NTRL — Machine■Readable Taxonomy & Admin API Contract (v1.3)

Production System · Authoritative Interface Specification

1. Purpose

This document defines the canonical, machine■readable taxonomy and Admin API contract for NTRL.

It ensures multiple engineers and systems can safely modify categorization behavior without introducing entropy or editorial drift.

2. Taxonomy Versioning Model

All taxonomy artifacts are versioned. No silent changes are permitted.

Every article stores the taxonomy_version used at categorization time.

```
{
  "taxonomy_version": "1.3.0",
  "effective_date": "2026-01-27",
  "previous_version": "1.2.0",
  "change_summary": "Initial production taxonomy and API contract"
}
```

3. Feed Category Schema

Feed categories are user-facing and configurable.

```
{
  "id": "technology",
  "label": "Technology",
  "enabled_by_default": true,
  "user_toggleable": true,
  "order": 4
}
```

4. Internal Domain Schema

Internal domains are system-only and exhaustive.

```
{
  "id": "law_justice",
  "label": "Law & Justice",
  "description": "Courts, litigation, regulation, and enforcement",
  "active": true
}
```

5. Domain → Feed Category Mapping

Mappings are deterministic and admin-controlled.

```json
{
  "domain_id": "law_justice",
  "feed_category_id": "united_states",
  "conditions": {
    "geography": "US"
  }
}
```

6. Tagging Axes Schema

Tags are structured across fixed axes.

```json
{
  "domain": "law_justice",
  "geography": {
    "scope": "national",
    "country": "US"
  },
  "actors": ["Supreme Court"],
  "action_type": "ruled",
  "temporal_state": "new",
  "evidence_state": "confirmed",
  "impact_scope": "national",
  "quantitative_signals": []
}
```

7. Admin API Endpoints

The Admin API is the sole interface for mutating taxonomy and categorization behavior.

Method	Endpoint	Purpose
GET	/admin/taxonomy	Fetch current taxonomy
POST	/admin/taxonomy	Create new taxonomy version
PUT	/admin/feed-categories	Update categories
PUT	/admin/domain-mapping	Update domain mappings
GET	/admin/review-queue	Fetch items needing review
POST	/admin/override	Apply categorization override

8. Override Contract

Overrides require explicit justification and are immutable.

```json
{
  "story_id": "abc123",
  "previous_category": "technology",
```

```
    "new_category": "united_states",
    "admin_id": "admin@ntrl",
    "reason": "Legal ruling scope is national, not technological",
    "timestamp": "2026-01-27T14:03:00Z"
}
```

9. Rollback & Safety

Taxonomy versions may be rolled back, but historical articles retain their original taxonomy_version.

Reprocessing content requires an explicit batch job and version bump.

10. Design Outcomes

This contract enables parallel engineering, auditability, and long-term stability.

It closes the loop between product intent, admin control, and backend execution.