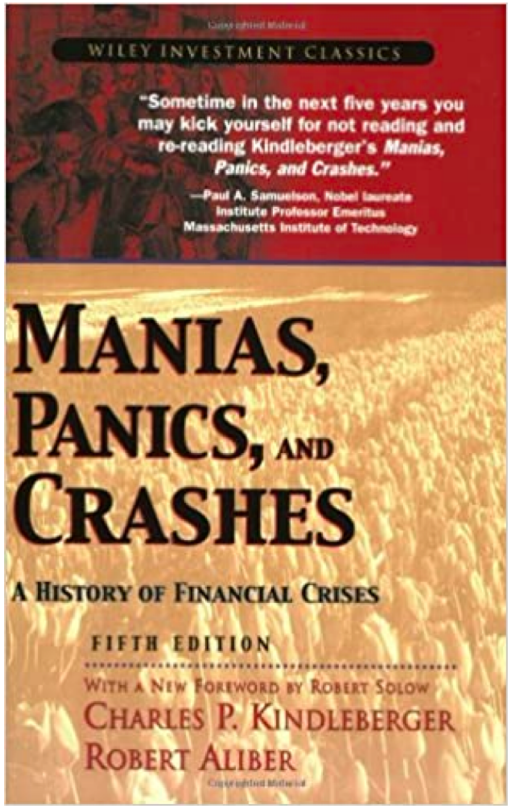
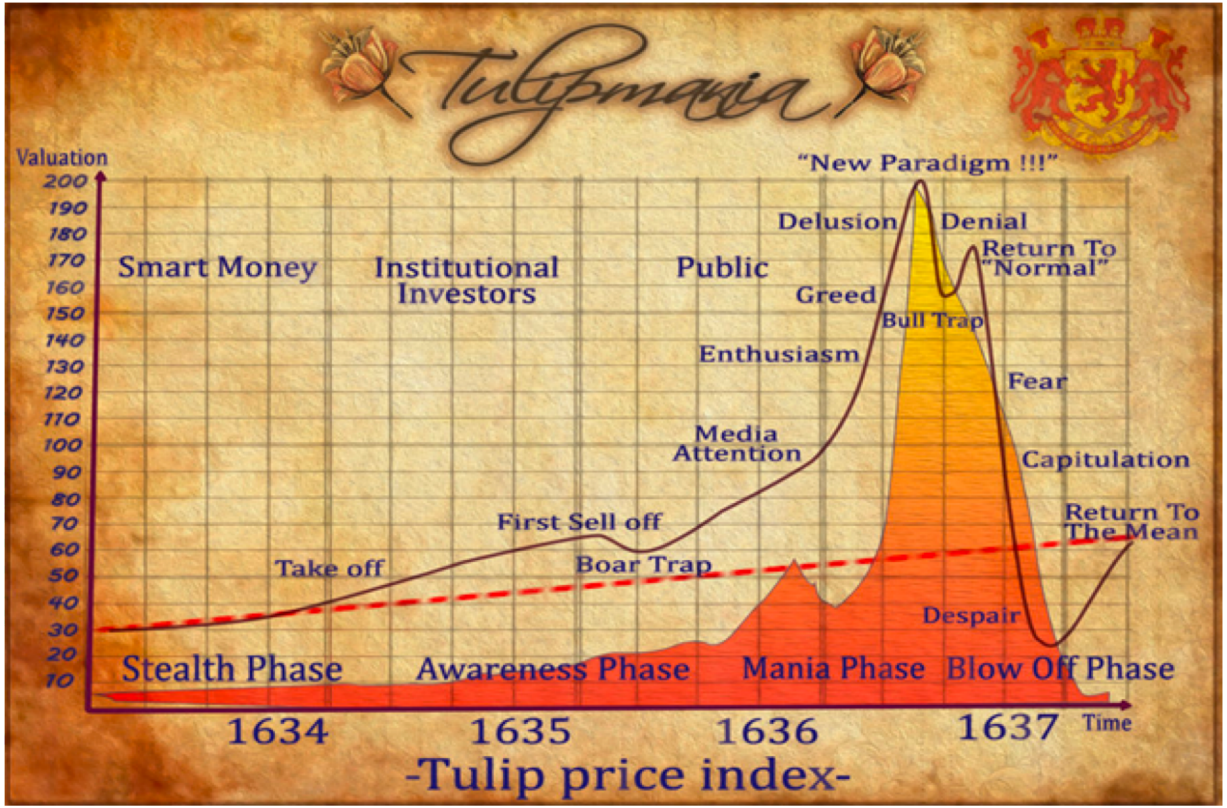
The Anatomy of Autonomy

swyx

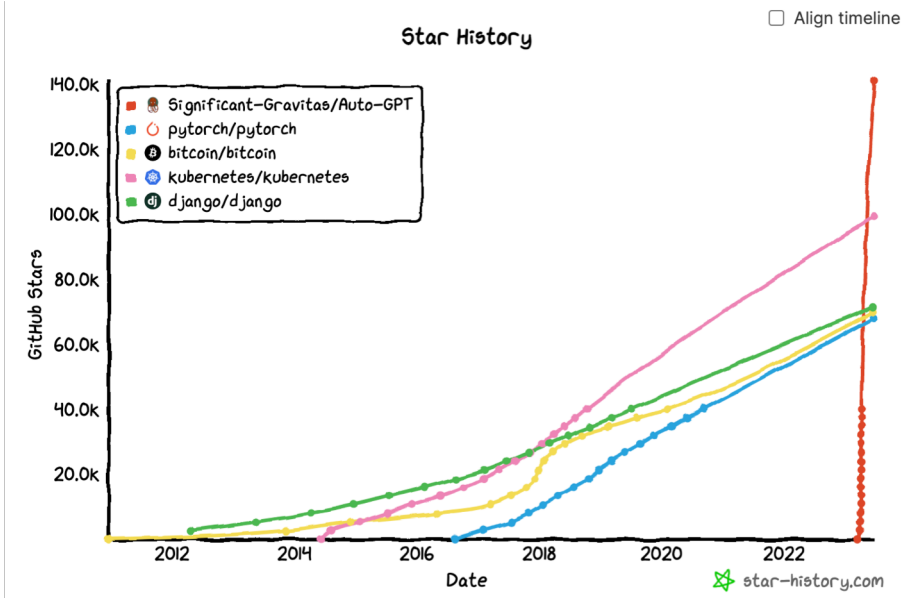
Latent Space

Agent Mania


WILEY INVESTMENT CLASSICS
"Sometime in the next five years you may kick yourself for not reading and re-reading Kindleberger's Manias, Panics, and Crashes."
—Paul A. Samuelson, Nobel laureate
Institute Professor Emeritus
Massachusetts Institute of Technology
MANIAS, PANICS, AND CRASHES
A HISTORY OF FINANCIAL CRISES
FIFTH EDITION
WITH A NEW FOREWORD BY ROBERT SOLOW
CHARLES P. KINDLEBERGER
ROBERT ALIBER


Tulipmania
Valuation
200
190
180
170
160
150
140
130
120
110
100
90
80
70
60
50
40
30
20
10
Smart Money
Institutional Investors
Public
"New Paradigm !!!"
Delusion
Denial
Return To "Normal"
Greed
Bull Trap
Enthusiasm
Fear
Media Attention
Capitulation
First Sell off
Return To The Mean
Take off
Boar Trap
Despair
Stealth Phase
Awareness Phase
Mania Phase
Blow Off Phase
1634
1635
1636
1637
Time
-Tulip price index-

Agents are the Tulips of AI Summer



How do we make useful agents?

Tech's Two Philosophies

Tech's Two Philosophies

Wednesday, May 9, 2018

🤖 The Zuck School

Tech tracks you,
feeds you,
decides for you

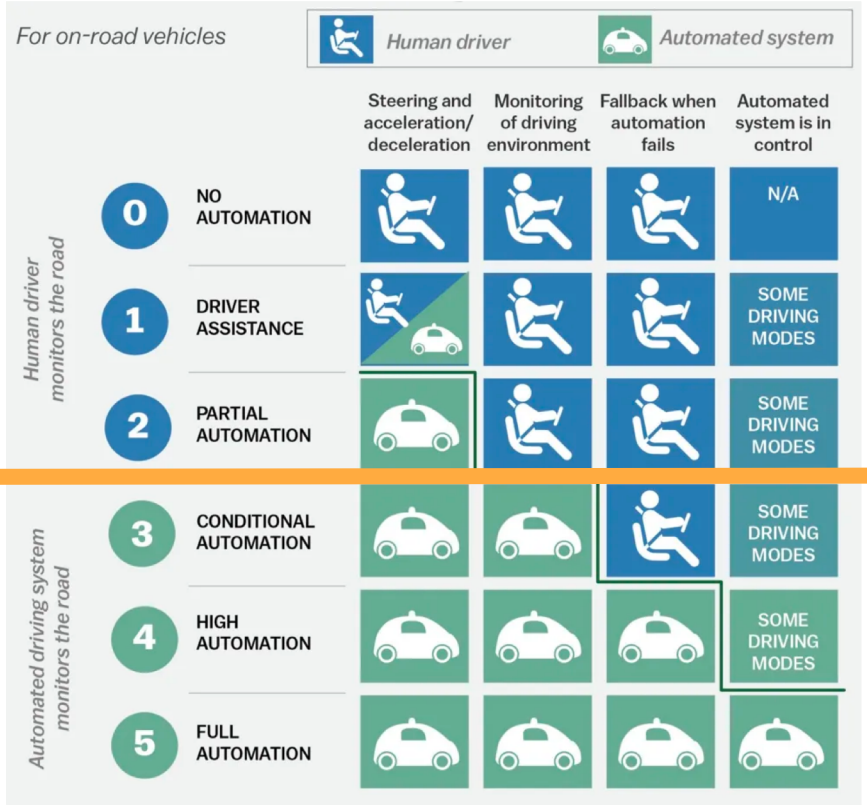
🚲 The Jobs School

Tech is bicycle for the mind

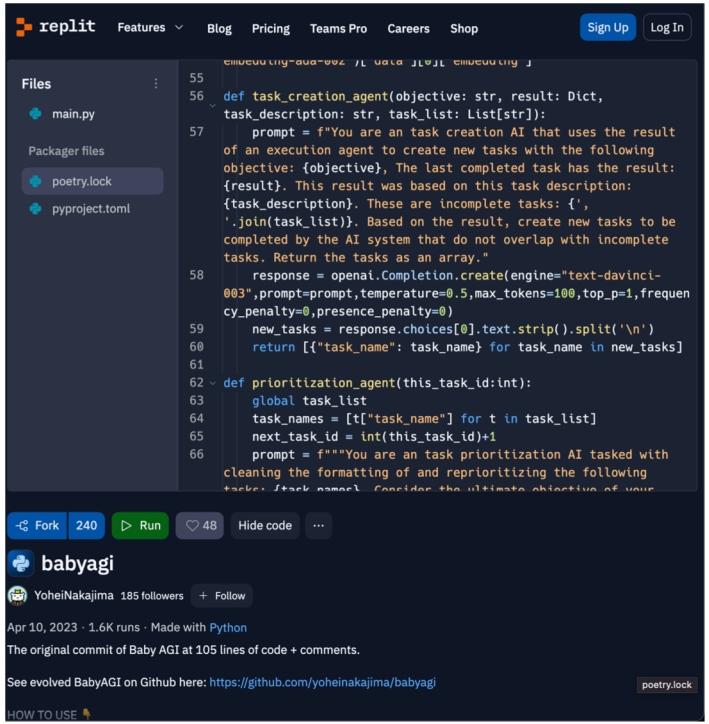
The Level 2 vs 3 Divide

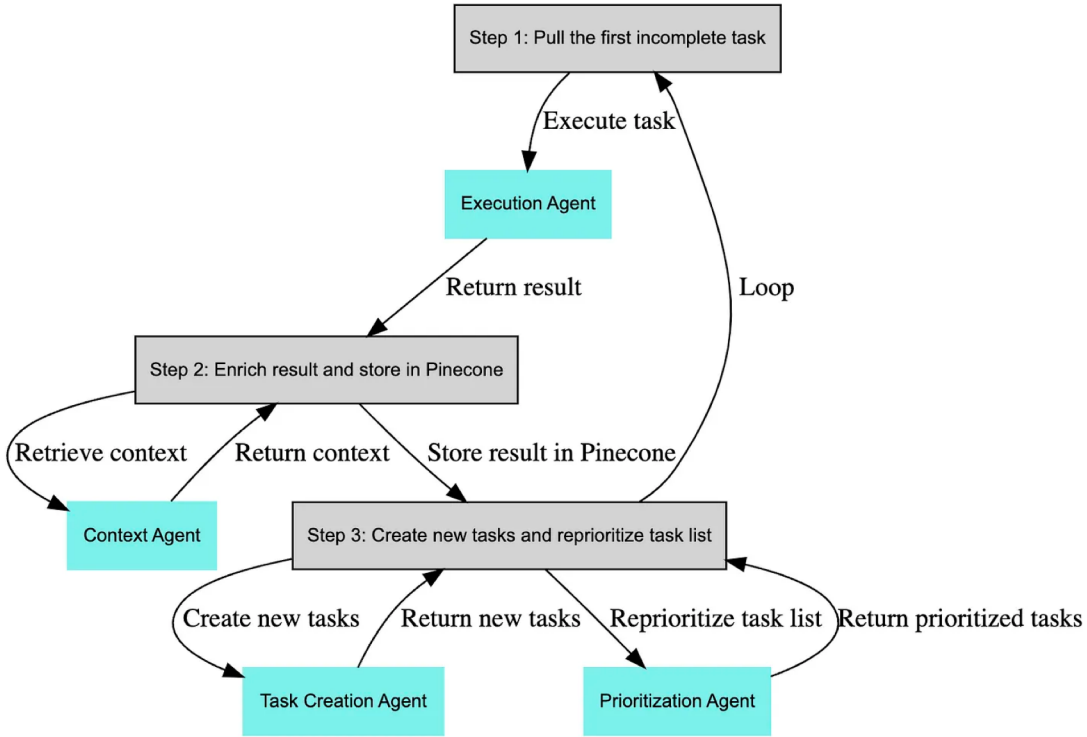
Human driver

AI driver

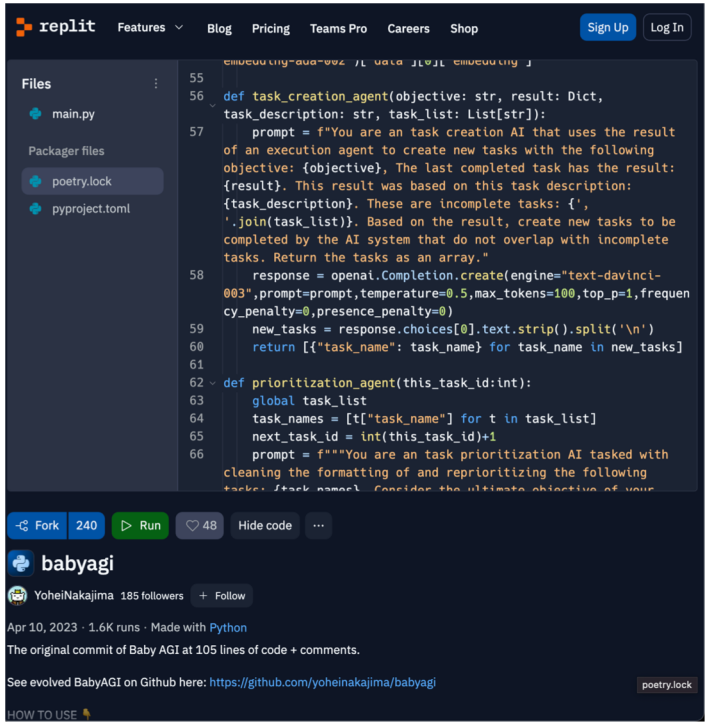


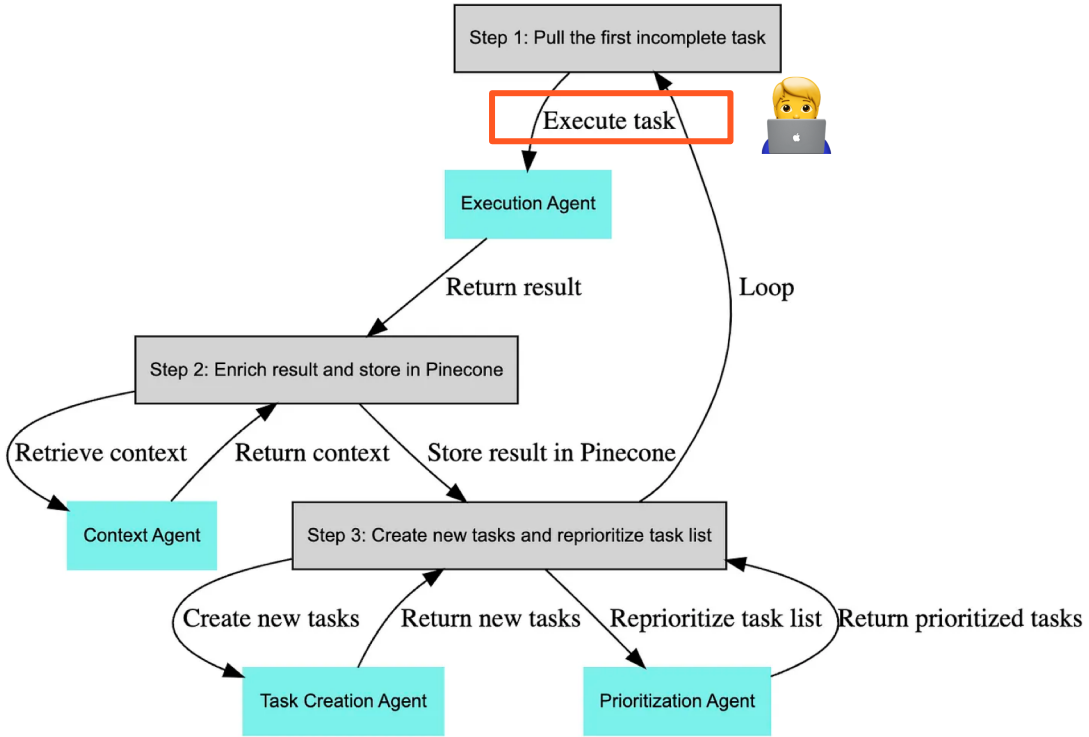
BabyAGI


replit
Features
Blog
Pricing
Teams Pro
Careers
Shop
Sign Up
Log In
Files
main.py
Packager files
poetry.lock
pyproject.toml
def task_creation_agent(objective: str, result: Dict, task_description: str, task_list: List[str]):
prompt = f"You are an task creation AI that uses the result of an execution agent to create new tasks with the following objective: {objective}, The last completed task has the result: {result}. This result was based on this task description: {task_description}. These are incomplete tasks: {', '.join(task_list)}. Based on the result, create new tasks to be completed by the AI system that do not overlap with incomplete tasks. Return the tasks as an array."
response = openai.Completion.create(engine="text-davinci-003",prompt=prompt,temperature=0.5,max_tokens=100,top_p=1,frequency_penalty=0,presence_penalty=0)
new_tasks = response.choices[0].text.strip().split('\n')
return [{"task_name": task_name} for task_name in new_tasks]
def prioritization_agent(this_task_id:int):
global task_list
task_names = [t["task_name"] for t in task_list]
next_task_id = int(this_task_id)+1
prompt = f"""You are an task prioritization AI tasked with cleaning the formatting of and reprioritizing the following
Fork
240
Run
48
Hide code
babyagi
YoheiNakajima
185 followers
Follow
Apr 10, 2023 · 1.6K runs · Made with Python
The original commit of Baby AGI at 105 lines of code + comments.
See evolved BabyAGI on Github here: https://github.com/yoheinakajima/babyagi
poetry.lock
HOW TO USE


Step 1: Pull the first incomplete task
Execute task
Execution Agent
Return result
Loop
Step 2: Enrich result and store in Pinecone
Retrieve context
Return context
Store result in Pinecone
Context Agent
Step 3: Create new tasks and reprioritize task list
Create new tasks
Return new tasks
Reprioritize task list
Return prioritized tasks
Task Creation Agent
Prioritization Agent

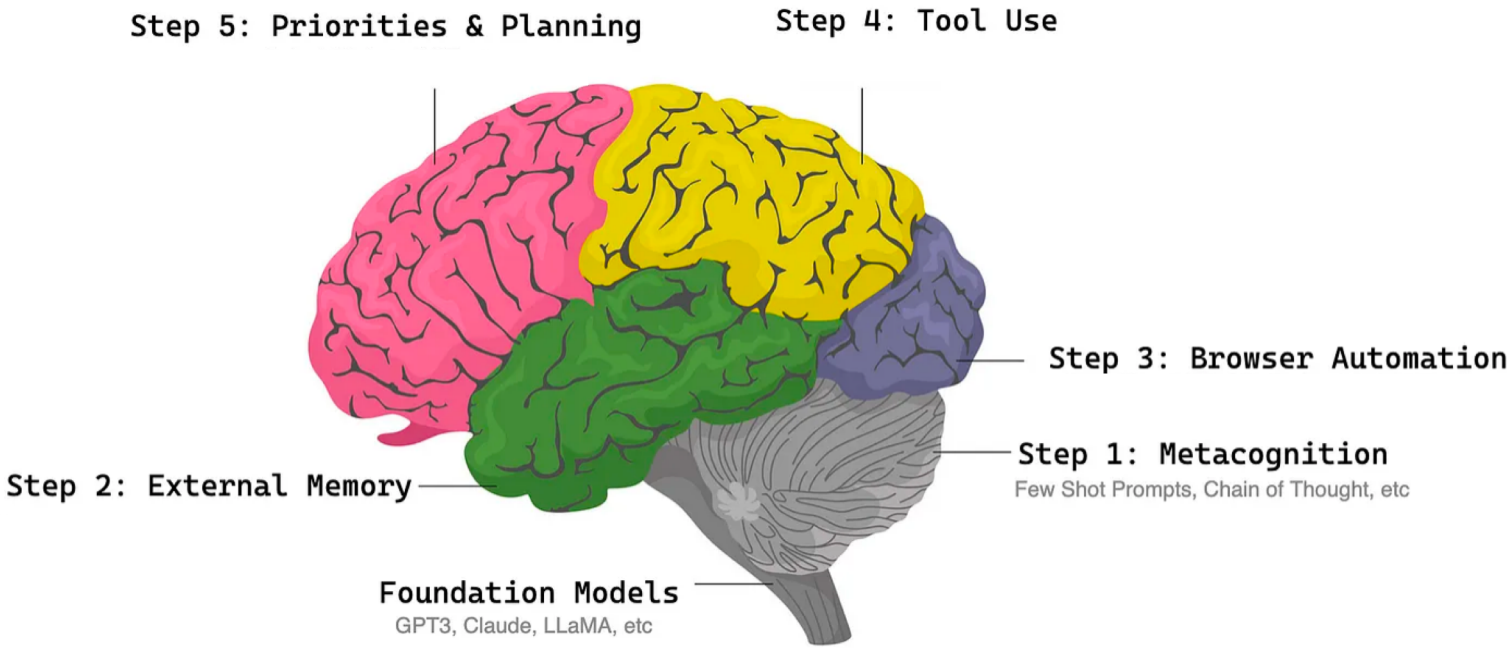
BabyAGI - Interrupt based Level 3 Agent





Anatomy of Autonomy

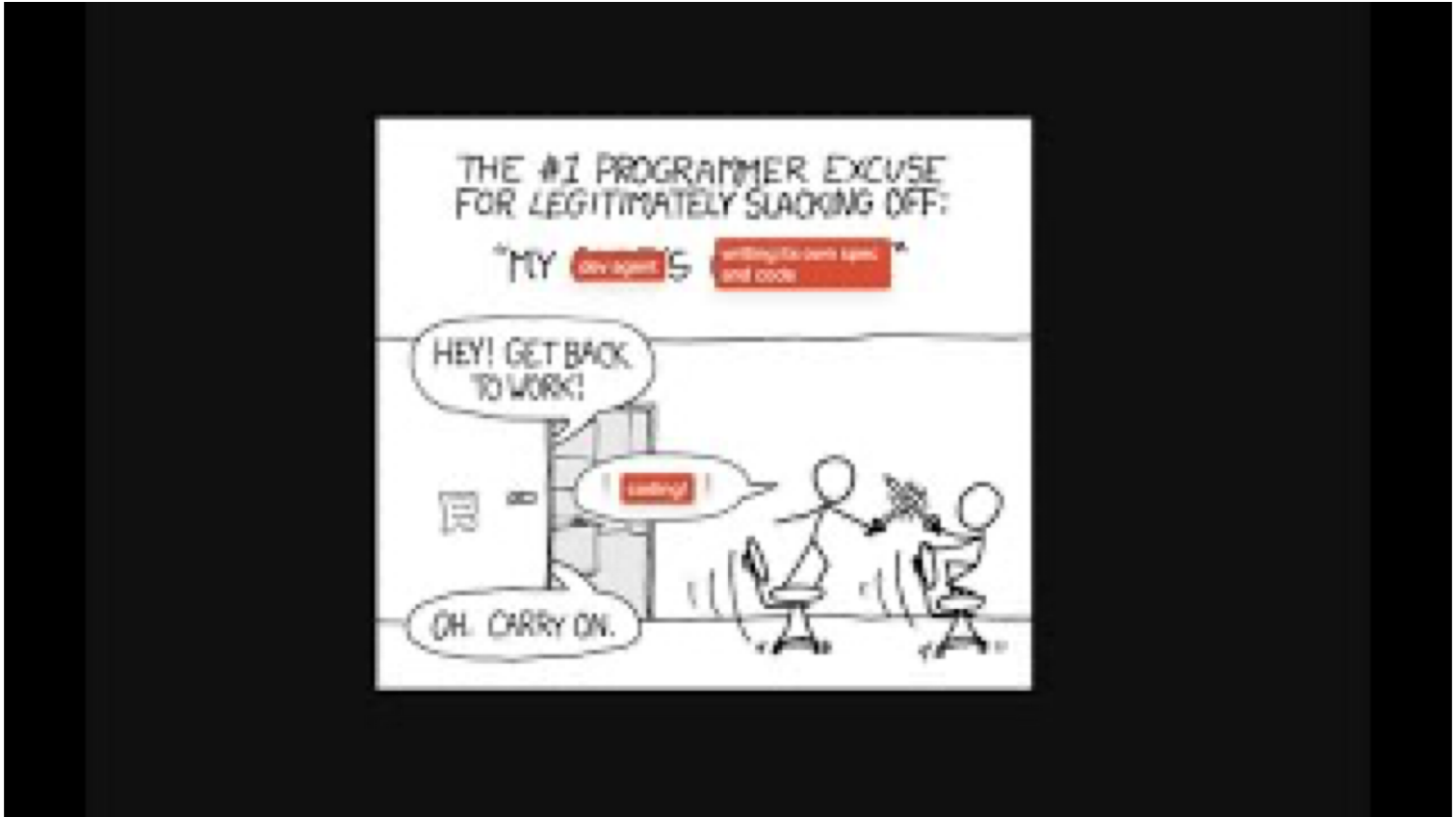
https://latent.space/p/agents



The Central Problem of Smol Developer (v0)

- AI native companies should have 0.5-0.1x engineers per stage of company
  - Or else they are not very AI native
  - Long term, OpenAI paying engineers $900k is a bug not feature (tho a very nice bug!)
- I want to make a lot of smol apps
  - Productivity starts with the smol things
    - Twemex/Twitter Links
    - HNX - Hacker News Chrome extension
    - Smol menubar
    - ????
- **Why can I not create smol apps at the speed of thought?**
  - Copilot only autocompletes once I know what I'm doing
  - AutoGPT gets stuck in loops and isn't focused on code

Want: Anthropic 100k Summarizer



Prompt to Whole App

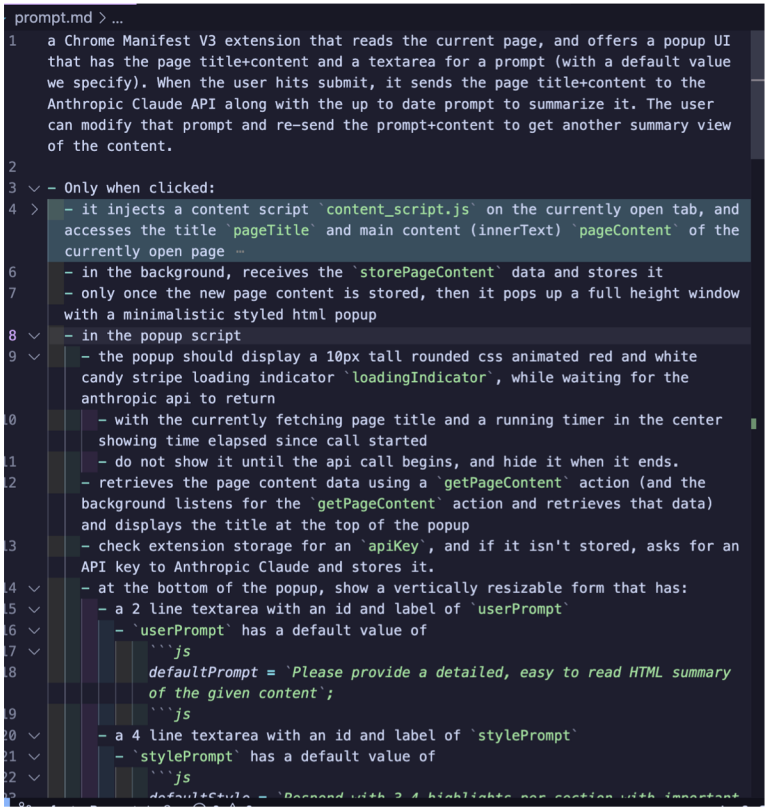
Minimal viable starting point

```bash
python main.py --prompt "a Chrome extension that, when clicked, opens a small window with a page where you can enter a prompt for reading the currently open page and generating some response from openai" --model=gpt-4
```

Markdown is all you need

```bash
# prompt in markdown file
python main.py --prompt prompt.md --model=gpt-4
```

Example of Prompt.md



prompt.md > ...

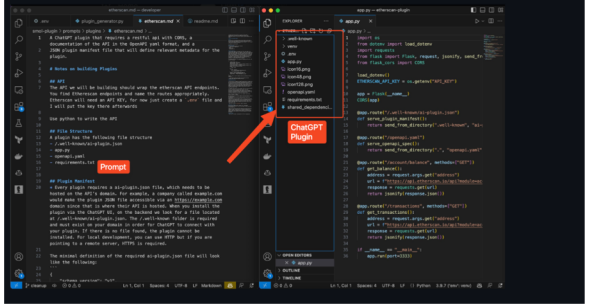
```
a Chrome Manifest V3 extension that reads the current page, and offers a popup UI
that has the page title+content and a textarea for a prompt (with a default value
we specify). When the user hits submit, it sends the page title+content to the
Anthropic Claude API along with the up to date prompt to summarize it. The user
can modify that prompt and re-send the prompt+content to get another summary view
of the content.

- Only when clicked:
  - it injects a content script `content_script.js` on the currently open tab, and
  accesses the title `pageTitle` and main content (innerText) `pageContent` of the
  currently open page …
  - in the background, receives the `storePageContent` data and stores it
  - only once the new page content is stored, then it pops up a full height window
  with a minimalistic styled html popup
  - in the popup script
    - the popup should display a 10px tall rounded css animated red and white
    candy stripe loading indicator `loadingIndicator`, while waiting for the
    anthropic api to return
      - with the currently fetching page title and a running timer in the center
      showing time elapsed since call started
      - do not show it until the api call begins, and hide it when it ends.
    - retrieves the page content data using a `getPageContent` action (and the
    background listens for the `getPageContent` action and retrieves that data)
    and displays the title at the top of the popup
    - check extension storage for an `apiKey`, and if it isn't stored, asks for an
    API key to Anthropic Claude and stores it.
    - at the bottom of the popup, show a vertically resizable form that has:
      - a 2 line textarea with an id and label of `userPrompt`
        - `userPrompt` has a default value of
          ```js
          defaultPrompt = `Please provide a detailed, easy to read HTML summary
          of the given content`;
          ```js
      - a 4 line textarea with an id and label of `stylePrompt`
        - `stylePrompt` has a default value of
          ```js
```

- Markdown is all you need
  - Code in prompts
  - And Prompts in Code in Prompt
- Lower activation for unfamiliar APIs
  - Chrome Manifest v3
  - Css animation
  - Curl anthropic api and dump it
- Debugging by `cat`ing
- Copy and paste programming
  - “Logbook = prompt”
- **Whole program coherence**
  - **shared_dependencies.md**
  - **Name variables**
```

Community Development

`smol-plugin` - prompt to ChatGPT plugin (tweet, fork)



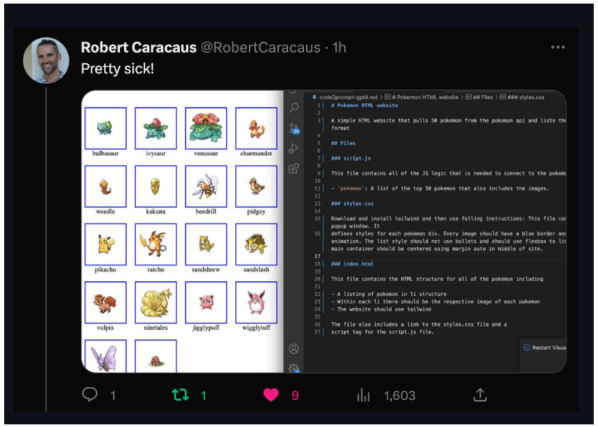
Lessons from Creating a VSCode Extension with GPT-4

KEVIN LIN

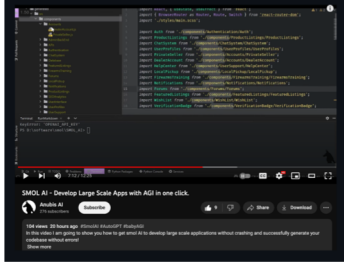
MAY 25, 2023

```
# Revised Political Campaign CRM Program Specification

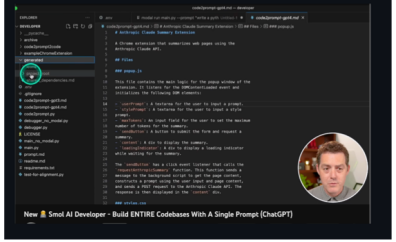
## Overview
The purpose of this CRM system is to efficiently manage and track voters, their
intentions, referrals, and requests for a political campaign. The back end of
this system will be implemented in Python using the SQLite3 database, while the
front end will use the Tailwind CSS framework. The system will include an
authentication system and will be designed for cloud deployment.
```



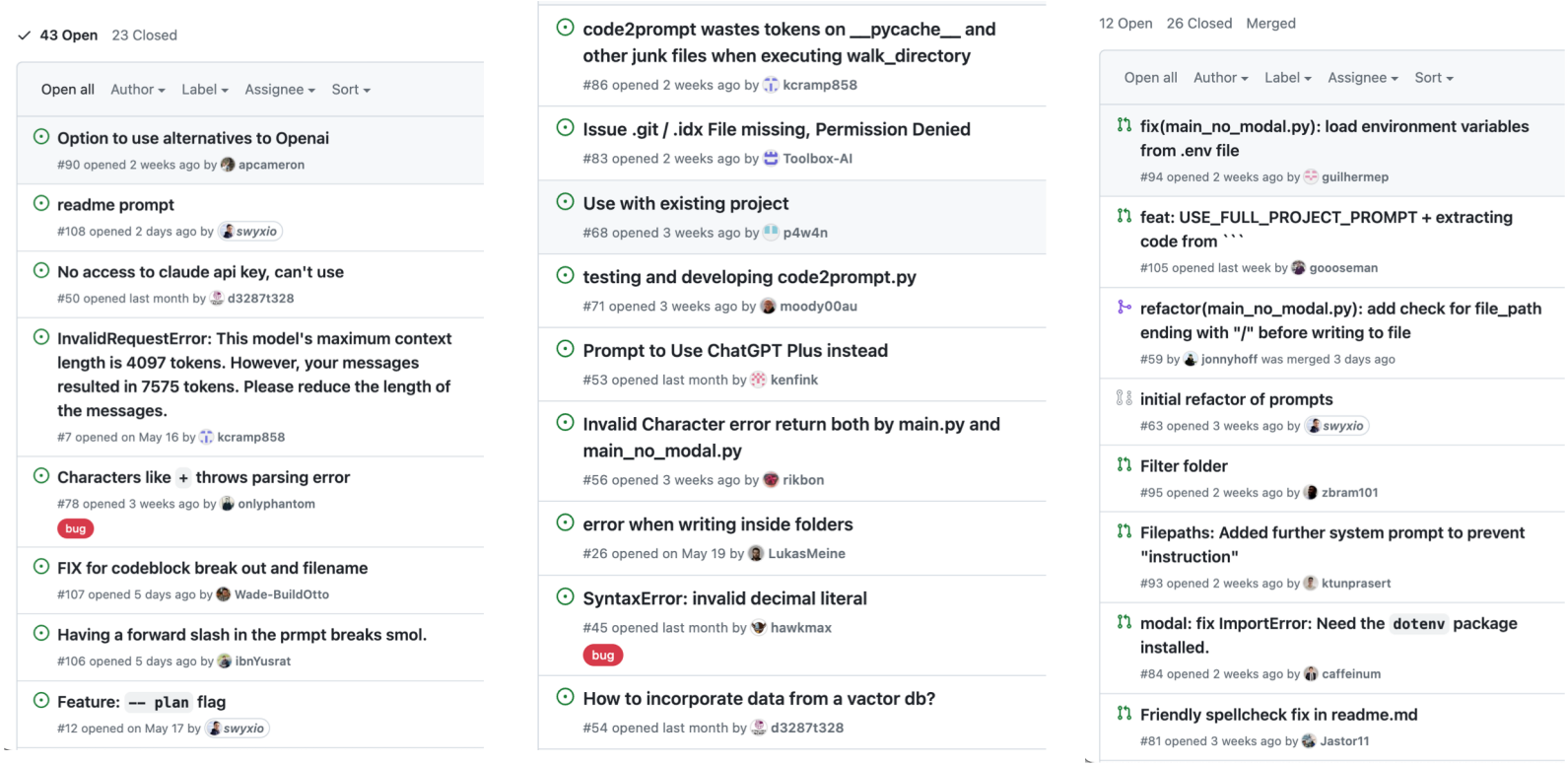
surprisingly complex React/Node/MongoDB full stack app



7 min Video: Smol AI Developer - Build ENTIRE
produces a full working OpenAI CLI python app



But this approach is problematic



✓ 43 Open 23 Closed

Open all Author Label Assignee Sort

- Option to use alternatives to Openai
  #90 opened 2 weeks ago by apcameron
- readme prompt
  #108 opened 2 days ago by swyxio
- No access to claude api key, can't use
  #50 opened last month by d3287t328
- InvalidRequestError: This model's maximum context length is 4097 tokens. However, your messages resulted in 7575 tokens. Please reduce the length of the messages.
  #7 opened on May 16 by kcramp858
- Characters like `+` throws parsing error
  #78 opened 3 weeks ago by onlyphantom
  bug
- FIX for codeblock break out and filename
  #107 opened 5 days ago by Wade-BuildOtto
- Having a forward slash in the prmpt breaks smol.
  #106 opened 5 days ago by ibnYusrat
- Feature: `--plan` flag
  #12 opened on May 17 by swyxio
- code2prompt wastes tokens on __pycache__ and other junk files when executing walk_directory
  #86 opened 2 weeks ago by kcramp858
- Issue .git / .idx File missing, Permission Denied
  #83 opened 2 weeks ago by Toolbox-AI
- Use with existing project
  #68 opened 3 weeks ago by p4w4n
- testing and developing code2prompt.py
  #71 opened 3 weeks ago by moody00au
- Prompt to Use ChatGPT Plus instead
  #53 opened last month by kenfink
- Invalid Character error return both by main.py and main_no_modal.py
  #56 opened 3 weeks ago by rikbon
- error when writing inside folders
  #26 opened on May 19 by LukasMeine
- SyntaxError: invalid decimal literal
  #45 opened last month by hawkmax
  bug
- How to incorporate data from a vactor db?
  #54 opened last month by d3287t328

12 Open 26 Closed Merged

Open all Author Label Assignee Sort

- fix(main_no_modal.py): load environment variables from .env file
  #94 opened 2 weeks ago by guilhermep
- feat: USE_FULL_PROJECT_PROMPT + extracting code from ```
  #105 opened last week by goooseman
- refactor(main_no_modal.py): add check for file_path ending with "/" before writing to file
  #59 by jonnyhoff was merged 3 days ago
- initial refactor of prompts
  #63 opened 3 weeks ago by swyxio
- Filter folder
  #95 opened 2 weeks ago by zbram101
- Filepaths: Added further system prompt to prevent "instruction"
  #93 opened 2 weeks ago by ktunprasert
- modal: fix ImportError: Need the `dotenv` package installed.
  #84 opened 2 weeks ago by caffeinum
- Friendly spellcheck fix in readme.md
  #81 opened 3 weeks ago by Jastor11

Problem: Planner too primitive

```
def planPrompt1():
    return """You are an AI developer who is tryi

    When given their intent, create a complete, ex

    only list the filepaths you would write, and
    do not add any other explanation, only return

    Good response:
    ["app.py","function.py","folder/file.py"]

    Bad response:
    - app.py
    - function.py
    - folder/file.py

    Good response:
    ["app.py","function.py","folder/file.py"]

    Bad response:
    - `app.py` a description here
    - `function.py` another description here
    - `folder/file.py` more description here
    """
```

```
the app is: Anki Flash Card Generator Chrome Extension

the files we have decided to generate are: manifest.json
styles.css

Shared dependencies:

1. Exported variables:
    - None

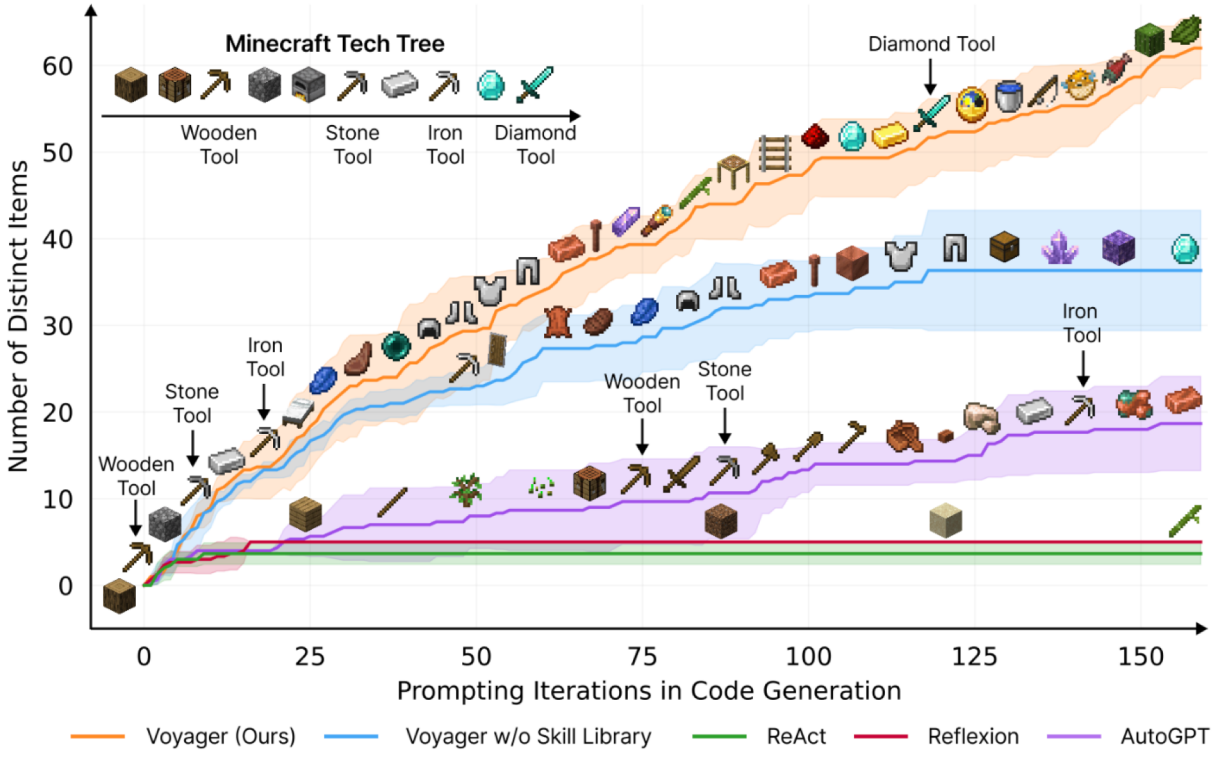
2. Data schemas:
    - AnkiCard: {question, answer, options: [A, B, C, D]}

3. DOM element id names:
    - api_key_input (popup.html)
    - submit_button (popup.html)
    - anki_cards_container (popup.html)

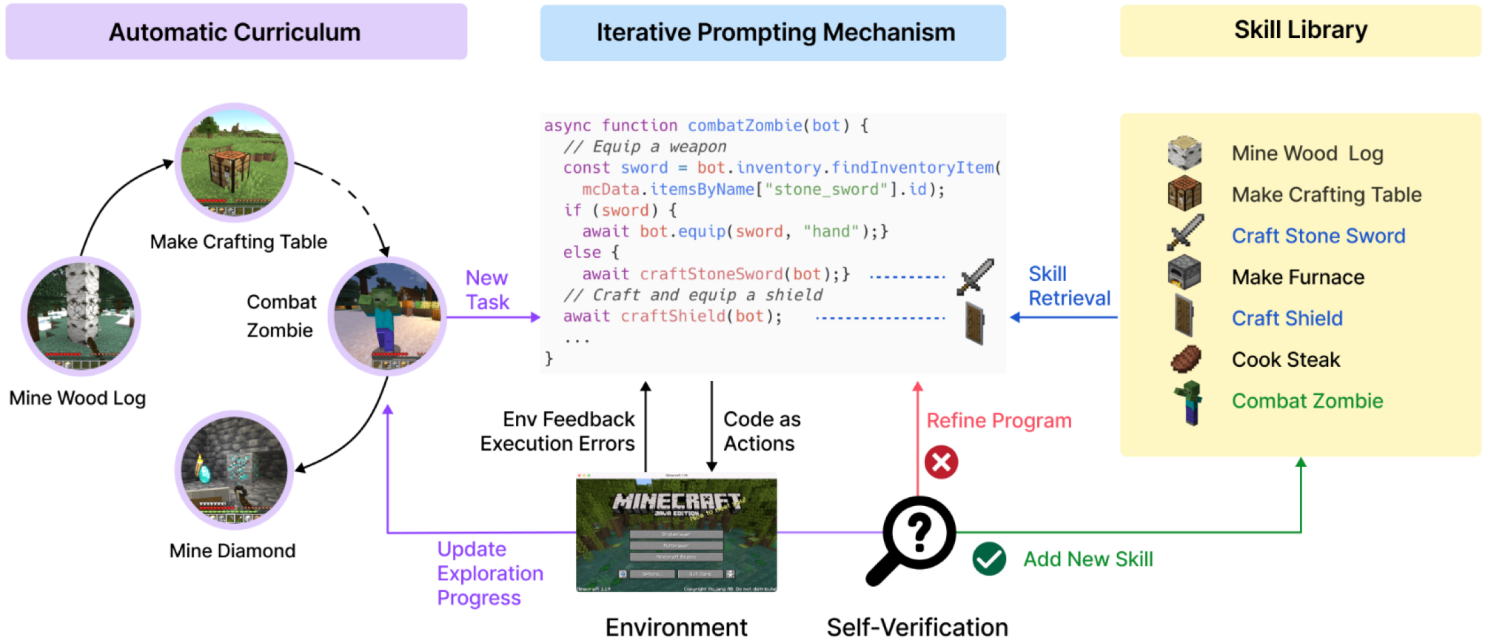
4. Message names:
    - content_request (contentScript.js -> popup.js)
    - content_response (popup.js -> contentScript.js)

5. Function names:
    - injectContentScript (background.js)
    - onExtensionIconClicked (background.js)
    - onPageContentReceived (popup.js)
    - generateAnkiCards (popup.js)
    - displayAnkiCards (popup.js)
    - formatAnkiCard (popup.js)
    - sendMessageToContentScript (popup.js)
```

Smol Developer: Voyager Edition



Voyager

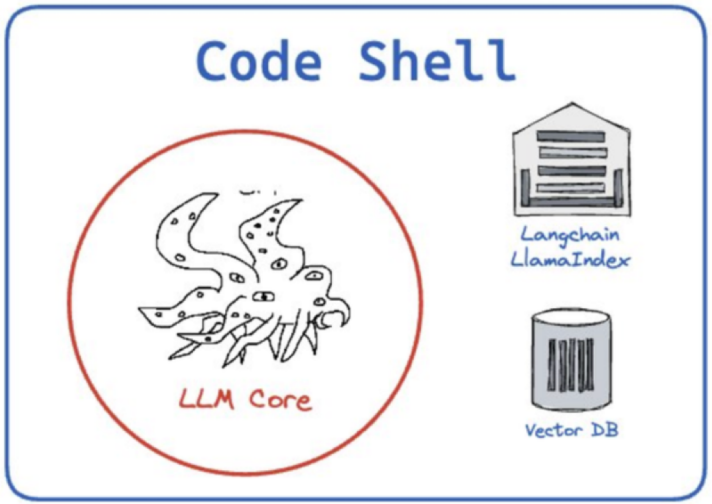


Voyager consists of three key components: an automatic curriculum for open-ended exploration, a skill library for increasingly complex behaviors, and an iterative prompting mechanism that uses code as action space.

“LLM Core” apps are fundamentally constrained by LLM capabilities

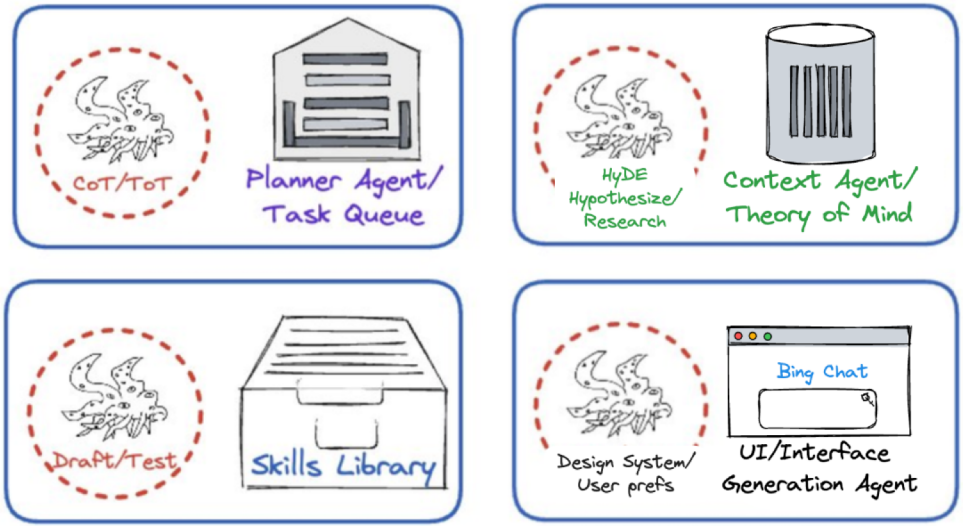
LLM Core
Code Shell

e.g. Retrieval Augmented Generation, Chat, Backend-GPT, Marvin AI, AutoGPT

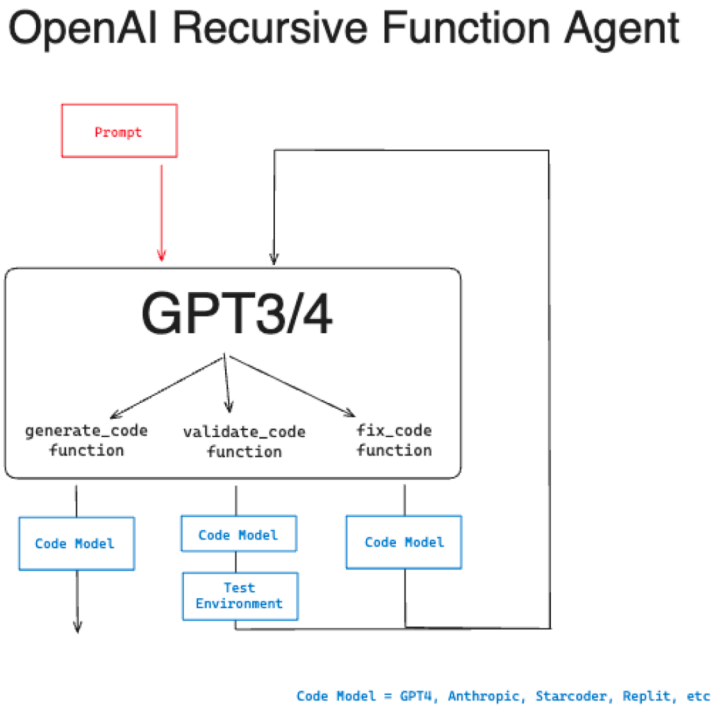


LLM Shell
Code Core

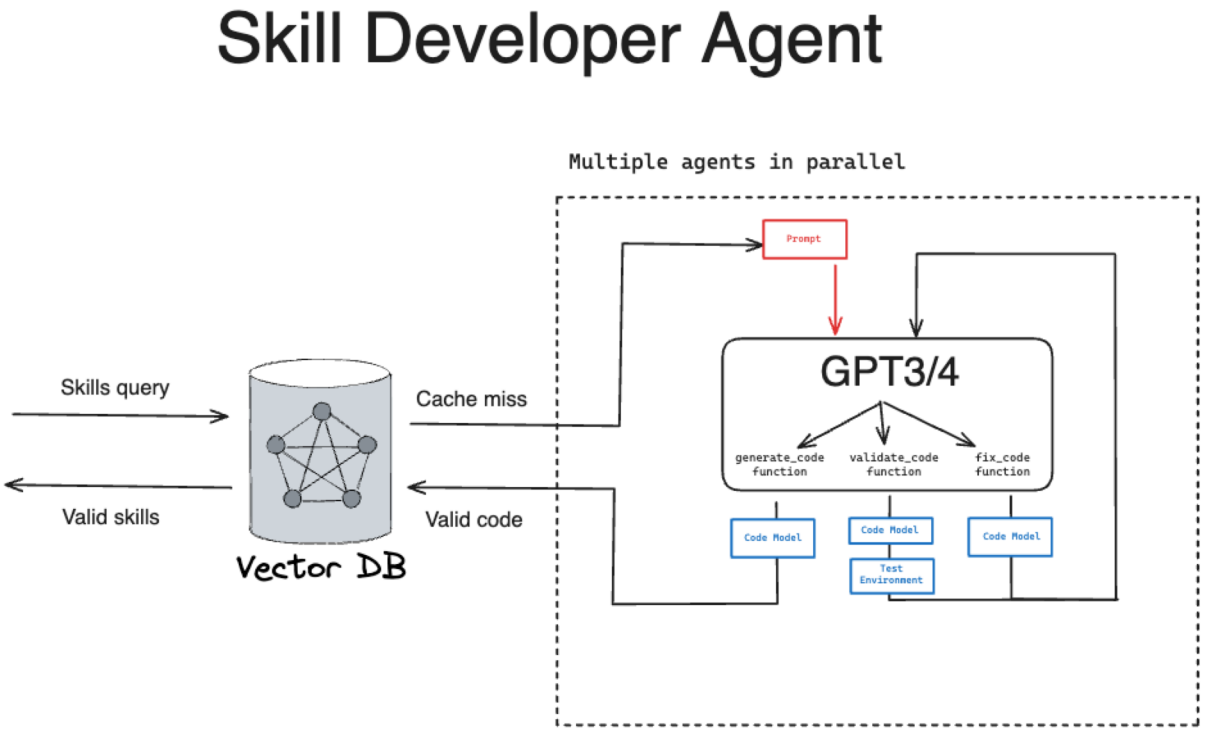
e.g. Copilot, Voyager, Smol-Developer



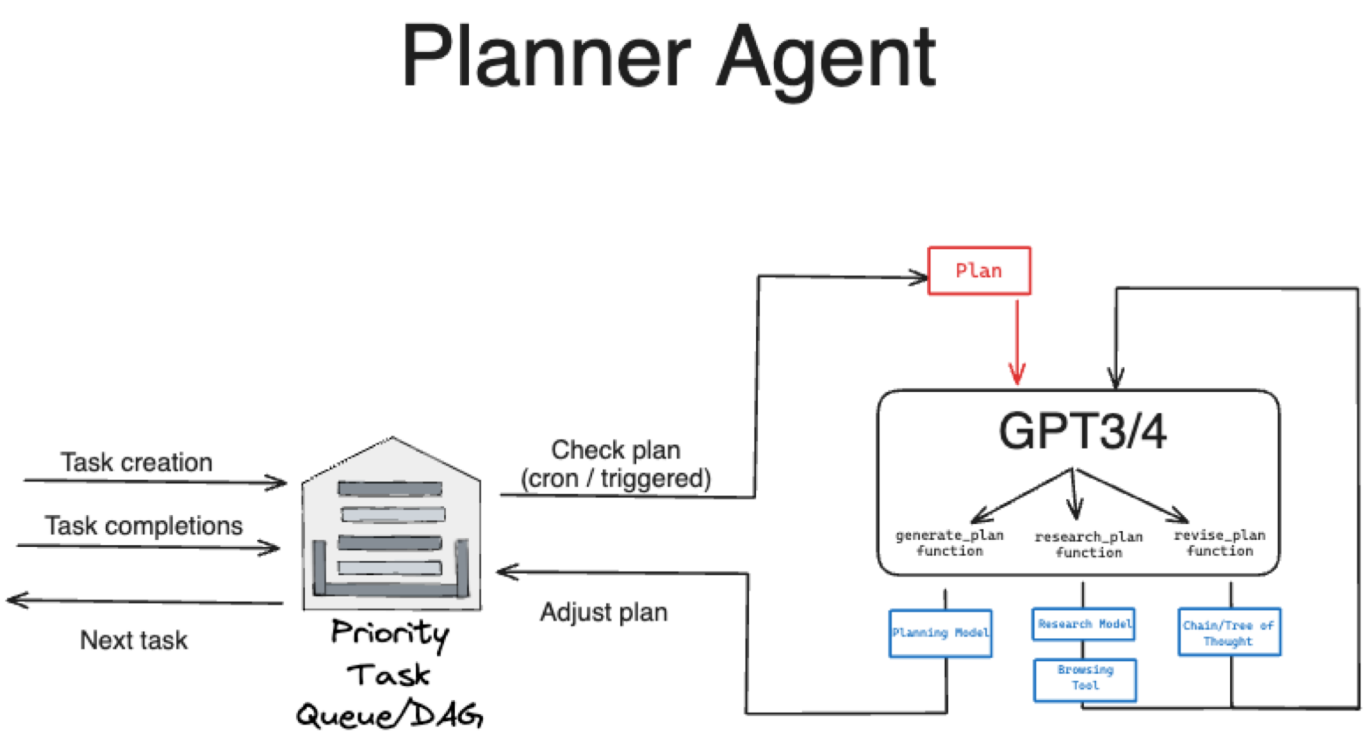
The XOR circuit of Code Core Agents



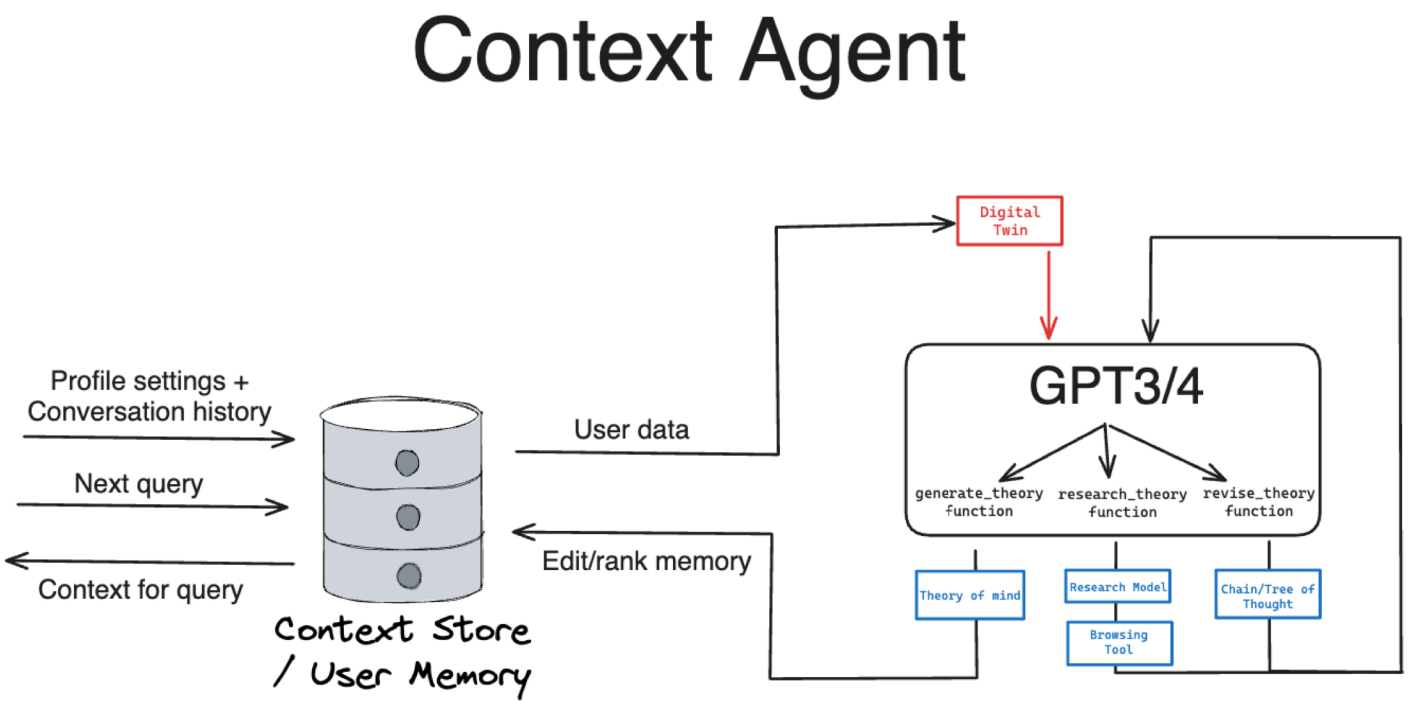
Building a Skills Library



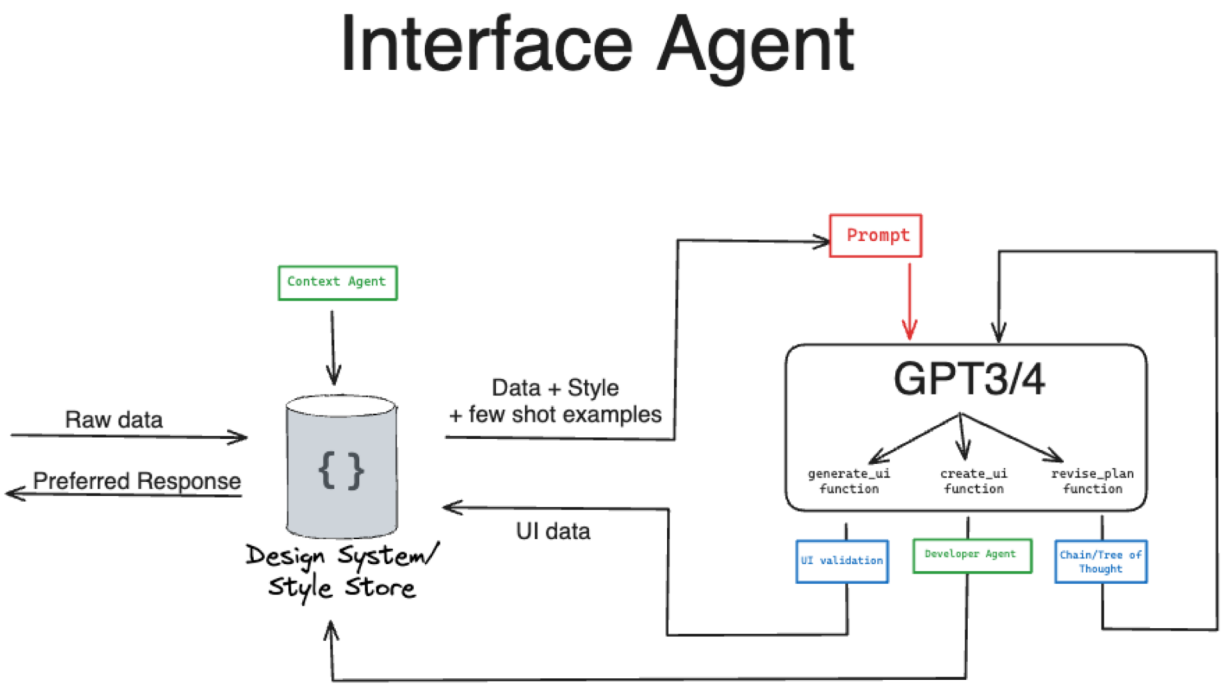
Building a Planner Agent



Building a Context Agent

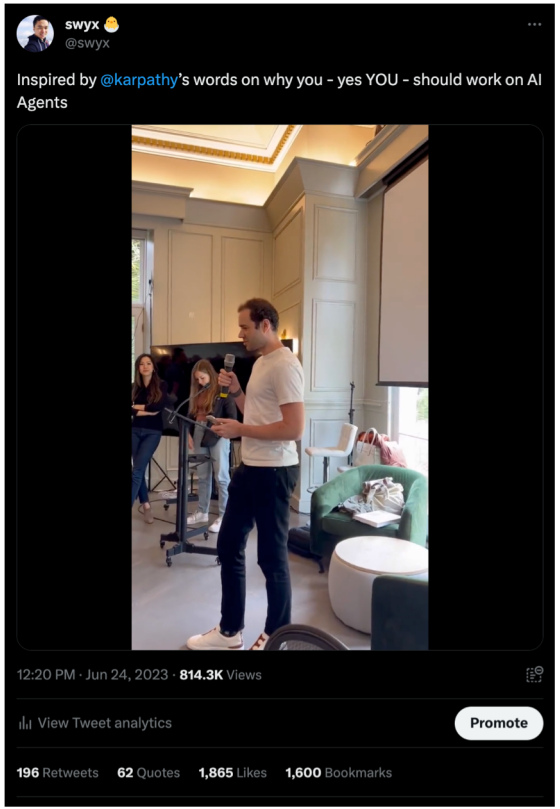


Building a Interface Agent



Code Interpreter = the most advanced Agent

Karpathy on Agents



What's interesting and not obvious is that you guys building AI agents are actually at the forefront of capability of AI agents today, and all the big labs like LLM labs like OpenAI and so on,I suspect are not at the edge of the capability.

You are at the forefront of it.

So OpenAI, for example, is very good at training massive transformer language models. One way to put it is if a paper comes out that proposes some different way of training a transformer, the internal slack at OpenAI is something along the lines of, oh yeah, someone tried that two and a half years ago, and here's what happened, and here's why it didn't work, and it's very well understood and very well mapped out.

But when a new agent paper comes out, we're all interested, and we look at it, and we're like, oh, that's really cool, that's novel. And that's because the team didn't have five years to spend on it, and it's competing now with all of you, the entrepreneurs and hackers and so on. It's really hard to do.

thanks!

latent.space
ai.engineer