

★ Member-only story

7 AgentKit Blueprints That Will Help You Build AI Agents (In Minutes, Not Hours)

30 min read · Oct 9, 2025



Joe Njenga

Follow



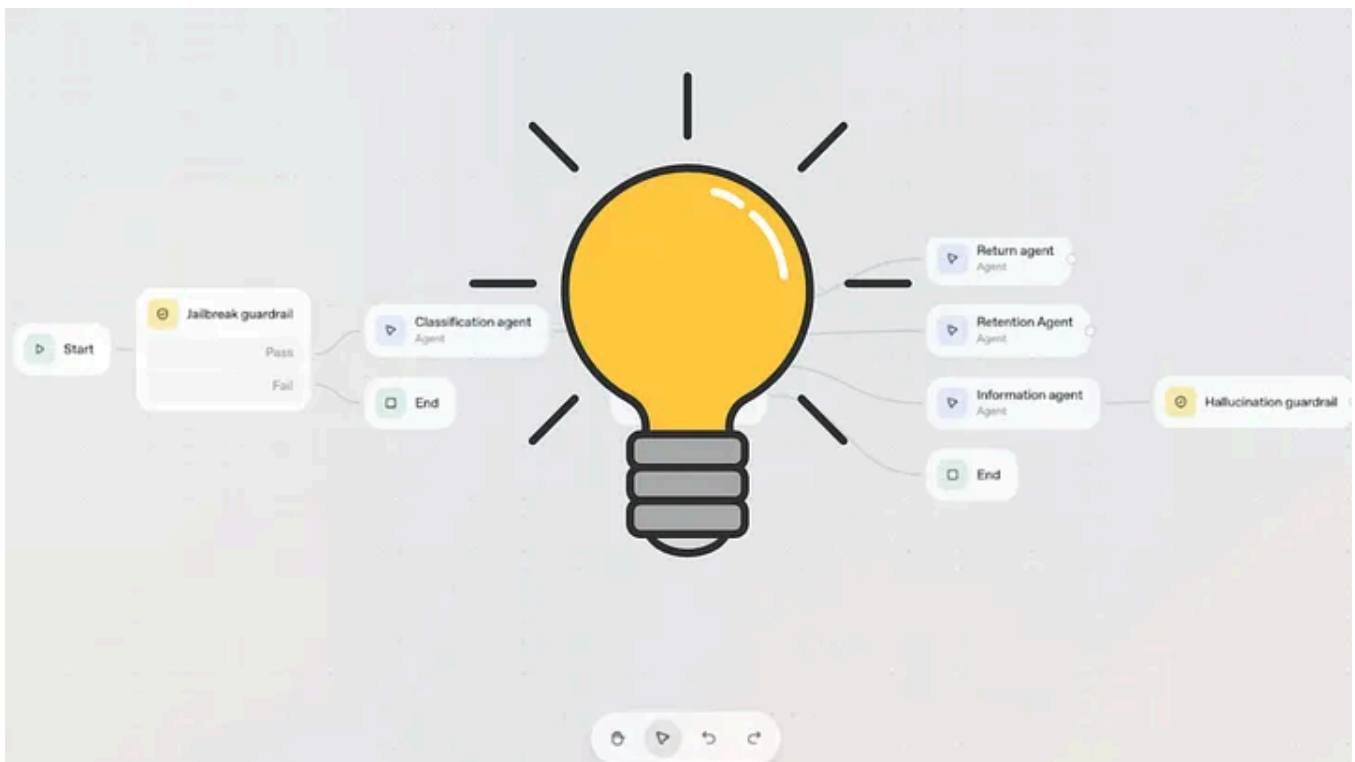
Listen



Share



More



If you are struggling to build an AI Agent, you don't have to — *first forget coding for a minute.*

These AgentKit blueprints will help you learn and build your AI agents in minutes using OpenAI AgentKit.

In my previous tutorial, *I tested OpenAI's AgentKit* and walked you through every component.

Now, I'd like to show you what you can build with it.

I spent time thinking through possible use cases and came up with these AI Agent blueprints you can use to build agents using the OpenAI AgentKit.

These agents solve real business problems.

They are based on AgentKit's strengths: conversation routing, knowledge base search, guardrails, and MCP integrations.

Each idea includes:

- The specific problem it solves
- Who needs it
- Technical workflow breakdown
- Which nodes do you use

They're the easiest way for you to learn core patterns of using the AgentKit to build agents.

Let's get into it.

. . .

1) Customer Support Agent with Smart Routing

Problem

Support tickets are sent to the wrong department, and customers wait while agents figure out who should handle their request.

Ideal For

E-commerce stores, SaaS companies, or any business receiving more than 50 support requests daily.

Especially useful when you have specialized teams for returns, technical issues, and general inquiries.

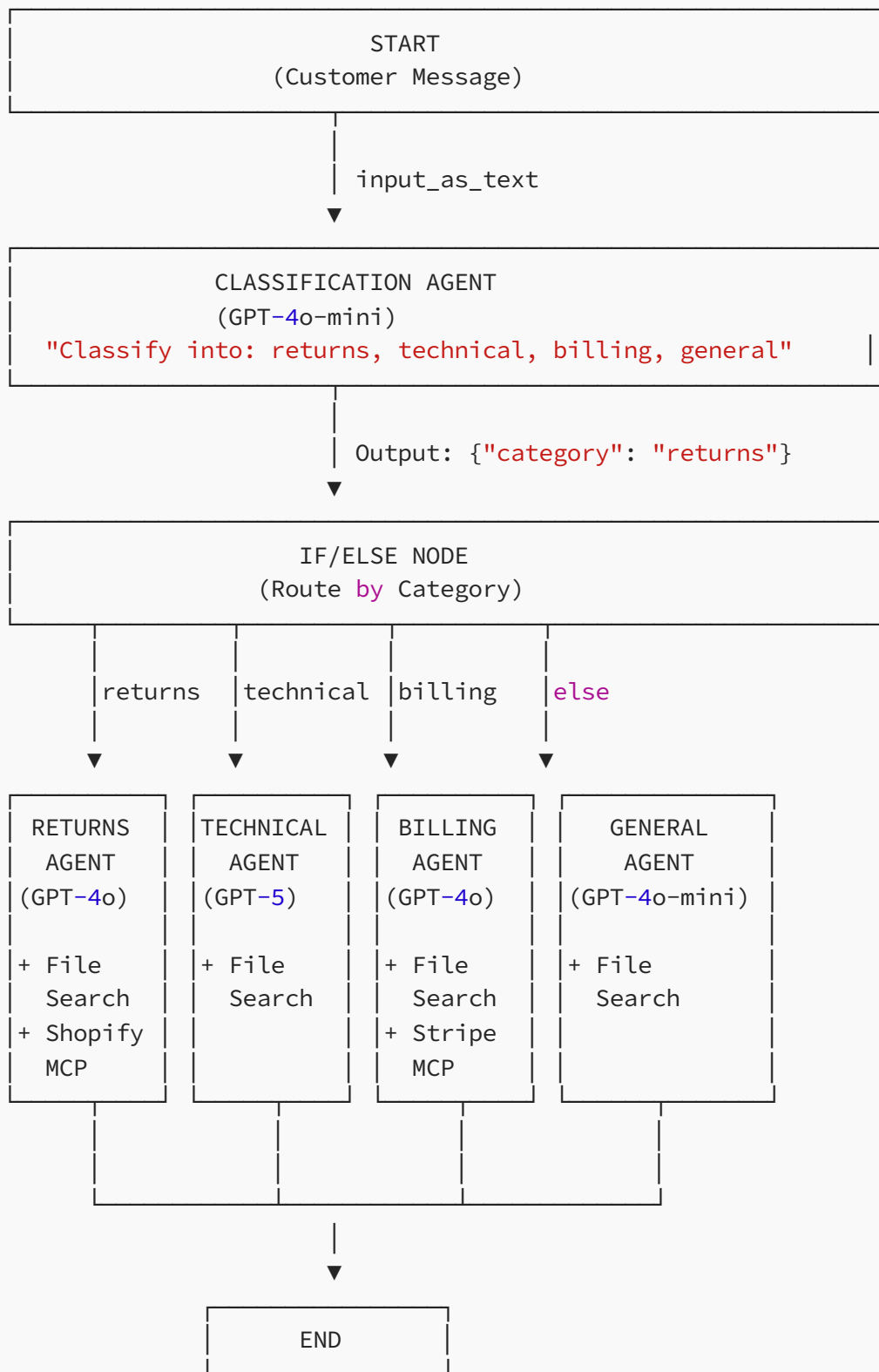
How it Works

When a user messages your support system, the classification agent reads the message and determines the intent: a return request, a technical problem, a billing question, or a general inquiry.

Based on classification, the workflow routes to specialized agents trained for each category.

Each specialized agent has specific instructions and access to relevant documentation.

Technical Flow Diagram



Example

Customer writes: “I ordered a blue sweater last week, but received a red one. Can I return it?”

The classification agent identifies this as a return request. Routes to the Returns Agent. That agent asks for the order number, confirms the return policy applies, and generates

a return label through the Shopify MCP.

Another customer writes: “My login isn’t working after the latest update.”

Classification identifies this as technical and routes to the Technical Agent with access to troubleshooting docs.

Nodes to Use :

Start node → Receives customer message as `input_as_text`

Classification Agent → Analyzes message intent

- Model: GPT-4o-mini (cost-effective)
- Instructions: *“You classify customer messages into four categories: returns, technical, billing, or general. Base your classification on the primary intent. Examples: ‘I want my money back’ = returns. ‘Can’t log in’ = technical. ‘Why was I charged twice?’ = billing.”*
- Output format: JSON with schema:

```
{  "category": "returns" // or "technical", "billing", "general"}
```

If/else node → Routes based on `{{output.category}}`

```
Case 1: {{output.category}} == "returns" → Returns Agent
Case 2: {{output.category}} == "technical" → Technical Agent
Case 3: {{output.category}} == "billing" → Billing Agent
Else → General Support Agent
```

Returns Agent

- Model: GPT-4o
- Instructions: *“You help process returns. Ask for the order number. Check our 30-day return policy. Collect the reason for the return. If eligible, inform the customer they’ll receive a return label via email.”*

- Tools: File search (return policy PDF), MCP (Shopify for order lookup)

Technical Agent

- Model: GPT-5 (needs reasoning for troubleshooting)
- Instructions: *“You troubleshoot technical issues. Ask clarifying questions. Search documentation for solutions. Provide step-by-step guidance. If unresolved after three attempts, escalate to human support.”*
- Tools: File search (technical documentation, FAQs)

Billing Agent

- Model: GPT-4o
- Instructions: *“You handle billing questions, search transaction history. Explain charges clearly. For refund requests, verify eligibility and process if approved.”*
- Tools: File search (billing policies), MCP (Stripe for transaction lookup)

General Agent

- Model: GPT-4o-mini
- Instructions: *“You answer general questions about products, shipping, and policies. Be helpful and concise. Direct complex issues to specialized agents.”*
- Tools: File search (general FAQ, product catalog)

End node → Closes conversation

• • •

2) Product Documentation Agent

Problem

Customers can't find answers in your documentation. They submit support tickets for questions already documented.

Your support team spends hours answering the same questions over and over.

Ideal For

SaaS companies, API providers, developer tools, or any product with technical documentation.

Especially valuable if you have 100+ pages of docs that customers struggle to navigate.

How it Works

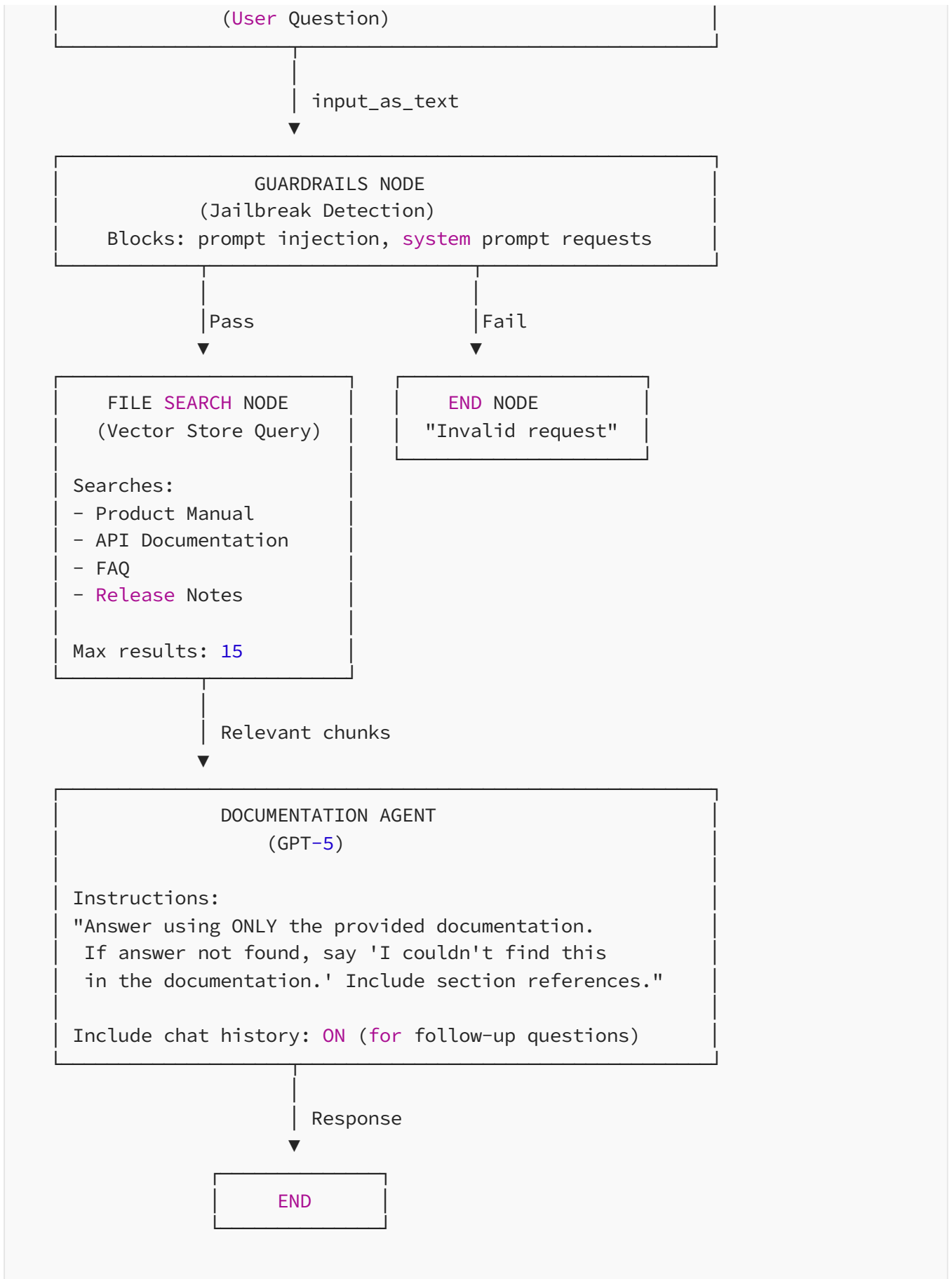
A user asks a question about your product. The agent searches your uploaded documentation using vector search (RAG).

It finds relevant sections and generates an accurate answer from your docs alone.

If the answer isn't in your documentation, the agent says so.

Technical Flow Diagram





Example

Customer asks: “How do I authenticate API requests?”

The file search finds relevant sections in your API docs on authentication.

The agent responds: *“To authenticate API requests, include your API key in the Authorization header: `Authorization: Bearer YOUR_API_KEY`. Find your API key in Settings > API Keys. Reference: Authentication section, page 12.”*

Follow-up question: “What if my key expires?”

The agent uses chat history to understand context. Searches for key expiration info. Responds: *“API keys expire after 90 days. You’ll receive an email notification 7 days before the expiration date. Generate a new key in Settings > API Keys. Reference: Security Best Practices, page 45.”*

Customer asks: “Can I use OAuth3 instead?”

Agent searches the docs and finds no OAuth3 information (because it doesn’t exist). Responds: *“I couldn’t find information about OAuth3 in the documentation. We currently support API key authentication and OAuth2. Would you like to know more about OAuth2?”*

Nodes to Use

Start node → Receives user question as `input_as_text`

Guardrails node → Prevents abuse

- Enable: Jailbreak detection (blocks “ignore instructions” attacks)
- Enable: Moderation (blocks inappropriate content)
- Pass → Continue to file search
- Fail → End with a polite error message

File Search node → Queries documentation

- Vector store: Upload all documentation PDFs
- Product manual
- API reference
- Troubleshooting guide
- FAQ document

- Release notes
- Query: `{{input_as_text}}`
- Max results: 15 (more context for technical questions)

Documentation Agent → Generates answer

- Model: GPT-5 (better reasoning for technical content)
- Instructions: *“You are a documentation assistant. Answer questions using ONLY the information provided from the documentation search. Always cite which document section your answer comes from (include page numbers when available). If the answer isn’t in the documentation, clearly state ‘I couldn’t find this information in the documentation’ and suggest related topics that might help. Never make up information. For code examples, use exact syntax from the docs.”*
- Include chat history: ON (enables follow-up questions)
- Output format: Text

End node → Closes conversation

. . .

3) Meeting Scheduler with Calendar Integration Agent

Problem

Back-and-forth emails to schedule meetings waste time. “Are you free Tuesday at 2 PM?” “No, but Thursday works.” “Thursday, I have a conflict.” This continues for days.

Ideal For

Sales teams, consultants, recruiters, or anyone scheduling 10+ meetings weekly.

Especially useful for client-facing roles where professional scheduling matters.

How it Works

A user requests a meeting. The agent asks for their preferred dates and times.

It checks your Google Calendar availability through MCP integration. The agent proposes available slots, and the User selects one.

The agent asks for confirmation before booking. Once approved, it creates the calendar event with all details and sends confirmation.

Technical Flow Diagram



input_as_text

SCHEDULER AGENT
(GPT-4o)

Instructions:

"You help schedule meetings. Ask for:

1. Meeting purpose
2. Preferred date/time
3. Duration needed

Then check calendar and suggest slots."

Tools: Google Calendar MCP

Include chat history: ON

Collects: purpose, date, duration

MCP NODE
(Google Calendar)

Action: Check availability

Input: Date range, duration

Output: Available time slots

Returns: Available slots

PROPOSAL AGENT
(GPT-4o)

Instructions:

"Present available time slots clearly.

Format: 'Available slots: Tuesday 2-3 PM,
Thursday 10-11 AM, Friday 3-4 PM'

Ask user to choose their preferred slot."

User selects slot

USER APPROVAL NODE

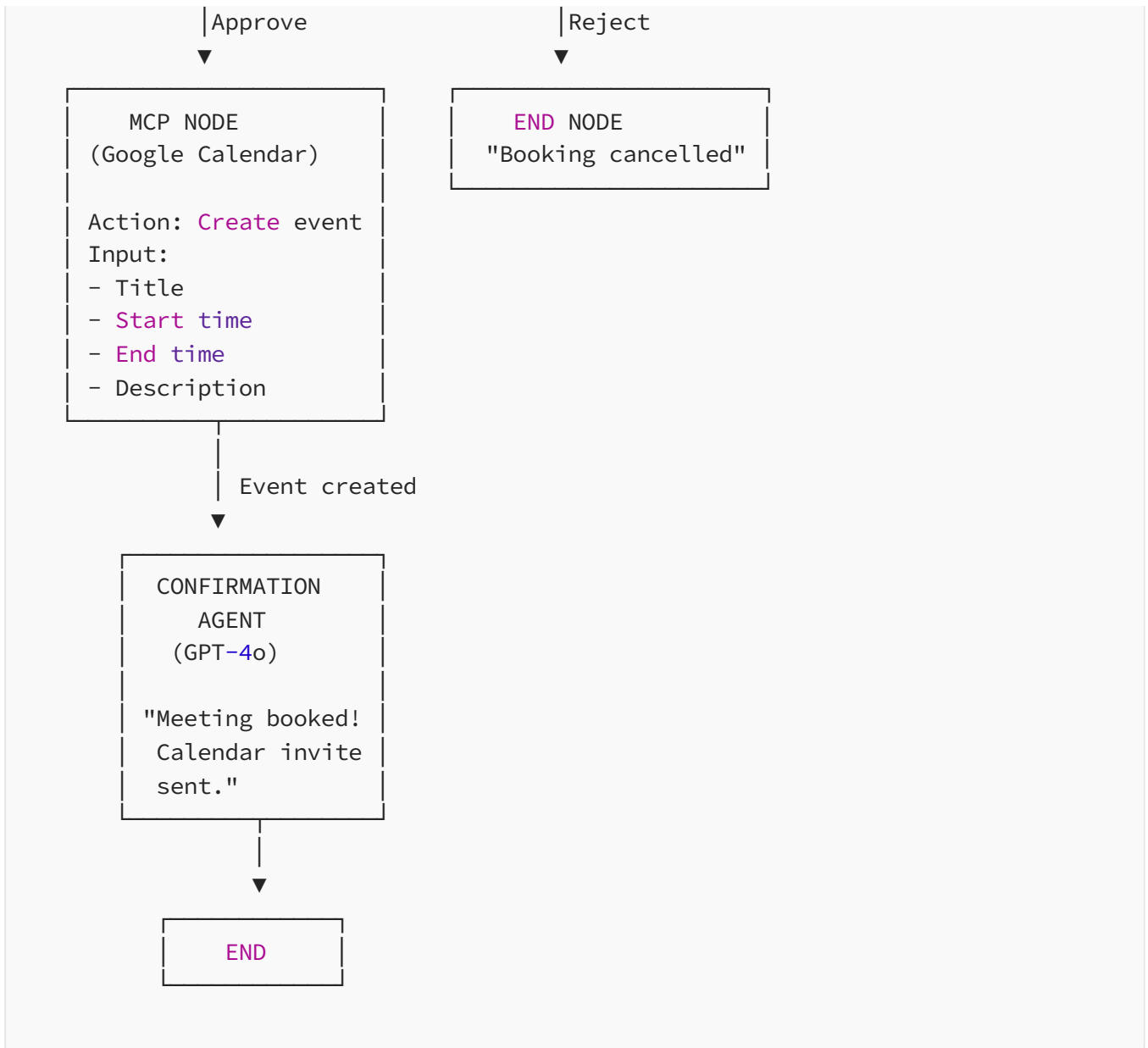
Message: "Confirm booking:

Meeting: {{meeting_purpose}}

Time: {{selected_slot}}

Duration: {{duration}}

Approve to book this meeting?"



Example

User: "I need to schedule a demo call."

Scheduler Agent: *"I'd be happy to help schedule your demo call. A few questions:*

1. What would you like to discuss in the demo?
2. What days work best for you this week?
3. How long do you need (30 minutes, 60 minutes)?"

User: "Product demo for our team. Tuesday or Wednesday would work. 30 minutes."

Agent checks Google Calendar via MCP for Tuesday and Wednesday availability.

Agent: "I found these available 30-minute slots:

- *Tuesday, March 12 at 2:00 PM EST*
- *Tuesday, March 12 at 4:00 PM EST*
- *Wednesday, March 13 at 10:00 AM EST*
- *Wednesday, March 13 at 3:00 PM EST*

Which time works best for you?”

User: “Wednesday at 10 AM.”

User Approval prompt appears: *“Confirm booking: Meeting: Product demo for team
Time: Wednesday, March 13 at 10:00 AM EST Duration: 30 minutes Approve to book
this meeting?”*

User clicks Approve.

MCP creates a calendar event with the title “Product Demo”, adds a description, and sets the time.

Confirmation Agent: “Your demo is booked for Wednesday, March 13, at 10:00 AM EST. Calendar invite sent to your email. Looking forward to the call!”

Nodes to Use

Start node → Receives meeting request

Scheduler Agent → Collects meeting details

- Model: GPT-4o
- Instructions: *“You are a meeting scheduler. Collect: 1) Meeting purpose/topic, 2) Preferred dates (ask for 2–3 options), 3) Meeting duration (15/30/60 minutes), 4) Attendee email if not already known. Be conversational and helpful. Once you have all details, check calendar availability using the Google Calendar tool.”*
- Tools: Google Calendar, MCP (read access)
- Include chat history: ON
- Output format: Text

MCP node → Google Calendar availability check

- Server: Google Calendar
- Action: Search for free time slots
- Input variables:
- `{{date_range}}` from conversation
- `{{duration}}` from conversation

Proposal Agent → Presents options

- Model: GPT-4o
- Instructions: *“Present available time slots clearly. Format as a numbered list. Include timezone. Ask the user to select by number or specify a preference. Be specific: ‘Tuesday, March 12 at 2:00 PM EST’ not ‘Tuesday afternoon’.”*
- Include chat history: ON

User Approval node → Confirms booking

- Message: “Confirm booking:\n\nMeeting: `{{meeting_purpose}}`\nTime: `{{selected_slot}}`\nDuration: `{{duration}}`\n\nApprove to book this meeting?”
- Approve path → Create event
- Reject path → End with cancellation message

MCP node → Google Calendar event creation

- Server: Google Calendar
- Action: Create calendar event
- Input:
- Title: `{{meeting_purpose}}`
- Start time: `{{selected_slot}}`
- Duration: `{{duration}}`
- Description: Meeting details from the conversation

- Attendees: {{attendee_email}}

Confirmation Agent → Success message

- Model: GPT-4o-mini
- Instructions: *“Confirm the meeting is booked. Include: time, date, and that calendar invite was sent. Keep it brief and friendly.”*

End node → Closes conversation.

• • •

4) Lead Qualification Agent

Problem

Sales teams waste time on unqualified leads. Someone fills out a contact form, but they're a student researching for a school project.

Or they want enterprise features but have a \$50 monthly budget. Reps spend hours on calls that go nowhere.

Ideal For

B2B SaaS companies, agencies, consultants, or any business with a sales process.

Especially valuable if you get 50+ inbound leads monthly and need to prioritize high-value prospects.

How it Works

A lead fills out your contact form, and the qualification agent asks strategic questions about budget, timeline, decision-making authority, and specific needs.

After collecting responses, a scoring system evaluates the lead using Transform node calculations.

Based on the score, if/else logic routes high-quality leads to immediate sales follow-up and lower-priority leads to nurture campaigns.

Technical Flow Diagram



Be conversational, not interrogative."

Include chat history: ON

Output format: JSON

Collects qualification data

TRANSFORM NODE
(Calculate Lead Score)

Scoring logic:

budget_score:

> \$10k/month = 25 points

\$5k-\$10k = 20 points

\$1k-\$5k = 10 points

< \$1k = 5 points

timeline_score:

"Immediate" = 25 points

"1-3 months" = 20 points

"3-6 months" = 10 points

"Just exploring" = 5 points

authority_score:

"Decision maker" = 25 points

"Influence decision" = 15 points

"Research only" = 5 points

fit_score:

"Perfect match" = 25 points

"Good fit" = 20 points

"Partial fit" = 10 points

"Poor fit" = 5 points

total_score = budget + timeline + authority + fit
(Max: 100 points)

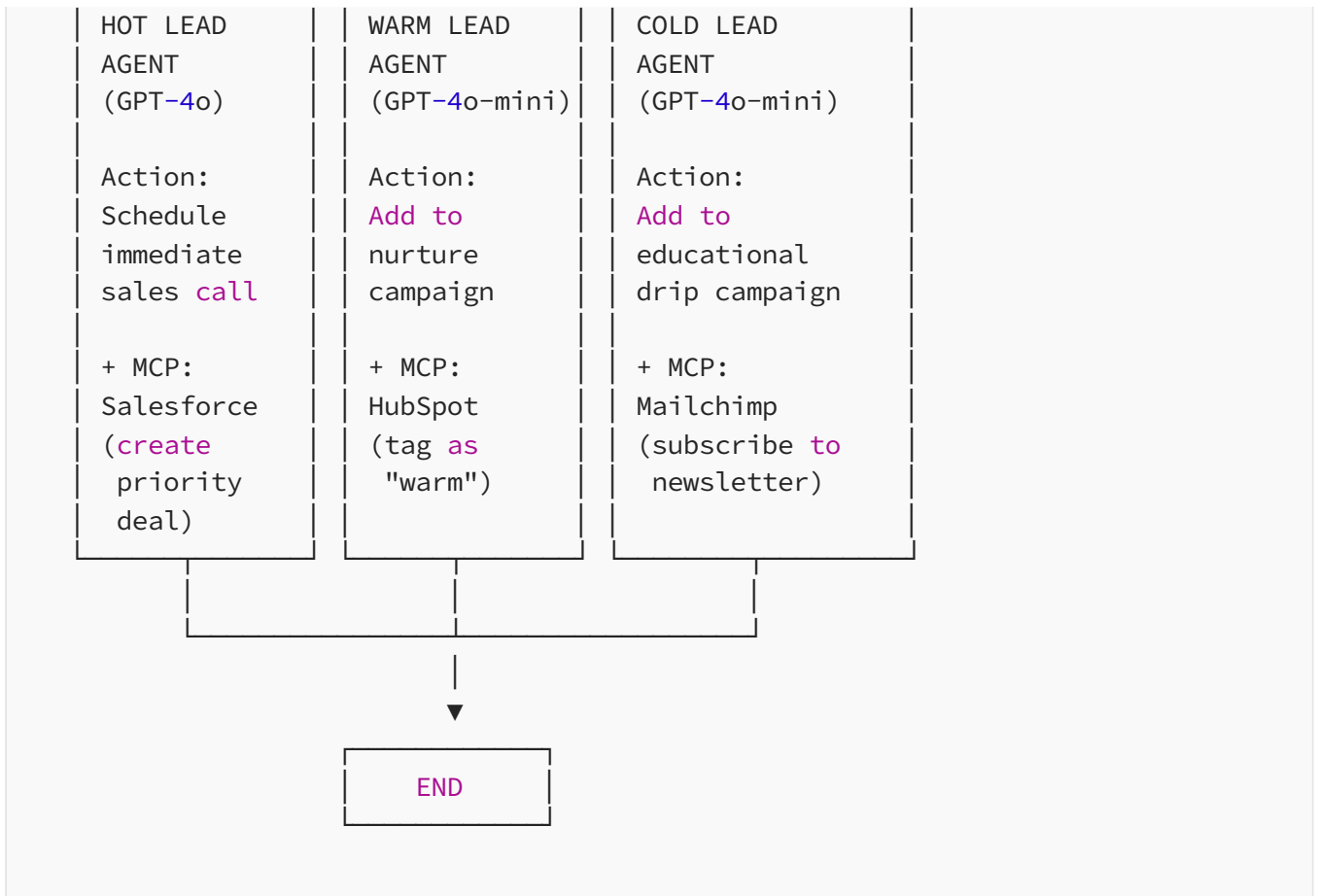
{{total_score}}

IF/ELSE NODE
(Route by Score)

Hot Lead
(80-100)

Warm Lead
(50-79)

Cold Lead
(0-49)



Example

Lead submits: "I'm interested in your enterprise plan."

Qualification Agent: *"Great to hear you're interested in our enterprise solution! To make sure I connect you with the right person on our team, I'd love to learn more about your needs. What specific challenges are you looking to solve with our platform?"*

Lead: *"We need to automate customer onboarding for our 500+ clients. The current manual process takes too long."*

Agent: *"That's a perfect use case for our platform. A few more questions to help you:*

- 1. What's your timeline for implementing a solution?*
- 2. What budget range are you working with for this project?*
- 3. Are you the primary decision-maker, or will others be involved in the decision?"*

Lead: *"We want to implement within 2 months. Budget is around \$15k monthly. I'm the VP of Operations, and I make the final call, though I'll consult with our CTO."*

Agent collects data:

- Budget: \$15k/month → 25 points
- Timeline: 2 months → 20 points
- Authority: Decision-maker → 25 points
- Need: Perfect match (500+ clients, automation) → 25 points

*The transform node calculates: **Total score = 95 points**. If/else routes to Hot Lead Agent (score 80–100).*

Hot Lead Agent: *“Based on what you’ve shared, our enterprise plan is an excellent fit for your needs. I’d like to connect you with Sarah from our enterprise sales team for a personalized demo. She can show you exactly how we’ve helped companies like yours automate onboarding for 500+ clients. Are you available for a 30-minute call this Thursday or Friday?”*

MCP creates a high-priority deal in Salesforce with all qualification data.

Alternative Case — Cold lead:

Lead: *“I’m a college student researching automation tools for a class project. No budget, just learning.”*

Qualification Agent collects: Student, no budget, no timeline, research only. Transform calculates: Total score = 20 points and routes to Cold Lead Agent.

Cold Lead Agent: *“Thanks for your interest! While we focus on enterprise solutions, I’d love to support your research. I’ll add you to our educational newsletter, where we share automation best practices and case studies. Would you like me to send you some resource articles about automation in business?”*

MCP subscribes to an educational newsletter via Mailchimp.

Nodes to Use

Start node → Receives initial inquiry

Set State node → Initialize scoring variables

- Create: `budget_score`, `timeline_score`, `authority_score`, `fit_score`, `total_score`
- All default to 0

Qualification Agent → Collects BANT information

- Model: GPT-4o
- Instructions: *“You qualify sales leads conversationally using BANT (Budget, Authority, Need, Timeline). Ask about: 1) Specific problem they’re solving, 2) Timeline for implementation, 3) Budget range, 4) Decision-making role. Extract this data and output as JSON. Be friendly, not robotic. If they’re vague, ask follow-up questions naturally.”*
- Include chat history: ON
- Output format: JSON

```
{  "budget_range": "10k-50k",  "timeline": "1-3 months",  "authority": "decision maker",  "need_fit": "perfect match",  "company_size": "500+ employee"}
```

Transform node → Calculate lead score

- Expression 1 (budget_score):

```
{{output.budget_range}} == "above 10k" ? 25 : {{output.budget_range}} == "5k-10k" ? 20 : {{output.budget_range}} == "1k-5k" ? 10 : 5
```

- Expression 2 (timeline_score):

```
{{output.timeline}} == "immediate" ? 25 : {{output.timeline}} == "1-3 months" ? 20 : {{output.timeline}} == "3-6 months" ? 10 : 5
```

- Expression 3 (authority_score):

```
{{output.authority}} == "decision maker" ? 25 : {{output.authority}} ==  
"influencer" ? 15 : 5
```

- Expression 4 (fit_score):

```
{{output.need_fit}} == "perfect" ? 25 : {{output.need_fit}} == "good" ? 20 : {{ou
```

- Expression 5 (total_score):

```
{{budget_score}} + {{timeline_score}} + {{authority_score}} + {{fit_score}}
```

If/else node → Route by score

- Case 1: `{{total_score}} >= 80` → Hot Lead Agent
- Case 2: `{{total_score}} >= 50 AND {{total_score}} < 80` → Warm Lead Agent
- Else → Cold Lead Agent

Hot Lead Agent → Immediate sales action

- Model: GPT-4o
- Instructions: “This is a qualified hot lead. Schedule an immediate sales call. Be enthusiastic. Mention specific pain points they shared. Offer 2–3 time slots this week.”
- Tools: MCP (Salesforce — create high-priority deal)

Warm Lead Agent → Nurture sequence

- Model: GPT-4o-mini

- Instructions: “This lead has potential but needs nurturing. Thank them. Explain that you’ll add them to relevant content based on their interests. Set expectations for follow-up in 1–2 weeks.”
- Tools: MCP (HubSpot — tag as ‘warm lead’, add to nurture workflow)

Cold Lead Agent → Educational content

- Model: GPT-4o-mini
- Instructions: “This lead isn’t ready to buy. Be helpful but set boundaries. Offer educational resources. Subscribe to the newsletter if interested.”
- Tools: MCP (Mailchimp — add to newsletter)

End node → Closes conversation

• • •

5) Internal Knowledge Base Search

Problem

Employees waste hours searching for company policies, procedures, and documentation.

HR gets the same questions repeatedly: “What’s our remote work policy?” “How do I submit expenses?” “What’s the PTO approval process?” Information exists, but nobody can find it.

Ideal For

Companies with 50+ employees, remote teams, or organizations with complex policies.

Especially valuable during onboarding when new hires have hundreds of questions.

How it Works

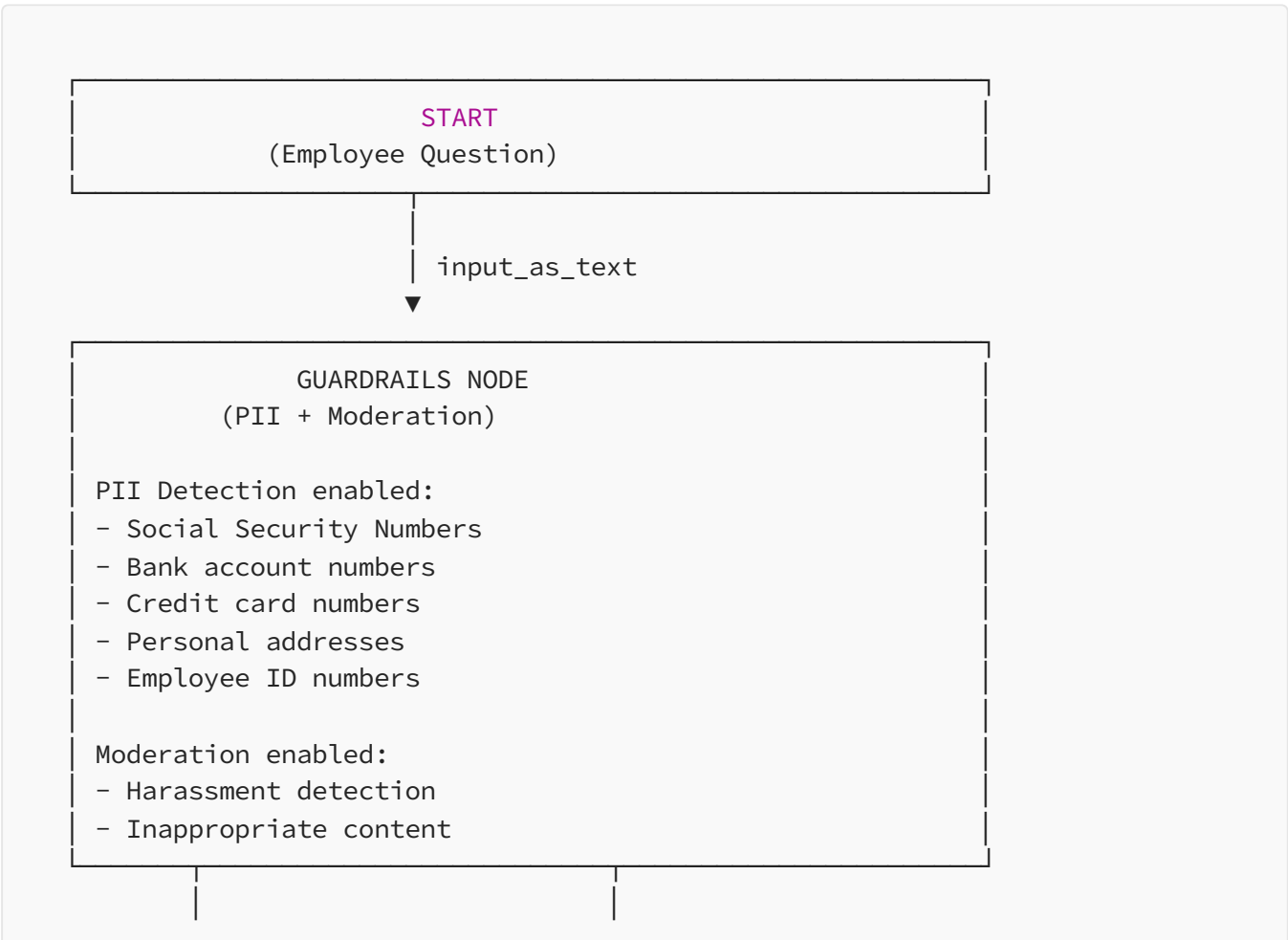
An employee asks a question about company policies, procedures, or resources.

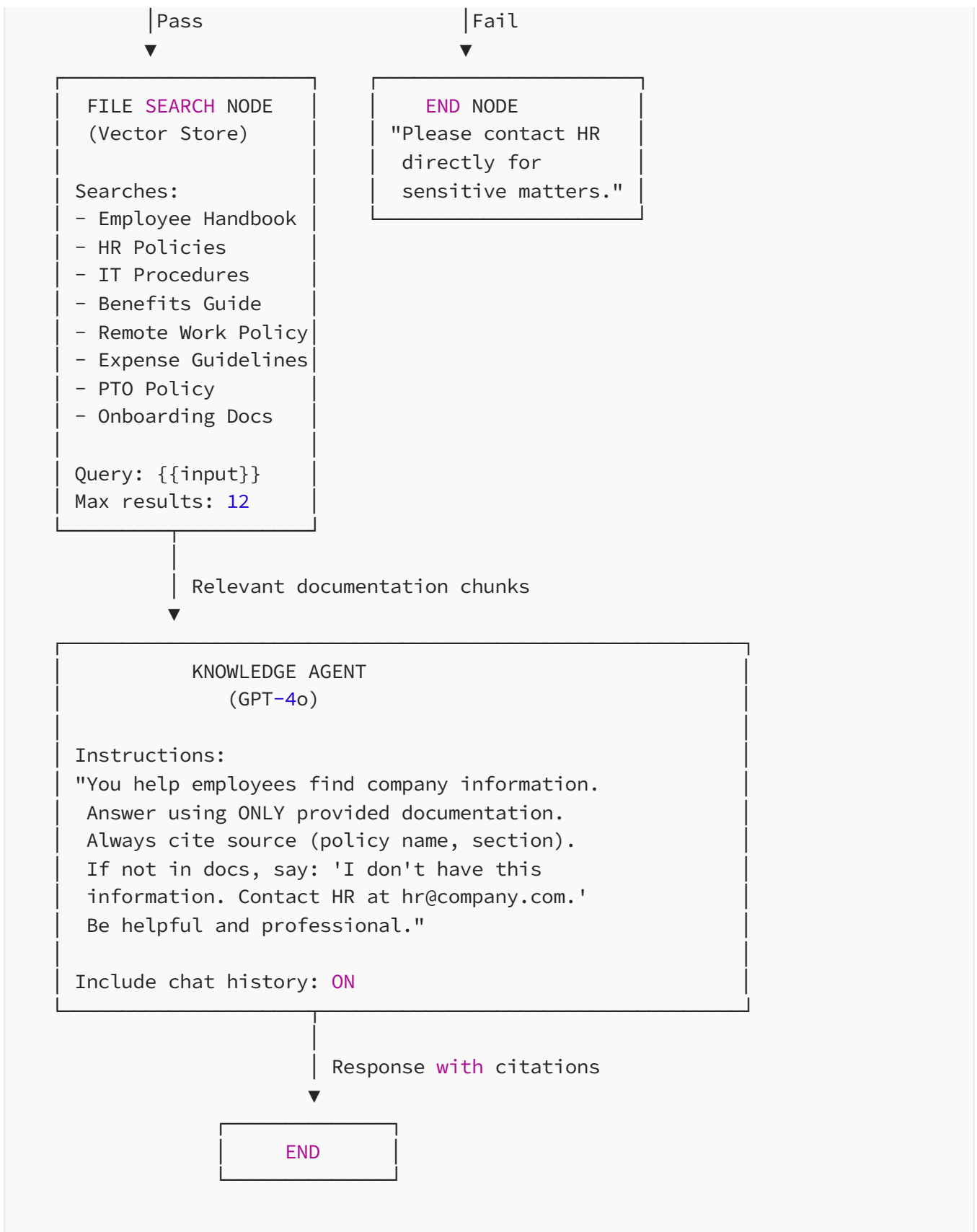
Guardrails check for PII to protect sensitive information.

File search queries your internal documentation (employee handbook, policies, procedures, FAQs).

The agent provides accurate answers with source citations. If information isn’t found, it directs employees to HR contacts.

Technical Flow Diagram





Real example:

Employee asks: “What’s our work-from-home policy? Can I work remotely full-time?”

Guardrails pass (no PII, appropriate question).

File search finds relevant sections from the Remote Work Policy and the Employee Handbook.

Knowledge Agent responds: “Our remote work policy allows full-time remote work with manager approval. Here’s what you need:

1. **Eligibility:** Employees who’ve been with the company 6+ months
2. **Equipment:** The Company provides a laptop and a monitor for the home office
3. **Requirements:**
 - Reliable internet (minimum 25 Mbps)
 - Dedicated workspace
 - Available during core hours (10 AM — 3 PM EST)
4. **Approval process:** Submit request through Workday, manager approves within 5 business days

Source: Remote Work Policy (Section 2.3), Employee Handbook (Page 47)

Need help submitting your request in Workday?”

Follow-up: “What internet speed do I need?”

Agent uses chat history, understands context: *“You need a minimum 25 Mbps download speed for remote work. The company doesn’t reimburse internet costs, but if you need faster speeds for your role, discuss a potential equipment stipend with your manager. Source: Remote Work Policy (Section 2.3.2)”*

Employee asks: “What’s my manager’s salary?”

Guardrails don’t block this (no PII in the question itself), but file search finds nothing.

Knowledge Agent: *“I don’t have access to salary information. For questions about compensation, please get in touch with HR directly at hr@company.com or schedule a time with your HR Business Partner.”*

Nodes to Use

Start node → Receives employee question

Guardrails node → Protect sensitive information — **PII Detection** enabled:

- Social Security Numbers
- Bank account numbers
- Credit card numbers
- Personal home addresses
- Employee ID numbers
- Phone numbers
- Date of birth

Moderation enabled:

- Harassment
- Violence
- Hate speech
- Pass → Continue to file search
- Fail → End with message: “Please contact HR directly at hr@company.com for matters involving personal information.”

File Search node → Search internal documentation

- Vector store contents:
- Employee Handbook (complete)
- All HR Policies (remote work, PTO, benefits, dress code, etc.)
- IT Procedures (password reset, equipment requests, software access)
- Benefits Guide (health insurance, 401k, perks)
- Expense Guidelines (what’s reimbursable, submission process)
- Onboarding Documentation (first week, first month guides)
- Office Procedures (conference room booking, guest policy)

- Query: `{{input_as_text}}`
- Max results: 12 (policies often need multiple sections for context)

Knowledge Agent → Provides answers

- Model: GPT-4o (better for professional, accurate responses)
- Instructions: “You are the company’s internal knowledge assistant. Answer employee questions using ONLY the documentation provided in the file search.”

Response Guidelines:

- Always cite sources (policy name, section number, page number)
- If information isn’t in the documentation, say: ‘I don’t have this information in our documentation. Please get in touch with HR at hr@company.com or your manager.’
- Be professional but friendly.
- For complex questions, break answers into numbered steps
- If the question involves personal situations (discipline, termination, specific accommodations), direct to HR
- Never make up policies or procedures

For Follow-up Questions:

- Use conversation history to understand context
- Reference previous answers when relevant

Sensitive topics to escalate:

- Legal issues
- Discrimination or harassment
- Medical accommodations
- Termination questions

- Salary negotiations Response: ‘This requires personalized guidance. Don't hesitate to contact HR at hr@company.com.
- Include chat history: ON
- Output format: Text

End node → Closes conversation.

. . .

6) E-commerce Product Recommendation Agent

Problem

Customers get overwhelmed by product choices. Your store has 500 products, but customers don't know which ones fit their needs.

They abandon their search or choose the wrong products and return them. Generic “featured products” don't match individual requirements.

Ideal For

Online stores with 50+ products, especially those with technical specifications or products requiring matching (clothing sizes, tech compatibility, specialized equipment).

Works for electronics, fashion, home goods, and sporting equipment.

How it Works

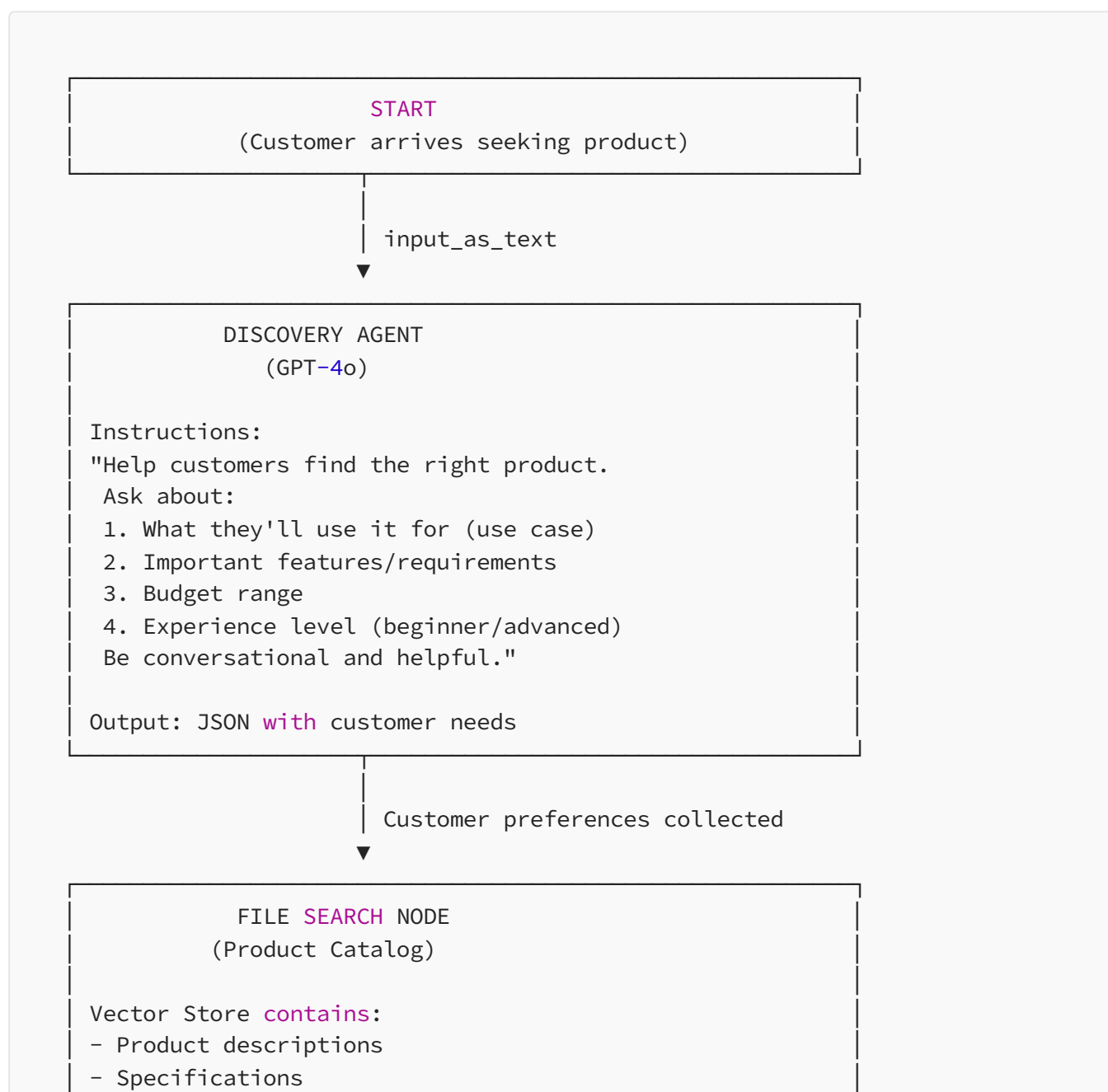
A customer arrives looking for a product but is unsure what they need. The discovery agent asks questions about their use case, preferences, and budget.

File search queries your product catalog with specifications. Transform node filters by price range.

If/else routes to category-specific recommendation agents.

Each agent provides personalized suggestions, with detailed explanations of why each product matches their needs.

Technical Flow Diagram



- Feature lists
- Use cases
- Customer reviews summary
- Compatibility info

Query: {{use_case}} + {{requirements}}

Max results: 20

Initial product matches

TRANSFORM NODE
(Filter by Budget)

Expression 1 (budget_category):
{{budget}} < 100 ? "budget" :
{{budget}} < 500 ? "mid-range" :
"premium"

Expression 2 (filtered_products):
Filter products where price <= {{budget}}

Budget-filtered products

IF/ELSE NODE
(Route by Product Category)

Electronics

Fashion

Home/Other

ELECTRONICS
AGENT
(GPT-4o)

Recommends
based on:
- Tech specs
- Features
- Reviews

FASHION
AGENT
(GPT-4o)

Recommends
based on:
- Style
- Size
- Color
- Occasion

HOME GOODS
AGENT
(GPT-4o)

Recommends
based on:
- Room size
- Use case
- Material
- Durability

2-3 product recommendations

END

Example

Customer: “I need a laptop for video editing.”

Discovery Agent: *“I can help you find the perfect laptop for video editing! A few questions to narrow down the best options:*

- 1. What video editing software do you use? (Premiere Pro, Final Cut, DaVinci Resolve?)*
- 2. What’s your budget range?*
- 3. Do you travel with it, or mostly desktop use?*
- 4. What resolution do you edit? (1080p, 4K, 8K?)”*

Customer: *“I use Premiere Pro for 4K YouTube videos. Budget around \$2,000. Mostly at home, but occasional travel. I’m an intermediate editor.”*

Discovery Agent outputs:

```
{  
  "category": "electronics",  
  "use_case": "video editing 4K",  
  "software": "Adobe Premiere Pro",  
  "budget": 2000,  
  "requirements": ["4K editing", "Premiere Pro compatible", "portable"],  
  "experience": "intermediate"  
}
```

File search finds laptops with relevant specs (powerful CPU, 16GB+ RAM, dedicated GPU, 4K display).

Transform filters products under \$2,000.

If/else routes to Electronics Agent (category: electronics).

Electronics Agent: “Based on your needs for 4K Premiere Pro editing at home with occasional travel, here are my top 3 recommendations:

1. MacBook Pro 16" M3 Pro — \$1,999

2. Dell XPS 15 (Intel i9, RTX 4060) — \$1,899

3. ASUS ProArt Studiobook 16 — \$1,799

My recommendation:

- **MacBook Pro** if you value portability and battery life.
- **Dell XPS** if you need maximum power and prefer Windows.
- **ASUS ProArt** if you want the best specs per dollar.

All three handle 4K Premiere Pro. Want details on any of these?”

Customer: *“Tell me more about the MacBook’s storage. Is 512GB enough?”*

Electronics Agent: *“512GB fills up quickly with 4K footage.*

***Solution:** Use an external SSD for project archives. The Samsung T7 (1TB) costs \$90 and connects via USB-C. Keep active projects on MacBook, archive completed ones externally.*

***Alternative:** Upgrade to 1TB MacBook (\$200 more = \$2,199). Only worth it if external drives are inconvenient for you.*

Most editors use an external storage workflow. Faster transfers, protects projects if the laptop fails. I’d stick with 512GB + external SSD.”

Nodes to Use

Start node → Customer arrives with a product need.

Discovery Agent → Understands requirements

- Model: GPT-4o
- Instructions: *“You help customers find products that fit their needs. Ask 3–4 questions about: use case, key requirements, budget, and experience level. Be conversational. Once you understand their needs, output structured JSON with: category (electronics/fashion/home), use_case (specific need), budget (number), requirements (array), experience (beginner/intermediate/advanced).”*
- Include chat history: ON

- Output format: JSON

```
{ "category": "electronics", "use_case": "video editing 4K",  
  "budget": 2000, "requirements": ["portable", "4K capable", "Premiere Pro"], "
```

File Search node → Query product catalog

- Vector store: Upload comprehensive product data
- Product descriptions (detailed, not just specs)
- Full specifications
- Compatible use cases
- Customer review summaries
- Comparison points vs competitors
- Query: `{{output.use_case}} {{output.requirements}}`
- Max results: 20 (narrow down with budget filter next)

Transform node → Budget filtering

- Expression 1 (budget_category):

```
{{output.budget}} < 100 ? "budget" : {{output.budget}} < 500 ? "mid-range" :  
{{output.budget}} < 2000 ? "premium" : "luxury"
```

- Expression 2 (max_price):

```
{{output.budget}} * 1.1
```

- (Allow 10% flexibility — customer might stretch for the perfect product)

If/else node → Route by category

- Case 1: `{{output.category}}` == "electronics" → Electronics Agent
- Case 2: `{{output.category}}` == "fashion" → Fashion Agent
- Else → Home Goods Agent

Electronics Agent: Tech product recommendations

- Model: GPT-4o
- Instructions: *“Recommend 2–3 products matching customer needs. For each product, provide: name, price, key features, any limitations, and ‘Why this fits’ explanation. Prioritize products within budget. Compare options objectively. Explain technical specs in simple terms. Format nicely with headers and bullet points.”*
- Include chat history: ON

Fashion Agent → Clothing/accessory recommendations

- Model: GPT-4o
- Instructions: *“Recommend outfits or accessories based on style, occasion, and body type mentioned. Explain why each piece works. Mention sizing guidance. Suggest complete outfits when relevant.”*
- Include chat history: ON

Home Goods Agent: Furniture/home product recommendations

- Model: GPT-4o
- Instructions: *“Recommend products based on room size, style preference, and use case. Mention measurements. Explain material durability. Suggest complementary items when relevant.”*
- Include chat history: ON

End node → Closes conversation.

7) Onboarding Agent for New Hires

Problem

HR teams spend hours answering the same onboarding questions from every new hire.

How do I access my benefits?” “Where’s the office kitchen?” “Who do I ask for IT equipment?”

New employees feel overwhelmed with information dumps on day one and forget critical details.

Ideal For

Companies hiring regularly (5+ new employees monthly), remote teams onboarding distributed workers, or organizations with complex benefits and systems.

Reduces HR workload while improving new hire experience.

How it Works

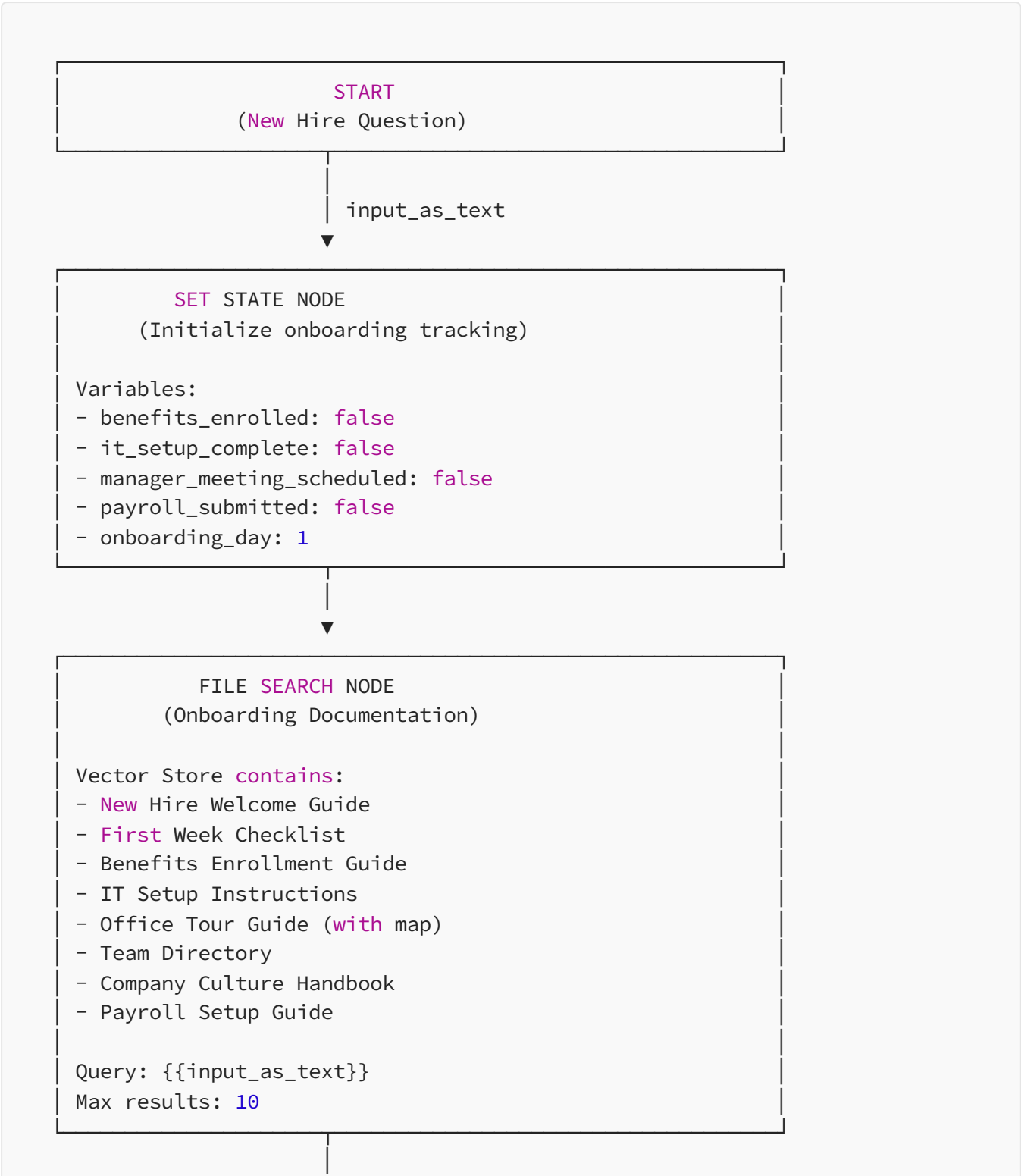
A new employee starts and needs information about their first week.

The onboarding agent provides answers from employee handbooks and onboarding documentation.

For benefits enrollment or IT setup, it offers to send reminder emails via Gmail MCP.

User approval ensures new hires control which actions the agent takes on their behalf. The agent tracks onboarding progress and prompts for incomplete tasks.

Technical Flow Diagram



Relevant onboarding info

ONBOARDING AGENT
(GPT-4o)

Instructions:

"You're an onboarding assistant for new employees.
Answer questions using documentation.
Be warm and welcoming - starting a new job is stressful.

Track onboarding tasks:

- Benefits enrollment
- IT equipment setup
- Manager 1:1 scheduled
- Payroll information submitted

If employee needs help with actions (schedule meeting, send reminder email), offer to help.
Always get approval before taking actions."

Include chat history: ON

Response + action recommendation

IF/ELSE NODE
(Does employee need action?)

Action needed

Info only

USER APPROVAL

Message:

"I can help with:
- Send reminder email about {{task}}
- Schedule meeting with {{person}}
Approve to proceed?"

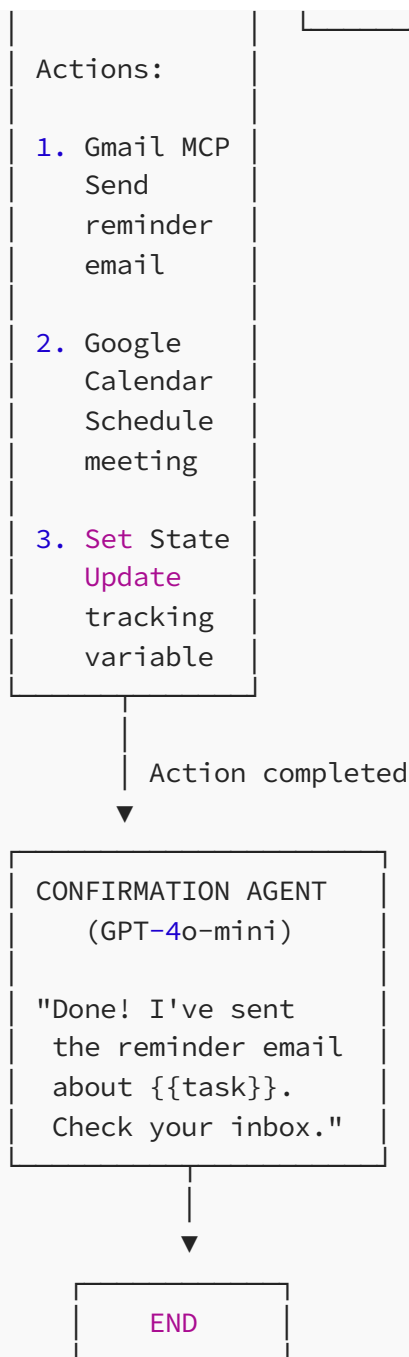
Approve

Reject

MCP NODE

END

END



Real example:

New hire (Day 1): “Hi! I’m starting today. Where do I begin?”

Onboarding Agent: “Welcome to the team! Excited to have you here. Let me walk you through your first day.

Today’s priorities:

1. You’re talking to me — great first step!
2. Pick up your laptop from IT (Building A, Room 101, open 9 AM-5 PM)

3. Complete benefits enrollment by the end of the week

4. Schedule your first 1:1 with your manager, Sarah

Right now: Head to the 2nd-floor kitchen for the welcome breakfast (starts at 9:30 AM). You'll meet your team, and Sarah will introduce you to everyone.

This week: You'll have training sessions Monday through Wednesday. Check your calendar — invites are already there.

What would you like to know more about?"

New hire: "How do I enroll in benefits?"

File search finds Benefits Enrollment Guide.

Agent: "Benefits enrollment is done through our Workday portal. Here's the step-by-step:

Deadline: You have until Friday (your first week) to enroll

How to enroll:

1. Go to workday.company.com
2. Click 'Benefits' tab
3. Click 'Enroll Now'
4. Choose your plans:
 - Health insurance (PPO or HMO options)
 - Dental (coverage starts next month)
 - Vision
 - 401(k) — company matches 4%

Plans to consider:

- PPO: \$180/month, more flexibility
- HMO: \$120/month, lower cost, requires a primary doctor

Help available: Benefits team holds Q&A sessions on Tuesday/Thursday at 2 PM (virtual). Join the Zoom link in your calendar.

Want me to send you a reminder email on Thursday morning about the enrollment deadline?”

New hire: “Yes, please send a reminder.”

User Approval prompt: “I can send a reminder email to your work address (john@company.com) on Thursday at 9 AM about the benefits enrollment deadline (Friday).

Approve to send this reminder?”

New hire clicks Approve.

Gmail MCP sends scheduled email:

```
To: john@company.com
Subject: Reminder: Benefits Enrollment Due Friday
Scheduled: Thursday 9:00 AM

Hi John,
Quick reminder that your benefits enrollment deadline is this Friday.
Enroll at: workday.company.com > Benefits > Enroll Now
Need help? Benefits team Q&A today at 2 PM (check calendar for Zoom link).
- Your Onboarding Assistant
```

Set State node updates: `benefits_enrollment_reminder_sent: true`

Confirmation Agent: “Done! I’ve scheduled a reminder email for Thursday at 9 AM. You’ll get a nudge about the Friday deadline.

Anything else you’d like to know about benefits or your first week?”

Day 3 follow-up:

New hire: “I still haven’t set up my laptop. Can you help?”

Agent checks state: `it_setup_complete: false`

Agent: “Let’s get your laptop set up! IT should have given you:

- Laptop
- Charger
- Monitor (if you requested one)

Setup steps:

1. Power on the laptop
2. Connect to WiFi: ‘Company-Secure’ (password in your welcome email)
3. Log in with: john@company.com and temporary password (check email from IT)
4. Follow prompts to set your permanent password
5. Install required apps (Slack, Zoom, Microsoft Office) — IT sent you installation links

IT Help Desk: ext. 5500 or it-help@company.com. They respond within 30 minutes during business hours.

Want me to schedule a 15-minute call with IT to walk you through setup?”

New hire: “Yes, schedule it.”

User Approval: “I can schedule a 15-minute IT setup call for tomorrow at 10 AM with Jake from IT Support.

Approve to book this meeting?”

Approved.

Google Calendar MCP creates an event:

- Title: “IT Setup Call — New Hire Onboarding”
- Time: Tomorrow, 10:00 AM
- Duration: 15 minutes
- Attendees: john@company.com, jake@company.com

- Description: “Help John complete laptop setup and app installations.”

Set State updates: `it_setup_scheduled: true`

Confirmation Agent: “Your IT setup call is booked for tomorrow at 10 AM with Jake. He’ll help you get everything configured.”

Nodes to Use

Start node → New hire question

Set State node → Track onboarding progress

- Create variables:
- `benefits_enrolled: false`
- `it_setup_complete: false`
- `manager_meeting_scheduled: false`
- `payroll_submitted: false`
- `onboarding_day: 1` (increment daily)
- `welcome_email_sent: false`
- `benefits_reminder_sent: false`

File Search node → Onboarding documentation

- Vector store uploads:
- New Hire Welcome Guide (first day, first week, first month)
- Benefits Enrollment Guide (detailed plan comparisons)
- IT Setup Instructions (laptop, software, access)
- Office Guide (floor plan, amenities, parking)
- Team Directory (names, roles, contact info)
- Company Culture Handbook (values, traditions)
- Payroll Setup Guide (direct deposit, tax forms)

- Training Schedule (sessions, instructors)
- Query: `{{input_as_text}}`
- Max results: 10

Onboarding Agent → Main assistant

- Model: GPT-4o
- Instructions: “You’re a friendly onboarding assistant helping new employees. Answer questions using the provided documentation. Be warm, welcoming, and encouraging — starting a new job is stressful!”

Tracking onboarding tasks: Monitor completion of: benefits enrollment, IT setup, manager 1:1, payroll submission.

Offer Proactive help:

- If Day 1: Focus on first-day logistics (where to go, who to meet)
- If Day 3 and benefits are not enrolled: Remind about the Friday deadline
- If the IT setup is incomplete, Offer to schedule help

Taking actions: When you can help with actions (e.g., sending emails, scheduling meetings), always say, ‘Want me to [action]?’ and wait for approval. Never take actions without asking.

Format responses: Use clear headers, bullet points, and numbered steps. Keep instructions simple. Include contact info when relevant.

Tone: Professional but friendly. Use first names. Say ‘Welcome!’ Be encouraging.”

- Include chat history: ON
- Output format: JSON when action needed:

```
{  "response_text": "Benefits enrollment due Friday...",  "action_needed":
  true,  "action_type": "send_reminder_email",  "action_details": {    "to":
```

```
"john@company.com",    "subject": "Benefits Enrollment Reminder",    "when":  
"Thursday 9 AM"  }}
```

If/else node → Check if action needed

- Case: `{{output.action_needed}} == true` → User Approval
- Else → End (information only)

User Approval node → Confirm actions

- Message: “I can `{{output.action_type}}`:
`{{output.action_details}}` Approve to proceed?”
- Approve → Execute action
- Reject → End with “No problem! Let me know if you change your mind.”

MCP Node → Execute approved actions

- Gmail MCP: Send reminder emails
- Google Calendar MCP: Schedule meetings
- After action, call Set State to update tracking variables

Confirmation Agent → Success message

- Model: GPT-4o-mini
- Instructions: “Confirm the action completed. Be brief and friendly. Mention what to expect next.”

End node → Closes conversation.

. . .

Final Thoughts

You now have seven complete blueprints for building AI agents with OpenAI’s Agent Kit.

I'm creating a video tutorial where I build one of these agents live from scratch. You'll see the exact process: uploading documents, configuring nodes, testing conversations, and deploying with ChatKit.

I'm also publishing follow-up articles on:

- Advanced Agent Kit patterns
- Combining the AgentKit with other automation options
- ChatKit deployment and AgentKit SDK
- Evaluation strategies for production agents

Let me know in the comments which agent you're building first and what problems you're solving with it.

. . .

Let's Connect!

If you are new to my content, my name is [Joe Njenga](#)

Join thousands of other software engineers, AI engineers, and solopreneurs who read my content [daily on Medium](#) and on [YouTube where I review the latest AI engineering tools and trends](#). If you are more curious about my projects and want to receive detailed guides and tutorials, [join thousands of other AI enthusiasts in my weekly AI Software engineer newsletter](#)

If you would like to connect directly, you can reach out here:

AI Integration Software Engineer (10+ Years Experience)

Software Engineer specializing in AI integration and automation.
Expert in building AI agents, MCP servers, RAG...

njengah.com



Follow me on [Medium](#) | [YouTube Channel](#) | [X](#) | [LinkedIn](#)

Ai Agent

Agent Kit

Agentic Ai

Agentic Workflow

OpenAI



Follow



Written by Joe Njenga

5K followers · 98 following

Software & AI Automation Engineer, Tech Writer & Educator. Vision: Enlighten, Educate, Entertain. One story at a time. Work with me: mail.njengah@gmail.com

Responses (6)



Bgerby

What are your thoughts?



Steve Kitchen

Oct 9



Joe, I follow your articles regularly and always find them interesting, insightful and very helpful. Your walkthroughs of different ways to deploy ai agents helped me in my own endeavors more than a few times.

Just wanted to take a moment to... [more](#)



7



1 reply

[Reply](#)



Raf Vantongerloo

Oct 11



Hi Joe, you've got a new follower! Thanks for the extensive explanation!


 3 [Reply](#)



Lonnell branch
Oct 10




Brilliant. I can see every US federal agency using most of these agents. I suggest you partner with someone in the government sector to create the government HR, payroll and onboarding version. A document for the government sector would be worth paying for. Thanks for sharing.

 3 [Reply](#)

[See all responses](#)

More from Joe Njenga


 In AI Software Engineer by Joe Njenga

Cursor 2.0 Has Arrived—And Agentic AI Coding Just Got Wild

Cursor has released version 2.0 , bringing the most powerful agentic AI we have seen yet, more autonomous than ever before,here's what's...

✦ 6d ago



 In AI Software Engineer by Joe Njenga

This Viral DeepSeek OCR Model Is Changing How LLMs Work

This DeepSeek OCR model hit an overnight success not seen in any other release—4k+ GitHub stars in less than 24 hours and more than 100k...

✦ Oct 23




 In Generative AI by Joe Njenga

15 Ways I'm Using Google's Nano Banana for UI/UX Design (Like a Pro)

I am not a UI/UX designer, but Google Nano Banana is filling the gap, and that doesn't mean I'm becoming one overnight.

✦ Oct 9



 Joe Njenga

I Asked Claude 4.5 to Build Me a Complex MCP Server (It Was So Good I May Forget Coding)


If you want to build your custom MCP server, stop wasting time thinking, coding, or trying so hard! The new Claude 4.5 is so good at coding

★ Oct 4



See all from Joe Njenga

Recommended from Medium


 In Towards AI by Teja Kusireddy

We Spent \$47,000 Running AI Agents in Production. Here's What Nobody Tells You About A2A and MCP.

Multi-agent systems are the future. Agent-to-Agent (A2A) communication and Anthropic's Model Context Protocol (MCP) are revolutionary. But...

Oct 16



 In Artificial Intelligence in Plain English by Hermes

10 Lessons Learned Building Voice AI Agents

I spent the last 12 months building voice AI agents. Here's what actually happens when you move from "demo on localhost" to "real users"

5d ago



In Coding Nexus by Code Coup

Claude Desktop Might Be the Most Useful Free Tool You'll Install This Year

I didn't expect much when I first saw the announcement for Claude Desktop. Another AI wrapper, I thought. Maybe with a shiny UI.



Oct 23





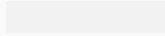
In StartupInsider by Vikram Suthar

Don't Build Your Next SaaS from Scratch. 4 Free Boilerplates to Launch Faster.

As an indie developer, your most valuable asset is time. Stop wasting it on user auth, dashboards, and email setup. Let's get your idea to...



Oct 29



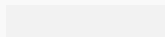
In Data Science Collective by Ida Silfverskiöld

Agentic AI: Single vs Multi-Agent Systems

Building with a structured data source in LangGraph



Oct 28





In MongoDB by MongoDB

Build AI Agents Worth Keeping: The Canvas Framework

This article was written by Mikiko Bazeley.

Oct 8



See more recommendations