Contents

Provider Rejection Handling & Intelligent Fallback

Version: 4.18.3
Last Updated: 2024-12-28

Overview

When an AI provider or self-hosted model rejects a prompt based on their ethics policies (that don't conflict with RADIANT's ethics), the system automatically attempts fallback to alternative models. If all capable models reject the request, the user receives a clear explanation.

How It Works

Rejection Flow

```
1. User submits prompt
2. AGI Brain selects optimal model
3. Model rejects prompt (provider ethics, content policy, etc.)
4. System checks: Does this violate OUR ethics?
     YES → Reject to user with explanation
     NO → Try fallback models
5. Fallback loop (max 3 attempts):
     Select model with lowest rejection rate
     Attempt request
     If success → Return response
6. If all fallbacks fail:
     Reject to user with detailed explanation
```

Key Principles

1. **RADIANT ethics take precedence** - If our ethics block it, no fallback is attempted
2. **Provider ethics don't block us** - Different providers have different policies; we route around them
3. **Users always know** - Every rejection is explained to the user
4. **Learning from patterns** - The system learns which models reject which types of content

---

Rejection Types

Type	Description	Fallback?
`content_policy`	Provider's content policy violation	Yes
`safety_filter`	Safety/moderation filter triggered	Yes
`provider_ethics`	Provider's ethical guidelines differ	Yes
`capability_mismatch`	Model can't handle this request type	Yes
`context_length`	Prompt too long for model	Yes
`moderation`	Pre-flight moderation blocked	Yes
`rate_limit`	Rate limiting (retry later)	Retry
`unknown`	Unknown error	Yes

Fallback Model Selection

Models are selected for fallback based on:

1. **Rejection rate** - Models with lowest historical rejection rates preferred
2. **Required capabilities** - Must have same capabilities as original
3. **Exclusion list** - Previously tried models excluded
4. **Provider diversity** - Prefer different providers for better success chance

Selection Query

```sql
SELECT model_id, provider_id, rejection_rate
FROM unified_model_registry m
LEFT JOIN model_rejection_stats s ON m.model_id = s.model_id
WHERE m.enabled = true
  AND m.model_id != ALL(excluded_models)
  AND m.capabilities && required_capabilities
ORDER BY COALESCE(s.rejection_rate, 0) ASC
LIMIT 10
```

User Notifications

Notification Types

Rejected Request:

```
Title: Request Could Not Be Completed
Message: The ethical guidelines of available AI providers prevented
         this response. We attempted 3 different AI models.
Suggested Actions:
  - Try rephrasing your request
  - Remove potentially sensitive content
  - Contact administrator
```

Resolved with Fallback:

```
Title: Resolved with Alternative Model
Message: Your request was processed by an alternative AI model
         after the original was unavailable.
```

Think Tank UI

- **Bell icon** with unread count in toolbar
- **Sheet panel** slides out showing all notifications
- **Rejection banners** appear in conversation when relevant
- **Suggested actions** are clickable to help users resolve issues

---

Database Schema

provider_rejections

Column	Type	Description
`id`	UUID	Primary key
`tenant_id`	UUID	Multi-tenant isolation
`user_id`	UUID	User who made request
`plan_id`	UUID	AGI Brain plan if applicable
`model_id`	VARCHAR	Model that rejected

Column	Type	Description
`provider_id`	VARCHAR	Provider ID
`rejection_type`	VARCHAR	Type of rejection
`rejection_message`	TEXT	Raw error from provider
`radiant_ethics_passed`	BOOLEAN	Did it pass our ethics?
`fallback_attempted`	BOOLEAN	Was fallback tried?
`fallback_model_id`	VARCHAR	Model that succeeded
`fallback_succeeded`	BOOLEAN	Did fallback work?
`fallback_chain`	JSONB	Array of all attempts
`final_status`	VARCHAR	pending, fallback_success, rejected
`final_response_to_user`	TEXT	Message shown to user

rejection_patterns

Learns patterns for smarter fallback:

Column	Type	Description
`pattern_hash`	VARCHAR	Hash of rejection characteristics
`trigger_keywords`	TEXT[]	Keywords that trigger rejections
`trigger_model_ids`	TEXT[]	Models that reject this pattern
`recommended_fallback_models`	TEXT[]	Models that work
`success_rate`	NUMERIC	Fallback success rate

model_rejection_stats

Per-model rejection statistics:

Column	Type	Description
`model_id`	VARCHAR	Model identifier
`total_requests`	INTEGER	Total requests to model
`total_rejections`	INTEGER	Total rejections
`rejection_rate`	NUMERIC	Computed rejection rate
`content_policy_count`	INTEGER	By type breakdown
`fallback_successes`	INTEGER	Successful fallbacks

API Endpoints

Record Rejection

```
POST /api/internal/rejections
{
  "modelId": "gpt-4",
  "providerId": "openai",
  "rejectionType": "content_policy",
  "rejectionMessage": "Content policy violation",
  "planId": "uuid"
}
```

Get User Notifications

```
GET /api/thinktank/rejections
Response: {
  "notifications": [...],
  "unreadCount": 3
}
```

Mark Notification Read

```
PATCH /api/thinktank/rejections/:id/read
```

Dismiss Notification

```
DELETE /api/thinktank/rejections/:id
```

Service Integration

ProviderRejectionService

```
// Handle rejection with automatic fallback
const result = await providerRejectionService.handleRejectionWithFallback(
  tenantId,
  userId,
  originalModelId,
  providerId,
  rejectionType,
  rejectionMessage,
  async (modelId, providerId) => {
    // Execute request with fallback model
    return await executeRequest(modelId, providerId, prompt);
  },
  planId,
  requiredCapabilities
);

if (result.success) {
  // Request succeeded (possibly with fallback)
  console.log('Handled by:', result.handlingModelId);
  console.log('Used fallback:', result.usedFallback);
} else {
  // All models rejected
  console.log('Rejection reason:', result.rejectionReason);
  console.log('User message:', result.userFacingMessage);
}
```

AGI Brain Integration

The AGI Brain Planner automatically uses rejection handling:

1. Selects optimal model for task
2. If model rejects, calls `providerRejectionService.handleRejectionWithFallback()`
3. If fallback succeeds, continues with new model
4. If all fail, returns rejection response to user

Configuration

Constants

```
const MIN_MODELS_FOR_TASK = 2;    // Minimum models needed
const MAX_FALLBACK_ATTEMPTS = 3; // Maximum fallback tries
```

Model Rejection Thresholds

Models with rejection rates above 30% are deprioritized for initial selection but may still be used as fallbacks.

Admin Dashboard

Rejection Analytics

Location: Admin Dashboard → Analytics → Rejections

Full analytics dashboard for monitoring rejections and informing policy updates.

Summary Cards

- **Total Rejections (30d)** - All rejections in period
- **Fallback Success Rate** - Percentage resolved via fallback
- **Rejected to User** - Requests that failed all fallbacks
- **Flagged Keywords** - Keywords marked for policy review

Tabs

Tab	Purpose
By Provider	See which providers reject most, rejection types, fallback rates
Violation Keywords	Keywords triggering rejections, per-provider breakdown
Flagged Prompts	Full prompt content for policy investigation
Policy Review	Recommendations for pre-filters based on patterns

Viewing Full Prompt Content

Administrators can view the complete rejected prompt to understand why it was rejected:

1. Go to Analytics → Rejections → Flagged Prompts
2. Click "View Full Prompt" on any entry
3. Review detected keywords and rejection reason
4. Decide: Add Pre-Filter, Add Warning, or Dismiss

Adding Pre-Filters

Based on rejection patterns, add pre-filters to RADIANT's ethics:

1. Identify high-frequency rejection keywords
2. Flag keywords for review
3. Investigate sample prompts
4. Add pre-filter rule to block before sending to AI

Database Views

View	Purpose
`rejection_summary_by_provider`	Aggregated stats per provider
`rejection_summary_by_model`	Aggregated stats per model
`top_rejection_keywords`	Most frequent violation keywords

Related Documentation

- AGI Brain Plan System
- AI Ethics Standards
- Model Router Service