# מסמך אפיון מערכת: Competitive Intelligence Multi-Agent System

**מוגש על ידי:** עידו כהן

## 1. תיאור הבעיה והפתרון

**הבעיה:** ארגונים נדרשים לאסוף, לנתח ולסנתז מידע עסקי ותחרותי (כגון שינויי תמחור, רכישות ואסטרטגיות מתחרים) ממקורות רבים ומפוזרים. התהליך הידני איטי, מועד לשגיאות וגוזל זמן יקר מאנליסטים.

**הפתרון:** מערכת אוטונומית מבוססת סוכנים (Multi-Agent System) המופעלת על גבי מנוע LangGraph. המערכת קולטת שאילתות מחקר, שולפת מידע רלוונטי ממסד נתונים וקטורי פנימי או ממקורות מידע בזמן אמת, מנתחת את התוכן ומפיקה דוחות מודיעין מדויקים. המערכת מאפשרת אינטגרציה אקטיבית למידע חיצוני באמצעות אוטומציה.

## 2. פירוט צמתים (Nodes) וזרימת המערכת

- **Input Guardrail ( `input_guardrail` ):** משמש כשומר סף. מקבל את קלט המשתמש ומוודא שאינו מכיל ניסיונות הזרקת פרומפטים (Prompt Injection), שפה פוגענית או בקשות שחורגות מתחום המערכת. במידה והקלט תקין, הזרימה ממשיכה; אחרת, הבקשה נדחית.
- **Agent Reasoning ( `agent_reasoning` ):** המוח המרכזי של המערכת. מנסח שאילתות אסטרטגיות, מחליט איזו פעולה נדרשת, וקובע באילו כלים (Tools) עליו להשתמש כדי להשלים את המשימה.
- **Tool Execution ( `execute_tools` ):** צומת ביצועי המופעל בהתאם להחלטת ה-Reasoning Node. מפעיל שאילתות ל-Vector DB או קריאות Webhook חיצוניות ל-n8n, ומחזיר את התוצאות לגרף.
- **Synthesizer ( `synthesizer` ):** מרכז את כלל התצפיות (Observations) שהתקבלו מהכלים השונים, ומעבד אותן לכדי מסמך תובנות ברור, מובנה ומקצועי.
- **Output Guardrail ( `output_guardrail` ):** בקרת איכות על התוצר המוגמר. מוודא שאין זליגת מידע רגיש, שאין הזיות (Hallucinations) ושתוכן הדוח מגובה בעובדות שנשלפו.

## 3. אסטרטגיית RAG (Retrieval-Augmented Generation)

- **מסד נתונים וקטורי ומודל:** שימוש ב-Qdrant Cloud בחישוב מרחק Cosine Similarity, מבוסס על מודל ההטמעה `text-embedding-3-small` של חברת OpenAI.
- **נתונים:** מסמך מקור פנימי המוזן ישירות מקובץ `internal_research_report.txt` .
- **אסטרטגיית Chunking:** חלוקת המסמכים מבוצעת באמצעות `RecursiveCharacterTextSplitter` עם מקטעים בגודל 500 תווים וחפיפה של 80 תווים לשמירה על הקשר רציף. בנוסף, מיושם חילוץ מטא-דאטה דינמי המזהה ומתייג את שם חברת המטרה מתוך הטקסט לטובת סינון וקטורי ממוקד.

## 4. פירוט האוטומציה (n8n) ושרת ה-Backend

- **שרת ה-Backend (Flask):** עוטף את הגרף של LangGraph ורץ על פורט 5000. חושף Endpoint בנתיב `/query` המקבל JSON ומחזיר את התשובה ל-UI של Streamlit שרץ על פורט 8501.
- **האוטומציה ב-n8n (Market Intelligence Dispatcher):**
  - **Webhook Input:** מאזין לבקשות POST בנתיב `/market-intelligence` ומקבל מהסוכן את שם החברה ושאילתת החיפוש.
  - **לוגיקה (Logic):** צומת HTTP Request המבצע קריאת POST חיצונית למנוע החיפוש Tavily API (עד 3 תוצאות ממוקדות).
  - **Response:** צומת Respond to Webhook מחזיר לסוכן תשובת JSON מובנית הכוללת סטטוס, ציון סנטימנט ומחרוזת מודיעין עדכנית.

## 5. תיאור כלים (Tools)

**search_internal_intelligence:**

```
"Search the internal competitive intelligence repository in Qdrant vector database. Use this tool FIRST to retrieve proprietary corporate dossiers, SWOT analyses, and confidential strategy notes. Optionally filter results by target company name."
```

**fetch_external_market_data:**

```
"Trigger external market intelligence gathering via n8n automation webhook. Use this tool when internal intelligence is insufficient, or when fresh real-time web search and live market signals are required."
```

## 6. סביבת הרצה (Docker)

המערכת כולה ארוזה בקונטיינרים דרך `docker-compose.yml` . להפעלת המערכת במלואה:

```
docker-compose up --build
```

לאחר העלייה, ממשק המשתמש נגיש בכתובת: `http://localhost:8501` .