

B2B Lead Discovery Agent - Comprehensive Technical Report

Executive Summary

A fully terminal-based B2B Lead Discovery Agent that automates the process of finding, validating, and enriching business leads. Built with Python, powered by Groq LLM (Llama 3.3 70B) and Apollo.io API.

System Architecture

```
graph TD
    subgraph Input
        CLI[Terminal CLI<br>agent.py]
    end

    subgraph Orchestration
        WF[Workflow<br>workflow.py]
    end

    subgraph Agents
        DA[Discovery Agent<br>Web Search + LLM]
        SA[Structure Agent<br>Size-based Mapping]
        RA[Role Agent<br>LinkedIn Search]
        EA[Enrichment Agent<br>Apollo API]
        VA[Verification Agent<br>Lead Scoring]
    end

    subgraph Services
        WS[Web Search<br>DuckDuckGo]
        LS[LinkedIn Search<br>DuckDuckGo]
        AC[Apollo Client<br>REST API]
    end

    subgraph Storage
        DB[(JSON Database<br>leads.json)]
    end

    CLI --> WF
    WF --> DA
    DA --> SA
    SA --> RA
    RA --> EA
    EA --> VA
    VA --> DB
    DA --> WS
    RA --> LS
    EA --> AC
    VA --> DB
```

Pipeline Stages

Stage 1: Discovery Agent

File: agents/discovery_agent.py

Purpose: Validate company suitability for B2B sales

Process:

1. Search web for company information (DuckDuckGo)
2. Collect news articles and LinkedIn company page
3. Send search results to LLM for analysis
4. Output: Company name, industry, size, location, growth signals

Data Sources:

- Web search results
- News articles
- LinkedIn company pages

Output Example:

```
{  
    "name": "Tata Consultancy Services (TCS)",  
    "industry": "IT Services",  
    "size": "enterprise",  
    "location": "India",  
    "growth_signals": ["Revenue Growth", "AI Investments"],  
    "status": "accepted"  
}
```

Stage 2: Structure Agent

File: agents/structure_agent.py

Purpose: Map decision-makers based on company size

Decision-Maker Mapping by Size:

Company Size	Typical Decision-Makers
--------------	-------------------------

Small	Founder, CEO, CTO, Owner
Medium	VP, Director, Head of
Large	SVP, VP, Senior Director
Enterprise	EVP, SVP, VP, Global Head

Process:

1. Receive company size from Discovery
 2. Load pre-defined decision-maker roles
 3. Use LLM to refine based on industry
 4. Output: Departments, hierarchy, recommended targets
-

Stage 3: Role Agent

File: agents/role_agent.py

Purpose: Find actual people at the company

Process:

1. Search LinkedIn via DuckDuckGo for people at company
2. Parse results to extract: Name, Title, Company
3. Score each person's decision-making power (1-10)
4. Accept people with power ≥ 6

Decision Power Scale:

Score	Roles
10	CEO, Founder, Owner
9	CTO, CFO, CMO, CIO
8	President, EVP
7	VP, SVP
6	Director, Head of
5	Senior Manager
4	Manager
1-3	Individual Contributors

Stage 4: Enrichment Agent

File: agents/enrichment_agent.py

Purpose: Get verified contact information

Data Provider: Apollo.io (FREE API)

API Used: api/v1/mixed_people/organization_top_people

Data Retrieved:

- First Name, Last Name
 - Job Title
 - Email (if available)
 - Phone (if available)
 - LinkedIn URL
-

Stage 5: Verification Agent

File: agents/verification_agent.py

Purpose: Final lead scoring and validation

Verification Rules:

1. Company must be "accepted"
2. At least one role with decision_power ≥ 7
3. Confidence score calculation

Confidence Score Factors:

- Company status: + 40%
- High-power roles: + 30%
- Enriched contacts: + 20%
- Multiple decision-makers: + 10%

Thresholds:

- $\geq 75\%$: Lead VERIFIED ✓
 - $< 75\%$: Lead REJECTED ✗
-

Tools & Technologies

Core Stack

Component	Technology	Purpose
Language	Python 3.x	Core development
LLM	Groq + Llama 3.3 70B	Natural language analysis
Web Search	DuckDuckGo (ddgs)	Company info, news
Contacts API	Apollo.io	Verified contact data
Terminal UI	Rich	Colored output, tables
Storage	JSON	Local data persistence

Dependencies

```
groq           # LLM API client
python-dotenv # Environment variables
rich           # Terminal UI
requests       # HTTP client
ddgs          # DuckDuckGo search
```

File Structure

```
Prototype/
├── agent.py               # Main CLI interface
├── workflow.py            # Pipeline orchestrator
├── database.py            # JSON storage operations
├── requirements.txt        # Python dependencies
├── .env                    # API keys (secret)
├── .env.example            # API keys template
|
└── agents/
    └── __init__.py
```

```

    └── discovery_agent.py      # Stage 1
    └── structure_agent.py     # Stage 2
    └── role_agent.py          # Stage 3
    └── enrichment_agent.py    # Stage 4
    └── verification_agent.py  # Stage 5

    └── services/
        └── __init__.py
        └── web_search.py       # DuckDuckGo search
        └── linkedin_search.py  # LinkedIn people search
        └── apollo_client.py    # Apollo API client

    └── data/
        └── leads.json          # Saved leads database

```

Data Flow

```

sequenceDiagram
    participant U as User
    participant A as Agent CLI
    participant W as Workflow
    participant D as Discovery
    participant S as Structure
    participant R as Roles
    participant E as Enrichment
    participant V as Verification
    participant DB as Database

    U->>A: analyze TCS
    A->>W: run_pipeline("TCS")

    W->>D: run("TCS")
    D-->>D: Web Search (DuckDuckGo)
    D-->>D: LLM Analysis (Groq)
    D->>W: {name, industry, size}

    W->>S: run(company_data)
    S-->>S: Map decision-makers by size
    S->>W: {departments, targets}

    W->>R: run(company, size)
    R-->>R: LinkedIn Search
    R-->>R: Score people
    R->>W: {people, scores}

    W->>E: run(company, roles)
    E-->>E: Apollo API call
    E->>W: {contacts}

    W->>V: run(all_data)
    V-->>V: Calculate confidence

```

```
V->>W: {status, score}  
  
W->>DB: save_lead(lead_data)  
W->>A: lead_result  
A->>U: Display result
```

API Integrations

1. Groq API (LLM)

- **Model:** llama-3.3-70b-versatile
- **Usage:** Company validation, role analysis
- **Cost:** Free tier available
- **Rate Limit:** ~30 requests/minute

2. DuckDuckGo (Web Search)

- **Library:** ddgs
- **Usage:** Company info, news, LinkedIn pages
- **Cost:** Free, no API key needed
- **Rate Limit:** Soft limits, may block heavy usage

3. Apollo.io (Contact Enrichment)

- **Endpoint:** /v1/mixed_people/organization_top_people
 - **Usage:** Get decision-makers at companies
 - **Cost:** Free tier (limited credits)
 - **Data:** Name, title, email, phone, LinkedIn
-

CLI Commands

Command	Example	Description
analyze	analyze Microsoft, Roles: CEO	Full pipeline
enrich	enrich Satya Nadella at Microsoft	Direct person lookup
history	history	View past leads
export	export	Save to CSV
clear	clear	Clear screen
help	help	Show commands
quit	quit	Exit agent

Sample Output

✓ Lead VERIFIED (Confidence: 90%)

Company: Wipro
Industry: IT Services

Size: enterprise
Location: India
Website: wipro.com
Growth Signals: top-performing stock
Sources: news

Decision Makers Found (5 accepted):

- ✓ [Target Role] - CEO (Power: 10/10) (suggested role)
- ✓ [Target Role] - CTO (Power: 9/10) (suggested role)
- ✓ [Target Role] - VP Sales (Power: 7/10) (suggested role)

Wipro is a verified B2B lead with multiple decision-makers
→ Proceed with outreach to the identified decision-makers

Configuration

Environment Variables (.env)

```
# Groq LLM API (required)
GROQ_API_KEY=gskxxxxxxxxxxxxxx

# Apollo Contact API (optional, enhances results)
APOLLO_API_KEY=xxxxxxxxxxxx
```

Getting API Keys

1. **Groq:** <https://console.groq.com> (free)
2. **Apollo:** <https://www.apollo.io> → API Settings → Create Key

- Select: api/v1/mixed_people/organization_top_people
-

Ethical Considerations

- ✓ Only collects publicly available data
 - ✓ Uses licensed data providers (Apollo)
 - ✓ No scraping of personal information
 - ✓ No data fabrication
 - ✓ Stores only: name, title, company, professional contact
-

Limitations

1. **LinkedIn Search:** DuckDuckGo may not find specific people (blocked by LinkedIn)
 2. **Apollo Free Tier:** Limited credits, may not return emails
 3. **LLM Knowledge:** Based on training data, not real-time
 4. **Rate Limits:** All APIs have usage limits
-

Future Enhancements

- [] Add Hunter.io for email verification
- [] Implement web scraping for company websites
- [] Add CRM integration (Salesforce, HubSpot)
- [] Create web UI dashboard
- [] Add email outreach automation