Discovery Agent: Mid-term Project Report

Thue T

December 9, 2025

Abstract

This report summarizes the progress on the Discovery Agent project, a system for automatically finding and structuring information about Danish industrial companies. It covers the research into data sources, orchestration tools, and AI API integration, and outlines the next steps for the project.

Contents

1 Introduction

The primary goal of this project is to create a ”Discovery Agent” that can build a database of Danish industrial companies, with a focus on steel and pipe fabricators. The agent will automatically discover companies, extract their capabilities, and store the information in a structured format.

The agent will follow a cyclical process to discover and process information, as illustrated in Figure 1.

2 Data Sources

A successful Discovery Agent relies on a variety of data sources. The following sources have been identified, ranked by their usefulness for this project.

1. **Official Business Registries (CVR):** The most reliable source for basic company information.
2. **Industry-Specific Portals & Directories:** High-value targets for finding companies in a specific sector.
3. **Company Websites:** The primary source for detailed manufacturing capabilities.
4. **Google Maps/Business Listings:** Good for finding locations and basic contact info.
5. **LinkedIn:** Useful for company size and focus.

2.1 Accessing the Danish Business Register (CVR)

The **CVR (Central Business Register)** is the official source of company information in Denmark. While a free official API exists, it can be complex to use. For ease of integration, a **third-party CVR API** is the recommended approach.

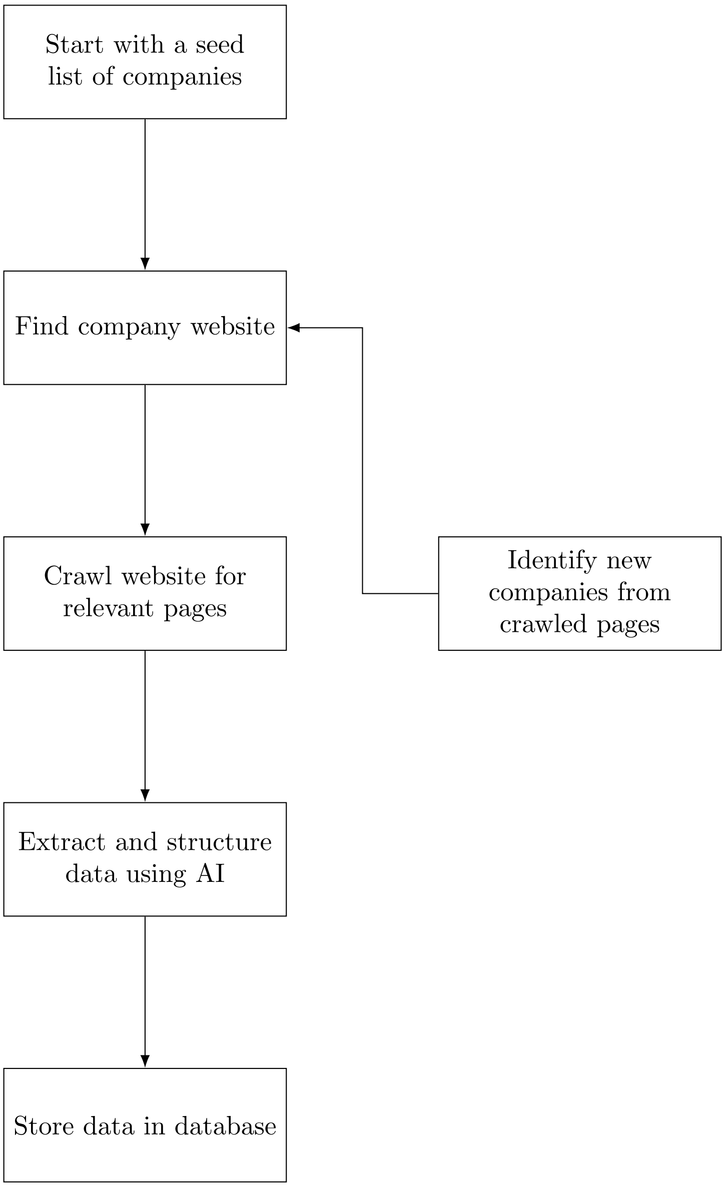
Start with a seed list of companies
Find company website
Crawl website for relevant pages
Identify new companies from crawled pages
Extract and structure data using AI
Store data in database

Figure 1: Discovery Agent Workflow

2.2 Legal Considerations: GDPR

Web scraping in the EU is subject to the **General Data Protection Regulation (GDPR)**. To minimize legal risks, the following guidelines should be followed:

- Focus on non-personal company data.
- Avoid collecting and storing information about specific employees.
- Always check the terms of service of the websites you are scraping.
- It is strongly advised to consult with a legal professional before starting any large-scale scraping project.

3 Workflow Orchestration

The Discovery Agent requires a tool to orchestrate the workflow (receiving requests, making API calls, etc.). We researched several self-hosted options suitable for a TrueNAS home server.

3.1 Recommendation: Node-RED

While n8n is a powerful tool, it may be overkill for this specific use case. **Node-RED is the recommended choice** for its simplicity, extremely low resource usage, and ease of setup. It provides all the necessary features

Tool	**Installation**	**Resource Usage**	**Learning Curve**	**AI Orchestration Capability**
n8n	Medium	Medium-High	Medium	Excellent (built-in nodes)
Node-RED	**Low**	**Very Low**	**Low**	Good (requires custom nodes)
Activepieces	Low	Low	Low	Good (growing library)
Express/FastAPI	Low (devs)	Very Low	Low (devs)	Excellent (custom code)

Table 1: Comparison of Workflow Orchestration Tools

to build the Discovery Agent's orchestration layer with minimal overhead.

4 AI API Integration

The core of the Discovery Agent is its ability to use AI to extract and structure data from unstructured text (like web pages). This requires orchestrating calls to multiple AI APIs in parallel for speed and resilience.

Feature	**Anthropic Claude**	**Google Gemini**	**OpenAI GPT**
Authentication	API Key	API Key or Google OAuth	API Key
JSON Mode	Yes (with prompting)	**Yes (built-in)**	**Yes (built-in)**
Best For	High-quality text generation	Multimodality, Google integration	General purpose, reasoning

Table 2: AI API Provider Comparison

4.1 Recommended Strategy for AI Orchestration

- **Use Python with 'asyncio':** For any custom code, Python's 'asyncio' library is perfect for making parallel API calls.
- **Use Official SDKs:** Always use the official Python SDKs for each AI provider.
- **Use JSON Mode:** To get structured, predictable data from the AI models, use the JSON mode offered by the APIs.
- **Implement Robust Error Handling:** Use a retry mechanism with exponential backoff for each API call.
- **Manage Costs:** Implement caching and set up billing alerts.

5 Next Steps

Based on the research conducted, the following next steps are recommended:

1. **Set up the development environment:**
   - Install and configure Node-RED on the TrueNAS server.
   - Set up a local Python development environment for the agent.
2. **Develop the Agent Prototype:**
   - Implement the CVR API integration to fetch basic company data.
   - Develop the web scraper for company websites.
   - Implement the AI integration for data extraction.
3. **Build the Backend and Frontend:**
   - Develop the backend API to serve the collected data.
   - Build the frontend search interface.

4. **Testing and Deployment:**
    - Thoroughly test the entire system.
    - Deploy the different components.