Contents

RAWS v1.1 User Documentation

API Guide for Developers and Integrators

Document Version: 1.1.0
API Version: v1
Last Updated: January 2026

---

1. Introduction

RAWS (RADIANT AI Weighted Selection) automatically selects the optimal AI model for your requests based on:

- **Quality**: How accurate the model is
- **Cost**: Price for your usage
- **Latency**: Response speed
- **Capabilities**: Features supported
- **Compliance**: Regulatory certifications

Why Use RAWS?

Without RAWS	With RAWS
Manually choose models	Automatic optimization
Risk compliance violations	Compliance-aware filtering
Static selection	Dynamic, context-aware
No fallback handling	Automatic fallback chain

---

2. Quick Start

```
curl -X POST https://api.radiant.example.com/v1/raws/select \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{
    "requiredCapabilities": ["chat", "streaming"],
    "estimatedInputTokens": 1000,
    "estimatedOutputTokens": 500
  }'
```

---

3. Domain-Specific Selection

RAWS supports 7 domains, each with appropriate compliance requirements:

3.1 Domain Overview

Domain	Use Case	Compliance	Min Quality
`healthcare`	Medical, clinical	HIPAA required	80
`financial`	Investment, accounting	SOC 2 required	75
`legal`	Contracts, litigation	SOC 2 required	80
`scientific`	Research, academic	Varies	70
`creative`	Content, marketing	None	-
`engineering`	Code, software	Varies	70
`general`	Default	None	-

3.2 Healthcare Domain

When to Use: Medical queries, patient data, clinical documentation

Compliance: HIPAA is **mandatory**. All models must be HIPAA-certified.

What Happens: - Only HIPAA-compliant models considered - Minimum quality score of 80 enforced - System 2 reasoning forced (no fast/cheap models) - Truth Engine verification required (ECD 0.05)

```
{
  "requiredCapabilities": ["chat", "tool_use"],
  "estimatedInputTokens": 2000,
  "estimatedOutputTokens": 1500,
  "domain": "healthcare"
}
```

Typical Models Selected: claude-sonnet-4-5, gpt-4o, gemini-2.5-pro (all HIPAA-certified)

3.3 Financial Domain

When to Use: Investment analysis, accounting, financial reporting, tax

Compliance: SOC 2 Type II is **mandatory**.

What Happens: - Only SOC 2 certified models considered - Minimum quality score of 75 enforced - System 2 reasoning forced - Truth Engine verification required (ECD 0.05)

```
{
  "requiredCapabilities": ["chat", "function_calling"],
  "estimatedInputTokens": 2000,
  "estimatedOutputTokens": 1500,
  "domain": "financial"
}
```

3.4 Legal Domain

When to Use: Contract analysis, legal research, compliance documentation

Compliance: SOC 2 Type II is **mandatory**. Source citations required.

What Happens: - Only SOC 2 certified models considered - Minimum quality score of 80 enforced - System 2 reasoning forced - Source citation verification enabled - Truth Engine required (ECD 0.05)

```
{
  "requiredCapabilities": ["chat", "tool_use"],
  "estimatedInputTokens": 3000,
  "estimatedOutputTokens": 2000,
  "domain": "legal"
}
```

3.5 Scientific Domain

When to Use: Research analysis, data interpretation, academic writing

Compliance: Varies by research type. FDA 21 CFR Part 11 for pharmaceutical research.

What Happens: - Source citation required - Minimum quality score of 70 - Slightly relaxed ECD threshold (0.08) - No forced compliance (specify if needed)

```
{
  "requiredCapabilities": ["chat", "reasoning"],
  "estimatedInputTokens": 3000,
  "estimatedOutputTokens": 2000,
  "domain": "scientific"
}
```

For FDA-regulated research, add compliance:

```
{
  "domain": "scientific",
  "requiredCompliance": ["FDA_21_CFR"]
}
```

3.6 Creative Domain

When to Use: Content writing, storytelling, marketing copy, brainstorming

Compliance: None required. Most flexible domain.

What Happens: - No compliance filtering - No minimum quality threshold - Cost and latency optimized (weights: C=0.25, L=0.20) - Learning dimension emphasized (E=0.10) - High ECD tolerance (0.20) - creative license allowed

```
{
  "requiredCapabilities": ["chat", "streaming"],
  "estimatedInputTokens": 500,
  "estimatedOutputTokens": 2000,
  "domain": "creative"
}
```

3.7 Engineering Domain

When to Use: Code generation, debugging, architecture design, DevOps

Compliance: Varies. SOC 2 recommended for sensitive applications.

What Happens: - Minimum quality score of 70 (code must work) - Capability dimension emphasized (K=0.20) - Prefers models with function_calling and tool_use - Moderate ECD threshold (0.10)

```
{
  "requiredCapabilities": ["chat", "function_calling"],
  "estimatedInputTokens": 2000,
  "estimatedOutputTokens": 1500,
  "domain": "engineering"
}
```

For medical device software, add FDA compliance:

```
{
  "domain": "engineering",
  "requiredCompliance": ["FDA_21_CFR", "SOC2"]
}
```

---

4. Compliance Options

4.1 Available Compliance Frameworks

Framework	Code	When Required
HIPAA	`HIPAA`	Healthcare/medical data
SOC 2 Type II	`SOC2`	Financial, legal, enterprise
GDPR	`GDPR`	EU data subjects
FDA 21 CFR Part 11	`FDA_21_CFR`	Pharma, medical devices
PCI-DSS	`PCI_DSS`	Payment card data
CCPA	`CCPA`	California consumer data
ISO 27001	`ISO_27001`	Enterprise security

4.2 Specifying Compliance

Single Framework:

```
{
  "requiredCapabilities": ["chat"],
  "estimatedInputTokens": 1000,
  "estimatedOutputTokens": 500,
  "requiredCompliance": ["HIPAA"]
}
```

Multiple Frameworks:

```
{
  "requiredCapabilities": ["chat"],
  "estimatedInputTokens": 1000,
  "estimatedOutputTokens": 500,
  "requiredCompliance": ["SOC2", "GDPR", "ISO_27001"]
}
```

4.3 Domain vs. Explicit Compliance

Using `domain` automatically sets compliance:

```
// These are equivalent:
{ "domain": "healthcare" }
{ "requiredCompliance": ["HIPAA"] }

// Domain also sets quality threshold, system type, Truth Engine
// So domain is preferred over explicit compliance alone
```

---

5. Optimization Strategies

5.1 Use Optimization Preferences

```
// Cost-optimized
{ "optimizeFor": "cost" }

// Quality-optimized
{ "optimizeFor": "quality" }

// Latency-optimized
{ "optimizeFor": "latency" }

// Balanced (default)
{ "optimizeFor": "balanced" }
```

5.2 Combine Domain with Optimization

```
{
  "requiredCapabilities": ["chat"],
  "estimatedInputTokens": 1000,
  "estimatedOutputTokens": 500,
  "domain": "engineering",
  "optimizeFor": "cost"  // Cost-optimize within engineering constraints
}
```

5.3 Set Hard Constraints

```
{
  "requiredCapabilities": ["chat"],
  "estimatedInputTokens": 1000,
  "estimatedOutputTokens": 500,
  "maxPrice": 0.01,        // Max $0.01 per request
  "minQuality": 75,        // At least 75 quality score
  "maxLatencyMs": 1000     // Under 1 second
}
```

---

6. Understanding Selection Results

6.1 Response Structure

```json
{
  "selection": {
    "modelId": "claude-sonnet-4-5",
    "providerId": "anthropic",
    "displayName": "Claude Sonnet 4.5",
    "score": 85.2,
    "estimatedPrice": 0.0115,
    "estimatedLatencyMs": 450,
    "reason": "Selected for engineering domain. HIPAA compliant. High capability score."
  },
  "fallbacks": [...],
  "scoring": {
    "dimensionScores": {
      "quality": 83,
      "cost": 70,
      "latency": 85,
      "capability": 100,
      "reliability": 95,
      "compliance": 100,
      "availability": 100,
      "learning": 60
    },
    "weightsUsed": {
      "Q": 0.30, "C": 0.15, "L": 0.15, "K": 0.20,
      "R": 0.10, "P": 0.00, "A": 0.05, "E": 0.05
    },
    "weightProfileId": "ENGINEERING"
  },
  "metadata": {
    "systemType": "SYSTEM_2",
    "domain": "engineering",
    "selectionTimeMs": 23
  }
}
```

6.2 Compliance Score

The compliance score (P) in `dimensionScores` reflects: - 100: Model has all required compliance certifications - 0: Model filtered out (you won't see this - it's excluded)

If you request HIPAA compliance, only HIPAA-certified models are returned.

---

7. SDK Examples

7.1 JavaScript/TypeScript

```
import { RAWSClient } from '@radiant/raws-client';

const raws = new RAWSClient({ apiKey: process.env.RADIANT_API_KEY });

// Healthcare selection (automatic HIPAA compliance)
const healthcareResult = await raws.select({
  requiredCapabilities: ['chat', 'tool_use'],
  estimatedInputTokens: 2000,
  estimatedOutputTokens: 1500,
  domain: 'healthcare',
});

// Engineering selection
const engineeringResult = await raws.select({
  requiredCapabilities: ['chat', 'function_calling'],
  estimatedInputTokens: 2000,
  estimatedOutputTokens: 1000,
  domain: 'engineering',
});

// Creative selection (cost-optimized)
const creativeResult = await raws.select({
  requiredCapabilities: ['chat', 'streaming'],
  estimatedInputTokens: 500,
  estimatedOutputTokens: 2000,
  domain: 'creative',
  optimizeFor: 'cost',
});
```

7.2 Python

```
from radiant_raws import RAWSClient

raws = RAWSClient(api_key="your-api-key")

# Healthcare (HIPAA enforced)
result = raws.select(
    required_capabilities=["chat", "tool_use"],
    estimated_input_tokens=2000,
    estimated_output_tokens=1500,
    domain="healthcare",
)

# Financial (SOC 2 enforced)
result = raws.select(
    required_capabilities=["chat", "function_calling"],
    estimated_input_tokens=2000,
    estimated_output_tokens=1500,
    domain="financial",
)
```

---

8. Best Practices

8.1 Always Specify Domain for Regulated Use Cases

```
//  Don't rely on auto-detection for regulated domains
const result = await raws.select({
  requiredCapabilities: ['chat'],
  estimatedInputTokens: 1000,
  estimatedOutputTokens: 500,
  // Missing domain - might not get HIPAA compliance
});

//  Explicitly specify domain
const result = await raws.select({
  requiredCapabilities: ['chat'],
  estimatedInputTokens: 1000,
  estimatedOutputTokens: 500,
  domain: 'healthcare',  // Guarantees HIPAA compliance
});
```

8.2 Check Compliance in Response

```
const result = await raws.select({ domain: 'healthcare', ... });

// Verify compliance was applied
console.log(result.scoring.dimensionScores.compliance); // Should be 100
console.log(result.metadata.domain); // Should be 'healthcare'
```

8.3 Use Appropriate Domain for Your Use Case

Use Case	Recommended Domain
Patient chatbot	`healthcare`
Investment advisor	`financial`
Contract review	`legal`
Research assistant	`scientific`
Blog writer	`creative`
Code assistant	`engineering`
General Q&A	`general`

---

9. FAQ

Q: What happens if I request a domain but don't have models with required compliance?

A: You'll receive error `RAWS_005: Compliance requirement not met`. This means no models in your tier have the required certification. Contact support to upgrade.

Q: Can I use healthcare models for non-healthcare purposes?

A: Yes, HIPAA-certified models can be used for any purpose. The certification means they *can* handle PHI, not that they *must*.

Q: How do I know which models have which compliance?

A: The selection response includes the model's compliance. You can also query the model registry:

```
curl https://api.radiant.example.com/v1/raws/models?compliance=HIPAA
```

Q: Is the engineering domain appropriate for medical device software?

A: Use `engineering` domain with explicit `requiredCompliance: ["FDA_21_CFR", "SOC2"]` for medical device software development.

---

10. Error Reference

Code	Description	Resolution
`RAWS_001`	No eligible models	Reduce requirements
`RAWS_005`	Compliance not met	Check tier/requirements
`RAWS_006`	Tier restriction	Upgrade subscription

---

11. Contact

Documentation: https://docs.radiant.example.com/raws

Compliance Questions: compliance@radiant.example.com

Support: support@radiant.example.com

End of User Documentation

Version 1.1.0 | January 2026