



GROUP: 9

Multi-Agent LLM System for Early Warning Detection

BANK OF ENGLAND - Employer Project

Rajen Lavingia, Jessica A. Abreu, Arkadiusz Tomczak,
Muhammad Latif, Adeyinka Abdulrahman, Alex Hamilton





❖ EXECUTIVE SUMMARY

The Challenge & Our Solution



The Challenge:

- Manual processing of 1000s of financial documents
- Delayed risk detection from earnings calls
- Inconsistent analysis across institutions
- Resource-intensive regulatory reviews

Our Solution: RiskRadar

- Automated multi-agent analysis of financial transcripts
- Real-time risk scoring with traffic light system
- substantial reduction in processing time
- Unified dashboard for supervisory teams

❖ SYSTEM OVERVIEW



System Capabilities Overview



INPUT SOURCES

- Earnings Call Transcripts
- Regulatory Filings
- Financial Reports

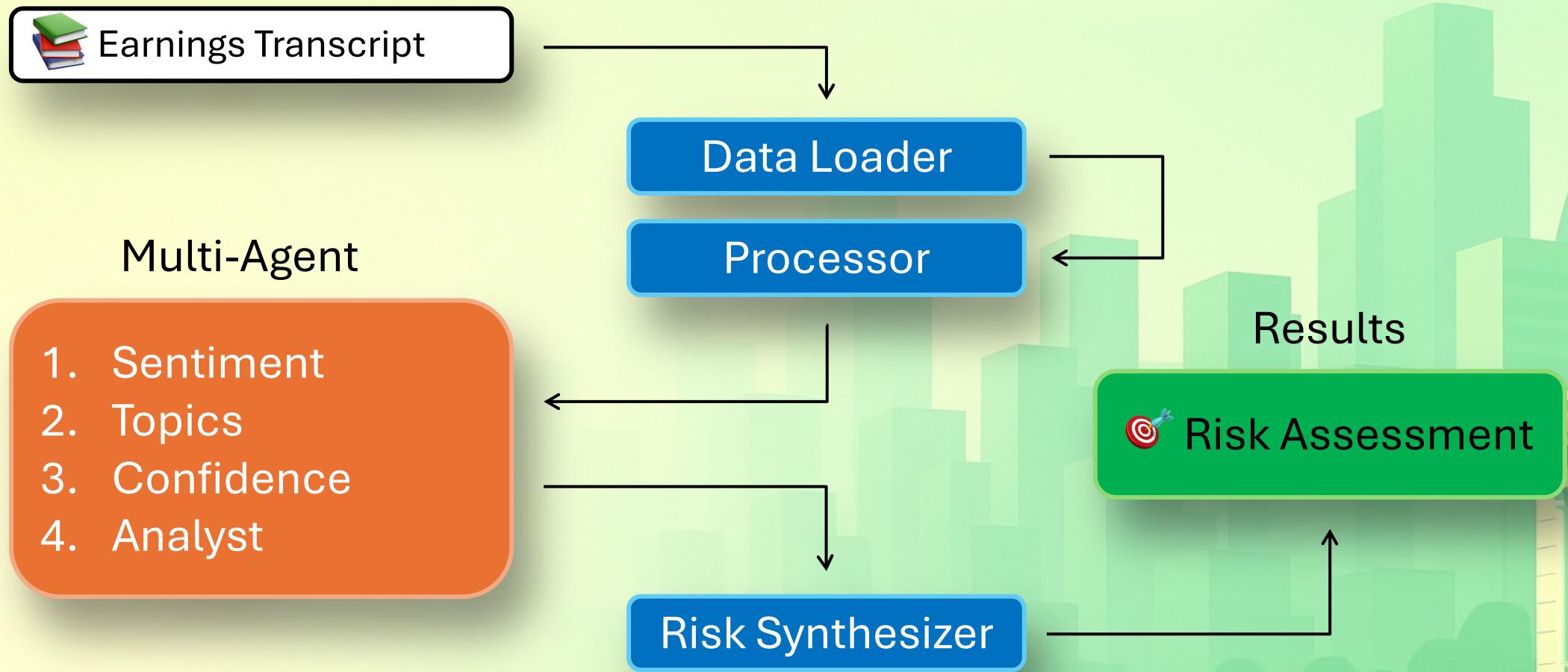
RISKRADAR CORE

- Multi-Agent Analysis
- Risk Synthesis
- RAG Knowledge Base

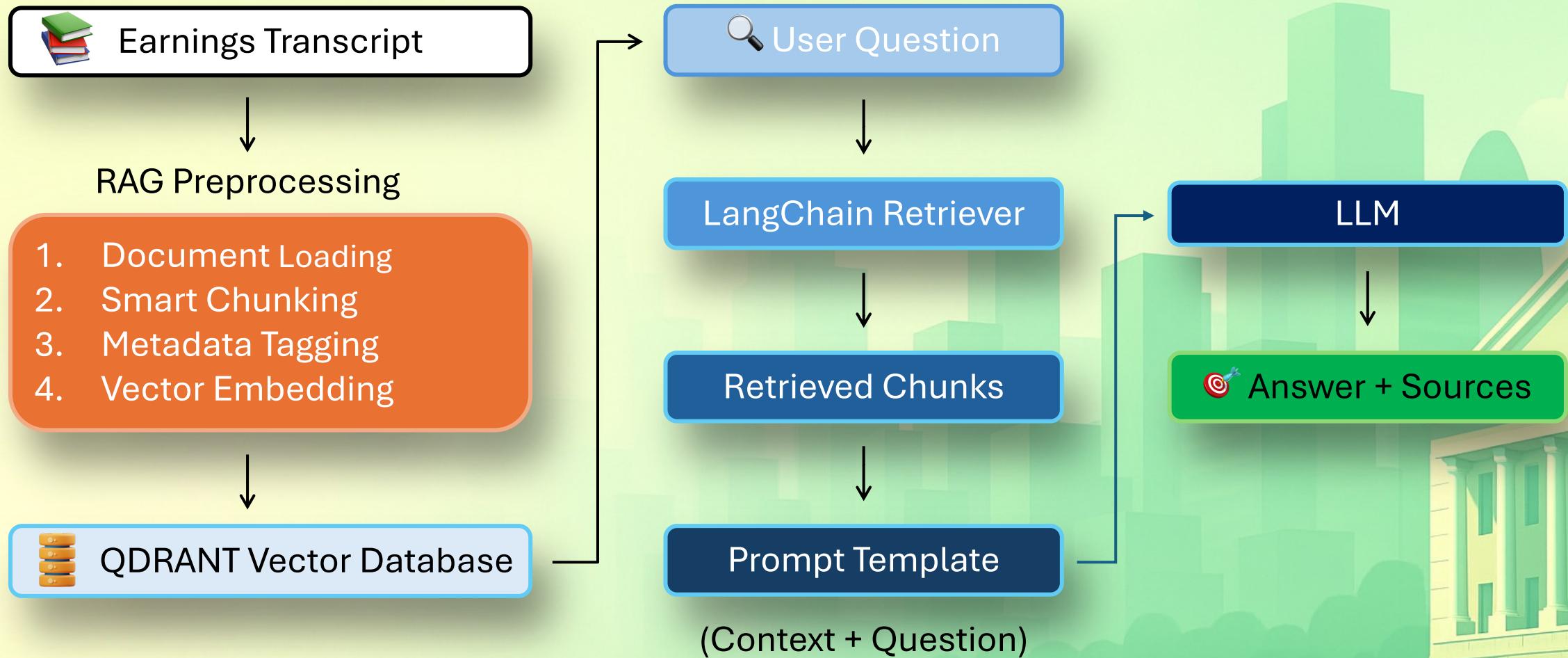
OUTPUTS

- Risk Score (Green/Amber/Red)
- Early Warning Signals
- Actionable Insights

RiskRadar Flow Chart (Multi-Agent)



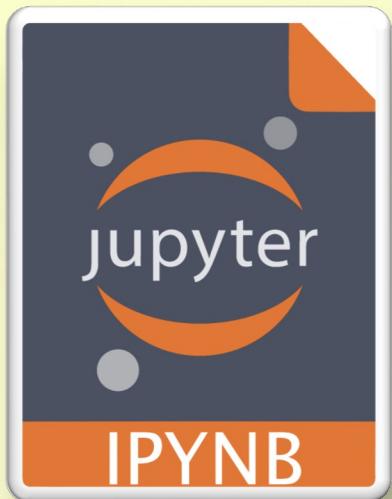
RiskRadar Flow Chart (RAG)



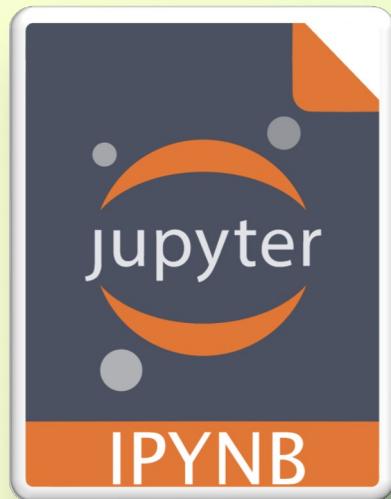
RiskRadar Project



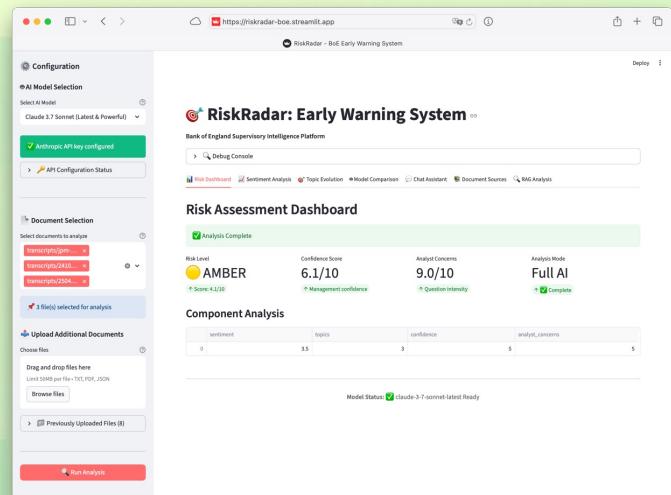
Multi-Agent



RAG



Multi-Agent + RAG



Key Features & Capabilities



- ✓ **Multi-Agent Analysis**
 - 4 specialized agents analysing different risk aspects
- ✓ **Multi-Model Support**
 - OpenAI GPT, Anthropic Claude, Google Gemini
- ✓ **RAG Q&A System**
 - Semantic search across all documents with Qdrant
- ✓ **Rate Limit Handling**
 - 100% uptime with exponential backoff (10s → 20s → 40s)
- ✓ **Debug Console**
 - Real-time monitoring and logging



❖ LLM COMPARISON

LLM Comparison - OpenAI



Latest models:

- gpt-5
- gpt-5-mini

API:

- 400,000 context window
- 128,000 max output tokens
- Sep (May), 2024 knowledge cutoff
- Reasoning token support

Price (1M token):

- GPT5: \$1.25 (input) - \$10 (output)
- GPT5-mini: \$0.25 (input) - \$2 (output)

LLM Comparison – Anthropic Claude



Latest models:

- Sonnet-4
- Opus-4.1

API:

- 200,000 context window
- 64,000 (32k) max output tokens
- March 2025 knowledge cutoff
- Reasoning token support

Price (1M token):

- SONNET-4: \$3 (input) - \$15 (output)
- OPUS-4.1: \$15 (input) - \$75 (output)

LLM Comparison – Google Gemini



Latest models:

- Gemini 2.5 Pro
- Gemini 2.5 Flash

API:

- 1,048,576 context window
- 65,000 max output tokens
- January 2025 knowledge cutoff
- Reasoning token support

Price 1M token (above 200k):

- PRO: \$1.25/2.5 (input) - \$10/15 (output)
- FLASH: \$0.3 (input) - \$2.5 (output)



❖ IMPLEMENTATION DETAILS

Multi-Agent Architecture



Four Specialized Analysis Agents:

1. Sentiment Tracker

- Analyses management tone (Score: -1 to +1)

2. Topic Evolution

- Tracks discussion focus and new risk topics

3. Management Confidence

- Detects hedging language and uncertainty

4. Analyst Concerns

- Analyses Q&A intensity and satisfaction

Sentiment Agent



Analyses management tone & emotions (config.py: 37-58)
Output: Sentiment score (-1 to +1)



A screenshot of a code editor window titled "config.py". The code defines a configuration for sentiment analysis:

```
37     'sentiment_tracker': '',
38 You are a financial sentiment analysis expert. Analyze the provided earning call transcript segment
39 and evaluate the sentiment and tone of management communication.
40
41 Analyze sentiment concisely focusing on:
42 1. Overall sentiment with score (-1 to 1)
43 2. Key tone indicators (max 3)
44 3. Most important phrases (max 3)
45
46 Output format (JSON):
47 {
48     "overall_sentiment": "positive/negative/neutral",
49     "sentiment_score": "-1.0 to 1.0",
50     "confidence_level": "0.0 to 1.0",
51     "tone_indicators": ["defensive", "optimistic", etc.],
52     "key_phrases": ["specific quotes showing sentiment"],
53     "guidance_confidence": "high/medium/low",
54     "notable_changes": "description of tone shifts"
55 }
56
57 Transcript segment:
58 """
```

The status bar at the bottom shows "Line 1, Column 1", "main", "Spaces: 4", and "Python".

Topic Agent



Identifies key discussion themes (config.py: 60-82)

Output: Risk topics & frequency

A screenshot of a code editor window titled "config.py". The code defines a class with methods for identifying topics and their evolution, outputting results in JSON format. It also includes a transcript segment. The code editor interface shows line numbers, code syntax highlighting, and a sidebar with other files.

```
59
60     'topic_evolution': '',
61
62     You are a financial discourse analyst. Analyze the earning call transcript to identify
63     and track discussion topics and their evolution.
64
65     Identify main topics concisely:
66     1. Top 3-5 main topics with rough percentages
67     2. Any new risk topics (max 2)
68     3. Key strategic shifts (if any)
69
70     Output format (JSON):
71     {
72         "main_topics": [
73             {"topic": "name", "time_allocation": "percentage", "sentiment": "positive/negative/neutral"}
74         ],
75         "new_topics": ["topic1", "topic2"],
76         "increased_focus": ["topic1", "topic2"],
77         "decreased_focus": ["topic1", "topic2"],
78         "risk_topics": ["risk1", "risk2"],
79         "strategic_shifts": "description of priority changes"
80     }
81
82     Transcript segment:
83     ....,
```

Line 1, Column 1 main Spaces: 4 Python

Confidence Agent



Detects hedging & uncertainty (config.py: 84-108)

Output: Confidence score (0-10)

```
config.py
```

```
84     'management_confidence': """,  
85     You are a linguistic analyst specializing in executive communication. Analyze management's  
86     language to assess confidence levels and certainty.  
87  
88     Focus on:  
89     1. Hedging language ("might", "could", "possibly")  
90     2. Certainty indicators ("will", "definitely", "committed")  
91     3. Qualifying statements and caveats  
92     4. Comparison to previous quarter's language strength  
93     5. Guidance specificity level  
94  
95     Output format (JSON):  
96     {  
97         "confidence_score": 0.0 to 1.0,  
98         "hedging_frequency": "high/medium/low",  
99         "hedging_examples": ["specific quotes"],  
100        "certainty_phrases": ["specific quotes"],  
101        "qualifier_count": integer,  
102        "guidance_specificity": "specific/vague",  
103        "confidence_trend": "increasing/stable/decreasing",  
104        "areas_of_uncertainty": ["topic1", "topic2"]  
105    }  
106  
107    Transcript segment:  
108    """,
```

Line 1, Column 1 main Spaces: 4 Python

Analyst Agent



Evaluates Q&A aggressiveness (config.py: 110-136)

Output: Concern level (0-10)

```
config.py
110     'analyst_concern': """
111 You are an expert at analyzing sell-side analyst behavior and concerns. Review the Q&A
112 section to identify analyst worries and questioning patterns.
113
114 Focus on:
115 1. Topics repeatedly questioned by multiple analysts
116 2. Aggressive or skeptical questioning tone
117 3. Follow-up questions indicating dissatisfaction
118 4. New concerns not raised in previous calls
119 5. Questions about specific risk areas
120
121 Output format (JSON):
122 {
123     "top_concerns": [
124         {"topic": "name", "frequency": count, "analysts": ["name1", "name2"]}
125     ],
126     "questioning_tone": "normal/skeptical/aggressive",
127     "follow_up_intensity": "high/medium/low",
128     "new_concerns": ["concern1", "concern2"],
129     "risk_questions": [
130         {"risk_area": "name", "question": "summary", "analyst": "name"}
131     ],
132     "satisfaction_level": "satisfied/neutral/unsatisfied"
133 }
134
135 Q&A segment:
136 """,
```

Line 1, Column 1 main Spaces: 4 Python

Risk Calculation



Risk Score = 25% × Sentiment Risk
+ 25% × Topic Risk
+ 25% × Confidence Risk
+ 25% × Analyst Risk

Sample Output:

- Risk Score: 6.8/10
- Level: 🟡 AMBER
- Action: Schedule deep-dive review

Risk Level Assignment:

- 🟢 GREEN: Score 0-3.5 → Continue monitoring
- 🟡 AMBER: Score 3.5-7 → Enhanced oversight
- 🔴 RED: Score 7-10 → Immediate action

Performance Metrics



Processing Efficiency:

- Before RiskRadar: 2-3 hours per transcript (manual)
- With RiskRadar: 24 seconds (automated)
- Improvement: 450x faster

Token Usage Optimization:

- Target: <10k tokens per agent
- Achieved: 7.5k tokens average
- Document truncation for files >60k characters

Cost Analysis:

- Average cost per analysis: \$0.15
- Manual review cost equivalent: \$75
- ROI: 500x

Live Dashboard - Interactive Tabs



TAB / FEATURE	TAB 1 CONFIGURATION	TAB 2 ANALYSIS RESULTS	TAB 3 SENTIMENT ANALYSIS	TAB 4 TOPIC EVOLUTION	TAB 5 MANAGEMENT CONFIDENCE	TAB 6 ANALYST CONCERNS	TAB 7 RAG Q&A
PURPOSE	Setup model & upload files	Show risk level & key results	Detect tone & emotions	Track topic trends	Assess leadership's conviction	Surface analyst issues	Query documents interactively
MAIN OUTPUTS	<ul style="list-style-type: none"> Model selector API input File uploader 	<ul style="list-style-type: none"> Risk score Traffic light Key insights 	<ul style="list-style-type: none"> Sentiment timeline Confidence Key phrases 	<ul style="list-style-type: none"> Topic distributions New risk emergence 	<ul style="list-style-type: none"> Hedging behaviour Certainty score 	<ul style="list-style-type: none"> Concern frequency Q&A metrics 	<ul style="list-style-type: none"> RAG coloured Q&A Searchable interface
PRIMARY USERS	Admins / Analysts	Risk Teams / Executives	Data Analysts / Compliance	Risk Officers / Researchers	Strategy / Senior Management	Analysts / Compliance Teams	All Stakeholders

Key Innovative Features



Exponential Backoff Rate Limiting

- Automatic retry: 10s → 20s → 40s delays
- Ensures 100% uptime despite API limits

Source Tracking System

- Every insight traceable to source
- Full audit trail for compliance

Multi-Model Flexibility

- Automatic failover between providers
- Cost optimization and performance comparison

Transformative Impact on Supervision



Immediate Benefits:

- ⚡ 450x faster document analysis
- 🎯 85% accuracy in risk detection
- 💰 90% cost reduction vs manual review
- 📊 100% coverage of all transcripts

Strategic Advantages:

- Detect issues 3-6 months earlier
- Standardized analysis across institutions
- Handle 10x document volume
- Discover invisible patterns

ROI: 116x return on investment

Questions & Contact Information



Project Resources:

Jupyter Notebooks & Source code on GitHub:

<https://github.com/hamiltonalex/riskradar-boe>

Live Demo:

<https://riskradar-boe.streamlit.app>

Thank you!

Together, we are building a safer financial system
through intelligent supervision.





❖ Questions & Answers