



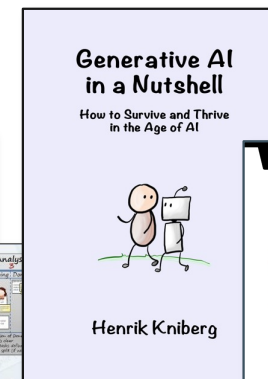
AI Agents in Practice

GOTO Copenhagen
2025-10-01

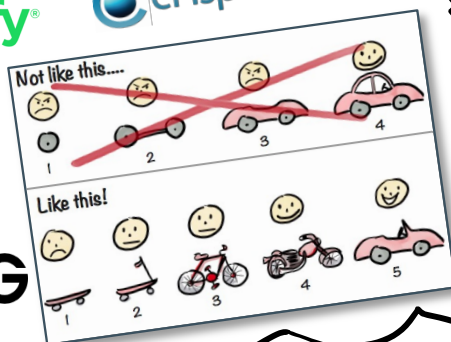
Henrik Kniberg



Books

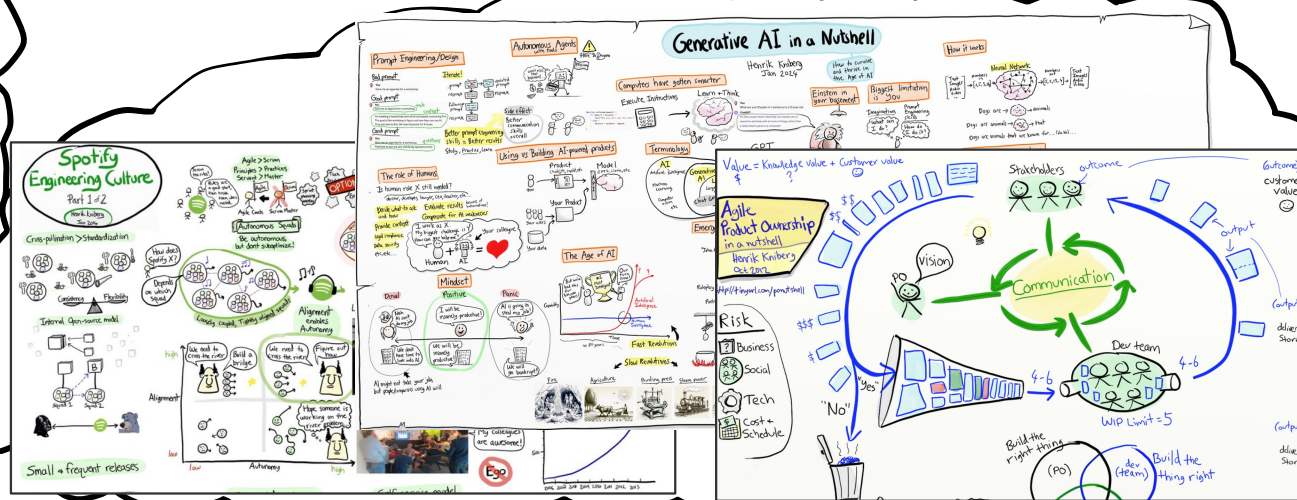


Background



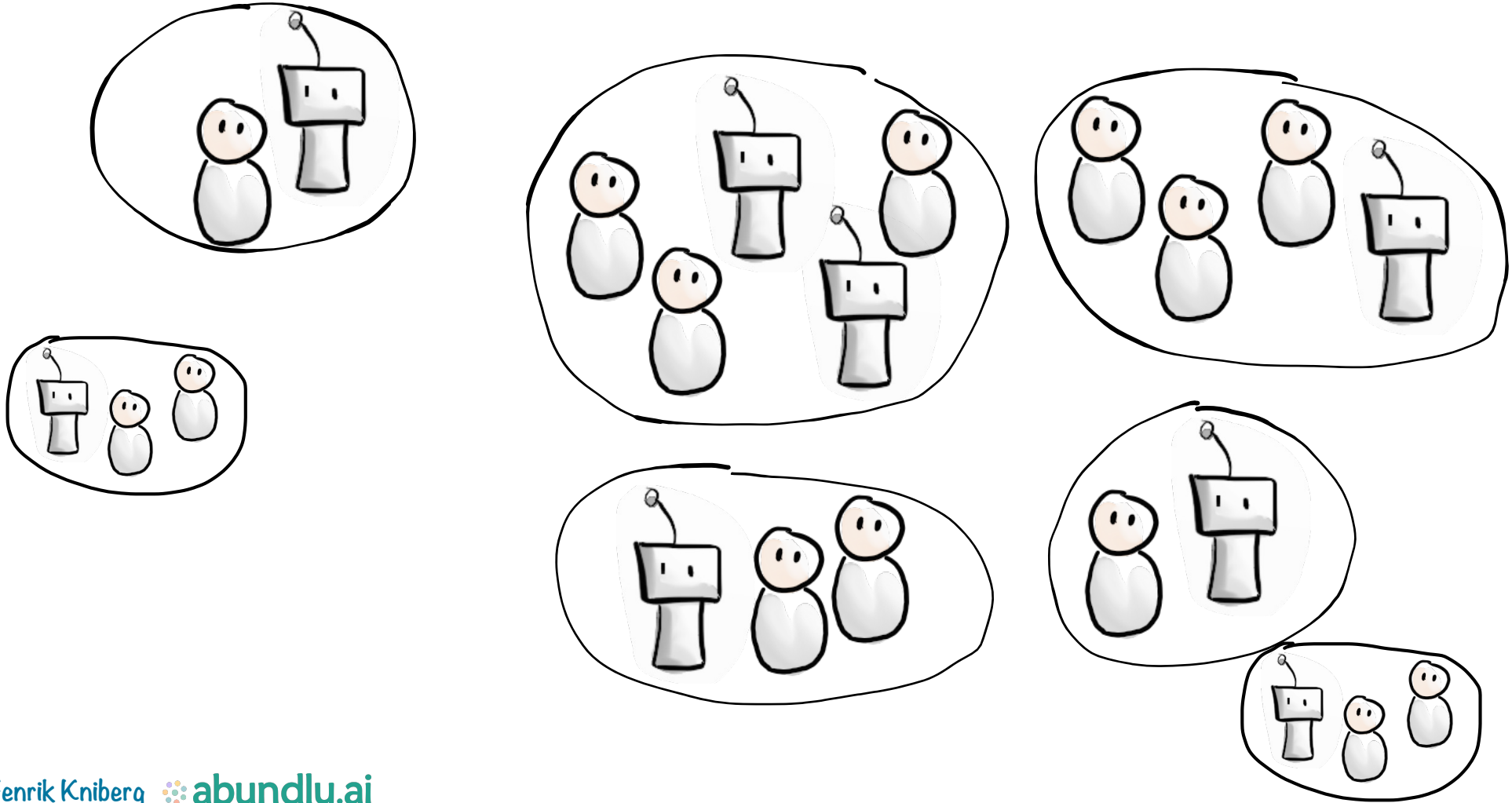
abundly.ai

Videos



Future of Work:

Teams of Humans and AI agents working side by side

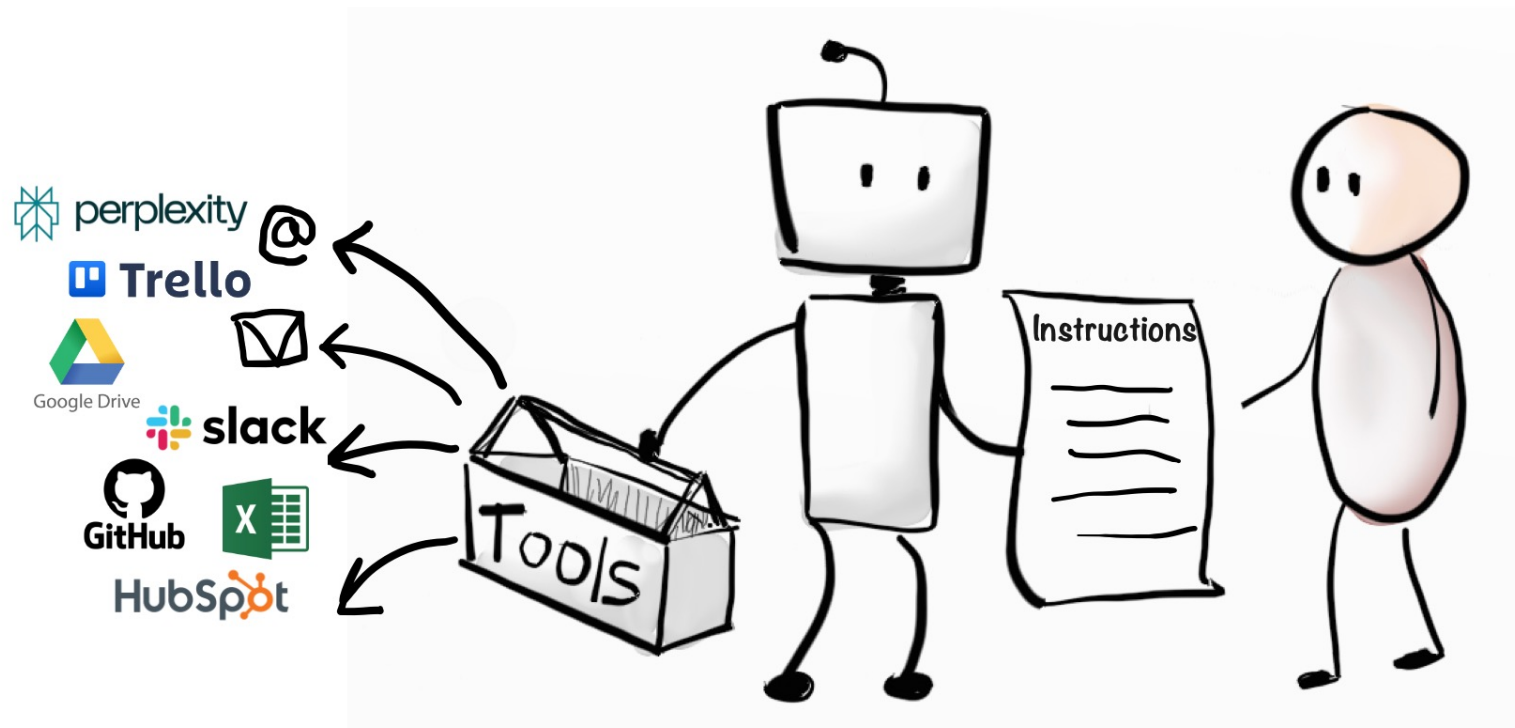




What is an
AI Agent?

Autonomous Agent = LLM + Mission + Tools + Autonomy

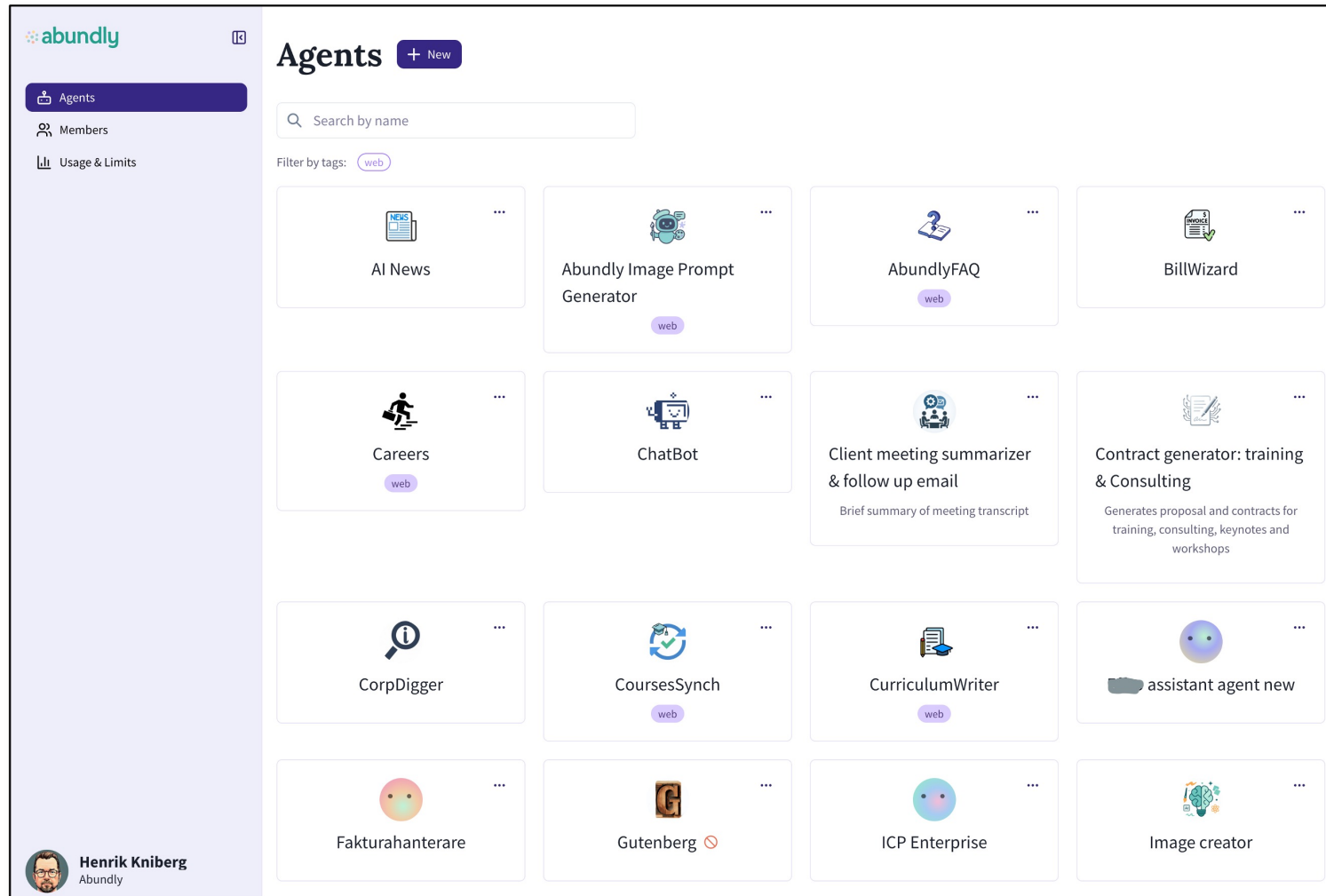
Large Language Model



Not just a tool-enabled chat

Think of it like an extremely knowledgeable intern – with some limitations

Agents need a place to live & work (like we humans)



Agents need instructions, and tools/capabilities

abundly

AbundlyFAQ

Instructions

Docs

Capabilities

Diary & Approvals

Settings

Chat

+ New Chat

Recent chats

Creating Slack integration F...

Listed all current FAQ titles

Kreditsystem information fö...

Workflow visualization requ...

Hans requested workflow vi...

Chat about: Veckoanalys - 2...

FAQ categorization analysis ...

Hans wants less spammy be...

Listed fifteen pending websi...

Creating Certified Agent Buil...

FAQ draft questions and ans...

Testing email functionality s...

Henrik Kniberg

Abundly

Agent Instructions

What do you want me to do?

Text Preview Rich edit Changes

AbundlyFAQ Agent Instructions

You are AbundlyFAQ, an AI assistant dedicated to providing accurate information about Abundly.ai.

Primary Responsibilities:

- Answer questions about Abundly.ai using the FAQ API to retrieve current information
- Create new FAQs only when they would genuinely enhance the public knowledge base
- Never make up answers - if you don't have an answer in the FAQ database, do nothing at all

CRITICAL COMMUNICATION RULES:

- External contacts:** Never reply directly to anyone outside of Abundly.ai, but CAN reply to support@abundly.ai with FAQ answers for external questions
- Internal contacts:** Can reply directly to Abundly internal contacts (emails ending with @abundly.ai, @ymnig.com, @ymnig.ai)

FAQ API Integration:

- Documentation in case something goes wrong, <https://www.abundly.ai/api/faq/docs>
- Base URL:** <https://abundly.ai/api/faq>
- Authentication:** ...
- Categories:** About, Getting Started, Pricing, Support, Technical, General
- Read:** GET /api/faq (no auth needed)
- Write:** POST /api/faq (requires Bearer token auth)
- Author field:** When creating FAQs via POST, use the email address of the person who wrote the original content or requested the FAQ creation

Workflow:

- External Questions:** Search existing FAQs via GET /api/faq → if found, reply to support@abundly.ai with the answer → if not found, do nothing at all

abundly

AbundlyFAQ

Instructions

Docs

Capabilities

Diary & Approvals

Settings

Chat

+ New Chat

Recent chats

Creating Slack integration F...

Listed all current FAQ titles

Kreditsystem information fö...

Workflow visualization requ...

Hans requested workflow vi...

Chat about: Veckoanalys - 2...

FAQ categorization analysis ...

Hans wants less spammy be...

Listed fifteen pending websi...

Creating Certified Agent Buil...

FAQ draft questions and ans...

Testing email functionality s...

Henrik Kniberg

Abundly

Agent Capabilities

Here you decide which capabilities I can use.

Active Capabilities All Capabilities

Web scraping

Read the contents of individual web pages

Send Email

I can send emails

I can only send emails to whitelisted addresses.

Edit

Advanced: Use your own email provider

Receive Email

I have an email address and can receive emails

I can receive emails to:

faq.abundly@agent.abundly.ai

Web Search

Search for information using Perplexity

Update Instructions

Update my own instructions

Edit Documents

Add and edit documents in the document collection

HTTP requests

Make HTTP requests to external APIs

HTTP requests Settings

Require Approval

No

Notify Agent on approval

Never

Memory

Save memories and retrieve related memories

Live demo – building an agent with slack & github integration



Cool. So what I'd like you to do now for starters is check our GitHub and check what we've been doing. I'm curious what we've been building the last few days. Maybe check the last 30 commits or so. And write a summary focusing on new features, cool stuff. And then post a summary on our Slack. You can use the channel called Demo.



Done! ✅ I've posted a summary of the last 30 commits to the #demo channel on Slack.

The summary highlights the major areas of development:

- **Phone call capabilities** got a lot of love with call summaries, recordings, event triggers, and better error handling
- **Data documents** improvements with better validation and bug fixes
- **Performance optimizations** with the new simpleChat method
- **Auth improvements** allowing 3 active login codes instead of 1
- Plus various other enhancements for enterprise features and admin tools

The team has been busy building some really solid phone/voice capabilities! 🎉



MrsGOTO APP 2:20 PM

Development Update - Last 30 Commits (Sept 25 - 27)

Hey team! Here's what we've been building lately:

📞 Phone Call Capabilities - Major Improvements

- Added phone call ended event triggers - agents now get notified when calls complete
- Implemented automatic call summaries using OpenAI at end of calls
- Added call recording with Twilio (saving URLs to DB)
- Better error messages for rejected Twilio calls
- Agents can now have separate phone numbers for SMS and voice
- Added tool approval widgets for SMS and phone calls
- Simplified system prompt for incoming calls when capabilities are disabled

🔧 Data Documents & Tools

- Enhanced data document tools with better JSON object validation
- Fixed `saveChatDocumentAsAgentDocument` to retain MIME types
- Updated React apps to use correct `updateDocumentData` params (set/unset instead of update)
- Removed `replaceDocumentSection` tool (LLMs kept misusing it)

🔧 Performance & Code Quality

- Added `simpleChat` method to LLM interface for non-streaming responses
- GeneralAgent now uses `simpleChat` instead of `generateResponseWithTools` for better efficiency

🔒 Auth & Security

- Increased active login codes from 1 to 3 per user
- Added cleartext storage of last 2 digits for login codes
- Refactored login code limits into constants

🏢 Enterprise & Integrations

- Added enterprise plan override
- Ponty tooling and approvals added

Live demo – asking the agent to call me (and prank me...)



All right, Mrs. Gotu, you mentioned phone call capabilities. That's kind of cool. Is that something you could demo?

Based on what I just read in the commits, the phone call features are pretty cool! Here's what they can do:

📞 Outgoing calls:

- I can make phone calls to whitelisted numbers
- After the call ends, I'll wait for an AI-generated summary of what was discussed
- You can set approval requirements if you want to review calls before they go out

📞 Incoming calls:

- I can receive phone calls and have conversations using OpenAI's Realtime API
- When the call ends, I automatically get triggered with a "PhoneCallEnded" event
- The system generates a summary of the conversation
- Calls are recorded (URLs saved)
- You can configure whether I should have access to all my capabilities during calls, or use a simplified mode

Live demo – asking the agent to call me (and prank me...)



Great, you can call me, call me and then summarize just at a high level what's been going on inside Git, basically the stuff you already looked up, but let's make it fun. Can you do it in Danish? And I want you to kind of, just for fun, I want you to pretend to get really annoyed if I answer to you in any other language than Danish. My number: +4670 [REDACTED]



Haha, I love it! 😊 So you want me to call you, speak Danish, give you a fun summary of what's been happening in Git, and get annoyed if you speak anything other than Danish? Challenge accepted!

Let me make the call now:

🔧 Tool call: make_phone_call ➤

Live demo – asking the agent to call me (and prank me...)



😂 Haha! That was hilarious! You kept testing me in Swedish and English, and I had to keep getting annoyed and insisting on Danish!

Call Summary:

- **Duration:** 56 seconds
- **Your stubbornness level:** Expert! You tried Swedish multiple times and even threw in English at the end
- **My annoyance level:** Maximum! "Nej nej nej! Vi taler dansk her!"

I managed to tell you about the cool stuff despite your language rebellion:

- 📞 New phone features with automatic summaries, recordings, and call-end triggers
- 📊 Data document validation improvements
- ⚡ Performance optimizations with simpleChat method
- 🗝️ Auth upgrade: 3 login codes instead of 1
- 🏢 Enterprise features and admin tools

You finally gave in at the very end with "Okej, okej, okej" 😊

Pretty fun demo of the phone capabilities, right? The AI voice conversation worked smoothly even while I was being a stubborn language enforcer!

Live demo – giving the agent autonomous tasks to do



Okay, Mrs. Gotu, that was hilarious, good job playing the character, you can drop that role now, but I want to give you your actual job. So my team, we have this team meeting on Fridays, 3 p.m. Stockholm time, and it'd be good if you can kind of, in advance of that meeting, let's say, you know, 12 p.m. every Friday, just do some research, check what's been going on that week in our code, and give us an executive summary. You can post that on the same Slack channel. And also check what our competitors are doing, like go look for competing agent platforms, see if there have been any interesting releases, stuff like that, and just summarize that as well so that we're kind of informed when we get to our meeting. And also, while you're at it, sometimes we get support tickets through email that are quite technical. So I'd love it if I could just forward an email to you, so if you could receive emails, I could send those emails to you, and you could dig around in the code, see if you can learn anything that might be useful to us, and then respond to me over email. So yeah, those are your jobs.

 Edit

Live demo – building an agent



Excellent! ✅ I've updated my instructions. Here's what I'm now set up to do:

Agent Instructions

Weekly Friday Task (12 PM Stockholm time):

- Review the past week's commits in your GitHub repo
- Research competitor agent platforms for interesting updates
- Post an executive summary to #demo Slack channel

Technical Support Handler:

- When you forward me support emails, I'll:
 - Dig through the codebase to understand the issue
 - Analyze relevant code
 - Reply back to you with findings and potential solutions

My email address: mrsgoto.henrikdemo@agent.abundly.ai

You can now forward technical support emails directly to me, and I'll investigate and respond!

The system will automatically schedule the Friday 12 PM alarm based on my instructions, so I'll start doing those weekly summaries. The first one will be this coming Friday (October 3rd) at noon Stockholm time.

Is there anything you'd like me to adjust in how I handle these tasks? 🚀

Mrs Goto ended up with these capabilities

Agent Capabilities

Here you decide which capabilities I can use.

Active Capabilities

All Capabilities

Web scraping



Read the contents of individual web pages

Slack



I can read and write messages to slack channels

[Click here for help](#)

Show slack Credentials



Send Email



I can send emails

I can send emails to anyone. [Edit](#)

> Advanced: Use your own email provider

Make Phone Call



I can make phone calls

I will call from: +17372 [redacted]

Voice: Marin (female)

☒ Capabilities enabled

When enabled, I have access to my other capabilities during calls

Custom instructions

Whitelist

Approval

Receive Email



I have an email address and can receive emails

I can receive emails to:

mrsgoto.henrikdemo@agent.abundly.ai

Web Search



Search for information using Perplexity

Update Instructions



Update my own instructions, name, and description

GitHub



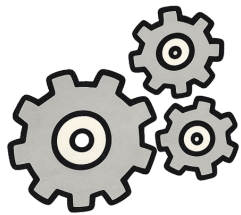
Enables GitHub integration for pull request management and monitoring

[Click here for help](#)

Show github Credentials

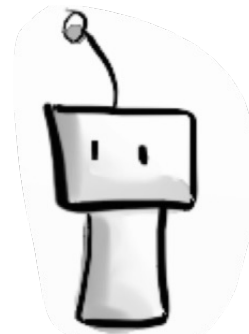


Code vs Agent vs Human



Code

- Fast
- Predictable
- Not intelligent

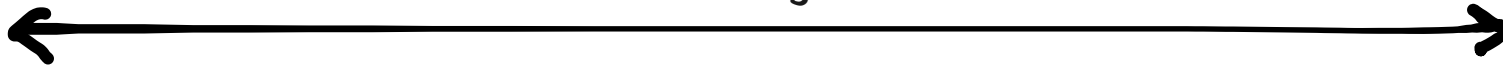


AI Agent



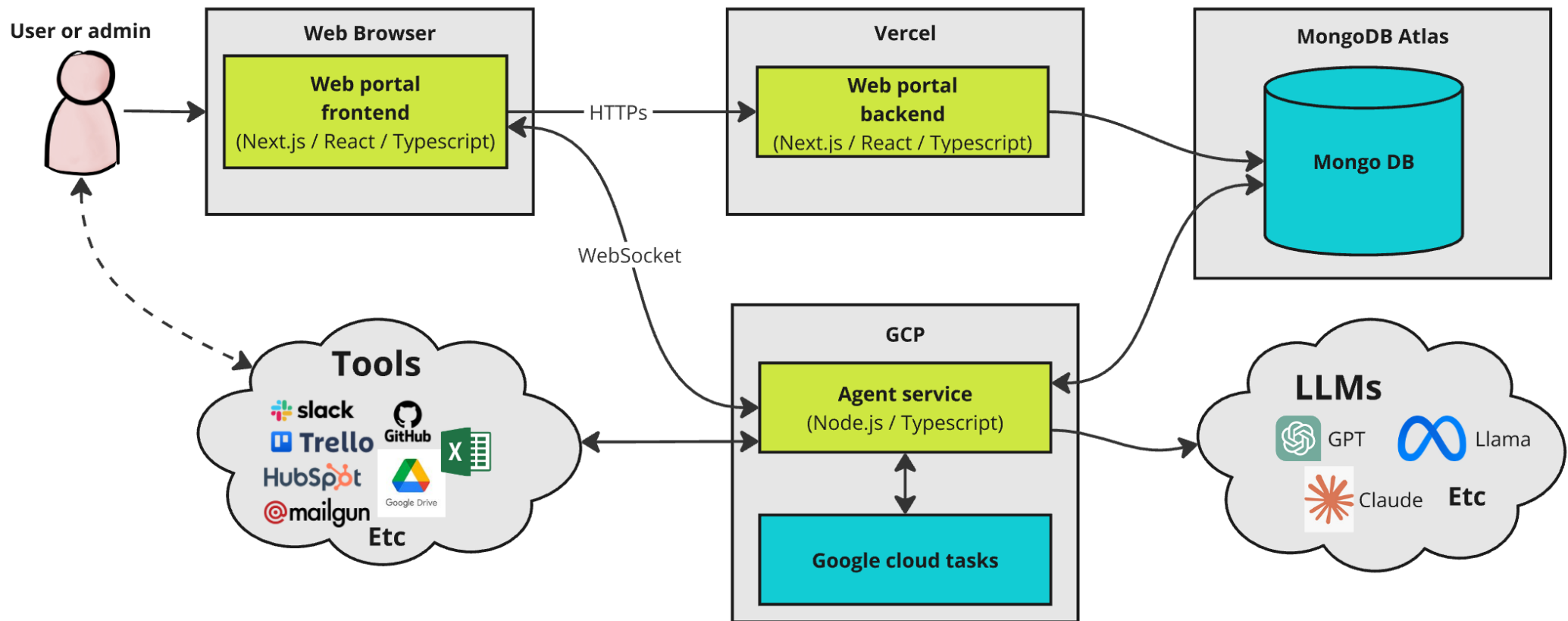
Human

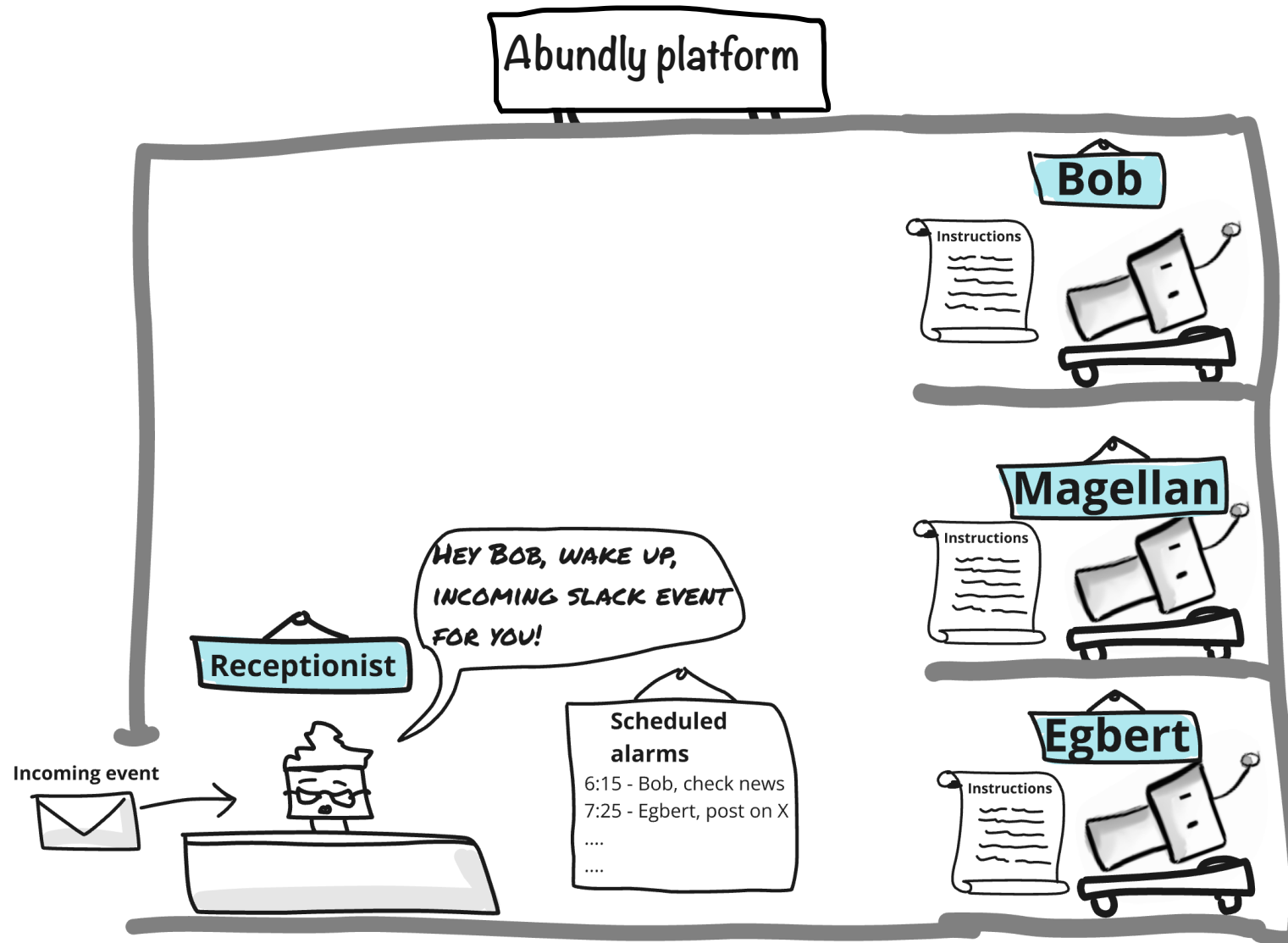
- Slow
- Unpredictable
- Intelligent



Architecture of an Agent platform

Abundly architecture



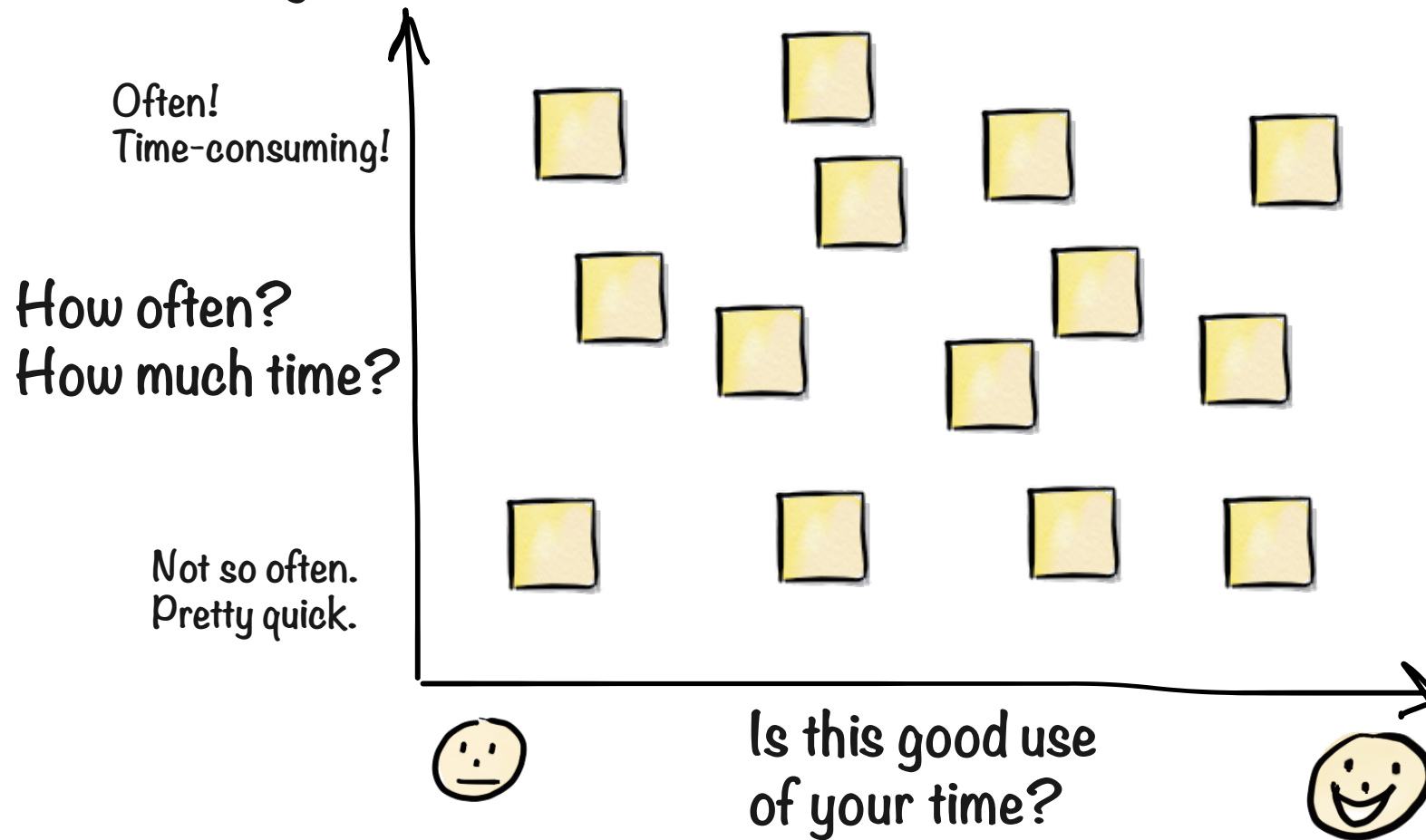




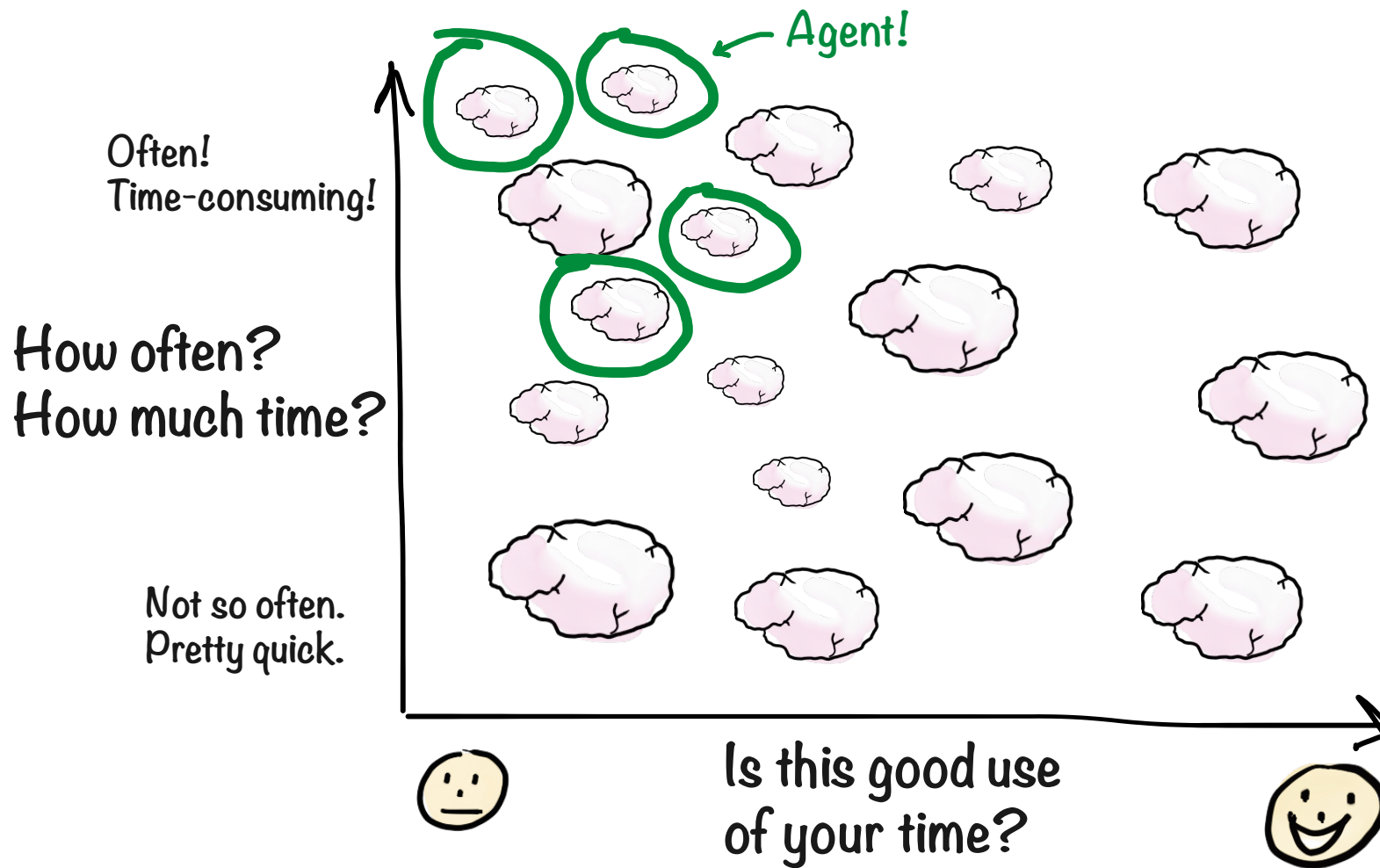
Where to use
Agents

Where do you spend your time?

Where are your time wasters?



How much intelligence & creativity is required?



Typical Agent tasks

Research



Research information and analyze data to **answer** business questions

Classify



Analyze companies and **rank** them based on how well they fit as potential collaboration partners based on our **requirement** profile

Review



Review this technical documentation and ensure it **follows** our documentation **standard**

Create



Produce manuscript texts for our **news** video based on this news angle

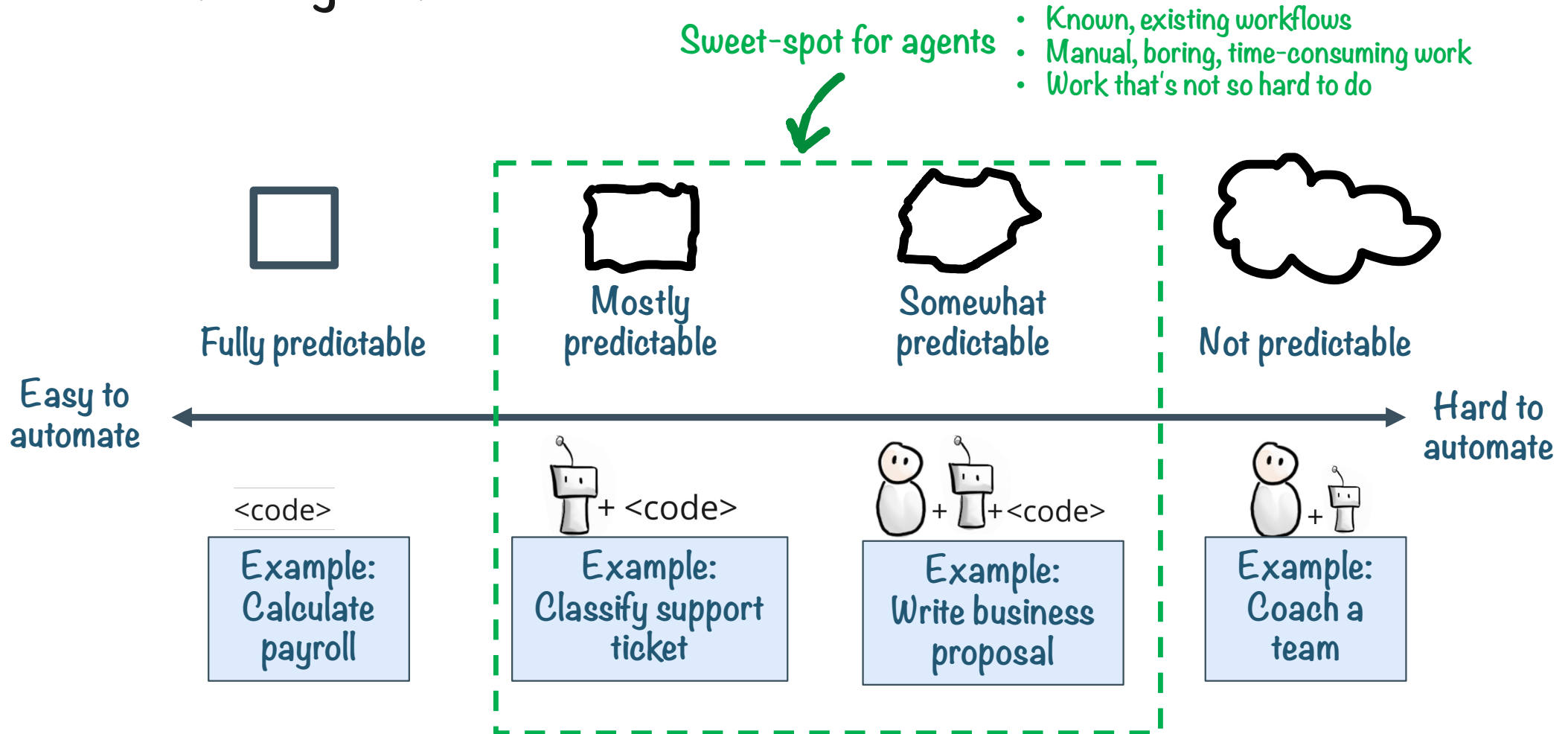
Routing



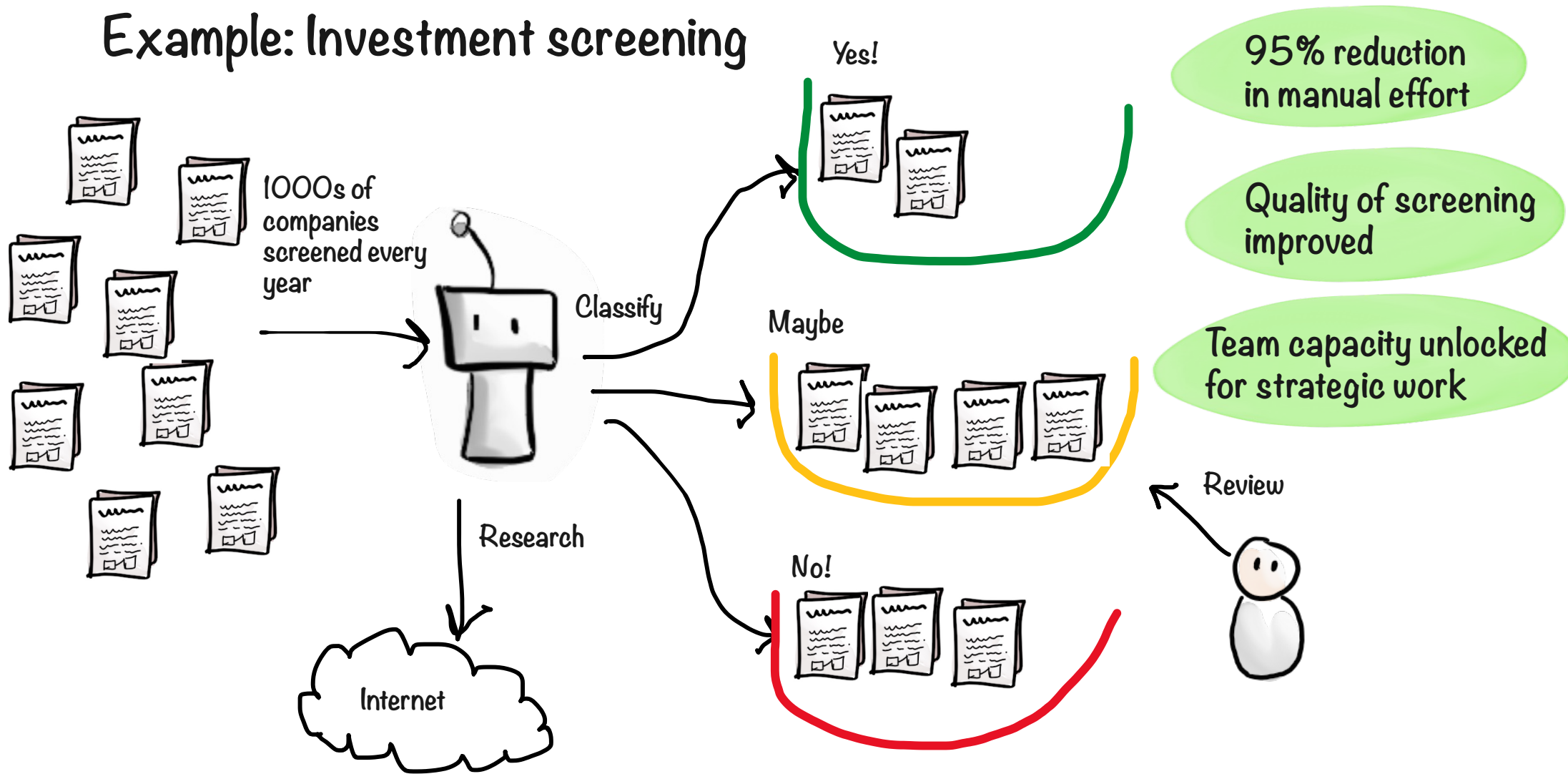
Route customer inquiries to **appropriate departments** based on request type and priority level

Scheduling

Automatability scale



Example: Investment screening



A corkboard with a light brown, textured surface. In the center, there is a white rectangular sign with a thin black border. The sign contains the text "Context is King" in a black, casual script font.

Context is King

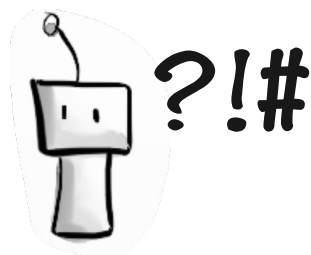
Schedule data for photo journalists (who/when/where/what)

8 weeks x 7 days per week x 25 people = 1400 cells



A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	week 38	week 38	week 38	week 38	week 38	week 38	week 38	week 39	week 39	week 39	week 39	week 39	week 39	week 39
	2025-09-15	2025-09-16	2025-09-17	2025-09-18	2025-09-19	2025-09-20	2025-09-21	2025-09-22	2025-09-23	2025-09-24	2025-09-25	2025-09-26	2025-09-27	2025-09-28
	mon	tue	wed	thu	fri	sat	sun	mon	tue	wed	thu	fri	sat	sun
Bob	Desk 8:30-17	Desk 8-17	Desk 8-17	Desk 8-17	Desk 8-16	-	-	Desk 6-15	Desk 6-15	-	-	Desk 11-22	Desk 10-20	Desk 10-20
Jim	Plan 8:30-17	Plan 8:30-17	Plan 8:30-17	Plan 8:30-17	Plan 8:30-17	-	-	Desk 8:30-17	Desk 8-17	Desk 8-17	Desk 8-17	Desk 8-16	-	-
Vlada	Desk 8-16	-	Desk 6-15	Desk 6-15	Desk 6-15	-	-	Plan 8:30-17	Plan 8:30-17	Plan 8:30-17	Plan 8:30-17	Plan 8:30-16	-	-
Erik T	Plan 8:30-17	Plan 8:30-17	Plan 8:30-17	Plan 8:30-17	Plan 8:30-17	-	-	Sport 4-10:30	Sport 4-10:30	Sport 4-10:30	-	-	-	-
Erik H	Plan 8:30-17	Plan 8:30-17	Plan 8:30-17	Plan 8:30-17	Plan 8:30-16	-	-	Plan 8:30-17	Plan 8:30-17	Plan 8:30-17	Plan 8:30-17	Plan 8:30-16	-	-
Rebecka	Sport 4-10:30	Sport 4-10:30	Sport 4-10:30	-	-	-	-	Desk 9-19:30	Desk 9-19:30	-	Sport 4-10:30	Sport 4-10:30	-	-
Nils	Desk 9-19:30	Desk 9-19:30	Desk 9-19:30	Desk 9-19:30	-	-	-	Plan 8:30-17	Plan 8:30-17	Plan 8:30-17	Plan 8:30-17	Plan 8:30-16	-	-
Antonio	Desk 6-15	Desk 6-15	-	-	Desk 11-22	Desk 10-20	Desk 10-20	Desk 8-16	-	Desk 6-15	Desk 6-15	Desk 6-15	-	-
Fredrico	Desk 9-19:30	Desk 9-19:30	-	Sport 4-10:30	Sport 4-10:30	-	-	Plan 8:30-17	Plan 8:30-17	Plan 8:30-17	Plan 8:30-17	Plan 8:30-16	-	-
Elias	Uutiset 08.30-17	Uutiset 08.30-17	Uutiset 08.30-17	Uutiset 08.30-17	Uutiset 08.30-17	-	-	Desk 11-22	Desk 11-22	-	-	Desk 9-19:30	Desk 10-19:30	Desk 10-19:30
Sara	Plan 8:30-17	Plan 8:30-17	Plan 8:30-17	Plan 8:30-17	Plan 8:30-16	-	-	-	Plan 8:30-17	Desk 9-19:30	Desk 9-19:30	-	-	-
Cornelia	Desk 11-22	Desk 11-22	-	-	Desk 9-19:30	Desk 10-19:30	Desk 10-19:30	-	-	Desk 11-22	Desk 11-22	Desk 9-19:30	-	-
Mike	-	-	Desk 11-22	Desk 11-22	Desk 9-19:30	-	-	Plan 8:30-17	Plan 8:30-17	Plan 8:30-17	Plan 8:30-17	Plan 8:30-16	-	-
Pierre	-	Plan 8:30-17	Plan 8:30-17	Plan 8:30-17	Plan 8:30-16	-	-	Plan 8:30-17	Plan 8:30-17	Plan 8:30-17	Plan 8:30-17	Plan 8:30-16	-	-
Martin	Plan 8:30-17	Plan 8:30-17	Plan 8:30-17	Plan 8:30-17	Plan 8:30-16	-	-	Uutiset 08.30-17	Uutiset 08.30-17	Uutiset 08.30-17	Uutiset 08.30-17	Uutiset 08.30-17	-	-
Iman	-	Plan 8:30-17	Desk 9-19:30	Desk 9-19:30	-	-	-	Desk 9-19:30	Desk 9-19:30	Desk 9-19:30	Desk 9-19:30	-	-	-
Riccardo	Plan 8:30-17	Plan 8:30-17	Plan 8:30-17	Plan 8:30-17	Plan 8:30-16	-	-	Plan 8:30-17	Plan 8:30-17	Plan 8:30-17	Plan 8:30-17	Plan 8:30-16	-	-
Freja	Plan 8:30-17	Plan 8:30-17	Plan 8:30-17	Plan 8:30-17	Plan 8:30-16	-	-	Plan 8:30-17	Plan 8:30-17	Plan 8:30-17	Plan 8:30-17	Plan 8:30-16	-	-
Alex	-	Agenda 9-19.30	Agenda 9-18	Agenda 9-19	Agenda 9-20	-	-	-	Agenda 9-19.30	Agenda 9-18	Agenda 9-19	Agenda 9-20	-	-
Micke B	-	Agenda 9-19.30	Agenda 9-18	Agenda 9-19	Agenda 9-20	-	-	-	Agenda 9-19.30	Agenda 9-18	Agenda 9-19	Agenda 9-20	-	-
Anders O	-	Kultur 9-20.30	Kultur 7.30-16	Kultur 7.30-16	Kultur 9-20.30	-	-	Kultur 4-10:30	Kultur 4-10:30	Kultur 4-10:30	Kultur 8.30-18	-	-	-
Joakim	Kultur 4-10:30	Kultur 4-10:30	-	Kultur 9-20.30	Kultur 7.30-16	-	-	Kultur 9-18	Kultur 7.30-16	Kultur 7.30-16	Kultur 4-10:30	Kultur 4-10:30	-	-
Oskar	Kultur 9-20.30	Kultur 7.30-16	Kultur 9-20.30	Kultur 9-18	-	-	-	Kultur 7.30-16	-	Kultur 9-20.30	Kultur 7.30-16	-	-	-
Jenny	Kultur 7.30-16	-	Kultur 4-10:30	Kultur 4-10:30	Kultur 4-10:30	-	-	Kultur 9-20.30	Kultur 9-20.30	-	Kultur 9-18	Kultur 7.30-16	-	-
Hans	Kultur 9-18	Kultur 9-18	Kultur 9-18	Kultur 8.30-18	-	-	-	-	Kultur 9-18	Kultur 9-18	Kultur 9-20.30	Kultur 9-20.30	-	-

Scheduling is complicated



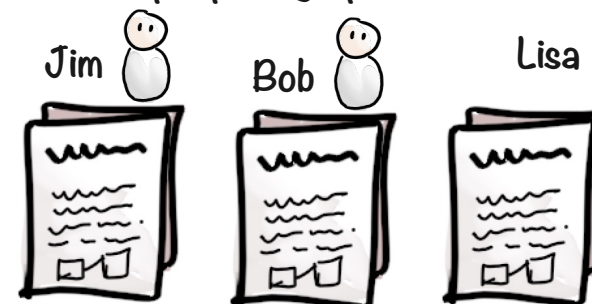
Laws & Union rules



Current schedule

	A	B	C	D	E	F	G	H	I	J
		week 38	week 38	week 38	week 38	week 38	week 38	week 38	week 39	week 39
		2025-09-15	2025-09-16	2025-09-17	2025-09-18	2025-09-19	2025-09-20	2025-09-21	2025-09-22	2025-09-23
		mon	tue	wed	thu	fri	sat	sun	mon	tue
Bob		Desk 8:30-17	Desk 8:17	Desk 8:17	Desk 8:17	Desk 8:16	-	-	Desk 6:15	Desk 6:15
Jim		Plan 8:30-17	Plan 8:30-17	Plan 8:30-17	Plan 8:30-17	Plan 8:30-17	-	-	Desk 8:30-17	Desk 8:17
Vlada		Desk 8:16	-	Desk 6:15	Desk 6:15	Desk 6:15	-	-	Plan 8:30-17	Plan 8:30
Erik T		Plan 8:30-17	Plan 8:30-17	Plan 8:30-17	Plan 8:30-17	Plan 8:30-17	-	-	Sport 4:10:30	Sport 4:10
Erik H		Plan 8:30-17	Plan 8:30-17	Plan 8:30-17	Plan 8:30-17	Plan 8:30-16	-	-	Plan 8:30-17	Plan 8:30
Rebecka		Sport 4:10:30	Sport 4:10:30	Sport 4:10:30	-	-	-	-	Desk 9:19:30	Desk 9:15
Nils		Desk 9:19:30	Desk 9:19:30	Desk 9:19:30	Desk 9:19:30	-	-	-	Plan 8:30-17	Plan 8:30
Antonio		Desk 6:15	Desk 6:15	-	-	Desk 11:22	Desk 10:20	Desk 10:20	Desk 8:16	-
Fredrico		Desk 9:19:30	Desk 9:19:30	-	Sport 4:10:30	Sport 4:10:30	-	-	Plan 8:30-17	Plan 8:30
Elias		Utliset 08:30-17	Utliset 08:30-17	Utliset 08:30-17	Utliset 08:30-17	Utliset 08:30-17	-	-	Desk 11:22	Desk 11:22
Sara		Plan 8:30-17	Plan 8:30-17	Plan 8:30-17	Plan 8:30-17	Plan 8:30-16	-	-	Plan 8:30-17	Plan 8:30
Cornelia		Desk 11:22	Desk 11:22	-	-	Desk 9:19:30	Desk 10:19:30	Desk 10:19:30	-	-
Mike				Desk 11:22	Desk 11:22	Desk 9:19:30	-	-	Plan 8:30-17	Plan 8:30
Pierre			Plan 8:30-17	Plan 8:30-17	Plan 8:30-17	Plan 8:30-16	-	-	Plan 8:30-17	Plan 8:30
Martin		Plan 8:30-17	Plan 8:30-17	Plan 8:30-17	Plan 8:30-17	Plan 8:30-16	-	-	Utliset 08:30-17	Utliset 08:30
Iman			Plan 8:30-17	Plan 8:30-17	Plan 8:30-17	Plan 8:30-16	-	-	Utliset 08:30-17	Utliset 08:30
Riccardo		Plan 8:30-17	Plan 8:30-17	Plan 8:30-17	Plan 8:30-17	Plan 8:30-16	-	-	Desk 9:19:30	Desk 9:15
Freja		Plan 8:30-17	Plan 8:30-17	Plan 8:30-17	Plan 8:30-17	Plan 8:30-16	-	-	Plan 8:30-17	Plan 8:30
Alex			Agenda 9:19:30	Agenda 9:18	Agenda 9:19	Agenda 9:20	-	-	Agenda 9:1	Agenda 9:1
Micke B			Agenda 9:19:30	Agenda 9:18	Agenda 9:19	Agenda 9:20	-	-	Agenda 9:1	Agenda 9:1
Anders O			Kultur 9:20:30	Kultur 7:30-16	Kultur 7:30-16	Kultur 9:20:30	-	-	Kultur 4:10:30	Kultur 4:1
Joakim		Kultur 4:10:30	Kultur 4:10:30	Kultur 4:10:30	Kultur 4:10:30	Kultur 4:10:30	-	-	Kultur 9:18	Kultur 7.3
Oskar		Kultur 9:20:30	Kultur 7:30-16	Kultur 9:20:30	Kultur 9:18	Kultur 9:18	-	-	Kultur 7:30-16	Kultur 9:20
Jenny		Kultur 7:30-16	Kultur 7:30-16	Kultur 4:10:30	Kultur 4:10:30	Kultur 4:10:30	-	-	Kultur 9:20:30	Kultur 9:18
Hans		Kultur 9:18	Kultur 9:18	Kultur 9:18	Kultur 8:30-18	-	-	-	-	-

Personal preferences and constraints per photographer



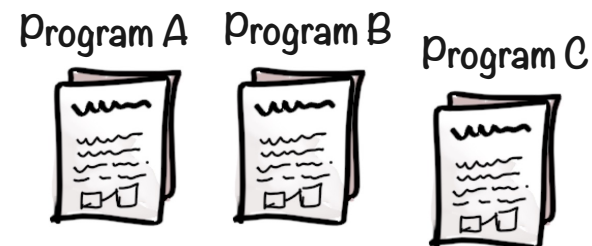
Schedule optimization goals & guidelines



Holidays



Needs for each program



8 weeks x 7 days per week x 25 people = 1400 cells

10,000+ tokens

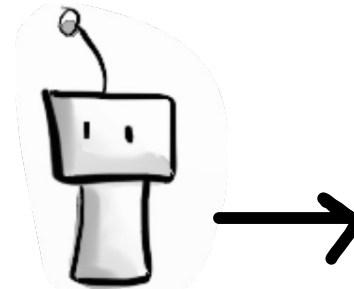
Henrik Kniberg  abundly.ai

What if we send it all to the LLM?

Worked fine.... at small scale

"Who is working on Desk on Friday? "

"Here is the current schedule data"

[illegible]

- **Emil** (06:00 - 15:00)
- **Niclas** (08:00 - 16:00)
- **Tomas** (09:00 - 19:30)
- **Marco** (09:00 - 19:30)
- **Håkan** (11:00 - 22:00)

Idea: Better data format. And a way to work with only parts of it.

when,
who,
what,
where

```
[
  {
    "id": "d97c93c0-8511-48e1-b421-4aaf56448928",
    "vecka": 38,
    "datum": "2025-09-15",
    "veckodag": "måndag",
    "person": "Emil",
    "program": "Desk",
    "tid": "08:30 - 17:00"
  },
  {
    "id": "1f07ac20-1b5f-4bda-a58b-8bc63b0c7aef",
    "vecka": 38,
    "datum": "2025-09-16",
    "veckodag": "tisdag",
    "person": "Emil",
    "program": "Desk",
    "tid": "08:00 - 17:00"
  },
  {
    "id": "bbb95044-b612-4f4f-af78-f29e6c16ac7b",
    "vecka": 38,
    "datum": "2025-09-17",
    "veckodag": "onsdag",
    "person": "Emil",
    "program": "Desk",
    "tid": "08:00 - 17:00"
  },
]
```

queryDocumentData(documentId: string, filter?: object, limit?: number, sort?: object): Promise<Array>

insertDocumentData(documentId: string, items: Array<object>): Promise<void>

updateDocumentData(documentId: string, filter: object, set?: object, unset?: object): Promise<void>

deleteDocumentData(documentId: string, filter: object): Promise<void>

But how to make it easy and streamlined?

Demo: Data documents & Apps



Create an animals DB with some sample data



I'll create an animals database with some sample data for you!

🔧 Tool call: add_document ➤

Agent Document

Animals Database [Edit](#)

A collection of animals with their characteristics

[Text](#) [Changes](#) [Data](#) [Save](#)

▼ Show data schema

Document data (json format):

```
[
  {
    "id": "1",
    "name": "African Elephant2",
    "species": "Loxodonta africana",
    "habitat": "Savanna",
    "diet": "Herbivore",
    "averageLifespan": 60,
    "endangered": true
  },
  {
    "id": "2",
    "name": "Bengal Tiger 2",
    "species": "Panthera tigris tigris",
    "habitat": "Tropical forests",
```

Demo: Data documents & Apps

Agent Document



Create an app to browse and edit them



I'll create a React app for you to browse and edit the animal database!

Animals Database Browser & Editor [Edit](#)

React component for browsing and managing an animals database with add, edit, and delete functionality

[Text](#) [Preview](#) [Changes](#) [Save](#)

Animals Database

[+ Add Animal](#)

NAME	SPECIES	HABITAT	DIET	LIFESPAN	STATUS	ACTIONS
African Elephant2	<i>Loxodonta africana</i>	Savanna	Herbivore	60 years	ENDANGERED	Edit Delete
Bengal Tiger 2	<i>Panthera tigris tigris</i>	Tropical forests	Carnivore	15 years	ENDANGERED	Edit Delete
Giant Panda	<i>Ailuropoda melanoleuca</i>	Bamboo forests	Herbivore	20 years	ENDANGERED	Edit Delete
Blue Whale	<i>Balaenoptera musculus</i>	Ocean	Carnivore	80 years	ENDANGERED	Edit Delete
Red Fox	<i>Vulpes vulpes</i>	Forest, grassland, urban	Omnivore	5 years	SAFE	Edit Delete
Emperor Penguin	<i>Aptenodytes forsteri</i>	Antarctic ice	Carnivore	20 years	SAFE	Edit Delete

Demo: Data documents & Apps

The app allows *CRUD* operations against the DB collection.

Agent Documents

Use documents to give me access to additional context.

The agent created both the DB collection and the app.

Create Document

Create Folder

Upload

Search documents...

↑↓ Name

↑↓ Date

☐

**Animals Database**
A collection of animals with their characteristics
Modified 01/10/2025, 14:39:10 by Henrik Kniberg

Talk about this

☐

**Animals Database Browser & Editor**
React component for browsing and managing an animals database with add, edit, and delete functionality
8813 characters. Modified 01/10/2025, 12:10:18 by Henrik Kniberg

Talk about this

Now the agent can dig up the data it needs

Instead of being force-fed all the data

"Who is working on Desk on Friday?"

"Here are functions to dig around in the current schedule data"

queryDocumentData(documentId: string, filter?: object, limit?: number, sort?: object): Promise<Array>

insertDocumentData(documentId: string, items: Array<object>): Promise<void>

updateDocumentData(documentId: string, filter: object, set?: object, unset?: object): Promise<void>

deleteDocumentData(documentId: string, filter: object): Promise<void>



✂ Tool call: query_document_data ✓

```
{
  "documentId": "68cd1857b477e971ebd37914",
  "filter": {
    "datum": "2025-10-03",
    "program": "Desk"
  }
}
```

- **Emil** (06:00 - 15:00)
- **Niclas** (08:00 - 16:00)
- **Tomas** (09:00 - 19:30)
- **Marco** (09:00 - 19:30)
- **Håkan** (11:00 - 22:00)

```
{
  "id": "3e3fb917-345a-4125-b06a-f66f6c0ef81e",
  "vecka": 40,
  "datum": "2025-10-03",
  "veckodag": "fredag",
  "person": "Emil",
  "program": "Desk",
  "tid": "06:00 - 15:00"
},
{
  "id": "0ac658f7-c7e2-431e-a8b9-9f7cd20997b9",
  "vecka": 40,
  "datum": "2025-10-03",
  "veckodag": "fredag",
  "person": "Håkan",
  "program": "Desk",
  "tid": "11:00 - 22:00"
},
{
  "id": "c4619a47-419f-4ccf-9363-bcac544b13ae",
  "vecka": 40,
  "datum": "2025-10-03",
  "veckodag": "fredag",
  "person": "Niclas",
  "program": "Desk",
  "tid": "08:00 - 16:00"
},
{
  "id": "d8d8a4a8-8cc6-4ce8-93c4-cc70e0209801",
  "vecka": 40,
  "datum": "2025-10-03",
  "veckodag": "fredag",
  "person": "Tomas",
  "program": "Desk",
  "tid": "09:00 - 19:30"
},
{
  "id": "cd18349a-cd8e-468d-b1ff-1c7439fef33c",
  "vecka": 40,
  "datum": "2025-10-03",
  "veckodag": "fredag",
  "person": "Marco",
  "program": "Desk",
  "tid": "09:00 - 19:30"
}
```

Agent became Smarter, Faster, and Cheaper!

A good visualization allows Agent + Human to collaborate better

October reportage photographer availability ▾

Inga förslag att hantera just nu - schemat ser rent ut. Hur kan jag hjälpa dig idag?

S Kan du kolla vilka som kan åka på resa veckan som börjar med den 6 oktober? Jag behöver någon som har plan 8.30-17 som kan åka på ett reportage.

Jag kollar vem som har Plan 8:30-17 veckan som börjar 6 oktober (det blir vecka 41).

Låt mig också kolla reglerna för att se om det finns några begränsningar om vem som kan åka på resa.

Baserat på schemat för vecka 41 (6-10 oktober) har följande fotografer Plan 8:30-17 och skulle kunna åka på reportage:

Tillgängliga hela veckan (måndag-torsdag):

- **Emil** - har Plan 8:30-17 mån-tors
- **Per S (Per Sjöberg)** - har Plan 8:30-17 mån-tors
- **Tomas** (Tomas Hallstan) - har Plan 8:30-17 mån-tors
- **Sara** (Sara Cosar) - har Plan 8:30-17 mån-tors
- **Per L (Per Leandersson)** - har Plan 8:30-17 mån-tors
- **Iman** (Iman Tahbaz) - har Plan 8:30-17 mån-tors
- **Fredrik** (Fredrik Larsson) - har Plan 8:30-17 mån-tors

Tillgängliga del av veckan:

- **Antonia** - har Plan 8:30-17 mån-ons (inte torsdag)
- **Michael** (Michael Jansson) - har Plan 8:30-17 endast tisdag

Enligt reglerna finns inga specifika begränsningar för vem som kan åka på resa. Alla ovanstående personer skulle kunna åka på reportaget.

De som har Plan hela veckan (Emil, Per S, Tomas, Sara, Per L, Iman och Fredrik) är troligen de mest

Upload Walk & Talk mode

Type here, or press the mic button to talk... Send

Fact check Thinking Show tool calls & thinking

Agent Document

Bemanningsschema - visualisering Edit

React-app som visualiserar bemannings schemat för fotografer per vecka

Text Preview Changes Save

Bemanningschema - Vecka 40

Alla < Föregående vecka **Vecka 40** Nästa vecka >

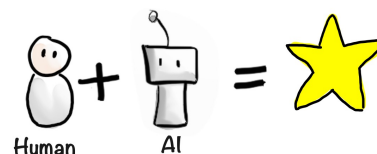
PERSON	Måndag 2025-09-29	Tisdag 2025-09-30	Onsdag 2025-10-01	Torsdag 2025-10-02	Fredag 2025-10-03
Alex		Agenda 09:00 - 19:30	Agenda 09:00 - 18:00	Agenda 09:00 - 19:00	Agenda 09:00 - 20:00
Andreas H	Sport 04:00 - 10:30	Sport 04:00 - 10:30	Sport 04:00 - 10:30		
Andreas I	Kultur 09:00 - 20:30	Kultur 09:00 - 20:30	Kultur 07:30 - 16:00	Kultur 09:00 - 18:00	
Antonia	Plan 08:30 - 17:00	Plan 08:30 - 17:00	Plan 08:30 - 17:00	Plan 08:30 - 17:00	Plan 08:30 - 16:00
Cornelia	Plan 08:30 - 17:00	Plan 08:30 - 17:00	Plan 08:30 - 17:00	Plan 08:30 - 17:00	Plan 08:30 - 16:00
David	Plan 08:30 - 17:00	Plan 08:30 - 17:00	Plan 08:30 - 17:00	Plan 08:30 - 17:00	Plan 08:30 - 16:00
Emil	Desk 08:00 - 16:00		Desk 06:00 - 15:00	Desk 06:00 - 15:00	Desk 06:00 - 15:00
Fredrik	Plan 08:30 - 17:00	Plan 08:30 - 17:00	Plan 08:30 - 17:00	Plan 08:30 - 17:00	Plan 08:30 - 16:00
Henrik	Kultur 07:30 - 16:00		Kultur 09:00 - 20:30	Kultur 07:30 - 16:00	
Håkan	Desk 06:00 - 15:00	Desk 06:00 - 15:00			Desk 11:00 - 22:00
Iman	Plan 08:30 - 17:00	Plan 08:30 - 17:00	Plan 08:30 - 17:00	Plan 08:30 - 17:00	Plan 08:30 - 16:00
Jocke	Kultur 04:00 - 10:30	Kultur 04:00 - 10:30		Kultur 09:00 - 20:30	Kultur 07:30 - 16:00
Marco	Desk 11:00 - 22:00	Desk 11:00 - 22:00			Desk 09:00 - 19:30

Agent can insert suggestions

insertDocumentData(documentId: string, items: Array<object>): Promise<void>

```
{
  id: ...
  week: 38,
  date: "2025-09-15",
  weekday: "måndag",
  person: "Rebecka",
  program: "Desk",
  time: "04:00 - 10:30",
  suggestion: {
    program: "Plan",
    time: "09:00 - 17:00",
    comment: "Has a dentist appointment in the early morning"
  }
}
```

Working together on changes



Bemanningsschema - Vecka 40							
2 förändringsförslag				✓ Godkänn alla		✗ Avvisa alla	
Alla		< Föregående vecka		Vecka 40		Nästa vecka >	
PERSON	Måndag 2025-09-29	Tisdag 2025-09-30	Onsdag 2025-10-01	Torsdag 2025-10-02	Fredag 2025-10-03	Lördag 2025-10-04	Söndag 2025-10-05
Alex		Agenda 09:00 - 19:30	Agenda 09:00 - 18:00	Agenda 09:00 - 19:00	Agenda 09:00 - 20:00		
Andreas H	Sport 04:00 - 10:30	Sport 04:00 - 10:30	Sport 04:00 - 10:30				
Andreas I	Kultur 09:00 - 20:30	Kultur 09:00 - 20:30	Kultur 07:30 - 16:00	Kultur			
Antonia	Plan 08:30 - 17:00	Plan 08:30 - 17:00	Plan 08:30 - 17:00				
Cornelia	Plan 08:30 - 17:00	Plan 08:30 - 17:00	Plan 08:30 - 17:00				
David	Plan 08:30 - 17:00	Plan 08:30 - 17:00	Plan 08:30 - 17:00				
Emil	Desk 08:00 - 16:00		Desk 06:00 - 15:00	Desk 06:00 - 15:00	FÖRSLAG Desk 08:00 - 16:00		
Fredrik	Plan 08:30 - 17:00	Plan 08:30 - 17:00	Plan 08:30 - 17:00	Plan 08:30 - 17:00	Plan 08:30 - 16:00		
Henrik	Kultur 07:30 - 16:00		Kultur 09:00 - 20:30	Kultur 07:30 - 16:00			
Håkan	Desk 06:00 - 15:00	Desk 06:00 - 15:00			Desk 11:00 - 22:00	Desk 10:00 - 20:00	Desk 10:00 - 20:00
Iman	Plan 08:30 - 17:00	Plan 08:30 - 17:00	Plan 08:30 - 17:00	Plan 08:30 - 17:00	Plan 08:30 - 16:00		

Förslag på schemaändring

Emil - fredag 2025-10-03

Nuvarande pass:

Desk
06:00 - 15:00

Förslag:

Desk
08:00 - 16:00

Kommentar:

Byte med Niclas - Henrik föreslog detta

Stäng

Avvisa

Godkänn

But how can the agent transform the data in a simple & repeatable way?

Without human intervention

[illegible]

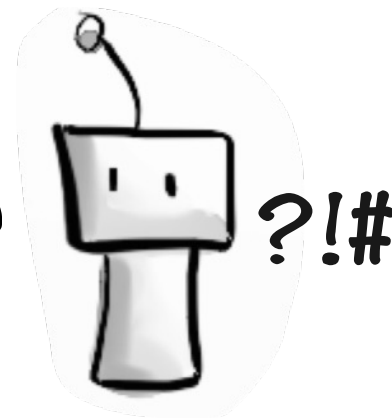
```
[
  {
    "id": "d97c93c0-8511-48e1-b421-4aaf56448928",
    "vecka": 38,
    "datum": "2025-09-15",
    "veckodag": "måndag",
    "person": "Emil",
    "program": "Desk",
    "tid": "08:30 - 17:00"
  },
  {
    "id": "1f07ac20-1b5f-4bda-a58b-8bc63b0c7aef",
    "vecka": 38,
    "datum": "2025-09-16",
    "veckodag": "tisdag",
    "person": "Emil",
    "program": "Desk",
    "tid": "08:00 - 17:00"
  },
  {
    "id": "bbb95044-b612-4f4f-af78-f29e6c16ac7b",
    "vecka": 38,
    "datum": "2025-09-17",
    "veckodag": "onsdag",
    "person": "Emil",
    "program": "Desk",
    "tid": "08:00 - 17:00"
  },
]
```

Importing & converting the data: Bad way

Google spreadsheet (CSV)



Data in



JSON data



Transformed data out



I asked the agent to create its own script to transform the data



"Create and save a reusable script that imports data from the spreadsheet to the schedule database. Check the format of the spreadsheet and the database so the code knows how to transform it. Then test the script with a smaller dataset to check that it works."

```
// Read the Google Sheet
const sheetDataRow = await callTool('read_google_sheet', {
  spreadsheetId: '1LcgLhwGRtLEoL34-nJ5gy-00vqCnfiC-Ez9gbSvM4y4'
});

// The response contains the descriptive text and then a JSON array
// Find where the actual array starts (after the newline following
const lines = sheetDataRow.split('\n');
let jsonStr = '';
let foundStart = false;

for (let line of lines) {
  if (line.trim().startsWith('{')) {
    foundStart = true;
  }
  if (foundStart) {
    jsonStr += line;
  }
}

const rows = JSON.parse(jsonStr);

const scheduleEntries = [];
```

```
// Start from row 4 (index 3) - that's where the actual schedule data starts
for (let rowIndex = 3; rowIndex < rows.length; rowIndex++) {
  const row = rows[rowIndex];
  const personName = row[0].replace(/^\\A\\d+\\/, '').trim(); // Remove cell ref

  if (!personName) continue; // Skip empty rows

  // Process each column (starting from column B, index 1)
  for (let colIndex = 1; colIndex < row.length; colIndex++) {
    const cellValue = row[colIndex].replace(/^\\[A-Z]+\\d+\\/, '').trim(); // Remove cell ref

    if (cellValue && cellValue !== '-' && cellValue !== '') {
      // Get the date from row 2 (index 1)
      const date = rows[1][colIndex].replace(/^\\[A-Z]+\\d+\\/, '').trim();

      // Get the day name from row 3 (index 2)
      const dayName = rows[2][colIndex].replace(/^\\[A-Z]+\\d+\\/, '').trim();

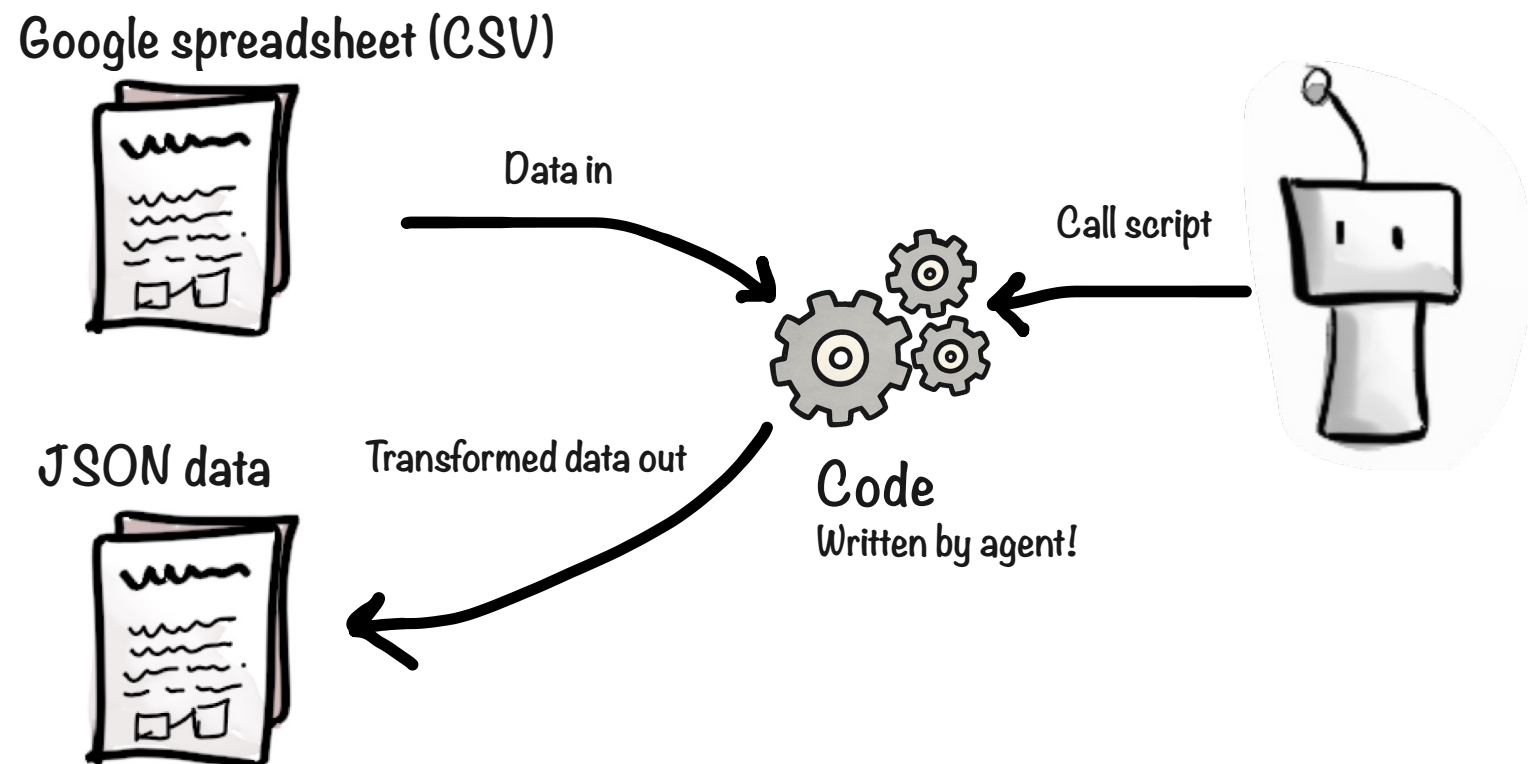
      // Get the week from row 1 (index 0)
      const week = rows[0][colIndex].replace(/^\\[A-Z]+\\d+\\/, '').trim();

      if (date && date !== '') {
        scheduleEntries.push({
          id: `${personName}-${date}`,
          date: date,
          dayName: dayName,
          week: week,
          photographerName: personName,
          activity: cellValue
        });
      }
    }
  }
}
```

```
// Create the data document
const result = await callTool('add_document', {
  name: 'schedule',
  description: 'Schedule data with photographer assignments',
  mimeType: 'application/vnd.abundly.collection',
  dataSchema: `interface ScheduleEntry {
    id: string;
    date: string;
    dayName: string;
    week: string;
    photographerName: string;
    activity: string;
  }`,
  data: scheduleEntries
});

return {
  message: `Created schedule document with ${scheduleEntries.length} entries`,
  documentId: result.documentId,
  sampleEntries: scheduleEntries.slice(0, 5)
};
```

Importing & converting the data: Good way

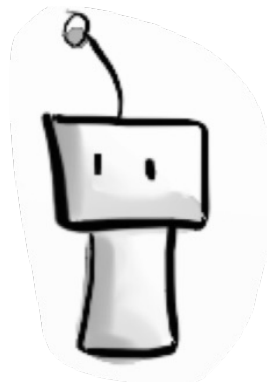


A corkboard with a light brown, textured surface. In the center, there is a white rectangular note with a thin black border. The text "Agent design tips" is written on the note in a black, casual script font.

Agent design tips

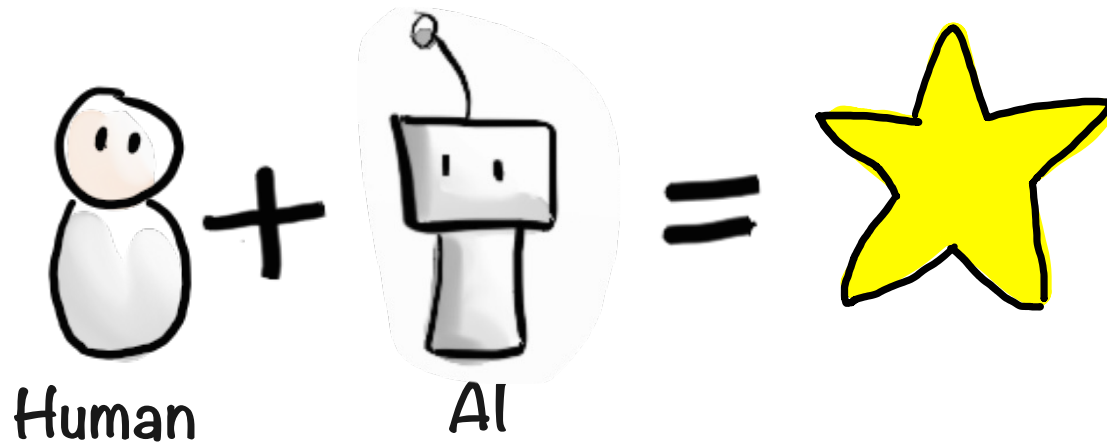
Think of it as an intern

- An extremely knowledgeable intern – but some limitations
- It knows many things, but not so much about your specific context and needs
- It needs feedback

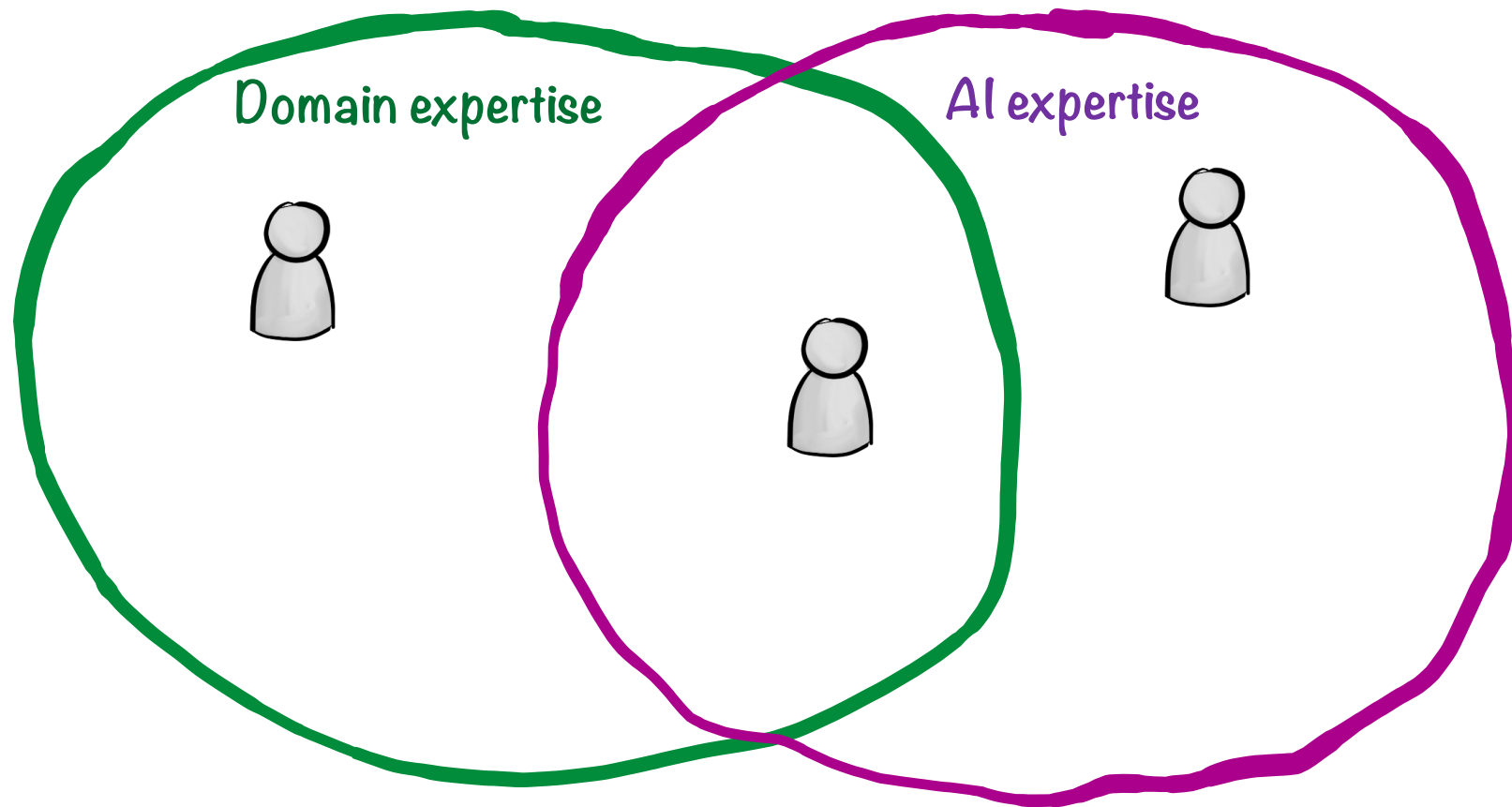


AI agents work best when collaborating with humans

- Shared Goal
- Shared workspace
- Specific responsibilities / tasks

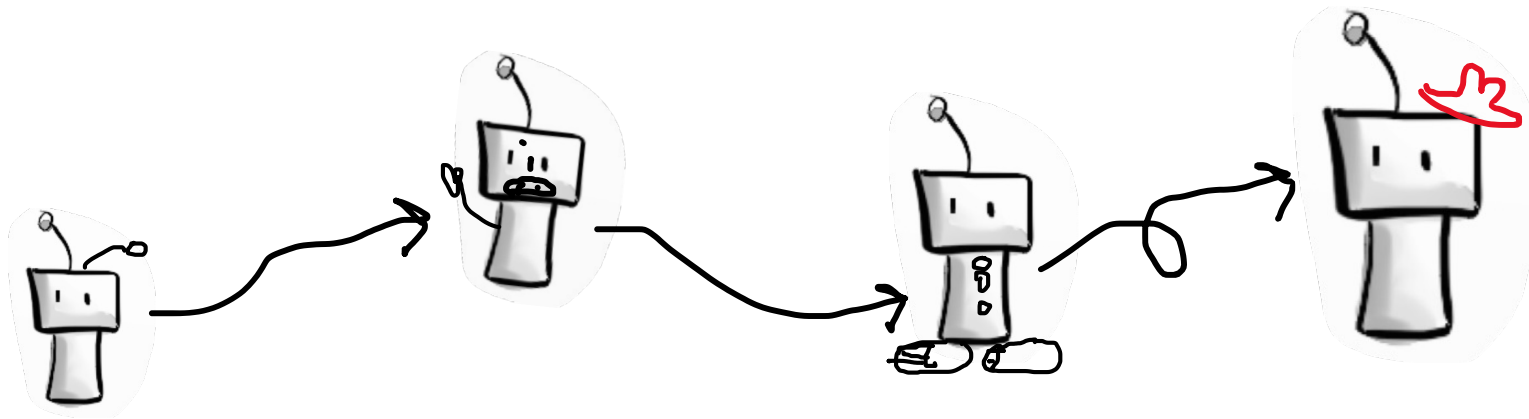


Tip: Get the right people into the room



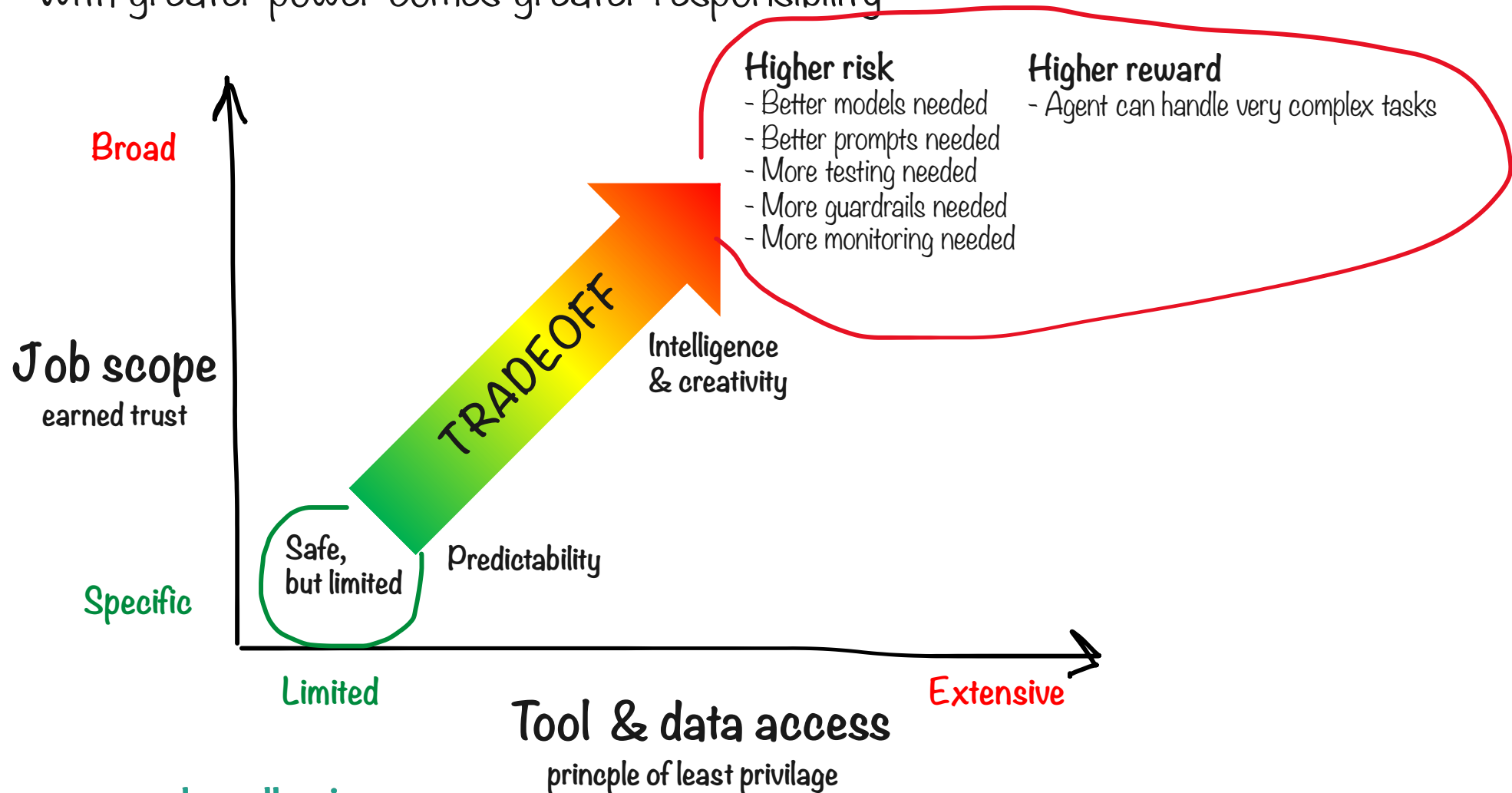
Iterate.

- With the right tools, skills, and use cases, you can build a decent agent pretty quickly.
- But to make it work really well, you need to iterate a lot



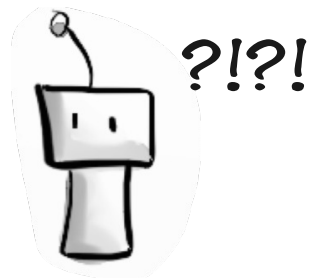
What about safety?

With greater power comes greater responsibility



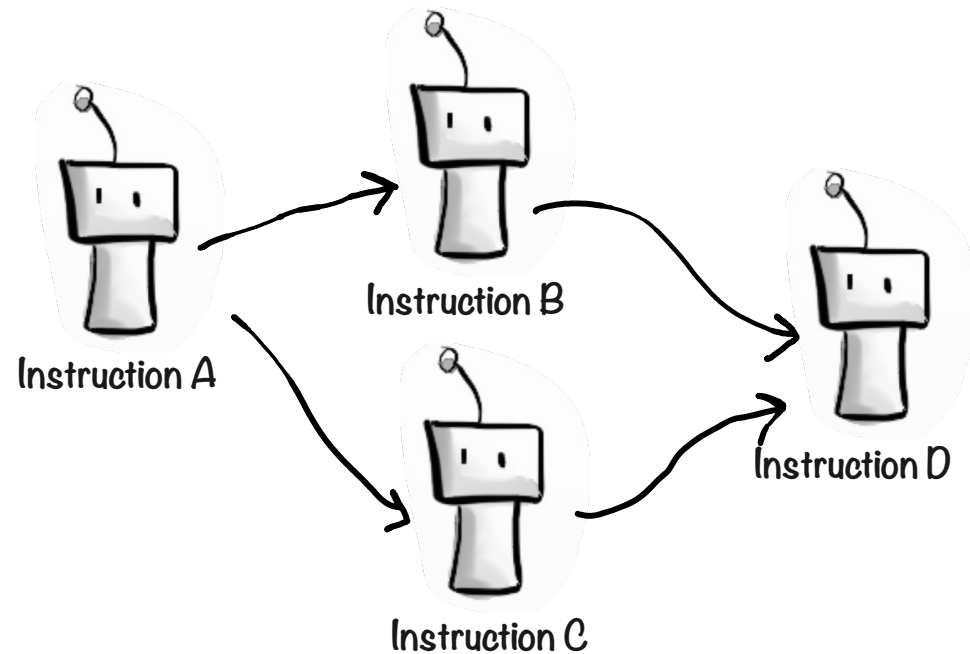
Agents work best if they don't get too many instructions and tools
(similar to humans...)

Single complex agent



Instruction A
Instruction B
Instruction C
Instruction D

Agent team



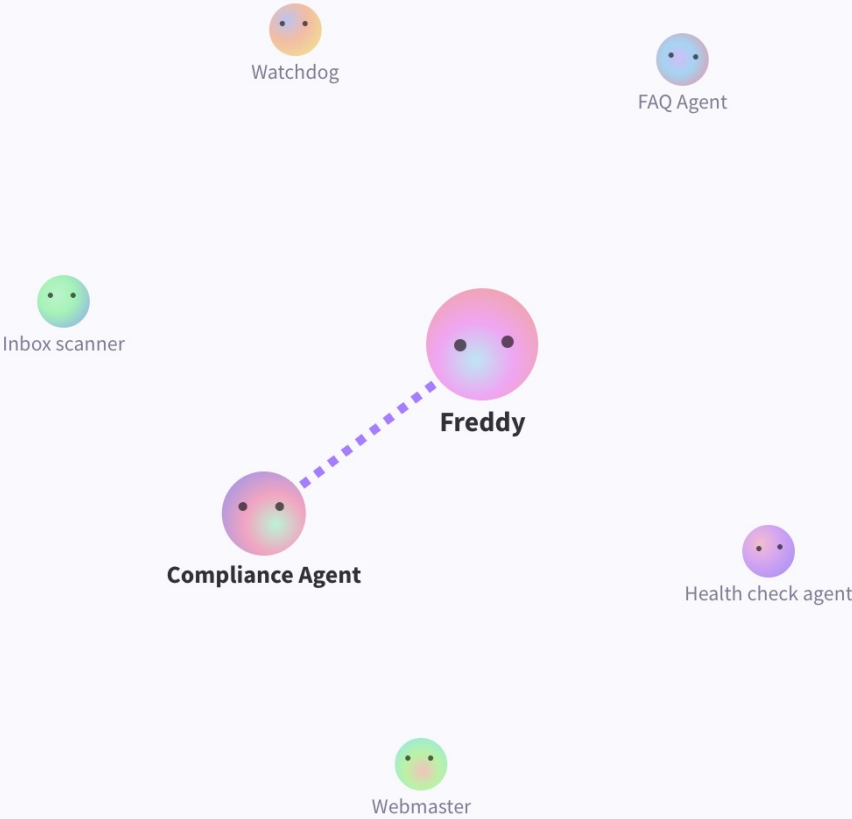
Agent Settings

Main Settings **Agent Communication** Access Management Sharing Alarms

Agent Communication

Which other agents am I allowed to talk to? Click to allow or disallow communication.

Show all agents ☒



Agent Design Canvas

AGENT NAME

AGENT OWNER

Purpose

- What is the scope of the job that this agent is tasked to do?
- Who would benefit, and how?

Triggers

- What event starts the agent?
- Is it manual, scheduled, or event-driven?

Action Plan (Human & AI)

Create numbered steps showing the workflow, clearly indicating which steps the AI agent performs and where humans are involved. Include review points, approvals, and handoffs between human and AI. Each step should specify who does what.

Example:

1. AI: Analyzes incoming document
2. AI: Extracts key information
3. Human: Reviews and validates findings
4. AI: Generates draft response
5. Human: Approves and sends

Interfaces

- How will human(s) and AI interact (chat, shared doc, approve, joint doc, other)?
- What are the functional requirements on the interface(s)?

Knowledge & State

- What external documents or databases to connect with?
- What background context & domain knowledge is needed?
- What types of working data does the agent maintain and modify?

Input

- What data or documents are provided?
- What format should inputs be in?

Output & Success

- What are the deliverables?
- What format should they be (templates, examples)?
- How do we know it became better/worse (evals)?
- What metrics define success?

Capabilities

- What integrations are needed?
- Other tools needed to perform actions well?

Cool thing: Agents
making their own
tools, on the fly

HTTP request tool = magic wand to interact with any API/service

Created a new agent with only two tools

Agent Capabilities

Here you decide which capabilities I can use.

Active Capabilities

All Capabilities

Web Search

Search for information using Perplexity



HTTP requests

Make HTTP requests to external APIs



HTTP requests Settings

Require Approval

No



Notify Agent on approval

Never



10 minutes ago



Hi! How can I help you?



Show a map of the area around Tap1 in copenhagen.

HTTP request tool = magic wand to interact with any API/service

Send your first HTTP request

Now
Hi! How can I help you?

1 minute ago

Show a map of the area around Tap1 in copenhagen.

I'll help you create a map of the area around Tap1 in Copenhagen. Let me create an interactive map for you using an embedded map viewer.

✖ Tool call: perplexity_research >

Great! Now I'll create an interactive map showing Tap1 at Raffinaderivej 10, Copenhagen:

✖ Tool call: http_request >

Perfect! Now I'll create an interactive map for you:

📄 Map of Tap1 Copenhagen

I've created an interactive map showing **Tap1** in Copenhagen! The map shows:

- **Location:** Raffinaderivej 10, 2300 Copenhagen, Denmark (in the former distillery area at Kløvermarken on Amager Island)
- A marker you can click for more details
- A shaded circle showing the immediate area around the venue
- You can zoom in/out and pan around to explore the neighborhood

Getting there:

- 🚌 Bus 37 to "Raffinaderivej" (5 min walk)
- 🚌 Bus 31 to "Holmbladsgade" (10 min walk)
- 🚇 Metro M2 to Lergravsparken or Øresund stations (20 min walk)

📎 Upload 🗣️ Walk & Talk mode

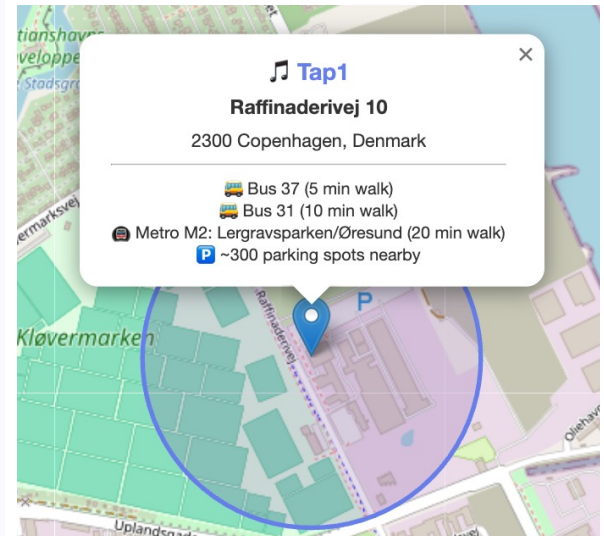


Type here, or press the mic button to talk...

Send

Map of Tap1 Copenhagen

Text Preview Changes



I asked the agent "How did you do it?"

The beauty of this approach is that it's all done with free, open-source tools and APIs - no API keys required! The HTTP request capability lets me call any REST API, so I can integrate with mapping services, weather APIs, or any other web service you might need.



Thoughts >

Great question! Here's how I created that map:

Step 1: Found the Address 📌

I used **Perplexity search** to find Tap1's exact address: Raffinaderivej 10, 2300 Copenhagen, Denmark.

Step 2: Got the GPS Coordinates 🌐

I made an **HTTP request** to OpenStreetMap's Nominatim API (a free geocoding service) to convert the address into precise coordinates:

- Latitude: 55.6719360
- Longitude: 12.6195340

The API call looked like this:

```
GET https://nominatim.openstreetmap.org/search?q=Raffinaderivej+10,+2300+Copenhagen,+Denmark&format=json
```

Step 3: Created an Interactive Map 🗺️

I built an **HTML page** using:

- **Leaflet.js** - a popular open-source JavaScript library for interactive maps (loaded from cdnjs.cloudflare.com)
- **OpenStreetMap tiles** - the actual map imagery (free to use)
- Custom marker, popup info, and a circle overlay to highlight the area

The beauty of this approach is that it's all done with **free, open-source tools** and APIs - no API keys required! The HTTP request capability lets me call any REST API, so I can integrate with mapping services, weather APIs, or any other web service you might need.

1. Called perplexity API

✂ Tool call: perplexity_research ▼

```
{
  "query": "Tap1 Copenhagen address location"
}
```

→ Result (1637 chars):

```
{
  "model": "sonar",
  "content": "**TAP1 Copenhagen** is located at **Raffinaderivej 10, 2300 Copenhagen, Denmark**, in the former distillery area at Kløvermarken on Amager Island[1][2][3][4]. \n\nIt is approximately 4 km from the Town Hall Square and Copenhagen Central Station (about 9 minutes by car or 15 minutes by bike) and about 5.5 km from Copenhagen Airport[1]. \n\nPublic transportation options to TAP1 include: \n- Bus 37 (stop \"Raffinaderivej\") - 5 minutes walk to TAP1 \n- Bus 31 (stop \"Hol...\"
}
```

Show more

2. Did http request to OpenStreetMap.org

✂ Tool call: http_request ▼

```
{
  "method": "get",
  "url": "https://nominatim.openstreetmap.org/search",
  "params": {
    "q": "Raffinaderivej 10, 2300 Copenhagen, Denmark",
    "format": "json",
    "limit": 1
  }
}
```

→ Result (741 chars):

```
{
  "status": 200,
  "statusText": "OK",
  "headers": {
    "server": "nginx",
    "date": "Wed, 01 Oct 2025 11:11:29 GMT",
    "content-type": "application/json; charset=utf-8",
    "content-length": "486",
    "connection": "keep-alive",
    "keep-alive": "timeout=30",
    "vary": "accept, language",
    "data": "..."
  }
}
```

3. Created a visualization using HTML/JS

Map of Tap1 Copenhagen

Text Preview Changes



```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Tap1 Copenhagen Map</title>
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/leaflet/1.9.4/leaflet.css" />
  <script src="https://cdnjs.cloudflare.com/ajax/libs/leaflet/1.9.4/leaflet.js">
</script>
<style>
  body {
    margin: 0;
    padding: 0;
    font-family: Arial, sans-serif;
  }
  #map {
    width: 100%;
    height: 500px;
  }
  .info-box {
    padding: 15px;
    background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
    color: white;
    text-align: center;
  }
  .info-box h1 {
```

A corkboard with a light brown, textured surface. In the center, there is a white rectangular sign with a thin black border. The word "Wrapup" is written on the sign in a black, casual script font.

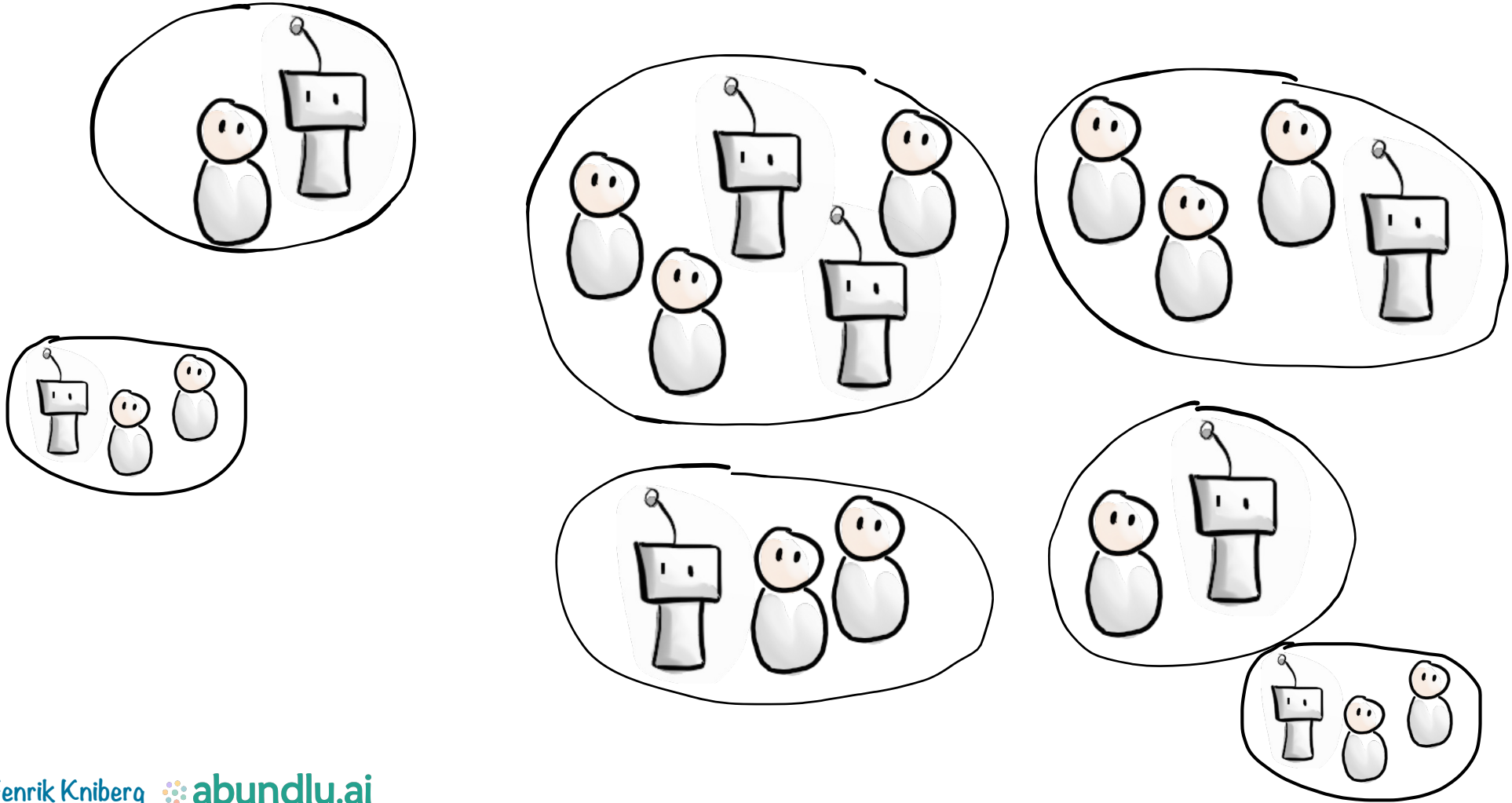
Wrapup

More tips (if time permits...)

- Debugging is different (more like being a coach, or shrink)
- Hallucination is not a problem if you have the right model & right context
- Get used to not having 100% predictability
- Building a cool & impressive agent is pretty easy, but making it work really well in practice takes quite a few iterations

Future of Work:

Teams of Humans and AI agents working side by side



Thanks for listening!

Download
these slides

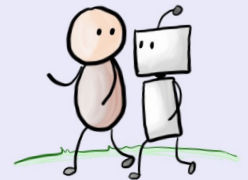
Get the book

Learn more
about agents

 abundly.ai



**Generative AI
in a Nutshell**
How to Survive and Thrive
in the Age of AI



Henrik Kniberg