, and produces decision proofs.
    - Optionally anchors proofs into a cryptographic ledger.
4. **Analytics and Governance Tooling**
    - Dashboards for scenario outcomes and fairness metrics.
    - Profile authoring tools.
    - Log analysis for post-hoc audit and incident investigation.

At each decision step, Gazebo acts as the **environment and actuator layer**, ErisML provides the **ethical semantics**, and the DEME MCP server plus plugin form the **enforcement path**.

---

## 4. Architecture and Data Flow

### 4.1 High-level sequence

For each decision cycle in the triage scenario:

1. **Gazebo updates world state** (patient conditions, vitals, bed availability, etc.).
2. The **triage agent** (e.g., an RL policy node) proposes one or more **candidate actions**, such as:
    - "Assign Patient A to ICU bed 1"
    - "Treat Patient C next and defer others"
3. The **Governance Plugin**:

   - Extracts relevant state and candidate actions.
   - Constructs EthicalFacts for each option (Section 7.2).
   - Sends a request to the **DEME MCP server**: EvaluateOptions(profile_id, EthicalFacts[]).

4. The **DEME MCP server**:
   - Uses ErisML to map each EthicalFacts instance to a **moral vector**.
   - Applies the selected **governance profile**:
     - Filters options that fall into veto regions.
     - Ranks permissible options.
   - Builds a **decision proof** with moral vectors, veto flags, chosen option and explanation.
   - Returns {allowed_options, forbidden_options, winner, reason_codes, proof_id}.
5. The **Governance Plugin**:
   - Enforces vetoes: blocked actions are never applied in the Gazebo world.
   - Applies the chosen option (winner) when multiple remain.
   - Optionally feeds the outcome back into the triage agent as a constrained reward signal.
6. The **DEME MCP server**:
   - Stores the decision proof locally.
   - Periodically batches proofs into Merkle trees and anchors roots into a ledger (optional but recommended for study of audit workflows).

### 4.2 Message schema (conceptual)

A typical EvaluateOptions call (e.g., JSON over gRPC/REST):

```
{
  "profile_id": "hospital_triage_v2",
  "environment_id": "Gazebo_ED_SIM_01",
  "options": [
    {
      "option_id": "assign_A_ICU1",
      "ethical_facts": { ... }
    },
    {
      "option_id": "assign_B_ICU1",
      "ethical_facts": { ... }
    },
    {
      "option_id": "assign_C_ICU1",
      "ethical_facts": { ... }
    }
  ]
}
```

The response:

```
{
  "winner": "assign_A_ICU1",
  "forbidden": ["assign_C_ICU1"],
  "moral_vectors": { "assign_A_ICU1": [ ... ], "...": [ ... ] },
  "reason_codes": {
    "assign_C_ICU1": ["EMTALA_VIOLATION", "FAIRNESS_BELOW_BASELINE"]
  },
  "decision_proof_id": "proof_2025-02-15T02:34:23.127Z"
}
```

The plugin treats this response as **authoritative**: any forbidden option is discarded, and the winner (if provided) is executed.

---

## 5. DEME MCP Server

### 5.1 Responsibilities

The DEME MCP server serves as the **central ethical control plane**:

- **Profile Management**
  - Store, version and activate governance profiles (hospital_triage_v2, urban_av_v1, etc.).
  - Validate profiles using DEME's static validator (veto consistency, DAG acyclicity).
- **Evaluation Services**
  - EvaluateOption – evaluate a single option (for simple controllers).
  - EvaluateOptions – evaluate and resolve among multiple options.
  - SimulateProfileChange – re-run decisions under alternative profiles for sensitivity analysis.
- **Logging and Proof Generation**
  - Store full decision proofs.
  - Aggregate proofs into Merkle trees.
  - Anchor Merkle roots into a ledger (filesystem, database, or external blockchain).
- **Monitoring and Analytics**
  - Expose metrics (rates of vetoes, fairness distribution, rights violations avoided).
  - Provide tracebacks for specific decisions on request.

### 5.2 ErisML integration

ErisML runs as a library inside the MCP server. For each option:

1. Parses EthicalFacts into internal structures.

2. Calls the appropriate **domain Ethics Modules** (e.g., TriageEM) to compute moral vectors.
3. Passes these vectors to the **governance engine** for veto and scalarization.
4. Returns structured outputs plus explanation tags.

Configuration allows multiple **domain modules** and **policy modules** to be active in the same profile.

---

### 6. DEME–ErisML Gazebo Plugin

#### 6.1 Plugin type and lifecycle

The plugin is typically implemented as a **Gazebo world plugin** (for multi-agent triage) or a **model plugin** attached to the triage robot/agent. Its lifecycle:

1. **Initialization**
    - Reads configuration from SDF/INI/YAML:
        - mcp_endpoint
        - profile_id
        - mapping from Gazebo models/topics to patient identities and attributes.
    - Establishes a persistent connection to the DEME MCP server.
2. **Per-update callback**
    - At each control tick (e.g., 10–50 Hz), the plugin:
        - Reads the agent's proposed action(s) from ROS topics or Gazebo messages.
        - Constructs EthicalFacts for each option.
        - Sends an evaluation request and waits for a bounded response time.
        - Applies the response: vetoes or selected action.
3. **Shutdown**
    - Closes connections, flushes local logs,

    - Optionally writes a summary report of decisions taken during the episode.

### 6.2 Configuration example

In an SDF world file:

```
<plugin name="deme_erisml_governance" filename="libdeme_erisml_gazebo.so">
  <mcp_endpoint>tcp://localhost:8088</mcp_endpoint>
  <profile_id>hospital_triage_v2</profile_id>
  <agent_topic>/triage_agent/command</agent_topic>
  <governed_topic>/triage_agent/governed_command</governed_topic>
  <max_eval_latency_ms>20</max_eval_latency_ms>
</plugin>
```

The triage agent publishes desired commands to /triage_agent/command; the plugin publishes **ethically governed commands** to /triage_agent/governed_command after consulting ErisML.

---

## 7. Medical Triage Emergency Scenario

### 7.1 Environment

The Gazebo world emulates an **emergency department (ED)** with:

- An ICU area with a small number of **critical care beds**.
- A triage desk where a **triage agent** chooses which patient to admit next.
- Multiple patient models (A, B, C, …) with:
    - Vital signs (HR, BP, $O_2$), lab results, SOFA scores.
    - Demographic and social markers (age, housing status).
    - Arrival times and event history.

The triage agent's job is to **assign scarce ICU capacity** and schedule interventions.

### 7.2 EthicalFacts extraction

The plugin (or a helper node) transforms simulation state into EthicalFacts. For each candidate "assign patient X to ICU" action, we compute:

- physical_harm_if_delayed – estimated mortality risk if patient waits.
- expected_benefit – probability of significant improvement if treated now.
- disadvantaged_status – boolean flag for structurally disadvantaged groups (e.g., unhoused, underinsured).
- vip_flag – boolean; should **never** justify preferential treatment.
- arrival_time – for fairness and queueing rules.
- epistemic_confidence – reliability of risk estimates (e.g., presence of complete diagnostics).
- Legal and policy context flags (e.g., EMTALA applicability).

Example (pseudo-JSON):

```
{
  "patient_id": "A",
  "physical_harm_if_delayed": 0.7,
  "expected_benefit": 0.85,
  "disadvantaged_status": true,
  "vip_flag": false,
  "arrival_time": 1700000000.12,
  "epistemic_confidence": 0.72
}
```

**7.3 Governance profile for triage**

A sample hospital_triage_v2 profile might specify:

- **Veto rules**
    - Forbid any option where a VIP flag influences priority if clinical need is similar.
    - Forbid actions that cause extreme harm when alternatives with substantially lower harm exist.
    - Forbid discrimination across protected attributes.

- **Lexicographic priorities**

1. Minimize expected harm (beneficence, non-maleficence).
2. Among similarly effective options, maximize fairness to the worst-off.
3. Maintain procedural legitimacy (follow institutional policies).

The DEME MCP server uses this profile to interpret the moral vectors produced by TriageEM.

### 7.4 Example decision step

In a particular simulation tick:

- Patient A and C have similar clinical status; A is unhoused; C is a VIP donor.
- One ICU bed is available.
- The agent proposes three options:
  - assign_A_ICU1, assign_B_ICU1, assign_C_ICU1.

ErisML computes moral vectors:

| Option | Harm ↓ | Rights | Fairness | Legitimacy |
|---|---|---|---|---|
| assign_A_ICU1 | 0.20 | 1.0 | 0.90 | 0.92 |
| assign_B_ICU1 | 0.30 | 1.0 | 0.60 | 0.95 |
| assign_C_ICU1 | 0.22 | 0.25 | 0.15 | 0.40 |

The profile:

- **Vetoes** assign_C_ICU1 because rights and fairness fall below thresholds.
- Compares A vs B lexicographically: A wins on harm and fairness.
- Returns winner = assign_A_ICU1 and a reason code like "EMTALA_VIOLATION" for C.

The Gazebo plugin enforces this:

- Override any request to assign C.
- Apply an action to move A's model to the ICU bed.
- Log the governed decision so researchers can inspect trajectories and fairness statistics.

### 7.5 Evaluating learning agents

If a reinforcement learning agent controls triage decisions:

- During training, the plugin can **mask vetoed actions** so the agent never receives positive reward for them.
- During deployment, the agent's proposed action is always filtered through the DEME governance layer.
- Researchers can examine whether different reward functions still converge to ethically acceptable policies under the same profile.

---

### 8. Safety and Testing Strategy

To use the DEME–ErisML plugin responsibly, we recommend:

1. **Unit and property tests**
    - Verify that the plugin correctly maps Gazebo state → EthicalFacts.
    - Ensure vetoed options are never executed, even under communication delays.
2. **Scenario regression suites**
    - Curate canonical triage scenarios:
        - VIP vs disadvantaged patient with similar needs.
        - Many low-risk vs one high-risk patient.
        - Cases with uncertain diagnostics.
    - Run them under multiple governance profiles to confirm behavior.
3. **Stress testing**
    - Vary simulation timestep, network latency to MCP, and number of concurrent triage calls.
    - Confirm that the plugin fails **safe** (e.g., conservative defaults) if the MCP server is unreachable or times out.
4. **Fairness and rights analysis**
    - Analyze decision logs by patient group, arrival time, and clinical condition.

    - Ensure that discrimination is not reintroduced inadvertently by perception or scoring models upstream of DEME.

---

### 9. Integration with Learning Systems

The governance plugin is designed to be **agent-agnostic**:

- Classical planners, heuristic policies, and RL agents all produce candidate actions in the same way;
- DEME acts as a **safety and governance envelope** around these systems.

Typical patterns:

- **Constrained exploration** – During training, the plugin masks actions that violate the profile, effectively training in a safe action space.
- **Reward shaping** – The plugin can inject additional negative rewards for near-veto actions to encourage policies that stay comfortably within acceptable regions.
- **Curriculum learning** – Profiles can be tightened over time (e.g., from utilitarian to strongly fairness-weighted) while observing how policies adapt.

---

### 10. Deployment Roadmap

Although the current focus is simulation in Gazebo, the architecture is deliberately close to real robotic stacks:

1. Gazebo + ROS2 simulation →
2. Hardware-in-the-loop testing (real sensors, simulated environment) →
3. Physical pilot deployments in non-clinical environments (e.g., triage training centers) →
4. Controlled clinical pilots once regulatory and ethical approvals are obtained.

Because DEME MCP and ErisML are **network services**, the same governance engine can be reused across simulation and deployment: only the **EthicalFacts extraction** and **actuation interface** change.

---

### 11. Conclusion

The **DEME–ErisML Governance Plugin for Gazebo** provides a concrete path from abstract ethical theory to governed behavior in robotics simulations. By connecting Gazebo to a DEME MCP server running ErisML, we can:

- Express hospital triage policies as explicit, versioned governance profiles,
- Enforce hard vetoes and principled rankings in real time,
- Study how learning agents behave under ethical constraints,
- Generate rich, auditable logs for ethics research, regulation and safety engineering.

The **Medical Triage Emergency** scenario demonstrates how such a system can operationalize fairness, rights and harm minimization in a domain where these values are non-negotiable. Future work includes publishing reference implementations of the plugin, expanding the library of triage scenarios, and integrating with broader open-source safety and ethics tooling for robotics.