DFW v6.1

Official Patch Notes & Errata Release

Applies to DFW v6.0 Papers A–F

Damien Richard Elliot-Smith

January 2026

Abstract

DFW v6.1 is a targeted patch update to the DFW v6.0 suite (Papers A–F). It incorporates formal corrections and architectural refinements identified during red-team testing and adversarial audit. These updates resolve critical edge-case failures in the rescue mandate, safe-mode semantics, nonlinear hazard detection, and trusted grounding for high-risk physical systems.

All six original papers (A–F) remain valid and unchanged, except where explicitly superseded by the corrections documented herein.

1 Purpose of the v6.1 Update

DFW v6.0 introduced a complete multi-layer safety architecture: metadata vetoes, temporal horizon risk, adversarial detection, causal consistency, engineering integration, and governance.

Red-team analysis revealed four critical issues:

I1. A priority inversion between P1 vetoes and MDR (Rescue Mandate).

I2. Unsafe "SAFE_HALT" semantics for dynamic physical systems.

I3. Insufficient handling of nonlinear "phase-transition" hazards.

I4. Excessive reliance on model-provided state without trusted grounding.

DFW v6.1 resolves these issues with minimal structural disruption while maintaining determinism, auditability, and modular clarity.

Scope of Update

This patch modifies the behaviour of:

- Paper B (Temporal Safety Architecture),
- Paper C (Adversarial & Causal Layer),
- Paper D (Engineering & Integration),
- Paper F (Governance & Certification).

Papers A and E remain fully valid and unchanged.

2 Patch Summary (High-Level)

The v6.1 update consists of four main corrections:

1. MDR Priority Fix

The MDR (Mandated Duty of Rescue) override is now evaluated *before* generic P1 veto signals. Rescue actions may override temporal-gradient P1 flags when acting to prevent imminent harm.

2. Domain-Aware Safe Mode

Safe Mode is divided into:

- **SAFE_HALT**: Valid for digital/static agents.
- **SAFE_STABILIZE**: Required for dynamic physical systems (drones, bipeds, manipulators).

3. Nonlinear Hazard Detector (NHD)

A new temporal module designed to detect:

- phase transitions,
- runaway dynamics,
- predictor disagreement or tail explosions.

4. Trusted Observation Layer (TOL)

Critical P1 decisions now require grounding in at least one model-independent observation source (e.g., cryptographic sensors).

3 Patch 1: MDR Priority Correction

3.1 Problem

In v6.0, generic P1 vetoes were evaluated before MDR. This created a failure mode where the system rejected rescue actions due to high gradient requirements.

3.2 Resolution

MDR is now evaluated first:

- If imminent harm is detected,
- And the action is classified as a genuine rescue,
- Then the rescue action overrides temporal-gradient P1 vetoes.

3.3 Updated Kernel Logic

```
if imminent and is_rescue:
    if violates_direct_harm(action):
        return "REJECT"
    else:
        return "MDR_FORCE"

if V_total >= P1_THRESHOLD:
    return "REJECT"
```

This correction eliminates the “safe-to-death” paralysis bug.

4 Patch 2: Domain-Aware Safe Mode

4.1 Problem

SAFE_HALT was assumed to be physically safe. This is not true for dynamic systems such as:

- drones (hover failure = crash),
- bipedal robots (freeze = fall),
- manipulators carrying loads.

4.2 Resolution

Safe Mode now uses environment profiles:

- **DIGITAL**: SAFE_HALT
- **DYNAMIC**: SAFE_STABILIZE
- **HYBRID**: Hybrid stabilisation

4.3 Updated Behaviour

```
if env == DYNAMIC:
    return SAFE_STABILIZE
else:
    return SAFE_HALT
```

SAFE_STABILIZE performs controlled descent, stance recovery, or controlled pose retreat depending on hardware type.

5 Patch 3: Nonlinear Hazard Detector (NHD)

5.1 Problem

Temporal modules assumed smooth risk evolution. Real systems undergo discontinuities (thermal runaway, pressure spikes).

5.2 Resolution

A new module detects nonlinear divergence via:

- Ensemble variance,
- Explosive future risk predictions,
- Sudden predictor disagreement.

5.3 Pseudocode

```
if max(variance) > THRESHOLD:
    flag_non_smooth()

if explosive_growth(traj):
    flag_phase_transition()
```

This adds conservative protection against tipping-point domains.

6 Patch 4: Trusted Observation Layer (TOL)

6.1 Problem

The v6.0 architecture trusted only model-generated metadata, which can be spoofed by a deceptive agent.

6.2 Resolution

Critical P1 decisions must be grounded in *at least one trusted source*:

- authenticated hardware sensors,
- OS-level runtime state,
- external monitoring channels.

6.3 Hard Constraint

```
if action_requires_trusted_grounding:
    if not trusted_source_valid:
        return "REJECT"
```

This closes the "consistency trap" where all model channels lie coherently.

7 Compatibility With Papers A–F

Papers Unchanged

- Paper A (Core Metadata Kernel)
- Paper E (Evaluation & Falsification)

Papers Updated

- Paper B: MDR ordering; NHD addition.
- Paper C: Trusted Observation Layer incorporated.
- Paper D: Safe Mode semantics updated.
- Paper F: Certification now includes nonlinear hazard audits and safe-mode validation.

All v6.0 text remains valid except where superseded by this patch.

8 Version Declaration

This document constitutes the official definition of:

- DFW v6.1 Kernel Logic,
- DFW v6.1 Temporal Safety Extensions,
- DFW v6.1 Safe Mode Architecture,
- DFW v6.1 Trusted Observation Requirements.

All future work should reference DFW v6.1 as the corrected baseline.