

# Complete n8n Agent Workflows Analysis

## Automotive & Email Automation Systems

---

### Executive Summary

This comprehensive analysis covers **20 n8n workflow automation files** divided into two categories:

- **10 Car Agent Workflows** - Automotive industry applications
- **10 Email Agent Workflows** - Email automation and management

All workflows leverage LangChain, vector databases, and various AI services (OpenAI, Anthropic Claude, HuggingFace, Cohere) to provide intelligent automation using RAG (Retrieval-Augmented Generation) architecture.

---

## PART 1: CAR AGENT WORKFLOWS

### Overview

10 AI-powered workflows designed for automotive and vehicle-related applications including safety systems, fleet management, insurance, and connected vehicles.

### Car Agent Inventory

#### 1. ADAS Event Annotator

**Purpose:** Advanced Driver Assistance Systems event annotation

**File:** `adas_event_annotator.json`

**Endpoint:** `/adas_event_annotator`

**Nodes:** 11

#### Tech Stack:

- Embedding Provider: Cohere
- LLM: OpenAI Chat
- Vector Store: Supabase
- Logging: Google Sheets

**Use Case:** Annotates and categorizes events from ADAS sensors including lane departure warnings, collision alerts, adaptive cruise control events, and other safety system activations.

---

## 2. Autonomous Vehicle Log Summarizer

**Purpose:** Summarizes autonomous vehicle operation logs

**File:** `autonomous_vehicle_log_summarizer.json`

**Endpoint:** `/autonomous_vehicle_log_summarizer`

**Nodes:** 11

**Tech Stack:**

- Embedding Provider: HuggingFace
- LLM: OpenAI Chat
- Vector Store: Weaviate
- Logging: Google Sheets

**Use Case:** Processes large volumes of AV telemetry data and generates concise summaries highlighting critical events, disengagements, and operational patterns.

---

## 3. Car Insurance Quote Generator

**Purpose:** Generates personalized car insurance quotes

**File:** `car_insurance_quote_generator.json`

**Endpoint:** `/car_insurance_quote_generator`

**Nodes:** 11

**Tech Stack:**

- Embedding Provider: HuggingFace
- LLM: Anthropic (Claude)
- Vector Store: Pinecone
- Logging: Google Sheets

**Use Case:** Analyzes driver profiles, vehicle information, and historical data to generate accurate, personalized insurance quotes using RAG to reference policy documents and pricing models.

---

## 4. Connected Car Alert

**Purpose:** Processes and manages connected vehicle alerts

**File:** `connected_car_alert.json`

**Endpoint:** `/connected_car_alert`

**Nodes:** 11

**Tech Stack:**

- Embedding Provider: OpenAI
- LLM: OpenAI Chat
- Vector Store: Redis
- Logging: Google Sheets

**Use Case:** Monitors real-time telemetry from connected vehicles, categorizes alerts (maintenance, security, diagnostics), and generates appropriate notifications.

---

## 5. Dealer Lead Qualifier

**Purpose:** Qualifies potential customer leads for car dealerships

**File:** `dealer_lead_qualifier.json`

**Endpoint:** `/dealer_lead_qualifier`

**Nodes:** 11

**Tech Stack:**

- Embedding Provider: Cohere
- LLM: HuggingFace Chat
- Vector Store: Pinecone
- Logging: Google Sheets

**Use Case:** Analyzes incoming sales leads, scores them based on purchase intent and qualification criteria, and routes high-value leads to sales teams.

---

## 6. EV Battery Degradation Report

**Purpose:** Analyzes and reports on electric vehicle battery health

**File:** `ev_battery_degradation_report.json`

**Endpoint:** `/ev_battery_degradation_report`

**Nodes:** 11

**Tech Stack:**

- Embedding Provider: Cohere
- LLM: OpenAI Chat
- Vector Store: Redis
- Logging: Google Sheets

**Use Case:** Monitors EV battery performance metrics over time, predicts degradation patterns, and generates health reports for fleet managers or vehicle owners.

---

## 7. Fleet Fuel Efficiency Report

**Purpose:** Generates fuel efficiency reports for vehicle fleets

**File:** `fleet_fuel_efficiency_report.json`

**Endpoint:** `/fleet_fuel_efficiency_report`

**Nodes:** 11

**Tech Stack:**

- Embedding Provider: HuggingFace
- LLM: Anthropic (Claude)
- Vector Store: Weaviate
- Logging: Google Sheets

**Use Case:** Analyzes fleet vehicle fuel consumption, identifies inefficiencies, compares against benchmarks, and generates actionable optimization recommendations.

---

## 8. Recall Notice Tracker

**Purpose:** Tracks and manages vehicle recall notices

**File:** `recall_notice_tracker.json`

**Endpoint:** `/recall_notice_tracker`

**Nodes:** 11

**Tech Stack:**

- Embedding Provider: OpenAI
- LLM: Anthropic (Claude)
- Vector Store: Pinecone
- Logging: Google Sheets

**Use Case:** Monitors official recall databases, matches recalls to fleet or customer vehicles, and generates automated notification workflows.

---

## 9. Ride-Share Surge Predictor

**Purpose:** Predicts ride-sharing demand surges

**File:** ride#U2011share\_surge\_predictor.json

**Endpoint:** /ride-share\_surge\_predictor

**Nodes:** 11

#### Tech Stack:

- Embedding Provider: Cohere
- LLM: Anthropic (Claude)
- Vector Store: Supabase
- Logging: Google Sheets

**Use Case:** Analyzes historical demand patterns, events, weather, and real-time data to predict surge pricing periods and optimize driver positioning.

---

## 10. VIN Decoder

**Purpose:** Decodes Vehicle Identification Numbers

**File:** vin\_decoder.json

**Endpoint:** /vin\_decoder

**Nodes:** 11

#### Tech Stack:

- Embedding Provider: HuggingFace
- LLM: HuggingFace Chat
- Vector Store: Redis
- Logging: Google Sheets

**Use Case:** Decodes VINs to extract vehicle specifications, manufacturing details, recall history, and provides comprehensive vehicle information reports.

---

## Car Agent Technology Distribution

### Embedding Providers

- **OpenAI:** 3 workflows
- **Cohere:** 3 workflows
- **HuggingFace:** 4 workflows

## LLM Providers

- **OpenAI Chat:** 4 workflows
- **Anthropic (Claude):** 3 workflows
- **HuggingFace Chat:** 3 workflows

## Vector Databases

- **Redis:** 3 workflows
  - **Pinecone:** 3 workflows
  - **Supabase:** 2 workflows
  - **Weaviate:** 2 workflows
- 

# PART 2: EMAIL AGENT WORKFLOWS

## Overview

10 AI-powered workflows designed for email automation, marketing, customer service, and business process optimization.

## Email Agent Inventory

### 1. Auto Archive Promotions

**Purpose:** Automatically archives promotional emails

**File:** `auto_archive_promotions.json`

**Endpoint:** `/auto-archive-promotions`

**Nodes:** 12

#### Tech Stack:

- Embedding Provider: OpenAI
- LLM: OpenAI Chat
- Vector Store: Pinecone
- Logging: Google Sheets
- Alerts: Slack

**Use Case:** Identifies promotional and marketing emails using AI classification, automatically archives them to keep inbox clean while maintaining searchability.

---

## 2. Auto Reply to FAQs

**Purpose:** Automatically responds to frequently asked questions

**File:** `auto_reply_to_faqs.json`

**Endpoint:** `/auto-reply-to-faqs`

**Nodes:** 12

**Tech Stack:**

- Embedding Provider: Cohere
- LLM: Anthropic (Claude)
- Vector Store: Pinecone
- Logging: Google Sheets
- Alerts: Slack

**Use Case:** Uses RAG to match incoming questions against a knowledge base and generates contextually appropriate responses, reducing support team workload.

---

## 3. Daily Email Digest

**Purpose:** Generates daily email summaries

**File:** `daily_email_digest.json`

**Endpoint:** `/daily-email-digest`

**Nodes:** 12

**Tech Stack:**

- Embedding Provider: OpenAI
- LLM: Anthropic (Claude)
- Vector Store: Supabase
- Logging: Google Sheets
- Alerts: Slack

**Use Case:** Aggregates and summarizes important emails from the previous day, categorizes by priority, and sends a consolidated digest to reduce email overwhelm.

---

## 4. Follow-up Emails

**Purpose:** Manages automated follow-up email sequences

**File:** `follow-up_emails.json`

**Endpoint:** `/follow-up-emails`

**Nodes:** 12

**Tech Stack:**

- Embedding Provider: OpenAI
- LLM: Anthropic (Claude)
- Vector Store: Weaviate
- Logging: Google Sheets
- Alerts: Slack

**Use Case:** Tracks email engagement, identifies emails requiring follow-up, and generates personalized follow-up messages based on conversation context.

---

## 5. Forward Attachments

**Purpose:** Intelligently forwards email attachments

**File:** `forward_attachments.json`

**Endpoint:** `/forward-attachments`

**Nodes:** 12

**Tech Stack:**

- Embedding Provider: OpenAI
- LLM: OpenAI Chat
- Vector Store: Pinecone
- Logging: Google Sheets
- Alerts: Slack

**Use Case:** Analyzes email attachments, categorizes them by type and content, and automatically forwards to appropriate team members or systems based on rules.

---

## 6. Lead to HubSpot

**Purpose:** Syncs qualified leads to HubSpot CRM

**File:** `lead_to_hubspot.json`

**Endpoint:** `/lead-to-hubspot`

**Nodes:** 12

**Tech Stack:**

- Embedding Provider: Cohere
- LLM: Anthropic (Claude)

- Vector Store: Supabase
- Logging: Google Sheets
- Alerts: Slack

**Use Case:** Extracts lead information from emails, enriches with AI-generated insights, and automatically creates/updates HubSpot contacts with proper categorization.

---

## 7. Mailchimp Campaign Tracking

**Purpose:** Tracks and analyzes Mailchimp campaign performance

**File:** `mailchimp_campaign_tracking.json`

**Endpoint:** `/mailchimp-campaign-tracking`

**Nodes:** 12

**Tech Stack:**

- Embedding Provider: OpenAI
- LLM: Anthropic (Claude)
- Vector Store: Supabase
- Logging: Google Sheets
- Alerts: Slack

**Use Case:** Monitors Mailchimp campaign metrics, generates AI-powered performance analysis, and provides optimization recommendations.

---

## 8. Parse Invoice Emails

**Purpose:** Extracts and processes invoice data from emails

**File:** `parse_invoice_emails.json`

**Endpoint:** `/parse-invoice-emails`

**Nodes:** 12

**Tech Stack:**

- Embedding Provider: OpenAI
- LLM: Anthropic (Claude)
- Vector Store: Weaviate
- Logging: Google Sheets
- Alerts: Slack

**Use Case:** Automatically detects invoice emails, extracts key data (amounts, due dates, vendor info), and integrates with accounting systems.

---

## 9. Product Launch Email

**Purpose:** Manages product launch email campaigns

**File:** `product_launch_email.json`

**Endpoint:** `/product-launch-email`

**Nodes:** 12

**Tech Stack:**

- Embedding Provider: Cohere
- LLM: OpenAI Chat
- Vector Store: Supabase
- Logging: Google Sheets
- Alerts: Slack

**Use Case:** Coordinates product launch email sequences, personalizes messaging based on customer segments, and tracks engagement metrics.

---

## 10. SendGrid Bounce Alert

**Purpose:** Monitors and alerts on SendGrid email bounces

**File:** `sendgrid_bounce_alert.json`

**Endpoint:** `/sendgrid-bounce-alert`

**Nodes:** 12

**Tech Stack:**

- Embedding Provider: OpenAI
- LLM: Anthropic (Claude)
- Vector Store: Weaviate
- Logging: Google Sheets
- Alerts: Slack

**Use Case:** Tracks email delivery failures, categorizes bounce types (hard/soft), and generates actionable alerts for list cleaning and deliverability management.

---

## Email Agent Technology Distribution

### Embedding Providers

- **OpenAI:** 7 workflows (70%)
- **Cohere:** 3 workflows (30%)

### LLM Providers

- **Anthropic (Claude):** 7 workflows (70%)
- **OpenAI Chat:** 3 workflows (30%)

### Vector Databases

- **Supabase:** 4 workflows (40%)
- **Pinecone:** 3 workflows (30%)
- **Weaviate:** 3 workflows (30%)

### Additional Features

- **Slack Alerts:** 10/10 workflows (100%)
  - **Google Sheets Logging:** 10/10 workflows (100%)
- 

## COMBINED ANALYSIS

### Overall Statistics

#### Workflow Count

- **Total Workflows:** 20
- **Car Agent:** 10
- **Email Agent:** 10

#### Node Architecture

- **Car Agent:** 11 nodes per workflow
- **Email Agent:** 12 nodes per workflow (includes Slack alerting)

## Common Components

All 20 workflows share these core components:

1. Webhook Trigger (POST endpoint)
2. Text Splitter (400 char chunks, 40 char overlap)
3. Embedding Generation
4. Vector Store (Insert & Query operations)
5. Vector Store Tool
6. Chat Model (LLM)
7. Window Memory (conversation context)
8. RAG Agent
9. Google Sheets (logging)

## Email-Only Components

- Slack Alert node (error handling)
- 

## Technology Stack Comparison

### Embedding Providers (All 20 Workflows)

Provider	Car Agent	Email Agent	Total	Percentage
OpenAI	3	7	10	50%
HuggingFace	4	0	4	20%
Cohere	3	3	6	30%

### LLM Providers (All 20 Workflows)

Provider	Car Agent	Email Agent	Total	Percentage
Anthropic (Claude)	3	7	10	50%
OpenAI Chat	4	3	7	35%
HuggingFace Chat	3	0	3	15%

## Vector Databases (All 20 Workflows)

Database	Car Agent	Email Agent	Total	Percentage
Pinecone	3	3	6	30%
Supabase	2	4	6	30%
Redis	3	0	3	15%
Weaviate	2	3	5	25%

## RAG Architecture Explained

All workflows implement Retrieval-Augmented Generation (RAG):

### Data Flow

1. Webhook receives input
2. Text Splitter chunks the content
3. Embeddings converts text to vectors
4. Vector Store stores embeddings for retrieval
5. When querying:
  - a. User query is embedded
  - b. Similar chunks retrieved from Vector Store
  - c. Retrieved context + query sent to LLM
  - d. LLM generates contextualized response
6. Results logged to Google Sheets
7. (Email only) Errors sent to Slack

### Benefits of RAG

- **Accuracy:** Grounds responses in actual data
- **Context:** Maintains conversation history
- **Scalability:** Handles large knowledge bases
- **Flexibility:** Easily update knowledge without retraining

## Required Credentials & APIs

## Core Requirements (All Workflows)

1. **Google Sheets OAuth2 API** - Result logging
2. **Vector Database APIs:**
  - Pinecone API (6 workflows)
  - Supabase API (6 workflows)
  - Redis API (3 workflows)
  - Weaviate API (5 workflows)

## Embedding Services

3. **OpenAI API** - 10 workflows
4. **Cohere API** - 6 workflows
5. **HuggingFace API** - 4 workflows

## LLM Services

6. **Anthropic API** - 10 workflows
7. **OpenAI API** - 7 workflows (some overlap with embeddings)
8. **HuggingFace API** - 3 workflows (some overlap with embeddings)

## Email-Specific

9. **Slack API** - All 10 email workflows

## Configuration Placeholders

Each workflow contains these placeholders to update:

- `SHEET_ID` - Your Google Sheet document ID
  - `SHEETS_API` - Your Google Sheets credential ID
  - `PINECONE_API` / `REDIS_API` / `SUPABASE_API` / `WEAVIATE_API`
  - `ANTHROPIC_API` / `OPENAI_API` / `COHERE_API` / `HF_API`
  - `SLACK_API` - (Email workflows only)
-

## Use Case Matrix

### Car Agent Applications

Workflow	Industry Sector	Primary User	Business Value
ADAS Event Annotator	Automotive Safety	OEMs, Tier-1 Suppliers	Safety compliance, data analysis
AV Log Summarizer	Autonomous Vehicles	AV Developers, Fleet Ops	Debugging, compliance reporting
Insurance Quote	Insurance	Insurance Companies	Automated underwriting
Connected Car Alert	IoT/Telematics	Fleet Managers, OEMs	Proactive maintenance
Dealer Lead Qualifier	Sales/Marketing	Car Dealerships	Sales efficiency
EV Battery Report	Electric Vehicles	Fleet Managers, Owners	Asset management
Fleet Fuel Report	Fleet Management	Fleet Managers	Cost optimization
Recall Notice Tracker	Compliance	OEMs, Dealerships	Safety compliance
Ride-Share Predictor	Transportation	Ride-share Companies	Revenue optimization
VIN Decoder	General Automotive	Multiple	Data enrichment

### Email Agent Applications

Workflow	Department	Primary User	Business Value
Auto Archive Promotions	IT/Personal	All Users	Productivity
Auto Reply FAQs	Customer Service	Support Teams	Cost reduction
Daily Email Digest	General	Executives, Managers	Time management
Follow-up Emails	Sales	Sales Teams	Pipeline management
Forward Attachments	Operations	Admin Teams	Workflow automation
Lead to HubSpot	Marketing/Sales	Marketing Teams	CRM integration
Mailchimp Tracking	Marketing	Marketing Teams	Campaign optimization
Parse Invoice	Finance	Accounting Teams	AP automation

Workflow	Department	Primary User	Business Value
Product Launch	Marketing	Product Managers	Launch coordination
SendGrid Bounce	IT/Marketing	Email Admins	Deliverability

## Deployment Guide

### Prerequisites

1. n8n instance (self-hosted or cloud)
2. All required API credentials
3. Google Sheet for logging
4. Slack workspace (email workflows)

### Deployment Steps

#### 1. Import Workflows

```
bash

# Import each JSON file into n8n
- Navigate to n8n dashboard
- Click "Import from File"
- Select workflow JSON
- Repeat for all 20 workflows
```

#### 2. Configure Credentials

For each workflow:

- Open workflow in n8n editor
- Click on each node requiring credentials
- Add/select appropriate credential
- Save credentials

#### 3. Update Placeholders

Replace in each workflow:

- SHEET\_ID → Your Google Sheet ID
- SHEETS\_API → Your Sheets credential
- [DB]\_API → Your vector DB credential
- [LLM]\_API → Your LLM credential
- SLACK\_API → Your Slack credential (email only)

## 4. Test Webhooks

```
bash

# Test each endpoint
curl -X POST https://your-n8n-instance.com/webhook/[endpoint] \
-H "Content-Type: application/json" \
-d '{"test": "data"}'
```

## 5. Monitor Logs

- Check Google Sheets for execution logs
- Monitor Slack for error alerts (email)
- Review n8n execution history

## Best Practices

1. **Start Small:** Deploy 2-3 workflows first
2. **Test Thoroughly:** Verify each workflow independently
3. **Monitor Costs:** Track API usage across providers
4. **Set Limits:** Configure rate limiting on webhooks
5. **Backup Config:** Export workflows regularly
6. **Version Control:** Track changes to workflows
7. **Document Custom:** Add notes for any customizations

---

## Performance Considerations

### Latency Factors

1. **Embedding Generation:** 100-500ms
2. **Vector Search:** 50-200ms
3. **LLM Inference:** 1-5 seconds

4. **Total Expected:** 2-7 seconds per request

## Optimization Tips

1. **Chunk Size:** Adjust based on content (current: 400 chars)
2. **Vector DB Choice:** Redis fastest, Pinecone most scalable
3. **LLM Selection:** OpenAI fastest, Claude most capable
4. **Caching:** Implement for common queries
5. **Batch Processing:** Group similar requests

## Scalability

- **Concurrent Requests:** Depends on n8n instance
  - **Vector DB Limits:** Check provider quotas
  - **API Rate Limits:** Monitor usage
  - **Cost Scaling:** Linear with request volume
- 

## Cost Estimation

### Per-Request Costs (Approximate)

#### Embedding Costs

- OpenAI: \$0.0001 per request
- Cohere: \$0.0001 per request
- HuggingFace: Free tier available

#### LLM Costs

- Claude (Sonnet): \$0.003 per request
- GPT-4: \$0.005 per request
- GPT-3.5: \$0.0015 per request
- HuggingFace: Free tier available

#### Vector DB Costs

- Pinecone: \$70/month (starter)
- Supabase: Free tier available

- Redis: Variable by provider
- Weaviate: Free tier available

## Supporting Services

- Google Sheets: Free
- Slack: Free tier available
- n8n: Self-hosted free, Cloud \$20+/month

## Monthly Cost Example (1000 requests/workflow)

Workflow: Auto Reply to FAQs

- Embeddings (Cohere): \$1
- LLM (Claude): \$30
- Vector DB (Pinecone): \$70 (shared)
- Total: ~\$31/month per workflow

## Security Considerations

### API Key Management

- Use n8n credential system
- Never hardcode in workflows
- Rotate keys regularly
- Use environment variables
- Limit key permissions

### Data Privacy

-  Sensitive data in vector stores
-  LLM providers may train on data
-  Logs contain PII
- Implement data retention policies
- Use private vector DB instances
- Consider self-hosted LLMs for sensitive data

## Access Control

- Secure webhook endpoints
  - Implement authentication
  - Rate limiting
  - IP whitelisting
  - Monitor unusual activity
- 

## Troubleshooting Guide

### Common Issues

#### 1. Webhook Not Responding

Check:

- Webhook path configured correctly
- n8n workflow activated
- Firewall/network settings
- Request format (POST, JSON)

#### 2. Vector Store Errors

Check:

- API credentials valid
- Index/collection exists
- Quota not exceeded
- Embedding dimensions match

#### 3. LLM Timeouts

Solutions:

- Reduce chunk size
- Implement retry logic
- Check API status
- Monitor rate limits

#### 4. Slack Alerts Not Sending

#### Check:

- Slack API token valid
- Channel exists and bot invited
- Slack workspace permissions

## 5. Google Sheets Errors

#### Check:

- OAuth token valid
- Sheet ID correct
- Permissions (edit access)
- Sheet structure matches expected

## Customization Guide

### Modify Text Chunking

```
json

{
  "parameters": {
    "chunkSize": 400, // Increase for longer context
    "chunkOverlap": 40 // Increase for better continuity
  }
}
```

### Change LLM Provider

```
json

// Replace node type:
"type": "@n8n/n8n-nodes-langchain.lmChatAnthropic"
// With:
"type": "@n8n/n8n-nodes-langchain.lmChatOpenAi"
// Update credentials accordingly
```

### Add Custom System Prompts

```
json
```

```
{  
  "parameters": {  
    "options": {  
      "systemMessage": "Custom instructions here"  
    }  
  }  
}
```

## Modify Memory Window

```
json  
  
{  
  "type": "@n8n/n8n-nodes-langchain.memoryBufferWindow",  
  "parameters": {  
    "contextWindowLength": 5 // Adjust conversation history  
  }  
}
```

## Future Enhancements

### Potential Additions

1. **Multi-modal Support:** Image/audio processing
2. **Advanced RAG:** Multi-query, re-ranking
3. **Custom Tools:** Function calling integration
4. **A/B Testing:** Compare LLM providers
5. **Analytics Dashboard:** Performance metrics
6. **Feedback Loop:** Human-in-the-loop refinement
7. **Multi-language:** Translation integration
8. **Voice Interface:** Speech-to-text integration

### Integration Opportunities

- Salesforce CRM
- Zendesk
- Jira
- Microsoft Teams

- Discord
  - WhatsApp Business
  - Custom APIs
- 

## Conclusion

This collection of 20 n8n workflows represents a comprehensive, production-ready AI automation suite spanning automotive and email domains. Each workflow demonstrates:

- RAG Architecture:** Robust retrieval-augmented generation
- Error Handling:** Slack alerts for email workflows
- Logging:** Complete audit trail in Google Sheets
- Modularity:** Easily customizable components
- Scalability:** Cloud-native vector databases
- Flexibility:** Multiple LLM/embedding provider options

## Key Takeaways

1. **Consistent Architecture:** All workflows follow proven RAG patterns
2. **Technology Diversity:** Multiple providers allow optimization
3. **Production Ready:** Error handling and logging built-in
4. **Cost Effective:** Free tiers available for most services
5. **Extensible:** Easy to customize for specific needs

## Getting Started Recommendations

**For Car Industry:** Start with VIN Decoder and Connected Car Alert

**For Email Automation:** Start with Auto Reply FAQs and Daily Digest

**For Developers:** Study one workflow thoroughly, then replicate pattern

## Support Resources

- n8n Documentation: <https://docs.n8n.io>
  - LangChain Docs: <https://python.langchain.com>
  - Vector DB Tutorials: Provider-specific docs
  - Community: n8n Discord, Reddit
-

**Document Version:** 1.0

**Last Updated:** January 29, 2026

**Total Workflows Analyzed:** 20

**Total Pages:** Complete Analysis