Contents

RADIANT Specialty Ranking System 1
- Overview . . . 1
- Architecture . . . 2
- 20 Specialty Categories . . . 2
  - Domain Expertise . . . 2
  - Task Capabilities . . . 2
  - Modalities . . . 3
  - Performance Attributes . . . 3
- Tier System . . . 3
- Specialty Ranking Data Structure . . . 4
- Mode Rankings . . . 4
  - Orchestration Modes . . . 5
- Model Specialty Profiles . . . 5
  - Claude 3.5 Sonnet . . . 5
  - OpenAI o1 . . . 5
  - DeepSeek Coder . . . 6
  - GPT-4o . . . 6
  - Gemini 2.0 Flash . . . 7
- AI-Powered Research . . . 7
  - Research Process . . . 7
  - Research API . . . 8
  - Research Schedule . . . 8
- Admin Controls . . . 9
  - Admin Dashboard . . . 9
  - Admin API . . . 9
- Integration with Orchestration . . . 9
  - Brain Router Integration . . . 9
  - AGI Brain Planner Integration . . . 10
  - Combined Scoring Example . . . 10
- Database Schema . . . 10
- Related Documentation . . . 12
- API Reference . . . 12
  - Endpoints . . . 12

RADIANT Specialty Ranking System

Version: 4.18.0 Last Updated: 2024-12-28

Overview

The Specialty Ranking System is RADIANT's **AI-powered proficiency ranking** for models and orchestration modes. It provides domain-specific expertise scores that drive intelligent model selection.

Architecture

```
                         SPECIALTY RANKING SYSTEM



    AI Research             Specialty              Model
    Service                   Rankings DB              Selection




   • Benchmarks              • Per-model scores     • Brain Router
   • Community Reviews       • Per-specialty        • AGI Planner
   • Internal Data           • Tiered (S-F)         • Orchestration
```

---

20 Specialty Categories

Models are ranked across 20 specialty categories representing domain-specific expertise:

Domain Expertise

Category	Icon	Description	Example Tasks
`medical`		Medical & Healthcare	Diagnosis, treatment, clinical guidelines
`legal`		Legal & Compliance	Contract review, legal research, compliance
`finance`		Finance & Trading	Financial analysis, trading strategies
`science`		Scientific	Research methodology, scientific writing
`security`		Cybersecurity	Vulnerability analysis, security audits
`architecture`		System Architecture	System design, scalability planning

Task Capabilities

Category	Icon	Description	Example Tasks
`reasoning`		Reasoning & Logic	Complex deduction, logical analysis

Category	Icon	Description	Example Tasks
`coding`		Code Generation	Programming, debugging, refactoring
`math`		Mathematics	Calculations, proofs, statistics
`creative`		Creative Writing	Stories, poetry, marketing copy
`analysis`		Data Analysis	Data interpretation, patterns
`research`		Research & Synthesis	Literature review, synthesis
`debugging`		Debugging & QA	Bug finding, test generation
`conversation`		Conversational	Natural dialogue, engagement

Modalities

Category	Icon	Description	Example Tasks
`vision`		Vision & Images	Image analysis, OCR, diagrams
`audio`		Audio & Speech	Transcription, voice analysis

Performance Attributes

Category	Icon	Description	Example Tasks
`speed`		Low Latency	Real-time responses
`accuracy`		High Accuracy	Fact-critical tasks
`safety`		Safety & Alignment	Sensitive content handling
`instruction`		Instruction Following	Complex multi-step tasks

Tier System

Each model receives a tier rating (S-F) for each specialty:

Tier	Score Range	Description	Use Case
S	95-100	Elite - Best-in-class	Primary selection for this specialty
A	85-94	Excellent - Highly recommended	Strong choice, reliable
B	75-84	Good - Solid performance	Acceptable, cost-effective
C	65-74	Average - Acceptable	Use if better unavailable
D	50-64	Below Average - Use with caution	Fallback only
F	0-49	Poor - Not recommended	Do not use

Specialty Ranking Data Structure

```
interface SpecialtyRanking {
  rankingId: string;
  modelId: string;                    // e.g., 'anthropic/claude-3-5-sonnet'
  provider: string;                   // e.g., 'anthropic'
  specialty: SpecialtyCategory;       // e.g., 'medical', 'coding'

  // Scores (0-100)
  proficiencyScore: number;           // Overall weighted score
  benchmarkScore: number;             // From published benchmarks
  communityScore: number;             // From community reviews
  internalScore: number;              // From internal usage data

  // Rankings
  rank: number;                       // Global rank for this specialty
  percentile: number;                 // e.g., 95 = top 5%
  tier: 'S' | 'A' | 'B' | 'C' | 'D' | 'F';

  // Metadata
  confidence: number;                 // 0-1 confidence in assessment
  dataPoints: number;                 // Number of data points used
  lastResearched: string;             // ISO timestamp
  researchSources: string[];          // Sources used
  trend: 'improving' | 'stable' | 'declining';

  // Admin
  adminOverride?: number;             // Locked admin score
  isLocked: boolean;                  // Whether ranking is locked
  updatedAt: string;
}
```

Mode Rankings

In addition to specialty rankings, models are ranked for each **orchestration mode**:

```
interface ModeRanking {
  rankingId: string;
  mode: OrchestrationMode;            // e.g., 'extended_thinking', 'coding'
  modelId: string;
  provider: string;
  score: number;
  tier: 'S' | 'A' | 'B' | 'C' | 'D' | 'F';
  strengths: string[];                // What this model excels at
  weaknesses: string[];               // Where it falls short
  recommendedFor: string[];           // Task types recommended for
  notRecommendedFor: string[];        // Task types to avoid
  confidence: number;
  isLocked: boolean;
  adminOverride?: number;
  updatedAt: string;
}
```

Orchestration Modes

Mode	Icon	Description
`thinking`		Standard reasoning with step-by-step analysis
`extended_thinking`		Deep multi-step reasoning for complex problems
`research`		Information gathering and synthesis
`creative`		Divergent thinking and idea generation
`analytical`		Data analysis and pattern recognition
`coding`		Code generation and debugging
`conversational`		Natural dialogue and engagement
`fast`		Quick responses with minimal latency
`precise`		High accuracy with verification
`balanced`		Optimal cost/quality/speed tradeoff

Model Specialty Profiles

Claude 3.5 Sonnet

```
Specialty Scores (0-100):
  reasoning:     94 (S)
  coding:        95 (S)
  math:          88 (A)
  creative:      92 (A)
  analysis:      91 (A)
  research:      90 (A)
  medical:       92 (A)
  legal:         89 (A)
  finance:       88 (A)
  security:      91 (A)
  vision:        93 (A)
  safety:        95 (S)
  speed:         75 (B)

Best For: General-purpose, coding, creative, research, analysis
Mode Recommendations: thinking, extended_thinking, creative, research
```

OpenAI o1

```
Specialty Scores (0-100):
  reasoning:     98 (S)
  coding:        90 (A)
  math:          96 (S)
  creative:      75 (B)
  analysis:      94 (S)
  research:      88 (A)
  medical:       85 (B)
  legal:         88 (A)
  finance:       91 (A)
  security:      89 (A)
  safety:        92 (A)
  speed:         60 (D)

Best For: Complex reasoning, mathematics, analysis, multi-step problems
Mode Recommendations: extended_thinking, analytical, precise
```

DeepSeek Coder

```
Specialty Scores (0-100):
  reasoning:     85 (B)
  coding:        96 (S)
  math:          92 (A)
  creative:      65 (C)
  analysis:      82 (B)
  debugging:     94 (S)
  architecture: 88 (A)
  security:      85 (B)
  speed:         90 (A)

Best For: Code generation, debugging, system design
Mode Recommendations: coding, fast
```

GPT-4o

```
Specialty Scores (0-100):
  reasoning:     90 (A)
  coding:        88 (A)
  math:          85 (B)
  creative:      88 (A)
  analysis:      86 (A)
  research:      87 (A)
  vision:        95 (S)
  audio:         92 (A)
  conversation: 91 (A)
  speed:         88 (A)

Best For: Multimodal tasks, vision, audio, conversation
Mode Recommendations: conversational, fast, balanced
```

Gemini 2.0 Flash

```
Specialty Scores (0-100):
  reasoning:     82 (B)
  coding:        80 (B)
  math:          78 (B)
  analysis:      80 (B)
  research:      82 (B)
  vision:        85 (B)
  speed:         98 (S)
  conversation: 85 (B)

Best For: Fast responses, real-time applications
Mode Recommendations: fast, conversational
```

AI-Powered Research

The specialty rankings are maintained through **automated AI research**:

Research Process

```
    SPECIALTY RANKING RESEARCH FLOW




    STEP 1: Gather Data Sources
    • Published benchmarks (MMLU, HumanEval, MATH, GPQA, etc.)
    • Community reviews (Reddit, Twitter, Discord)
    • Academic papers and evaluations
    • Internal usage data and quality scores




    STEP 2: AI Analysis
    • Claude 3.5 Sonnet analyzes all sources
    • Generates per-specialty proficiency scores
    • Assigns tier ratings (S/A/B/C/D/F)
    • Calculates confidence levels




    STEP 3: Score Calculation
```

```
proficiencyScore = (benchmarkWeight × benchmarkScore) +
                   (communityWeight × communityScore) +
                   (internalWeight × internalScore)

Default Weights: benchmark=0.5, community=0.3, internal=0.2
```

```
STEP 4: Update Rankings
• Update specialty_rankings table
• Recalculate global ranks per specialty
• Calculate percentiles
• Record research log
```

Research API

```
// Research a specific model across all specialties
const result = await specialtyRankingService.researchModelProficiency(
  'anthropic/claude-3-5-sonnet'
);
// Returns: { modelsResearched: 1, specialtiesUpdated: 20, rankingsChanged: 20 }

// Research all models for a specific specialty
const result = await specialtyRankingService.researchSpecialtyRankings('medical');
// Returns: { modelsResearched: 50, specialtiesUpdated: 1, rankingsChanged: 45 }

// Get leaderboard for a specialty
const leaderboard = await specialtyRankingService.getSpecialtyLeaderboard('coding', 10);
// Returns: { specialty: 'coding', rankings: [{ rank: 1, modelId: '...', score: 96, tier: 'S' }

// Get best model for a specialty
const best = await specialtyRankingService.getBestModelForSpecialty('medical', { minScore: 85 }
// Returns: { modelId: 'anthropic/claude-3-5-sonnet', score: 92, tier: 'A' }
```

Research Schedule

```
interface ResearchSchedule {
  scheduleId: string;
  name: string;
  frequency: 'hourly' | 'daily' | 'weekly' | 'monthly' | 'manual';
  cronExpression?: string;
  enabled: boolean;
  lastRun?: string;
  nextRun?: string;
  targetScope: 'all' | 'specialty' | 'mode' | 'model';
  targetFilter?: string;
}

// Example schedules:
// - Daily research for new models
// - Weekly refresh of all specialty rankings
// - Monthly deep research with expanded sources
```

Admin Controls

Admin Dashboard

Path: Admin Dashboard → Orchestration → Specialty Rankings

Features: - **Leaderboards**: View top models per specialty - **Model Profiles**: See all specialty scores for a model - **Override Scores**: Lock a model's specialty score - **Trigger Research**: Manually refresh rankings - **Configure Weights**: Adjust scoring weights

Admin API

```
// Override a ranking (locks it from research updates)
await specialtyRankingService.adminOverrideRanking(
  'anthropic/claude-3-5-sonnet',
  'medical',
  95,  // New score
  'Internal evaluation showed higher medical accuracy'
);

// Unlock a ranking (allows research to update it again)
await specialtyRankingService.unlockRanking('anthropic/claude-3-5-sonnet', 'medical');

// Get model rankings
const rankings = await specialtyRankingService.getModelRankings('anthropic/claude-3-5-sonnet')
```

Integration with Orchestration

Brain Router Integration

The Brain Router uses specialty rankings for model selection:

```
// In brain-router.ts
const bestMedicalModel = await specialtyRankingService.getBestModelForSpecialty('medical', {
  minScore: 85,
  excludeModels: disabledModels
});

// Factor specialty score into routing decision
const domainMatchScore = await getSpecialtyScore(modelId, detectedSpecialty);
const finalScore = costScore * 0.3 + latencyScore * 0.2 + qualityScore * 0.3 + domainMatchScore
```

AGI Brain Planner Integration

The AGI Brain Planner uses specialty rankings to select models:

```
// In agi-brain-planner.service.ts
const { primary, fallbacks } = await this.selectModels(
  tenantId,
  promptAnalysis,
  domainResult,      // Contains detected domain/subspecialty
  orchestrationMode
);

// Models are selected based on:
// 1. Domain proficiency match (from domain taxonomy)
// 2. Specialty rankings (from specialty ranking service)
// 3. Mode rankings (how well model performs in the chosen mode)
```

Combined Scoring Example

```
Prompt: "Review this contract for liability issues"

Domain Detection:
  Field: Law → Domain: Contract Law → Subspecialty: Commercial Contracts
  Confidence: 0.89

Required Specialties: legal, accuracy, reasoning

Model Scoring:

  Model                  Legal    Accuracy   Reasoning  Combined

  Claude 3.5 Sonnet     89 (A)   91 (A)     94 (S)      91.3
  GPT-4o                85 (B)   88 (A)     90 (A)      87.7
  OpenAI o1             88 (A)   92 (A)     98 (S)      92.7


Selected: OpenAI o1 (highest combined score for legal + reasoning)
```

Database Schema

```
-- Specialty rankings table
CREATE TABLE specialty_rankings (
  ranking_id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  model_id TEXT NOT NULL,
  provider TEXT NOT NULL,
  specialty TEXT NOT NULL,
  proficiency_score NUMERIC(5,2) NOT NULL,
  benchmark_score NUMERIC(5,2),
```

```sql
  community_score NUMERIC(5,2),
  internal_score NUMERIC(5,2),
  rank INTEGER,
  percentile NUMERIC(5,2),
  tier TEXT NOT NULL CHECK (tier IN ('S', 'A', 'B', 'C', 'D', 'F')),
  confidence NUMERIC(3,2) DEFAULT 0.80,
  data_points INTEGER DEFAULT 0,
  last_researched TIMESTAMPTZ,
  research_sources TEXT[],
  trend TEXT DEFAULT 'stable' CHECK (trend IN ('improving', 'stable', 'declining')),
  admin_override NUMERIC(5,2),
  admin_notes TEXT,
  is_locked BOOLEAN DEFAULT false,
  created_at TIMESTAMPTZ DEFAULT NOW(),
  updated_at TIMESTAMPTZ DEFAULT NOW(),
  UNIQUE(model_id, specialty)
);

-- Mode rankings table
CREATE TABLE mode_rankings (
  ranking_id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  mode TEXT NOT NULL,
  model_id TEXT NOT NULL,
  provider TEXT NOT NULL,
  score NUMERIC(5,2) NOT NULL,
  tier TEXT NOT NULL CHECK (tier IN ('S', 'A', 'B', 'C', 'D', 'F')),
  strengths TEXT[],
  weaknesses TEXT[],
  recommended_for TEXT[],
  not_recommended_for TEXT[],
  confidence NUMERIC(3,2) DEFAULT 0.80,
  admin_override NUMERIC(5,2),
  is_locked BOOLEAN DEFAULT false,
  created_at TIMESTAMPTZ DEFAULT NOW(),
  updated_at TIMESTAMPTZ DEFAULT NOW(),
  UNIQUE(mode, model_id)
);

-- Research logs
CREATE TABLE specialty_research_logs (
  log_id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  research_type TEXT NOT NULL, -- 'model', 'specialty', 'full'
  target_id TEXT,
  models_researched INTEGER,
  specialties_updated INTEGER,
  rankings_changed INTEGER,
  duration_ms INTEGER,
  ai_confidence NUMERIC(3,2),
  sources_used TEXT[],
  created_at TIMESTAMPTZ DEFAULT NOW()
);
```

Related Documentation

- Orchestration Methods - Complete orchestration system documentation
- Domain Taxonomy - Domain detection and proficiency system
- AGI Brain Planner - Real-time planning system
- Model Router - Intelligent model selection

API Reference

Endpoints

Method	Path	Description
GET	`/api/admin/specialty-rankings`	List all rankings
GET	`/api/admin/specialty-rankings/:modelId`	Get model's rankings
GET	`/api/admin/specialty-rankings/specialty/:specialty`	Get specialty leaderboard
POST	`/api/admin/specialty-rankings/research/model/:modelId`	Research model
POST	`/api/admin/specialty-rankings/research/specialty/:specialty`	Research specialty
PATCH	`/api/admin/specialty-rankings/:modelId/:specialty`	Override ranking
DELETE	`/api/admin/specialty-rankings/:modelId/:specialty/lock`	Unlock ranking
GET	`/api/admin/mode-rankings`	List mode rankings
GET	`/api/admin/mode-rankings/:mode`	Get mode leaderboard