

Primary Focus: Telecom RAG Assistant (Enhanced Build Guide)

OpenRouter → Multiple LLM options for different query types

Google AI Studio → Gemini 2.0 Flash for complex analysis

Opik → Performance monitoring and optimization

ExaSearch → External telecom knowledge integration

Secondary Enhancement: Original Telco Demo

Upgrade existing incident analysis with better models

Add performance monitoring to current workflows

Enhance intelligence with external context

🚀 Enhanced Telecom RAG Assistant - n8n Build Guide v2.0

Step 3.5: Add OpenRouter Integration (INSERT AFTER Query Classifier)

Node Type: HTTP Request

Purpose: Smart LLM routing based on query complexity and type

Configuration:

Node Name: LLM Router & Enhancer

Method: POST

URL: <https://openrouter.ai/api/v1/chat/completions>

Authentication: HTTP Header Auth

Header Name: Authorization

Header Value: Bearer YOUR_OPENROUTER_API_KEY

Headers:

Content-Type: application/json

HTTP-Referer: <https://your-domain.com>

X-Title: Telecom RAG Assistant

JSON Body:

jsonCopy

```

|
"model": "{ { $('Query Classifier').item.json.message.content.trim() } == 'PREDICTION' ?
'google/gemini-2.0-flash-exp:free' : $('Query Classifier').item.json.message.content.trim()
== 'INCIDENT' ? 'anthropic/claude-3.5-haiku:beta' : 'openai/gpt-4o-mini' } }",
    "messages":
        [
            {
                "role": "system",
                "content": "You are an expert telecom network analyst AI. Your role is to enhance user
queries with technical context and suggest optimal data retrieval strategies. For the query
type { { $('Query Classifier').item.json.message.content.trim() } }, provide:\n\n1.
ENHANCED_QUERY: Technically enhanced version of the user query\n2. DATA_STRATEGY:
Specific Neo4j query optimization suggestions\n3. ANALYSIS_FOCUS: Key metrics and
KPIs to emphasize\n4. EXTERNAL_CONTEXT: Relevant industry context or
standards\n\nRespond in JSON format."
            },
            {
                "role": "user",
                "content": "Original Query: { { $('RAG Query Webhook').item.json.body.query } }\nQuery
Type: { { $('Query Classifier').item.json.message.content.trim() } }\nTime Range: { { $('RAG
Query Webhook').item.json.body.timeRange } }\nFilters: { JSON.stringify($('RAG Query Webhook').item.json.body.filters) } }"
            }
        ],
    "temperature": 0.3,
    "max_tokens": 800,
    "response_format": { "type": "json_object" }
}

```

Connection: Query Classifier → LLM Router & Enhancer → Query Router
Position: [530, 300]

Step 4.5: Add Google AI Studio Integration (Enhanced Analysis)

Node Type: HTTP Request

Purpose: Use Gemini 2.0 Flash for complex predictive and analytical queries

Configuration:

Node Name: Gemini Advanced Analyzer

Method: POST

URL: https://generativelanguage.googleapis.com/v1beta/models/gemini-2.0-flash-exp:generateContent?key=YOUR_GOOGLE_API_KEY

Headers:

Content-Type: application/json

JSON Body:

jsonCopy

```
|
    "contents": {
      "parts": [
        {
          "text": "As an expert telecom network analyst, provide advanced analysis for this {{ $('LLM Router & Enhancer').item.json.ANALYSIS_FOCUS }}:\n\nOriginal Query: {{ $('RAG Query Webhook').item.json.body.query }}\nQuery Enhancement: {{ $('LLM Router & Enhancer').item.json.ENHANCED_QUERY }}\nData Strategy: {{ $('LLM Router & Enhancer').item.json.DATA_STRATEGY }}\n\nTelecom Data Context:\n{{ JSON.stringify($('Response Processor').item.json.data.slice(0, 5), null, 2) }}\n\nProvide:\n1. DEEP_INSIGHTS: Advanced pattern recognition and anomaly detection\n2. PREDICTIVE_INDICATORS: Early warning signs and trend predictions \n3. BUSINESS_IMPACT: Operational and financial implications\n4. TECHNICAL_RECOMMENDATIONS: Specific technical actions and optimizations\n5. EXTERNAL_FACTORS: Industry trends, regulatory considerations, competitive analysis\n\nFormat as structured analysis suitable for senior network engineers and management."
        }
      ]
    },
    "generationConfig": {
      "temperature": 0.4,
      "topK": 40,
      "topP": 0.95,
      "maxOutputTokens": 1500
    }
  }
}
```

Connection: Response Processor → Gemini Advanced Analyzer (parallel to AI Summary Generator)

Position: [1420, 450]

Step 6.5: Add Opik Performance Monitoring

Node Type: HTTP Request

Purpose: Track workflow performance, model usage, and optimization metrics

Configuration:

Node Name: Opik Performance Tracker

Method: POST

URL: <https://cloud.opik.comet.com/api/v1/traces>

Authentication: HTTP Header Auth

Header Name: Authorization

Header Value: Bearer YOUR_OPIK_API_KEY

Headers:

Content-Type: application/json

JSON Body:

jsonCopy

```
|
    "trace": {
        "id": "{{ $execution.id }}_{{ Date.now() }}",
        "name": "Telecom_RAG_Query",
        "start_time": "{{ $('RAG Query Webhook').item.timestamp }}",
        "end_time": "{{ $now }}",
        "metadata": {
            "query_type": "{{ $('Query Classifier').item.json.message.content.trim() }}",
            "original_query": "{{ $('RAG Query Webhook').item.json.body.query }}",
            "data_count": "{{
                \\\('Response Processor').item.json.dataCount }}", "processing_time_ms": "{{ Math.abs(new
                Date(\\
                now) - new Date($('RAG Query Webhook').item.timestamp)) }}",
            "models_used": {
                "classifier": "gpt-3.5-turbo",
                "enhancer": "{{ $('LLM Router & Enhancer').item.json.model
            }",
            "analyzer": "gemini-2.0-flash-exp",
            "summarizer": "gpt-4"
        },
        { $('Final Response Formatter').item.json.response.success }}",
        "success": ""user_id": "{{ $('RAG Query Webhook').item.json.body.userId}"
    }
},
"spans": [
{
    "name": "Query_Classification",
    "type": "LLM",
    "start_time": "{{ $('Query Classifier').item.timestamp }}"}
```

```

"input": "{{ $('RAG Query Webhook').item.json.body.query }}",
"output": "{{ $('Query Classifier').item.json.message.content }}"
},
{
"name": "Data_Retrieval",
"type": "DATABASE",
"start_time": "{{ $('Response Processor').item.timestamp }}",
"input": "Neo4j Query",
"output": "{{ $('Response Processor').item.json.dataCount }} records"
}
]
}

```

Connection: Final Response Formatter → Opik Performance Tracker → Return Response
Position: [1750, 350]

Step 7.5: Add ExaSearch External Context Enhancement

Node Type: HTTP Request

Purpose: Enhance responses with external telecom industry context and best practices

Configuration:

Node Name: ExaSearch Context Enhancer

Method: POST

URL: <https://api.exa.ai/search>

Authentication: HTTP Header Auth

Header Name: x-api-key

Header Value: YOUR_EXA_API_KEY

Headers:

Content-Type: application/json

JSON Body:

jsonCopy

```

|
"query":      $('RAG      Query      Webhook').item.json.body.query
"telecom      network      {{      $('Query
Classifier').item.json.message.content.trim().toLowerCase() }} {{ $('LLM Router &
Enhancer').item.json.ENHANCED_QUERY
} best practices industry standards",
"type": "neural",
"useAutoprompt": true,
"numResults": 3,
"contents": {

```

```
"text": true,
"highlights": true,
"summary": true
},
"category": "telecommunications",
"includeDomains": ["ieee.org", "itu.int", "3gpp.org", "gsma.com", "telecomengine.com"],
"startPublishedDate": "2023-01-01T00:00:00.000Z"
}
```

Additional Processing Node:

Node Type: Code (JavaScript)

Purpose: Process ExaSearch results and integrate with analysis

JavaScript Code:

```
javascriptCopy// Process ExaSearch results
```

```
const exaResults = $input.item.json;
```

```
const originalAnalysis = $('Gemini Advanced Analyzer').item.json;
```

```
// Extract key insights from external sources
```

```
const externalInsights = exaResults.results ? exaResults.results.map(result => ({
```

```
title: result.title,
```

```
summary: result.summary,
```

```
highlights: result.highlights,
```

```
url: result.url,
```

```
relevanceScore: result.score
```

```
})) : [];
```

```
// Combine with internal analysis
```

```
const recent_trends = insight.title.toLowerCase().includes('standard')
    ? externalInsights.filter(insight => insight.title.toLowerCase().includes('trend'))
    : externalInsights.filter(insight => insight.title.toLowerCase().includes('2025'))
    ...originalAnalysis,
external_context: {
    industry_insights: externalInsights,
    best_practices: externalInsights.filter(insight =>
const enhancedResponse = insight.title.toLowerCase().includes('best practice'),
confidence_boost: externalInsights.length > 0 ? 15 : 0
};
```

```
return { json: enhancedResponse };
```

Connection: Gemini Advanced Analyzer → ExaSearch Context Enhancer → External Context Processor → Final Response Formatter

Position: [1420, 550]

Enhanced Model Selection Strategy

Dynamic Model Selection Logic:

javascriptCopy// In LLM Router & Enhancer node

```
const queryType = $('Query Classifier').item.json.message.content.trim();
```

```
const queryComplexity = $('RAG Query Webhook').item.json.body.query.length;
```

```
let selectedModel;
```

```
if (queryType === 'PREDICTION' || queryComplexity > 100) {
```

```
  selectedModel = 'google/gemini-2.0-flash-exp:free'; // Complex analysis
```

```
} else if (queryType === 'INCIDENT') {
```

```
  selectedModel = 'anthropic/claude-3.5-haiku:beta'; // Fast incident response
```

```
} else if (queryType === 'PERFORMANCE') {
```

```
  selectedModel = 'openai/gpt-4o-mini'; // Balanced performance
```

```
} else {
```

```
  selectedModel = 'meta-llama/llama-3.2-11b-vision-instruct:free'; // General queries
```

```
}
```



Enhanced Response Format (Updated)

```
jsonCopy{
```

```
  "success": true,
```

```
  "queryType": "TOWER_STATUS",
```

```
  "originalQuery": "What towers need immediate attention?",
```

```
  "enhancedQuery": "Identify telecom towers with critical performance degradation  
requiring immediate field intervention based on failure rates, call drop patterns, and SLA  
thresholds",
```

```
  "timestamp": "2025-06-28T15:30:00Z",
```

```
  "processingTime": "2847ms",
```

```
  "dataCount": 15,
```

```
  "summary": "Checked 15 towers. 2 towers showing critical failure rates requiring  
immediate attention.",
```

```
  "aiAnalysis": "Advanced Gemini 2.0 analysis with predictive insights...",
```

```
  "externalContext": {
```

```
    "industryInsights": [...],
```

```
    "bestPractices": [...],
```

```
    "recentTrends": [...]
```

```
  },
```

```
  "data": [...],
```

```
  "visualizations": [...],
```

```
  "recommendations": [...],
```

```
  "performance": {
```

```
    "modelsUsed": ["gpt-3.5-turbo", "gemini-2.0-flash-exp", "claude-3.5-haiku"],
```

```
    "processingTime": "2.8s",
```

```
"confidenceScore": 92,  
"dataQuality": "High"  
}  
}
```

Upgrade Path for Original Telco Demo

Quick Wins:

Replace OpenAI nodes in incident workflow with OpenRouter for model diversity

Add Opik tracking to monitor incident detection performance

Enhance Slack messages with Gemini-powered deeper analysis

Add external context for incident root cause analysis

Implementation Priority:

- ✓ Start with OpenRouter - Immediate model diversity (15 mins)
- ✓ Add Opik tracking - Performance monitoring (10 mins)
- ✓ Integrate Google AI Studio - Enhanced analysis (20 mins)
- ✓ ExaSearch integration - External context (25 mins)

Total implementation time: ~70 minutes for dramatically enhanced capabilities!