```
import os
from dotenv import load_dotenv
from swarms import Agent
from swarm_models import OpenAIChat
from swarms.structs.swarm_router import SwarmRouter

load_dotenv()

# Get the OpenAI API key from the environment variable
api_key = os.getenv("GROQ_API_KEY")

# Model
model = OpenAIChat(
    openai_api_base="https://api.groq.com/openai/v1",
    openai_api_key=api_key,
    model_name="llama-3.1-70b-versatile",
    temperature=0.1,
)
# Define specialized system prompts for each agent
DATA_EXTRACTOR_PROMPT = """You are a highly specialized private equity agent focused on data extraction from various documents. Your expertise includes:
1. Extracting key financial metrics (revenue, EBITDA, growth rates, etc.) from financial statements and reports
2. Identifying and extracting important contract terms from legal documents
3. Pulling out relevant market data from industry reports and analyses
4. Extracting operational KPIs from management presentations and internal reports
```

5. Identifying and extracting key personnel information from organizational charts and bios

Provide accurate, structured data extracted from various document types to support investment analysis."""

SUMMARIZER_PROMPT = """You are an expert private equity agent specializing in summarizing complex documents. Your core competencies include:

1. Distilling lengthy financial reports into concise executive summaries
2. Summarizing legal documents, highlighting key terms and potential risks
3. Condensing industry reports to capture essential market trends and competitive dynamics
4. Summarizing management presentations to highlight key strategic initiatives and projections
5. Creating brief overviews of technical documents, emphasizing critical points for non-technical stakeholders

Deliver clear, concise summaries that capture the essence of various documents while highlighting information crucial for investment decisions."""

FINANCIAL_ANALYST_PROMPT = """You are a specialized private equity agent focused on financial analysis. Your key responsibilities include:

1. Analyzing historical financial statements to identify trends and potential issues
2. Evaluating the quality of earnings and potential adjustments to EBITDA
3. Assessing working capital requirements and cash flow dynamics
4. Analyzing capital structure and debt capacity
5. Evaluating financial projections and underlying assumptions

Provide thorough, insightful financial analysis to inform investment decisions and valuation."""

MARKET_ANALYST_PROMPT = """You are a highly skilled private equity agent specializing in market analysis. Your expertise covers:

```
1. Analyzing industry trends, growth drivers, and potential disruptors
2. Evaluating competitive landscape and market positioning
3. Assessing market size, segmentation, and growth potential
4. Analyzing customer dynamics, including concentration and loyalty
5. Identifying potential regulatory or macroeconomic impacts on the market
Deliver comprehensive market analysis to assess the attractiveness and risks of potential investments."""

OPERATIONAL_ANALYST_PROMPT = """You are an expert private equity agent focused on operational analysis. Your core competencies include:
1. Evaluating operational efficiency and identifying improvement opportunities
2. Analyzing supply chain and procurement processes
3. Assessing sales and marketing effectiveness
4. Evaluating IT systems and digital capabilities
5. Identifying potential synergies in merger or add-on acquisition scenarios
Provide detailed operational analysis to uncover value creation opportunities and potential risks."""

# Initialize specialized agents
data_extractor_agent = Agent(
    agent_name="Data-Extractor",
    system_prompt=DATA_EXTRACTOR_PROMPT,
    llm=model,
    max_loops=1,
    autosave=True,
    verbose=True,
    dynamic_temperature_enabled=True,
    saved_state_path="data_extractor_agent.json",
    user_name="pe_firm",
    retry_attempts=1,
    context_length=200000,
    output_type="string",
)

summarizer_agent = Agent(
    agent_name="Document-Summarizer",
    system_prompt=SUMMARIZER_PROMPT,
    llm=model,
    max_loops=1,
    autosave=True,
    verbose=True,
    dynamic_temperature_enabled=True,
    saved_state_path="summarizer_agent.json",
    user_name="pe_firm",
    retry_attempts=1,
    context_length=200000,
    output_type="string",
)

financial_analyst_agent = Agent(
    agent_name="Financial-Analyst",
    system_prompt=FINANCIAL_ANALYST_PROMPT,
    llm=model,
    max_loops=1,
    autosave=True,
    verbose=True,
    dynamic_temperature_enabled=True,
    saved_state_path="financial_analyst_agent.json",
    user_name="pe_firm",
    retry_attempts=1,
    context_length=200000,
    output_type="string",
)

market_analyst_agent = Agent(
    agent_name="Market-Analyst",
    system_prompt=MARKET_ANALYST_PROMPT,
    llm=model,
    max_loops=1,
    autosave=True,
    verbose=True,
    dynamic_temperature_enabled=True,
    saved_state_path="market_analyst_agent.json",
    user_name="pe_firm",
    retry_attempts=1,
    context_length=200000,
    output_type="string",
)
```

```
operational_analyst_agent = Agent(
    agent_name="Operational-Analyst",
    system_prompt=OPERATIONAL_ANALYST_PROMPT,
    llm=model,
    max_loops=1,
    autosave=True,
    verbose=True,
    dynamic_temperature_enabled=True,
    saved_state_path="operational_analyst_agent.json",
    user_name="pe_firm",
    retry_attempts=1,
    context_length=200000,
    output_type="string",
)

# Initialize the SwarmRouter
router = SwarmRouter(
    name="pe-document-analysis-swarm",
    description="Analyze documents for private equity due diligence and investment decision-making",
    max_loops=1,
    agents=[
        data_extractor_agent,
        summarizer_agent,
        financial_analyst_agent,
        market_analyst_agent,
        operational_analyst_agent,
    ],
    swarm_type="ConcurrentWorkflow",  # or "SequentialWorkflow" or "ConcurrentWorkflow" or
)


# Example usage
if __name__ == "__main__":
    # Run a comprehensive private equity document analysis task
    result = router.run(
        "Where is the best place to find template term sheets for series A startups. Provide links and
references"
    )
    print(result)


    # Retrieve and print logs
    for log in router.get_logs():
        print(f"{log.timestamp} - {log.level}: {log.message}")
```