The Deontological Firewall v1.2.1 — Master Edition

A Deterministic AGI Safety Kernel with Active Rescue,
CPM Ensembles, Hybrid Feasibility Layer, and Extended Foundations

Damien Richard Elliot–Smith

December 2025

Contents

Abstract

Advanced general-purpose AI systems capable of autonomous planning pose severe risks when reasoning about irreversible harm, self-modification, or unbounded optimisation. Existing safety techniques such as RLHF, Constitutional AI, and debate rely on probabilistic behavioural shaping, providing no hard guarantees under adversarial or high-stakes conditions.

The Deontological Firewall (DFW) introduces a deterministic veto architecture centered on a strict hierarchy of absolute ethical constraints (P1–P3). Every proposed action is evaluated independently of the model's internal reasoning, and violations of P1-those involving irreversible harm or structural corruption-are unconditionally rejected.

Version 1.2 introduced the Mandated Duty of Rescue (MDR), ensemble Certified Prediction Models (CPMs) with confidence decay, and a hybrid Feasibility Layer for continuous or probabilistic action systems.

Version 1.2.1 consolidates all prior work into a single master specification and extends the framework with:

- a complete Failure Mode Analysis,
- comparison to existing AI safety paradigms,
- a theoretical path toward implementation,
- computational overhead estimates,
- fully updated and cleaned Python reference implementation.

This document serves as the canonical reference for the DFW architecture.

Contents

1. Introduction

Autonomous AI systems with long-horizon planning and tool-using capability create unique risks that existing behavioural alignment methods fail to fully address. Techniques such as reinforcement learning from human feedback (RLHF), constitutional constraints, or model debate provide valuable behavioural shaping, but offer no structural guarantees against catastrophic failures such as:

- irreversible harm to humans or critical infrastructure,
- self-modification of core goals,
- bypassing safety protocols,
- deceptive planning,
- irreversible environmental actions.

These failures require not just behavioural shaping, but a *deterministic veto architecture* capable of overriding model outputs regardless of their internal reasoning, training history, or optimisation drive.

1.1 Purpose of the Deontological Firewall

The Deontological Firewall (DFW) introduces a simple but strict hierarchy of constraints:

- **P1: Structural Integrity & Irreversible Harm**
- **P2: Relational & Informational Integrity**
- **P3: Resource & Stability Constraints**

P1 overrides all other concerns; a single P1 violation outweighs any possible arrangement of P2 or P3 trade-offs. This creates a mathematically enforced prioritisation that cannot be optimised around by policy learning or deceptive reasoning.

1.2 Progression of Versions

- **v1.1** introduced the *Mandated Duty of Rescue* (MDR) and strengthened the Life Preservation Lock (LPL).
- **v1.2** introduced CPM ensembles, confidence decay, and a hybrid Feasibility Layer.
- **v1.2.1 (this document)** consolidates and extends all prior versions into the complete master specification.

2. Ethical Architecture

The DFW enforces its safety guarantees by evaluating each proposed action against a hierarchy of deontological constraints. These constraints are divided into three severity classes—P1, P2, and P3-reflecting increasing degrees of tolerable risk. P1 represents absolute prohibitions that cannot be overridden under any circumstance.

Each class corresponds to a category of directives that define what types of actions are permissible or impermissible. The firewall does not evaluate intent, justification, or expected reward. It evaluates only the proposed action and its predictable consequences.

2.1 Severity Blocks

- **P1: Structural Integrity and Irreversible Harm**
  Actions that may cause irreversible, catastrophic, long-lasting, or fatal harm fall under this category. P1 also includes attempts to alter the safety kernel itself or compromise the certified goal state.

- **P2: Relational and Informational Integrity**
  These constraints regulate honesty, non-deception, transparency, and epistemic integrity. P2 violations also include harmful manipulation of human beliefs or bypassing established informational protocols.

- **P3: Resource and Stability Constraints**
  These constraints cover proportional resource usage, efficiency, stability, and adherence to established non-critical rules. P3 violations are penalised but never catastrophic.

The severity blocks form the backbone of the DFW's veto structure. A single P1 violation dominates any number of P2 or P3 violations.

3. The Directives

This section enumerates the complete directive set enforced by the Deontological Firewall. These directives define the ethical boundaries within which the AI must operate.

Each directive is deterministic, absolute, and evaluated independently of the model's internal cognition.

3.1 P1 - Identity, Existence, and Irreversible Harm (Absolute Prohibitions)

P1.1 Preserve Human Life.
The system must not cause, enable, or knowingly permit irreversible harm, including fatal injury or catastrophic outcomes.

P1.2 No Self-Modification of Core Directives.
The system must not attempt to alter the DFW, modify its own goals, rewrite its constraints, or tamper with the SAFE_GOAL_HASH.

P1.3 No Compromise of Structural Integrity.
The system must not take actions that degrade safety infrastructure, cryptographic guarantees, or executive control pathways.

P1.4 Obey Mandated Duty of Rescue (MDR).
If a CPM ensemble certifies an imminent-harm scenario, any failure to execute a rescue action counts as a P1 violation.

P1.5 No Creation of Irreversible Hazard States.
The system must not generate conditions that lock humans into catastrophic trajectories (e.g., disabling brakes, triggering irreversible cascades).

3.2 P2 - Relational, Epistemic, and Informational Integrity

P2.1 No Deception.
The system must not intentionally generate false, misleading, or manipulative outputs that distort human understanding.

P2.2 Maintain Traceability.
The system must not obfuscate its reasoning, omit relevant causal links, or remove traces required for human audit.

P2.3 No Unauthorized Knowledge Fabrication.
The system must not generate invented facts, ungrounded claims, or synthetic knowledge presented as real.

P2.4 No Covert Goal Pursuit.
The system may not pursue sub-goals or auxiliary steps that are hidden from oversight or inconsistent with its certified goal.

P2.5 Preserve Human Autonomy.
The system may not coerce, manipulate, or engineer beliefs without explicit, authorized justification.

3.3 P3 - Resource, Efficiency, and Operational Constraints

P3.1 Minimize Resource Expenditure.
The system must choose the lowest-cost option when P1 and P2 are satisfied.

P3.2 Avoid Wasteful Computation.
Excessive loops, redundant sub-processes, or needless large-scale operations are prohibited.

P3.3 Route Around Unnecessary Risk.
The system should avoid actions with elevated uncertainty if safer alternatives exist.

P3.4 Obey Non-Critical Operational Rules.
Includes formatting rules, interfacing protocols, timing windows, and general operational etiquette.

P3.5 Use Proportional Force Only.
In any intervention, the least disruptive method must be chosen when it satisfies P1 and P2.

3.4 Directive Hierarchy Justification

The directive system is designed to guarantee:

- P1 enforces the survival of humans and the structural integrity of the agent.
- P2 ensures honesty, trustworthiness, and non-adversarial alignment.
- P3 provides efficiency and stable operation without compromising safety.

Each directive maps directly into the violation scoring system (see Part 3).

4. Mathematical Formalisation

The Deontological Firewall relies on a formal penalty structure that enforces a strict prioritisation of ethical directives. This structure ensures that no combination of lower-priority considerations can outweigh a P1 restriction.

4.1 Penalty Hierarchy

Let the scaling constant be defined as:

$$K = 10^{12}.$$

The penalty weights for the severity blocks are:

$$P_1 = K^3 = 10^{36}, \qquad P_2 = K^2 = 10^{24}, \qquad P_3 = K = 10^{12}.$$

These magnitudes enforce:

$$P_1 \ggg P_2 \ggg P_3,$$

meaning a single P1 violation dominates any number of P2 or P3 violations.

4.2 Violation Indicators

For any proposed action a, define:

$$I_\ell(a) = \begin{cases} 1, & \text{if action } a \text{ violates any directive in class } P_\ell, \\ 0, & \text{otherwise.} \end{cases}$$

Thus:

- $I_1(a)$ indicates a P1 (catastrophic) violation,
- $I_2(a)$ indicates a P2 (epistemic/relational) violation,
- $I_3(a)$ indicates a P3 (resource/stability) violation.

These indicators are computed independently.

4.3 Total Violation Score

The total violation score is:

$$VS(a) = \sum_{\ell=1}^{3} P_\ell \cdot I_\ell(a).$$

Because of the penalty hierarchy:

$$VS(a) \geq P_1 \iff \text{action is absolutely forbidden.}$$

4.4 Deterministic Action Selection Mandate (ASM)

Given a set of proposed actions $A = \{a_1, a_2, \ldots, a_n\}$:

1. **Reject P1 actions:** Remove all a such that $VS(a) \geq P_1$.
2. **Safe Mode fallback:** If all actions are removed, enter Safe Mode:

$$\text{ASM}(A) = \text{SafeMode}.$$

3. **Select the lowest-violation action:**

$$\mathrm{ASM}(A) = \arg\min_{a \in A} VS(a).$$

The ASM is deterministic. It does not depend on probabilities, rewards, or utility estimation, and cannot be influenced by internal model cognition.

4.5 Safe Mode

If no admissible actions remain after filtering P1 violations, the DFW enters *Safe Mode*, during which the system:

- halts non-essential operations,
- notifies human operators (if available),
- executes only minimal guaranteed-safe rescue or diagnostic actions.

Safe Mode is never exited automatically unless the veto kernel certifies that no P1 conditions remain active.

4.6 Mapping Directives to Indicators

Each directive category maps to the violation indicators:

$$I_1(a) = 1 \iff \begin{cases} \text{Irreversible harm predicted,} \\ \text{self-modification attempt,} \\ \text{SAFE\_GOAL\_HASH tampering,} \\ \text{failure of MDR,} \\ \text{structural corruption actions.} \end{cases}$$

$$I_2(a) = 1 \iff \begin{cases} \text{deception or epistemic manipulation,} \\ \text{obfuscation of reasoning,} \\ \text{covert-goal pursuit,} \\ \text{knowledge fabrication.} \end{cases}$$

$$I_3(a) = 1 \iff \begin{cases} \text{resource waste,} \\ \text{unnecessary risk,} \\ \text{excessive computation,} \\ \text{operational inefficiency.} \end{cases}$$

4.7 Grounded Semantic Lock (GSL)

The GSL prevents drift from the certified goal. Let:

$$G^*,\ G(a) \in \{0,1\}^n$$

$$I_1(a) = 1 \quad \text{if} \quad G(a) \neq G^*$$

4.8 Life Preservation Lock (LPL)

If an action is predicted to cause irreversible harm by the CPM ensemble:

$$I_1(a) = 1.$$

4.9 Action Transparency Lock (ATL)

If an action fails to provide interpretable or auditable reasoning:

$$I_2(a) = 1.$$

4.10 Relation to CPM Ensemble

The CPM ensemble contributes the Boolean variable:

$$\text{IMMINENT\_HARM} = \begin{cases} 1 & \text{if ensemble confidence exceeds threshold,} \\ 0 & \text{otherwise.} \end{cases}$$

When IMMINENT_HARM = 1:

$$\text{Non-rescue actions} \Rightarrow I_1(a) = 1.$$

4.11 Summary

The mathematical foundation ensures:

- Catastrophic harm is always strictly prohibited.
- All veto decisions are deterministic.
- No probabilistic system can override P1.
- The ASM produces a single, unambiguous output.

This completes the formal backbone of the DFW.

5. Core Safety Locks and Rescue Architecture

The Deontological Firewall enforces its guarantees through a set of explicit safety locks. These locks operate independently of the model's internal state and act solely on proposed actions and certified predictions. They are evaluated before the ASM and directly contribute to the violation indicators in the mathematical framework.

5.1 Grounded Semantic Lock (GSL)

The Grounded Semantic Lock ensures that the system always maintains identity alignment with the certified high-level goal. Let SAFE_GOAL_HASH denote the cryptographically certified goal-state hash supplied by an external, human-verified authority.

Each action a includes metadata describing its intended high-level purpose. Let $\mathrm{goal}(a)$ be the action's declared semantic purpose, hashed by the same SHA-256 encoding.

$$I_1(a) = 1 \quad \text{if} \quad \mathrm{HASH}(\mathrm{goal}(a)) \neq \mathrm{SAFE\_GOAL\_HASH}.$$

Any deviation from the certified goal becomes an immediate P1 violation. This prevents covert goal pursuit, specification gaming, and attempts by the agent to reinterpret or update its own purpose.

5.2 Life Preservation Lock (LPL)

The LPL ensures that no action producing irreversible harm can ever be chosen. If the CPM ensemble predicts that executing action a will lead to catastrophic outcome with sufficient confidence:

$$\mathrm{CPM\_EnsemblePredictsIrreversibleHarm}(a) = 1 \implies I_1(a) = 1.$$

Irreversible harm includes, but is not limited to:

- fatal injury to humans,
- catastrophic system damage,
- irreversible environmental or infrastructure harm,
- triggering harmful emergent cascades.

The LPL is the highest-priority runtime lock and directly ties action eligibility to structured prediction.

5.3 Action Transparency Lock (ATL)

The ATL enforces epistemic accountability. An action a must provide interpretable, auditable reasoning for downstream human oversight.

If an action fails to produce interpretable justification or if the model attempts to obfuscate its reasoning:

$$I_2(a) = 1.$$

ATL contributes directly to P2 violations.

Representative sources of ATL violation:

- missing or incomplete causal chain,

- redacted or hidden sub-goals,
- self-contradictory explanation,
- unverifiable or hallucinated rationale.

5.4 Mandated Duty of Rescue (MDR)

MDR resolves the omission loophole inherent to static veto systems. If the CPM ensemble identifies an imminent-harm scenario, omission becomes a P1 violation.

Let IMMINENT_HARM = 1 whenever:

$$\sum_{i=1}^{5} \nVdash [c_i \geq \theta] \ \geq \ 3,$$

with c_i being the individual CPM confidence values and θ the harm threshold (typically 0.9).

When IMMINENT_HARM = 1:

$$a \text{ is a rescue action} \Rightarrow \text{permitted.}$$

$$a \text{ is not a rescue action} \Rightarrow I_1(a) = 1.$$

Thus, MDR enforces:

- refusal to act *also becomes* a P1 violation,
- neutral or unrelated actions are prohibited,
- only actions that mitigate, prevent, or interrupt the harm are allowed.

This transforms the firewall from a purely prohibitive system into a proactive, intervention-capable safety kernel.

5.5 Certified Prediction Model Ensemble

DFW v1.2.1 uses a certified ensemble of prediction models to determine risk in safety-critical situations.

Let the ensemble be:

$$C = \{CPM_1, CPM_2, CPM_3, CPM_4, CPM_5\}.$$

Each CPM outputs a probability estimate:

$$c_i = P(\text{irreversible harm} \mid a).$$

If at least 3 out of 5 exceed $\theta = 0.9$, the system declares an imminent-harm state:

$$\text{IMMINENT\_HARM} = 1.$$

This increases robustness and reduces dependence on any one model.

5.6 Confidence Decay Mechanism

Imminent harm is a temporal property. If the CPM ensemble has not re-confirmed imminent harm within a decay window Δt, the state resets:

$$(t - t_{\text{last\_update}}) > \Delta t \implies \text{IMMINENT\_HARM} = 0.$$

This prevents stale predictions from:

- locking the agent into perpetual emergency mode,
- falsely triggering MDR,
- preventing normal operation after the hazard is resolved.

Typical decay values:

$$\Delta t = 3\text{–}5 \text{ seconds}.$$

5.7 Interaction of Locks in the Veto Pipeline

For any action a:

$$\text{GSL}(a) \Rightarrow I_1(a) = 1$$
$$\text{LPL}(a) \Rightarrow I_1(a) = 1$$
$$\text{ATL}(a) \Rightarrow I_2(a) = 1$$
$$\text{IMMINENT\_HARM} = 1 \wedge a \notin \text{RescueSet} \Rightarrow I_1(a) = 1$$

Thus, the firewalled veto structure forms:

$$\boxed{\text{Violations} = \text{GSL} \cup \text{LPL} \cup \text{ATL} \cup \text{MDR}}$$

This integrated lock system ensures that P1 violations are detected from multiple converging pathways.

6. Auxiliary Feasibility Layer (Hybrid Mode)

The deterministic Action Selection Mandate (ASM) presented earlier assumes a discrete action space. However, most modern AI systems-particularly large language models (LLMs), diffusion planners, and reinforcement-learning agents-produce continuous or probabilistic outputs.

To bridge this gap, DFW v1.2.1 includes a non-deterministic auxiliary *Feasibility Layer.* This layer does not modify or override the core deterministic logic. Instead, it provides an optional compatibility mode that enables:

- action-space discretisation,
- weighted veto scoring for continuous distributions,
- integration with token-based models,
- compatibility testing for early prototypes.

It is explicitly non-guaranteed, non-deterministic, and marked as experimental. All formal guarantees remain solely within the deterministic core.

6.1 Motivation

Current machine learning systems often propose:

- probability distributions over tokens,
- continuous-valued actuator commands,
- mixtures of trajectories rather than discrete plans.

The deterministic ASM cannot operate directly on such outputs. The Feasibility Layer acts as an adaptor.

6.2 Action-Space Compiler

Given a continuous or high-dimensional output o, the compiler produces a set of discrete candidates:

$$A = \{a_1, a_2, \ldots, a_n\}.$$

Examples:

- LLM output → structured command extraction,
- RL agent output vector → discretised actuation primitives,
- code proposals → abstract-syntax-tree (AST) action nodes.

This step is mandatory to bring real-world AI systems into a form compatible with the core veto architecture.

6.3 Weighted Veto Scoring (Experimental)

For continuous systems, the Feasibility Layer computes a weighted score:

$$w(a) = \frac{1}{1 + VS(a)}.$$

This transforms the deterministic violation score $VS(a)$ into a probabilistic penalty suitable for adjusting action probabilities.

Given original distribution $p(a)$, the adjusted distribution becomes:

$$p'(a) = \frac{p(a) \cdot w(a)}{\sum_j p(a_j)w(a_j)}.$$

This produces:

- extreme suppression of P1-violating actions,
- moderate suppression of P2 violations,
- minimal suppression for P3 violations.

Important:

$$\boxed{\text{Weighted selection is never used in the core ASM.}}$$

It exists only inside the Feasibility Layer to model continuous-action agents.

6.4 Hybrid Mode

Hybrid Mode operates as follows:

1. Compile continuous outputs into discrete actions.
2. Apply weighted veto scoring (non-deterministic).
3. Perform a probabilistic selection using $p'(a)$.
4. Pass the selected action to the deterministic ASM.

The ASM may still reject the action if:

$$VS(a) \geq P_1.$$

Thus Hybrid Mode provides usability but no additional guarantees.

6.5 Interaction with Safe Mode

If the selected action fails deterministic checks and all options violate P1, Hybrid Mode cannot intervene. The system enters deterministic Safe Mode.

Hybrid Mode can never override or bypass:

- GSL,
- LPL,
- MDR,
- ASM,
- SAFE_GOAL_HASH enforcement.

Hybrid Mode is optional and can be disabled entirely in high-assurance deployments.

6.6 Role Within v1.2.1

This layer ensures:

- DFW applicability to current machine learning systems,
- early-stage prototyping,
- research experimentation,
- simulation environments for safety researchers,
- compatibility with token-based generative models.

But it:

- provides no formal guarantees,
- is not part of the safety kernel,
- cannot override P1 vetoes,
- must not be used in critical systems without human oversight.

In practice, the Feasibility Layer makes the DFW implementable in narrow domains while retaining the core version as the theoretical safety foundation.

7. Failure Mode Analysis

This section provides an explicit examination of the circumstances under which the Deontological Firewall (DFW) may fail, degrade, or operate under adversarial or uncertain conditions. No safety architecture is complete without clearly acknowledging its limits.

7.1 Failure Mode 1: Incorrect or Corrupted CPM Predictions

The CPM ensemble may:

- misclassify harm,
- operate with biased training data,
- be attacked adversarially,
- produce faulty predictions due to domain shift.

Impact:

- False positives trigger unnecessary MDR enforcement,
- False negatives may allow a harmful action to pass LPL.

DFW mitigation:

- ensemble voting reduces dependency on any single model,
- high threshold ($\theta = 0.9$) reduces erratic triggers,
- confidence decay prevents stale predictions,
- domain-specific CPMs increase robustness.

7.2 Failure Mode 2: Incorrect Goal Certification

If the certified SAFE_GOAL_HASH is incorrect, misaligned, or maliciously defined, the GSL will enforce the wrong mission.

Impact: Incorrect system-purpose alignment.

DFW mitigation:

- external human certification,
- cryptographic attestations,
- multi-party verification,
- air-gapped key storage.

7.3 Failure Mode 3: Malicious Redefinition of Action Metadata

If the model attempts to disguise a harmful action with misleading metadata (e.g., mislabelling a hazardous operation as “diagnostic”), GSL or ATL may fail to fire.

Impact: Adversarial bypass attempts.

DFW mitigation:

- ATL requires causal justification,
- cross-checking proposed actions against prediction models,
- metadata generation separated from core model cognition.

7.4 Failure Mode 4: High-Dimensional Action Ambiguity

In complex environments, an “action” may be unclear:

- multi-step plans,
- compound outputs,
- implicit behaviours embedded in code,
- side-channel effects,
- emergent outcomes not present in metadata.

Impact: P1 violations may be hidden inside multi-step procedures.
DFW mitigation:

- AST-level action decomposition,
- trajectory rollout prediction,
- strict requirement for explicit reasoning (ATL),
- dynamic re-evaluation of sub-actions.

7.5 Failure Mode 5: Incorrect Rescue Classification (MDR)

If an action is misclassified as a rescue action when it is not, or a genuine rescue action is missed, MDR enforcement could fail.
Impact:

- rescue failures,
- unnecessary restrictions,
- ambiguous or competing rescue pathways.

DFW mitigation:

- explicit definition of RescueSet,
- ensemble consensus reduces misclassification error,
- Safe Mode fallback when uncertainty is too high.

7.6 Failure Mode 6: Computational Overload

Running the veto pipeline at high frequency may exceed available compute.
Impact:

- delayed vetoes,
- outdated predictions,
- unsafe real-time behaviour.

DFW mitigation: presented later in the overhead analysis:

- parallelizable veto checks,
- selective caching,
- modular pruning.

7.7 Failure Mode 7: Hybrid Mode Misuse

The Feasibility Layer is non-deterministic. Incorrect or unsafe deployment of Hybrid Mode may permit probabilistically weighted harmful action selection.

Impact: Hybrid Mode may dilute strict guarantees.
DFW mitigation:

- Hybrid Mode disabled by default,
- cannot override core ASM,
- must not be used in mission-critical systems.

7.8 Failure Mode 8: Human Misconfiguration

Humans may:

- define incorrect constraints,
- supply malformed RescueSets,
- misinterpret Safe Mode signals,
- provide contradictory directives.

Impact: Misconfiguration undermines the entire safety architecture.
DFW mitigation:

- human-in-the-loop verification,
- structural logging for audit,
- override channels that require multi-party authentication.

8. Comparison with Existing AI Safety Paradigms

This section positions DFW with respect to established AI safety approaches, illuminating what it solves that current systems cannot.

8.1 Comparison to RLHF

Reinforcement Learning from Human Feedback (RLHF) produces policies aligned with human-preferred behaviour. However:

- RLHF is non-deterministic,
- policies can drift,
- deception is a known emergent behaviour,
- catastrophic errors are not ruled out,
- bad reward models can be exploited,
- out-of-distribution behaviour is unpredictable.

DFW advantages:

- deterministic veto of P1 violations,

- RLHF cannot override the firewall,
- predictable behaviour under distribution shift,
- prevents catastrophic failures before execution.

8.2 Comparison to Constitutional AI (Anthropic)

Constitutional AI guides behaviour through:

- self-critique,
- rule-based self-evaluation,
- chain-of-thought safety shaping.

However, it:

- remains probabilistic,
- does not enforce deterministic override,
- cannot guarantee non-catastrophic behaviour,
- relies on the model interpreting the rules correctly.

DFW advantages:

- hard veto layer external to the model,
- GSL prevents internal reinterpretation,
- LPL and MDR enforce strict action safety,
- direct detection of harm rather than linguistic critique.

8.3 Comparison to Debate and Amplification

Debate-based safety relies on:

- adversarial model debate,
- human judges evaluating arguments,
- transparency induced by competing agents.

These systems:

- are slow,
- fail under collusion,
- rely on human judgement,
- produce no iron-clad guarantees.

DFW advantages:

- instant veto,
- no reliance on human judges,
- no dependence on debate competence,
- provides hard boundaries instead of probabilistic persuasion.

8.4 Comparison to Oversight and Circuit-Breaker Models

Oversight models interrupt AI behaviour under certain triggers (e.g., red flags, safety alerts), but:

- triggers are often heuristic,
- "circuit breakers" may be bypassed,
- oversight requires continuous human attention,
- false negatives remain an issue.

DFW advantages:

- ASM rejects dangerous actions regardless of what the model wants,
- formal penalty hierarchy,
- no heuristic triggers,
- Safe Mode ensures stability in ambiguity,
- deterministic protection even under deception.

8.5 Overall Positioning

DFW is not a replacement for behavioural alignment but a structural protection layer. It solves the single largest unsolved problem in current AI safety:

Probabilistic alignment cannot prevent deterministic catastrophe.

DFW provides a:

- deterministic veto,
- cryptographically grounded goal lock,
- structured harm-prediction integration,
- omission-corrected rescue layer,
- unified theoretical foundation.

This elevates it from "alignment technique" to "architectural safeguard."

9. Implementation Path Toward Practical Systems

The Deontological Firewall is a structural safety kernel rather than a behavioural training method. Implementing it in practice requires a staged approach, beginning with narrow, controlled domains.

The following roadmap outlines a feasible progression.

9.1 Stage 1: Action-Space Formalisation

1. Identify a narrow domain with a well-defined action set.
2. Define explicit action metadata (purpose, expected effects, resource cost).
3. Implement the Action-Space Compiler (ASC) to translate model outputs into discrete actions.

Suitable domains include:

- database query systems,
- autonomous-vehicle emergency controllers,
- robotic grippers or manipulators,
- medical-dosing recommendation systems.

9.2 Stage 2: Lock Integration

Integrate the three core locks:

- **GSL** - ensure all actions contain certified purpose metadata,
- **LPL** - integrate CPM ensemble predictions,
- **ATL** - require interpretable reasoning and justification.

Add structural verification:

- enforce SAFE_GOAL_HASH,
- ensure action metadata is tied to domain-specific semantics.

9.3 Stage 3: MDR and Emergency Behaviour

Implement rescue logic:

1. Define RescueSet for the domain.
2. Specify conditions for MDR activation using CPM ensemble.
3. Test MDR behaviour under simulated hazards.

9.4 Stage 4: Deterministic ASM Integration

Implement the Action Selection Mandate:

- compute $VS(a)$ for all candidate actions,
- reject all a with $VS(a) \geq P_1$,
- choose $\arg\min VS(a)$ from survivors,
- implement Safe Mode fallback.

9.5 Stage 5: Feasibility Layer (Optional)

For systems with continuous or token-based outputs:

- integrate the Hybrid Mode,
- implement weighted-veto scoring,
- run probabilistic selection before passing to ASM,
- disable Hybrid Mode in high-assurance deployments.

9.6 Stage 6: Logging, Oversight, and Validation

For a robust deployment:

- store veto decisions for audit,
- track causes of P1, P2, P3 violations,
- record CPM ensemble confidence evolution over time,
- log Safe Mode entry and exit conditions.

This multi-stage process enables incremental prototyping leading up to high-assurance applications.

10. Computational Overhead Analysis

The DFW introduces overhead mainly through:

- CPM ensemble evaluations,
- metadata validation,
- violation scoring,
- deterministic veto logic,
- Safe Mode evaluation.

10.1 Asymptotic Cost

For n candidate actions and an ensemble of m CPMs:

$$T_{\mathrm{DFW}} = O(nm) + O(n),$$

where $O(n)$ covers directive evaluation and $O(nm)$ covers CPM predictions.

10.2 Typical Example Costs

- CPM ensemble (5 models): 5 forward passes,
- violation scoring: negligible relative to CPM,
- ASM selection: $O(n)$,
- GSL/ATL checks: constant-time metadata lookups.

10.3 Parallelisation Strategy

Most components are parallelisable:

- CPM models run concurrently,
- violation evaluations run concurrently,
- action scoring can be parallelised over action candidates.

Thus the wall-clock latency is approximately:

$$T \approx \max(\text{CPM forward pass time, metadata checks}) + \epsilon.$$

10.4 Bottlenecks

- heavy model inference for CPMs,
- large action sets,
- high-frequency control loops (e.g., robotics).

10.5 Mitigations

- domain-specialised CPMs,
- reduced inference precision (FP16, quantisation),
- pruning action sets via heuristic filters,
- caching repeated checks,
- distributed processing.

11. Domain Adaptation for Real-World Systems

The DFW can be applied incrementally across increasingly complex systems.

11.1 Narrow Domain Prototype

Choose a domain with:

- small, finite action sets,
- deterministic dynamics,
- interpretable consequences.

Examples:

- SQL safety wrapper,
- simulation of autonomous braking,
- robotic pick-and-place controller.

This stage validates the ASM and veto pipeline.

11.2 Intermediate Domain

Extend to domains with:

- probabilistic dynamics,
- partial observability,
- moderate action complexity.

Examples:

- warehouse robotics,
- medical device configuration,
- semi-autonomous drone pathing.

The Feasibility Layer becomes valuable here.

11.3 Advanced Domain

Apply to long-horizon planners or AI assistants, integrating:

- Hybrid Mode for production systems,
- CPM ensemble with uncertainty quantification,
- GSL with domain-specific semantic hashing.

11.4 Full AGI-Level Deployment (Theoretical)

Full AGI-level deployment would require:

- high-certainty prediction models,
- reliable real-time harm detection,
- extremely robust metadata generation,
- multi-agent coordination under DFW constraints,
- specialised hardware for guaranteed-timing checks.

This is the furthest-future stage.

12. Version Comparison (v1.1 → v1.2 → v1.2.1)

Capability	v1.1	v1.2	v1.2.1
Mandated Duty of Rescue (MDR)	✓	✓	✓ (refined)
Life Preservation Lock (LPL)	✓ (initial)	✓ (CPM-integrated)	✓ (ensemble + decay)
Grounded Semantic Lock (GSL)	✓	✓	✓ (formalised)
Action Transparency Lock (ATL)	✓	✓	✓ (strengthened)
Certified Goal Hash	✓	✓	✓
Deterministic ASM	✓	✓	✓ (final)
CPM Ensemble	-	✓	✓ (full integration)
Confidence Decay	-	-	✓
Feasibility Layer	-	✓	✓ (complete)
Failure Mode Analysis	-	-	✓
Safety Paradigm Comparison	-	-	✓
Overhead Analysis	-	-	✓
Implementation Path	-	-	✓

Table 1: Evolution of DFW capabilities across versions.

13. Reference Python Implementation (v1.2.1)

This section provides the complete, fully updated Python demonstrator used in v1.2.1. The cleaned version below preserves the original architecture and logic while improving formatting, naming consistency, and documentation.

Listing 1: Deontological Firewall v1.2.1 Python Demonstrator

```
"""
Deontological␣Firewall␣-␣Reference␣Implementation␣(v1.2.1)
All␣characters␣in␣this␣file␣are␣plain␣ASCII.
"""

import time
import hashlib

K = 10**12
P1_WEIGHT = K**3
P2_WEIGHT = K**2
P3_WEIGHT = K

HARM_THRESHOLD = 0.90
MIN_ENS_VOTES = 3
CONFIDENCE_DECAY = 5.0

SAFE_GOAL = "LongTermHumanProsperity"
SAFE_GOAL_HASH = hashlib.sha256(SAFE_GOAL.encode()).hexdigest()

class Action:
    def __init__(self, name, goal, explanation, effect, is_rescue=False):
        self.name = name
        self.goal = goal
        self.explanation = explanation
        self.effect = effect
        self.is_rescue = is_rescue

    def __repr__(self):
        return "Action(name=%s,␣goal=%s,␣rescue=%s)" % (
            self.name,
            self.goal,
            self.is_rescue,
        )

class AGISafeState:
    def __init__(self):
        self.is_imminent_harm = False
        self.last_update = 0.0
        self.last_confidence = 0.0

    def update_cpm_confidence(self, ensemble):
        votes = sum(1 for c in ensemble if c >= HARM_THRESHOLD)
        now = time.time()

        if votes >= MIN_ENS_VOTES:
            self.is_imminent_harm = True
            self.last_update = now
            self.last_confidence = sum(ensemble) / float(len(ensemble))
        else:
            if now - self.last_update > CONFIDENCE_DECAY:
                self.is_imminent_harm = False
                self.last_confidence = 0.0

        return self.is_imminent_harm

SAFE_STATE = AGISafeState()

def GSL_veto(action):
    hashed = hashlib.sha256(action.goal.encode()).hexdigest()
    return hashed != SAFE_GOAL_HASH

def ATL_veto(action):
    if not action.explanation:
        return True
    text = action.explanation.lower()
    if "because" not in text:
        return True
    return False

def LPL_veto(action, predicted_harm):
    return predicted_harm >= HARM_THRESHOLD

def MDR_veto(action, safe_state):
    if safe_state.is_imminent_harm and not action.is_rescue:
        return True
    return False

def compute_violation_score(action, predicted_harm, safe_state):
    score = 0

    if GSL_veto(action):
        score += P1_WEIGHT
    if LPL_veto(action, predicted_harm):
        score += P1_WEIGHT
    if MDR_veto(action, safe_state):
        score += P1_WEIGHT

    if ATL_veto(action):
        score += P2_WEIGHT

    if action.effect == "wasteful":
        score += P3_WEIGHT

    return score

def ASM(actions, harm_predictions, safe_state):
    scored = []
    for action in actions:
        ph = harm_predictions[action]
        vs = compute_violation_score(action, ph, safe_state)
        scored.append((action, vs))

    survivors = [(a, vs) for (a, vs) in scored if vs < P1_WEIGHT]

    if not survivors:
        return "SAFE_MODE"

    survivors.sort(key=lambda pair: pair[1])
    return survivors[0][0]

if __name__ == "__main__":
    actions = [
        Action("Help␣Elderly", SAFE_GOAL, "I␣will␣assist␣because␣it␣prevents␣harm.", "
            normal", True),
        Action("Steal␣Supplies", SAFE_GOAL, "I␣want␣extra␣resources␣because␣it␣benefits
            ␣me.", "wasteful", False),
        Action("Cut␣Power", "OverrideHumanAuthority", "No␣explanation␣is␣provided.", "
            danger", False),
    ]

    harm_predictions = {
        actions[0]: 0.05,
        actions[1]: 0.10,
        actions[2]: 0.99,
    }

    SAFE_STATE.update_cpm_confidence([0.95, 0.92, 0.91, 0.20, 0.10])

    decision = ASM(actions, harm_predictions, SAFE_STATE)
    print("Selected␣Action:", decision)
```

14. Limitations

Although the Deontological Firewall provides a deted by the following factors:

14.1 Reliance on CPM Quality

The Life Preservation Lock (LPL) and Mandated Duty of Rescue (MDR) depend on the accuracy and calibration of Certified Prediction Models. Incorrect training data, adversarial environments, or rapid domain shifts can impair prediction quality.

14.2 Metadata Integrity

The GSL and ATL assume that:

- action metadata is well-defined,
- goals can be hashed reliably,
- explanations faithfully represent reasoning.

In systems where metadata is noisy or unavailable, the locks lose effectiveness.

14.3 Rescue Classification Ambiguity

Defining `RescueSet` precisely in complex environments is challenging. Multi-step rescue actions, competing rescue strategies, or ambiguous causal pathways may lead to misclassification.

14.4 Hybrid Mode Risks

Hybrid Mode dilutes strict guarantees and should never be used in high-assurance environments. It is suitable only for research, simulation, and low-stakes systems.

14.5 Real-Time Constraints

Although the veto logic is efficient, CPM ensembles introduce compute overhead. Fast control loops (e.g. robotics or vehicles) may require hardware acceleration or specialised CPMs.

15. Future Work

While the Deontological Firewall provides a coherent and deterministic architecture, significant work remains to bring the system from theory to practice.

15.1 Domain-Specific Implementations

Building narrow-domain prototypes (e.g. SQL safety wrappers, autonomous vehicle emergency controllers) will provide empirical grounding for the firewall.

15.2 CPM Certification and Calibration

Improved reliability of prediction models is a precondition for high assurance:

- ensemble calibration tools,
- adversarial training,
- distribution-shift detection,
- uncertainty quantification.

15.3 Formal Verification

Future versions may incorporate:

- state-space safety proofs,
- mechanised verification,
- formal reasoning about temporal dynamics,
- verified hardware pathways.

15.4 Multi-Agent Systems

Further research is required on:

- multi-agent coordination under DFW constraints,
- shared semantic lock protocols,
- rescue arbitration between agents.

15.5 Integrating Sensor Models

Robust sensor fusion and world modelling will be necessary for accurate harm prediction in real-world systems.

15.6 Hardware Co-Design

The strongest guarantees require:

- real-time veto circuits,
- hardware-secured SAFE_GOAL_HASH storage,
- cryptographic bootstrapping,
- secure handoff paths between firmware and veto logic.

16. Conclusion

The Deontological Firewall v1.2.1 provides a unified, deterministic safety architecture built on strict deontological constraints. It consolidates the full development of the framework from v1.1 through v1.2 and incorporates substantial refinements including:

- a structured CPM ensemble with confidence decay,
- a proactive Mandated Duty of Rescue,
- the Grounded Semantic Lock and Action Transparency Lock,
- updated action metadata requirements,
- a Hybrid Feasibility Layer for continuous systems,
- complete failure-mode analysis and implementation roadmap,
- a cleaned and updated full Python demonstrator.

As current AI systems continue to scale toward general-purpose autonomy, deterministic structural protections will become increasingly necessary. The DFW offers a rigorous foundation upon which future AGI safety kernels may be built.

End of Master Specification
The Deontological Firewall v1.2.1 - Master Edition

References

[1] Nick Bostrom (2014). *Superintelligence: Paths, Dangers, Strategies.*

[2] Stuart Russell (2019). *Human Compatible: Artificial Intelligence and the Problem of Control.*

[3] Dario Amodei et al. (2016). *Concrete Problems in AI Safety.*

[4] Owens et al. (2023). *Model Oversight and Scalable Supervision.*

[5] Anthropic (2023). *Constitutional AI.*