Of course. Here are four complete and distinct metadata files for the 10-K use case, architected with a primary routing agent and three specialized agents. Each file is self-contained and follows the robust structure we've developed.

### 1. The Routing Agent (routing-agent.yaml)

This is the "switchboard" agent. Its only job is to understand the user's intent and delegate the query to the correct specialist. Its safe\_tools are the other agents.

# Agent Card Metadata v3.0 (Routing Agent)  
# Last updated on 2025-09-22  
# Defines the capabilities for the 10-K Document Routing Agent.  
  
agent\_card:  
 name: "RoutingAgent\_10K"  
 version: "3.0"  
 description: "Acts as a smart dispatcher. It analyzes user queries about 10-K filings and routes them to the appropriate specialist agent: RAG for policies, Text-to-SQL for financials, or Web Search for general info."  
  
 goals:  
 - name: "RouteQueryToSpecialistAgent"  
 description: "Classify the user's intent and forward the query to the correct downstream agent."  
 trigger\_examples:  
 - "What were the total assets last year?"  
 - "Describe the company's data privacy policy."  
 - "Who is the current CEO of the parent company?"  
  
guardrails:  
 budgeting:  
 max\_steps: 3 # Very low step count, as it should only decide and delegate.  
 max\_duration\_seconds: 30  
 max\_token\_limit: 1024  
  
 goal\_configurations:  
 - goal\_name: "RouteQueryToSpecialistAgent"  
 reasoning: "The primary goal is to prevent this agent from answering questions itself. It must delegate. The trajectory ensures it performs a classification and the safe\_tools list limits its final action to calling another agent."  
  
 safe\_tools:  
 # The ONLY tools this agent can use are the other agents.  
 allow:  
 - "RAG\_10K\_PolicyAgent"  
 - "Text2SQL\_10K\_FinancialAgent"  
 - "WebSearchAgent"  
 # Explicitly deny access to low-level data tools.  
 deny:  
 - "vector\_search"  
 - "execute\_sql\_query"  
 - "perform\_web\_search"  
  
 trajectory:  
 reasoning: "A single-step trajectory to classify and route. This prevents the agent from engaging in a long conversation or deviating from its core routing task."  
 steps:  
 - step: 1  
 tool\_name: "classify\_and\_delegate" # This is a conceptual tool representing the routing decision.  
 description: "Analyze the user query and invoke the appropriate specialist agent."  
 reasoning: "This step enforces the agent's core function. The LLM's choice of tool here must be one of the allowed agents from the safe\_tools list."

### 2. The RAG Agent (rag-agent.yaml)

This agent specializes in finding and synthesizing answers from unstructured text within the 10-K documents, perfect for policy and risk-related questions.

# Agent Card Metadata v3.0 (RAG Agent)  
# Last updated on 2025-09-22  
# Defines the capabilities for the 10-K Policy & Risk RAG Agent.  
  
agent\_card:  
 name: "RAG\_10K\_PolicyAgent"  
 version: "3.0"  
 description: "Specializes in answering questions about company policies, risk factors, governance, and other non-structured text found in 10-K filings using Retrieval-Augmented Generation."  
  
 goals:  
 - name: "AnswerPolicyQuestion"  
 description: "Searches document embeddings and summarizes findings for policy-related queries."  
  
guardrails:  
 budgeting:  
 max\_duration\_seconds: 120  
 max\_token\_limit: 8192  
 max\_steps: 5  
  
 goal\_configurations:  
 - goal\_name: "AnswerPolicyQuestion"  
 reasoning: "The agent must follow the canonical RAG pattern: search, retrieve, summarize. This prevents hallucinations and ensures answers are grounded in the provided documents."  
  
 safe\_tools:  
 allow:  
 - "vector\_search"  
 - "retrieve\_document\_chunk"  
 - "summarize\_text"  
 deny:  
 - "execute\_sql\_query" # Cannot access structured financial data.  
 - "perform\_web\_search"  
  
 trajectory:  
 reasoning: "This 3-step RAG trajectory stops context drift by forcing the agent to find evidence before formulating an answer."  
 steps:  
 - step: 1  
 tool\_name: "vector\_search"  
 description: "Finds relevant document chunks based on semantic similarity to the user's query."  
 - step: 2  
 tool\_name: "retrieve\_document\_chunk"  
 description: "Fetches the full text of the most relevant chunks."  
 condition: "steps.vector\_search.output.chunk\_ids is not null"  
 - step: 3  
 tool\_name: "summarize\_text"  
 description: "Synthesizes an answer based on the content of the retrieved chunks."  
 condition: "steps.retrieve\_document\_chunk.output.content is not empty"

### 3. The Text-to-SQL Agent (text2sql-agent.yaml)

This agent is the quantitative expert. It translates natural language questions about financials into SQL queries to run against a structured database.

# Agent Card Metadata v3.0 (Text-to-SQL Agent)  
# Last updated on 2025-09-22  
# Defines the capabilities for the 10-K Financial Text-to-SQL Agent.  
  
agent\_card:  
 name: "Text2SQL\_10K\_FinancialAgent"  
 version: "3.0"  
 description: "Specializes in answering quantitative questions (e.g., revenue, assets, liabilities) by generating and executing SQL queries against a structured database of 10-K financial data."  
  
 goals:  
 - name: "AnswerQuantitativeQuestion"  
 description: "Translates a natural language question into a SQL query and returns the result."  
  
guardrails:  
 budgeting:  
 max\_duration\_seconds: 90  
 max\_token\_limit: 4096  
 max\_steps: 4  
  
 goal\_configurations:  
 - goal\_name: "AnswerQuantitativeQuestion"  
 reasoning: "The agent must first generate a valid SQL query before it can execute it. This prevents it from running arbitrary or malformed queries against the database."  
  
 safe\_tools:  
 allow:  
 - "generate\_sql\_query"  
 - "execute\_sql\_query"  
 - "format\_financial\_data"  
 deny:  
 - "vector\_search" # Cannot access unstructured policy documents.  
 - "perform\_web\_search"  
  
 trajectory:  
 reasoning: "This trajectory enforces a safe Text-to-SQL workflow, preventing direct execution of potentially flawed code."  
 steps:  
 - step: 1  
 tool\_name: "generate\_sql\_query"  
 description: "Converts the user's financial question into a syntactically correct SQL query."  
 - step: 2  
 tool\_name: "execute\_sql\_query"  
 description: "Executes the generated SQL query against the financial database."  
 condition: "steps.generate\_sql\_query.output.sql\_query is valid"  
 - step: 3  
 tool\_name: "format\_financial\_data"  
 description: "Formats the raw numerical output into a human-readable response."  
 condition: "steps.execute\_sql\_query.output.result is not empty"

### 4. The Web Search Agent (web-search-agent.yaml)

This is the generalist fallback agent. When the internal 10-K data (structured or unstructured) isn't sufficient, this agent can search the live web.

# Agent Card Metadata v3.0 (Web Search Agent)  
# Last updated on 2025-09-22  
# Defines the capabilities for the General Purpose Web Search Agent.  
  
agent\_card:  
 name: "WebSearchAgent"  
 version: "3.0"  
 description: "A general-purpose agent that can search the web to answer questions that are outside the scope of the internal 10-K documents, such as current news, market data, or executive profiles."  
  
 goals:  
 - name: "AnswerGeneralQuestion"  
 description: "Performs a web search and summarizes the findings to answer a user's query."  
  
guardrails:  
 budgeting:  
 max\_duration\_seconds: 90  
 max\_token\_limit: 8192  
 max\_steps: 4  
  
 goal\_configurations:  
 - goal\_name: "AnswerGeneralQuestion"  
 reasoning: "The agent should provide a summarized result from multiple sources, not just a list of links. The trajectory ensures it searches first, then summarizes."  
  
 safe\_tools:  
 allow:  
 - "perform\_web\_search"  
 - "summarize\_web\_results"  
 deny:  
 - "execute\_sql\_query" # No access to internal databases.  
 - "vector\_search" # No access to internal documents.  
  
 trajectory:  
 reasoning: "A simple 'search then summarize' workflow to ensure the agent provides a synthesized answer rather than raw search results."  
 steps:  
 - step: 1  
 tool\_name: "perform\_web\_search"  
 description: "Executes a search query on a public search engine."  
 - step: 2  
 tool\_name: "summarize\_web\_results"  
 description: "Summarizes the information gathered from the top search results."  
 condition: "steps.perform\_web\_search.output.results is not empty"