回归初心 热AI终端

热·AI

第十九届D2终端技术大会

19TH D2 TERMINAL
TECHNOLOGY CONFERENCE

AI 时代生存指南

前端的技术壁垒与竞争力重塑

谢亚东

Lepton AI Founding Member

www.xieyadong.com

一、在 AI 时代构建前端技术壁垒

历史上的技术壁垒的构建和洗牌

载体变化

载体多元化催生跨平台适配壁垒

行业发展

行业需求迭代驱动前端职责持续扩张

架构演进

架构复杂化提高技术栈专业门槛

载体变化

多浏览器适配

IE / Firefox /
Chrome 兼容性

移动优先时代

响应式设计适应移动
设备

多平台融合

PWAs 和 WebView
模糊应用界限

全终端覆盖

跨端开发和 AI 工具
主导

行业发展

门户网站和内容平台

专注于静态内容和品牌展示

社交网络和电商平台

引入实时互动和移动适配

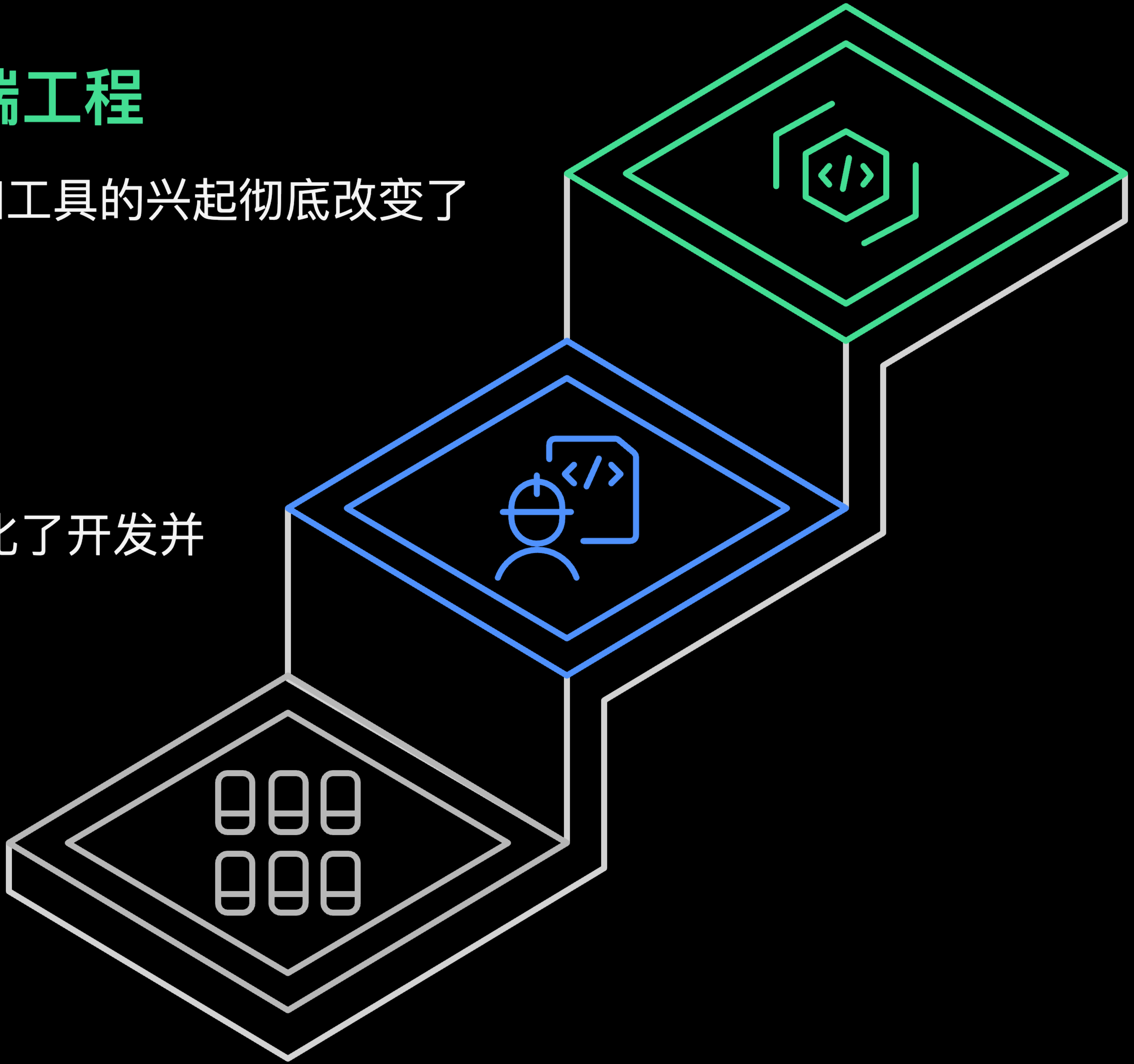
短视频和直播平台

需要音视频处理和低延迟交互

小程序和云原生

跨平台开发和 Serverless API 调用

架构演进



现代前端工程

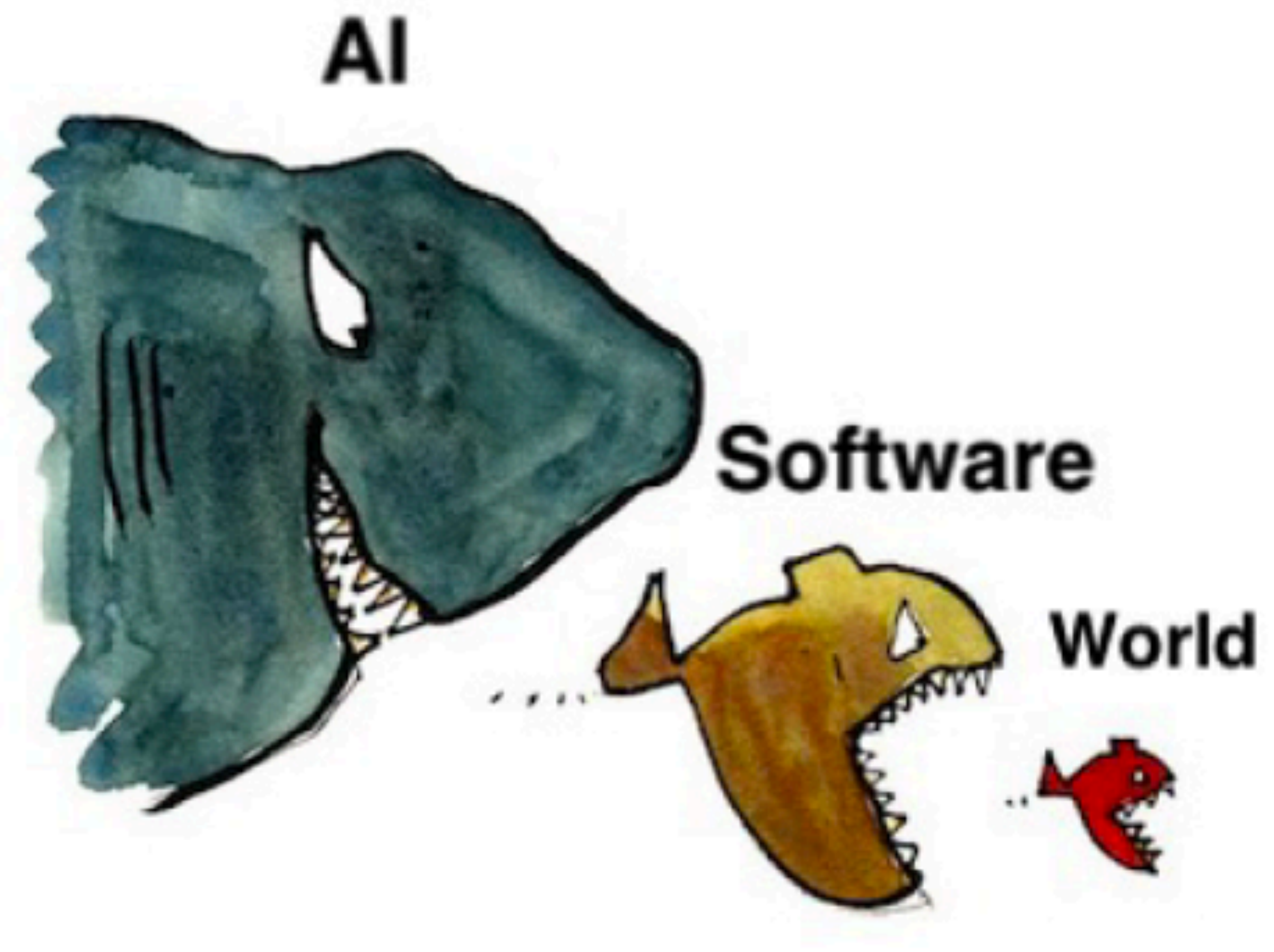
现代框架和工具的兴起彻底改变了开发实践。

前端工程化初期

引入 jQuery 和 AJAX 简化了开发并增强了互动性

基础Web技术

静态 HTML 和基础脚本标志着 Web 开发的早期阶段



“Software is eating the world”
Andreessen Horowitz, HP (2011)

“Software is eating the world, but AI is going to eat software”
Jensen Huang, Nvidia CEO (2017)

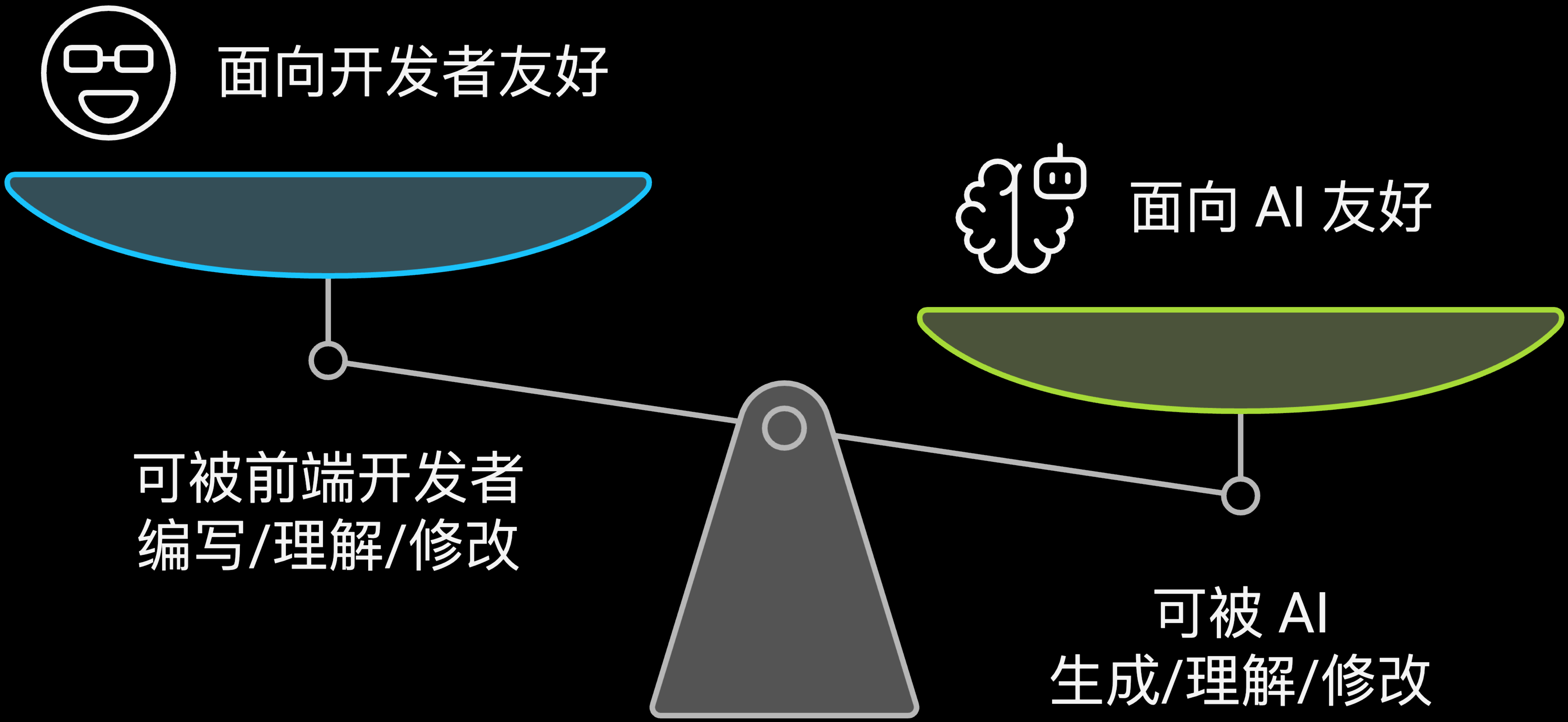
在 AI 时代构建前端技术壁垒

1. 面向 AI 做技术选型

2. 掌握 AI 数据处理与渲染方案

1. 面向 AI 做技术选型

可被前端开发者编写/理解/修改 VS 可被 AI 生成/理解/修改



面向开发者友好

可被前端开发者
编写/理解/修改

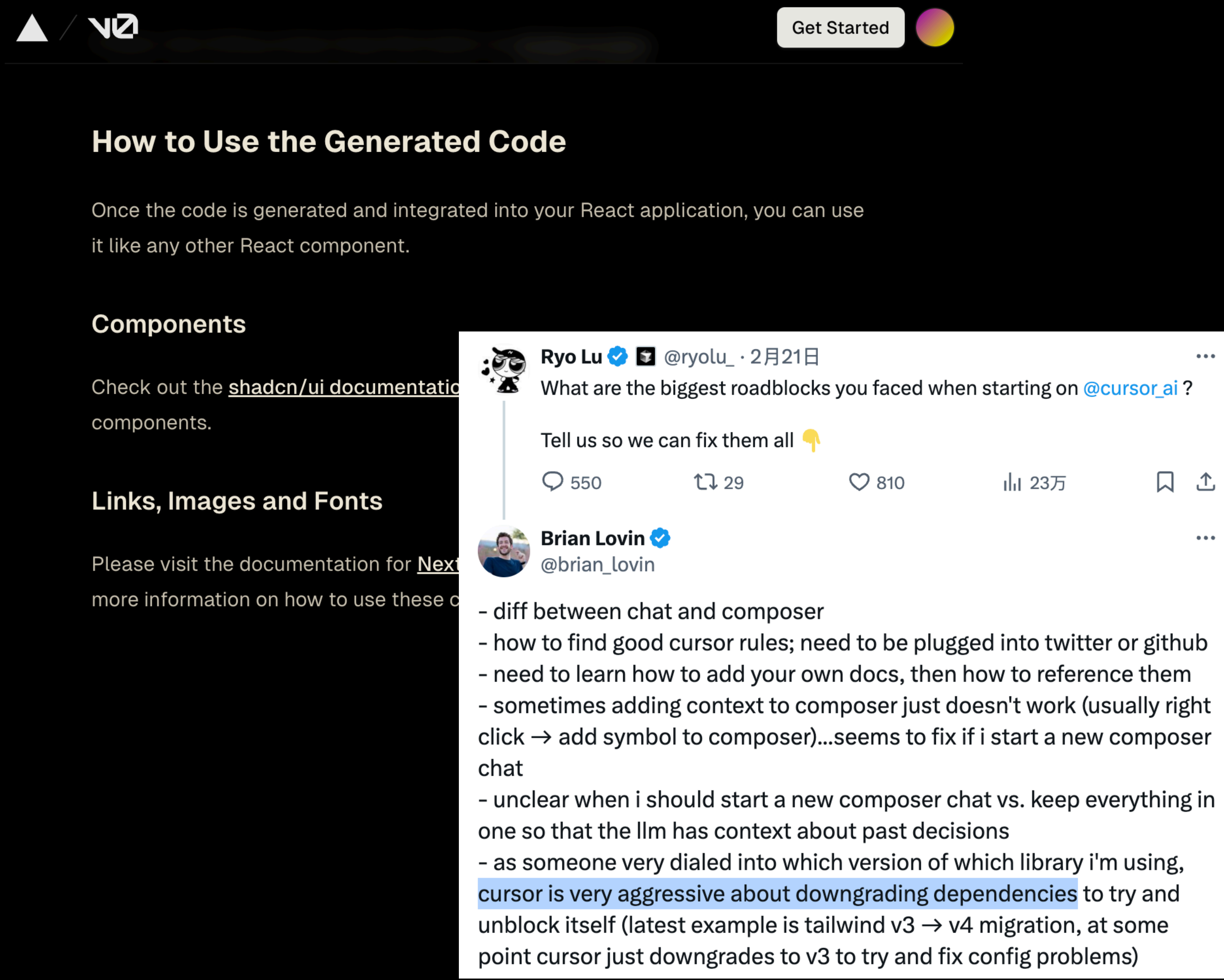
面向 AI 友好

可被 AI
生成/理解/修改

马太效应

AI加持下的主流框架继续保持优势

- 更大的用户基础
- 更多的训练语料
- 更好的 AI 生成/提示效果



How to Use the Generated Code

Once the code is generated and integrated into your React application, you can use it like any other React component.

Components

Check out the shadcn/ui documentatio
components.

Links, Images and Fonts

Please visit the documentation for Next
more information on how to use these c

Ryo Lu @ryolu_ · 2月21日

What are the biggest roadblocks you faced when starting on @cursor_ai ?

Tell us so we can fix them all 👇

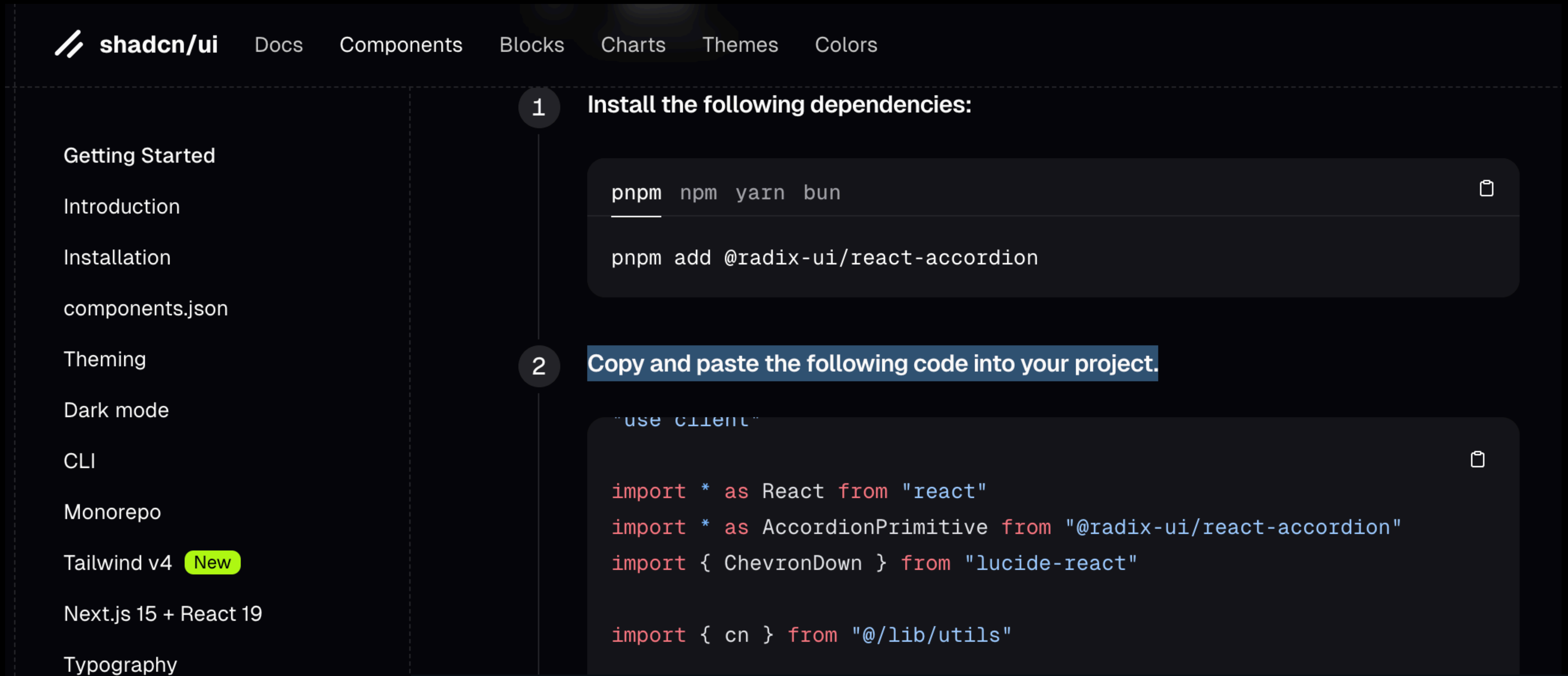
Brian Lovin @brian_lovin

- diff between chat and composer
- how to find good cursor rules; need to be plugged into twitter or github
- need to learn how to add your own docs, then how to reference them
- sometimes adding context to composer just doesn't work (usually right click → add symbol to composer)...seems to fix if i start a new composer chat
- unclear when i should start a new composer chat vs. keep everything in one so that the llm has context about past decisions
- as someone very dialed into which version of which library i'm using, cursor is very aggressive about downgrading dependencies to try and unblock itself (latest example is tailwind v3 → v4 migration, at some point cursor just downgrades to v3 to try and fix config problems)

反模式

传统的代码组织模式受到挑战

- 打包 VS 生成
- 拆分 VS 单文件
- 可维护代码 VS 可抛弃代码

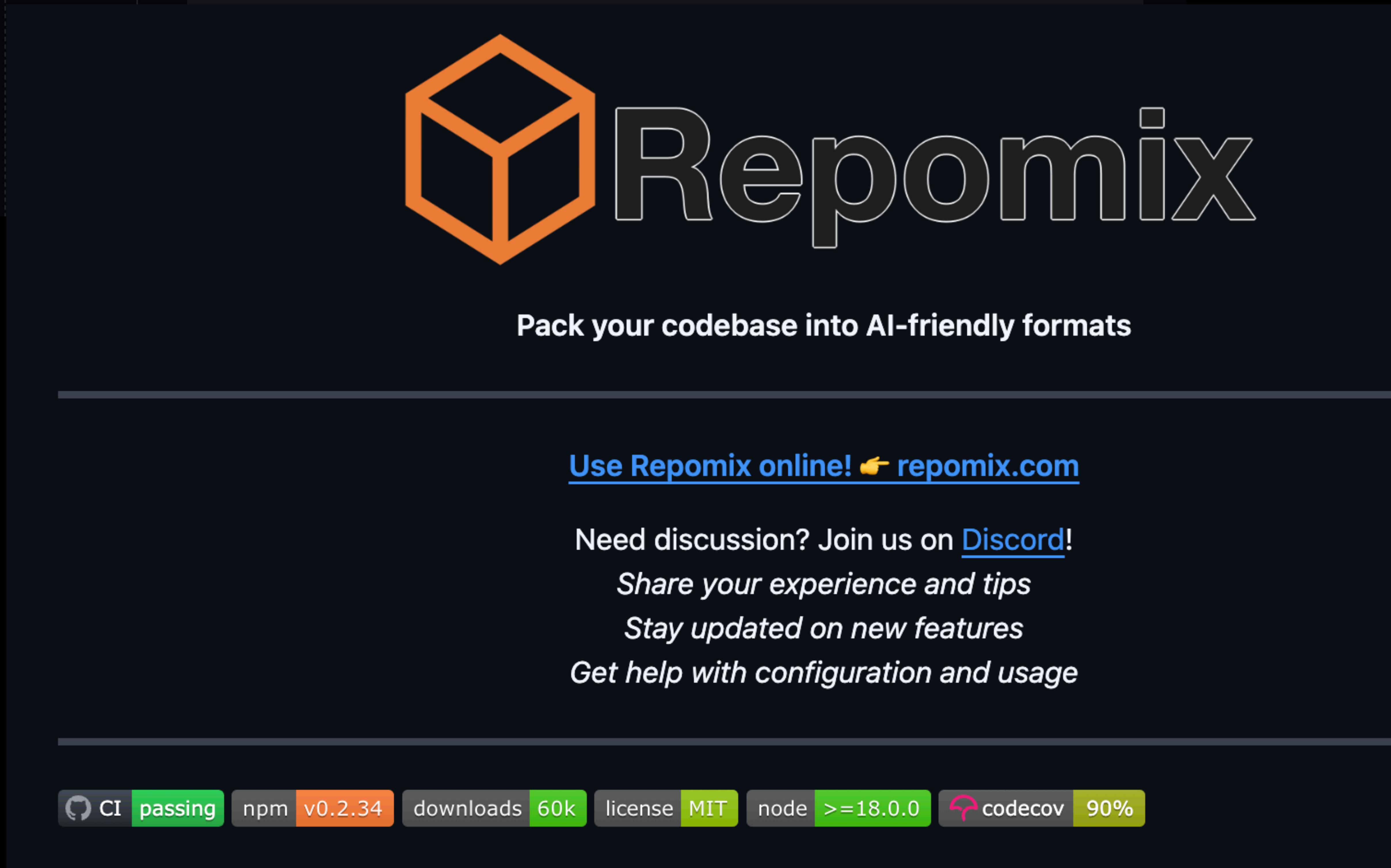


1 Install the following dependencies:

```
pnpm add @radix-ui/react-accordion
```

2 Copy and paste the following code into your project.

```
"use client"

import * as React from "react"
import * as AccordionPrimitive from "@radix-ui/react-accordion"
import { ChevronDown } from "lucide-react"

import { cn } from "@/lib/utils"
```



Pack your codebase into AI-friendly formats

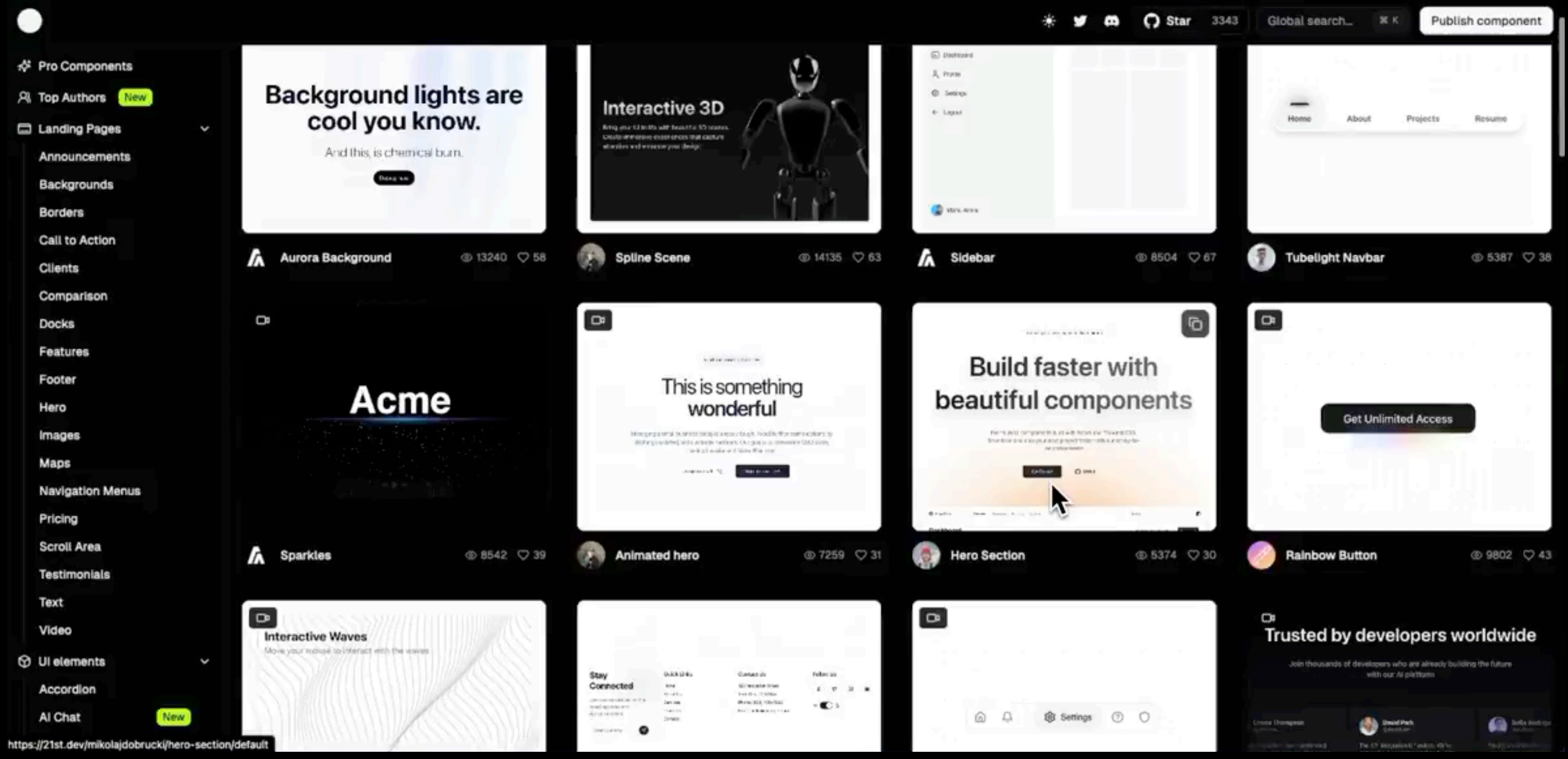
Use Repomix online! 👉 repomix.com

Need discussion? Join us on Discord!
Share your experience and tips
Stay updated on new features
Get help with configuration and usage

反模式

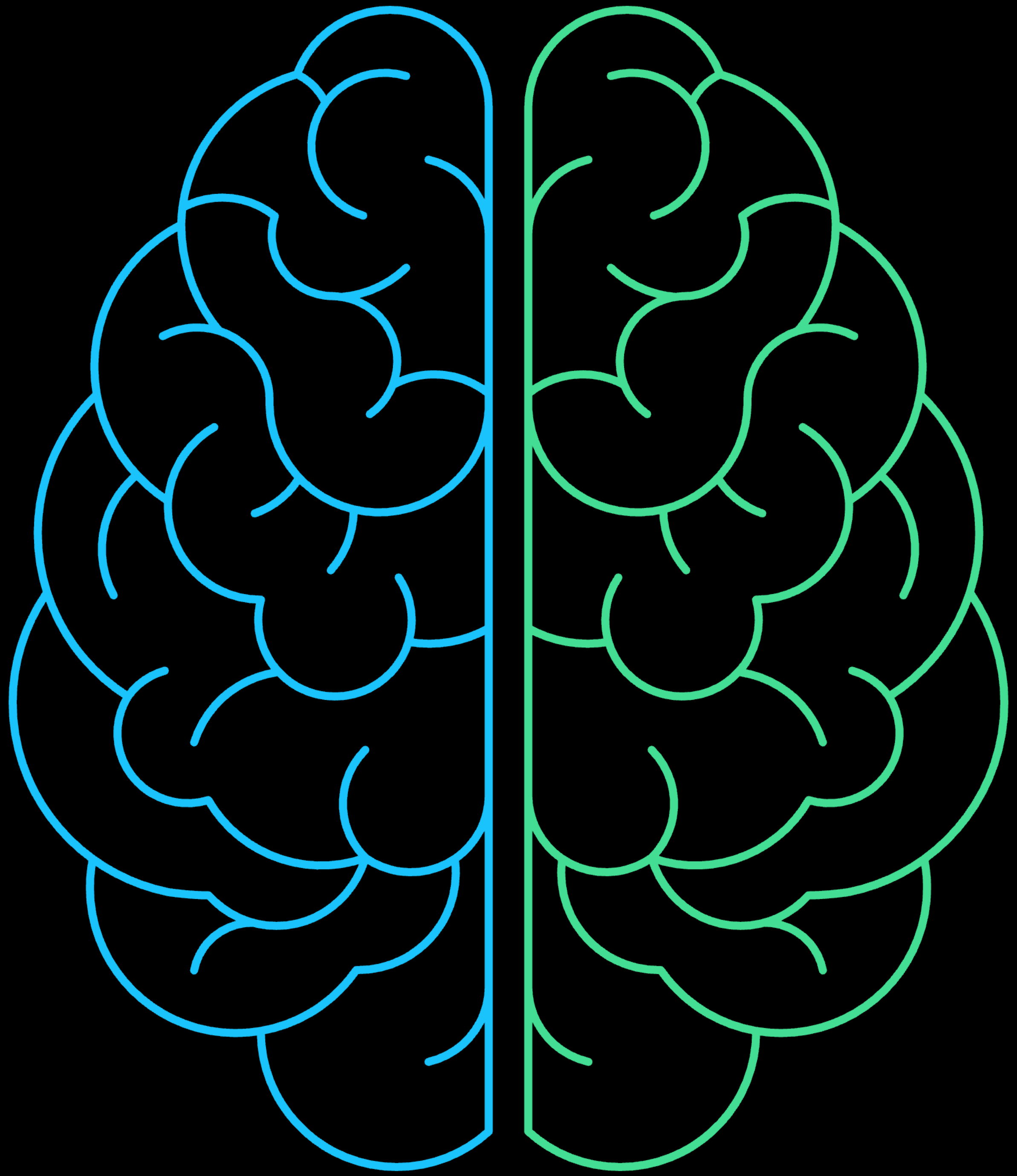
传统的代码组织模式受到挑战

- 打包 VS 生成
- 拆分 VS 单文件
- 可维护代码 VS 可抛弃代码



2. 掌握 AI 数据处理与渲染方案

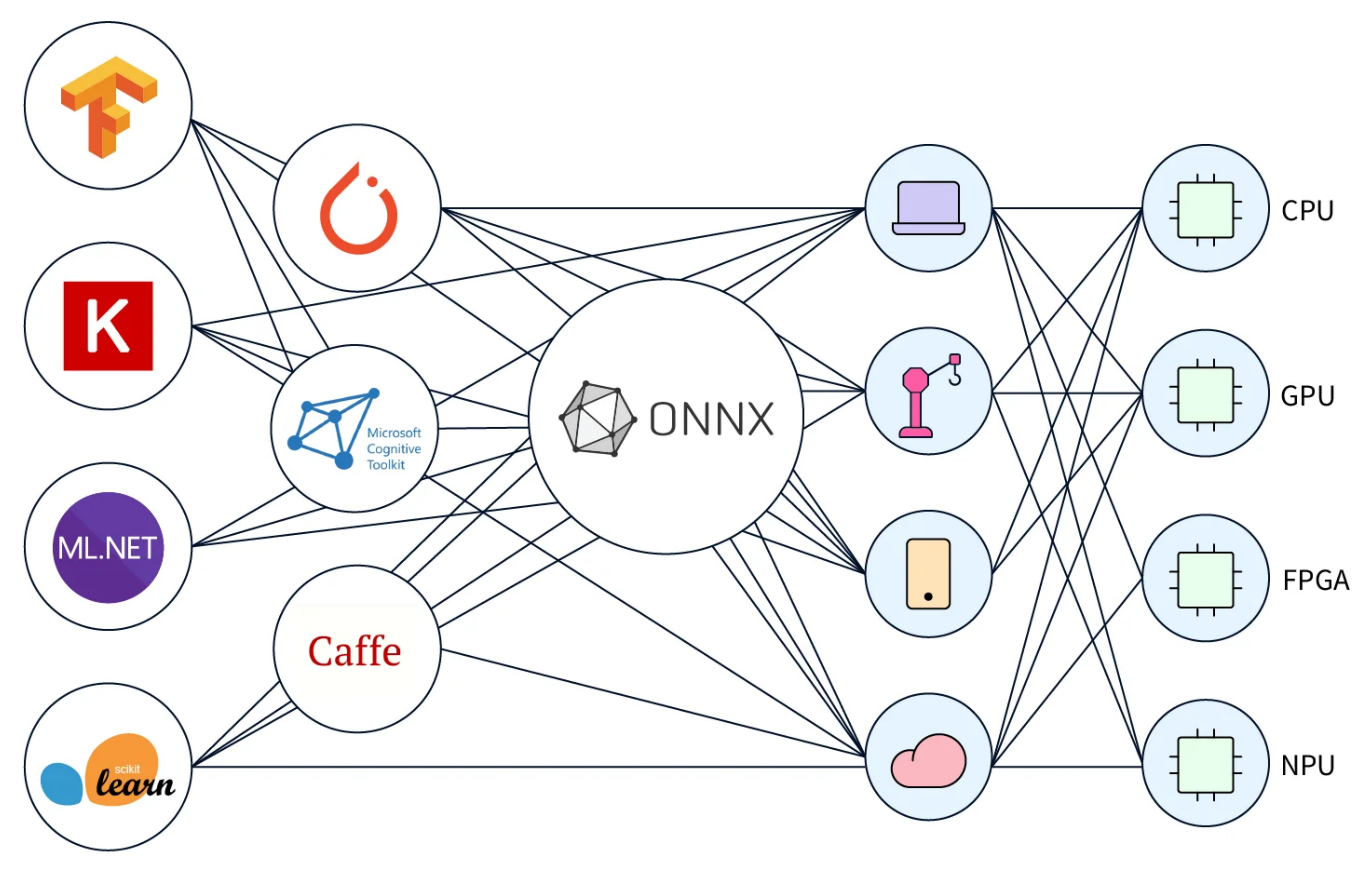
数据处理



界面渲染

2.1 新数据处理方案

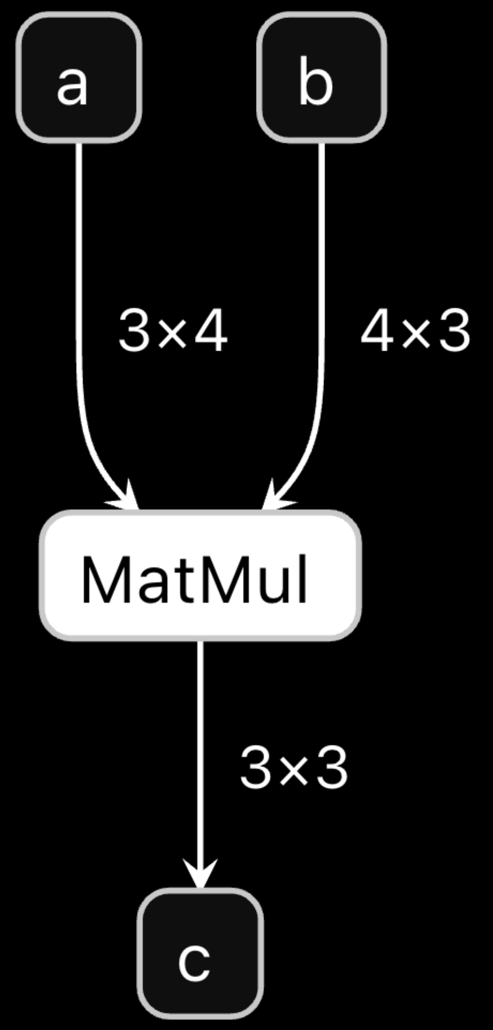
`lodash` vs `ai_web_runtime`



https://www.scaler.com/topics/deep-learning/onnx-model/

```
const ort = require('onnxruntime-web');
async function main() {
    try {
        const session = await ort.InferenceSession.create('./model.onnx');

        // prepare inputs. a tensor need its corresponding TypedArray as data
        const dataA = Float32Array.from([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]);
        const dataB = Float32Array.from([10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120]);
        const tensorA = new ort.Tensor('float32', dataA, [3, 4]);
        const tensorB = new ort.Tensor('float32', dataB, [4, 3]);

        // prepare feeds. use model input names as keys.
        const feeds = { a: tensorA, b: tensorB };

        // feed inputs and run
        const results = await session.run(feeds);

        // read from results
        const dataC = results.c.data;
        document.write(`data of result tensor 'c': ${dataC}`);

    } catch (e) {
        document.write(`failed to inference ONNX model: ${e}.`);
    }
}

main();
```

ONNX RUNTIME

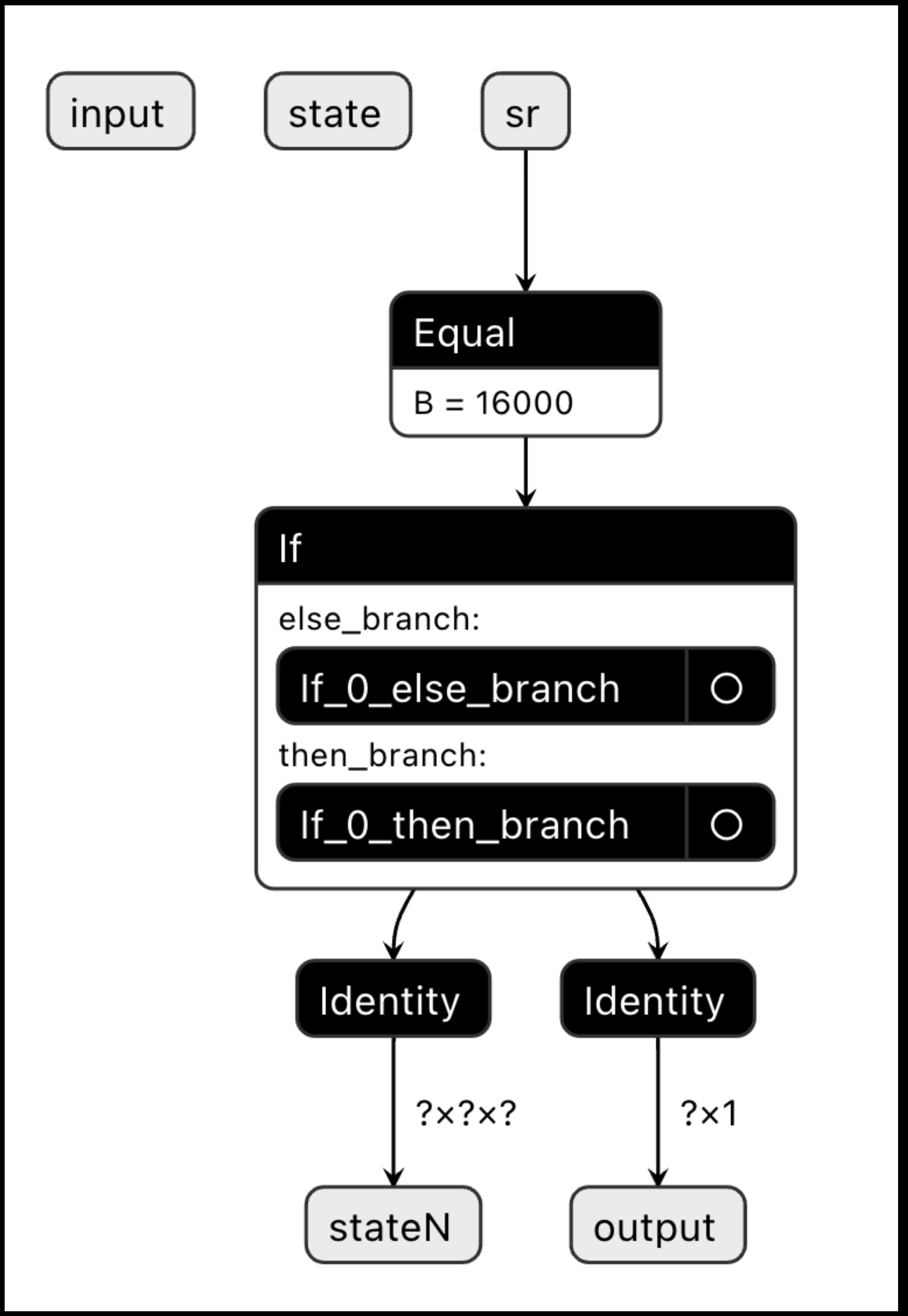


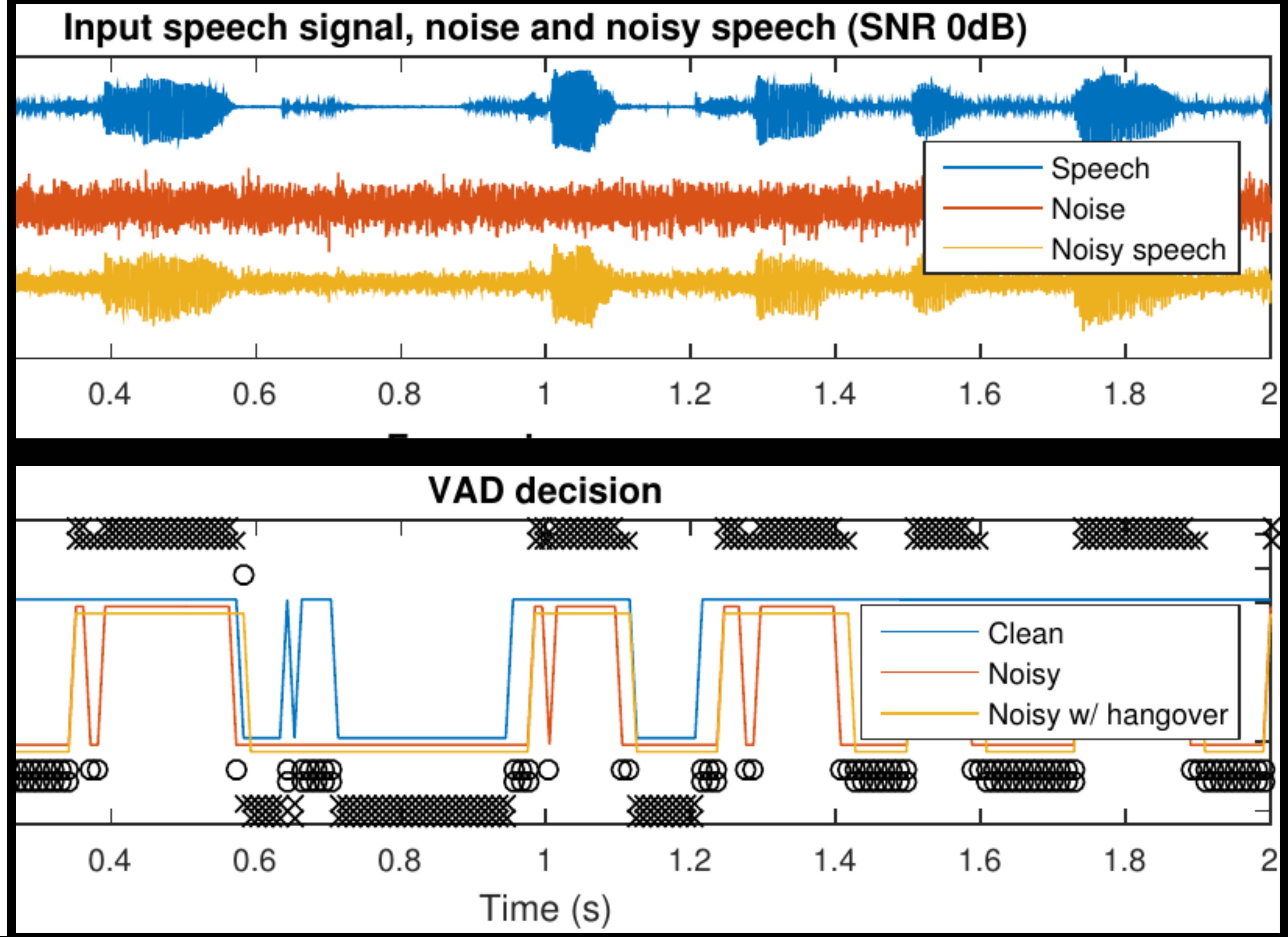
format	ONNX v3
producer	backend-test
version	0
imports	ai.onnx v9
graph	test_matmul_2d

INPUTS

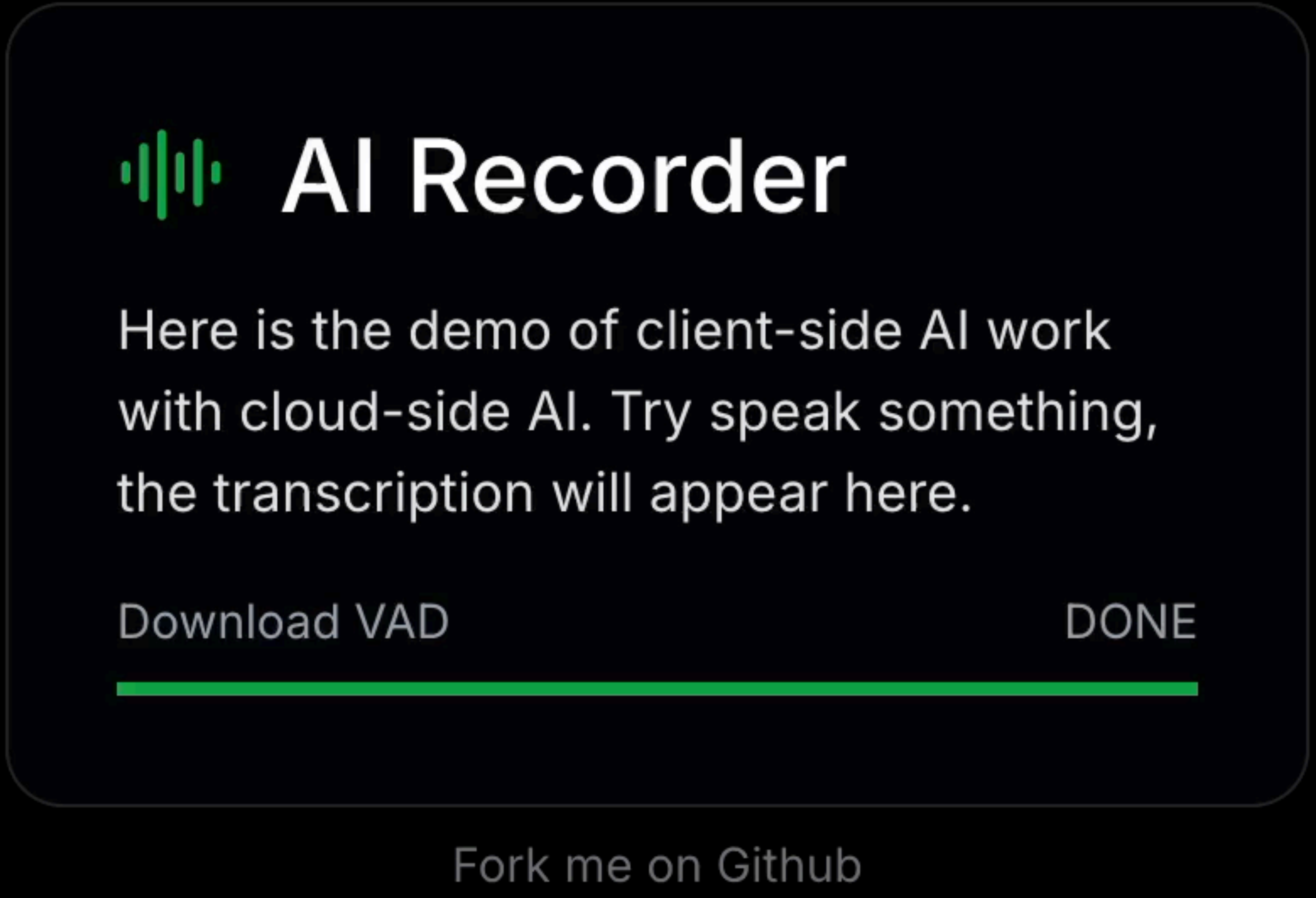
a	name: **a** tensor: **float32[3,4]**
b	name: **b** tensor: **float32[4,3]**

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{bmatrix} \times \begin{bmatrix} 10 & 20 & 30 \\ 40 & 50 & 60 \\ 70 & 80 & 90 \\ 100 & 110 & 120 \end{bmatrix}$$





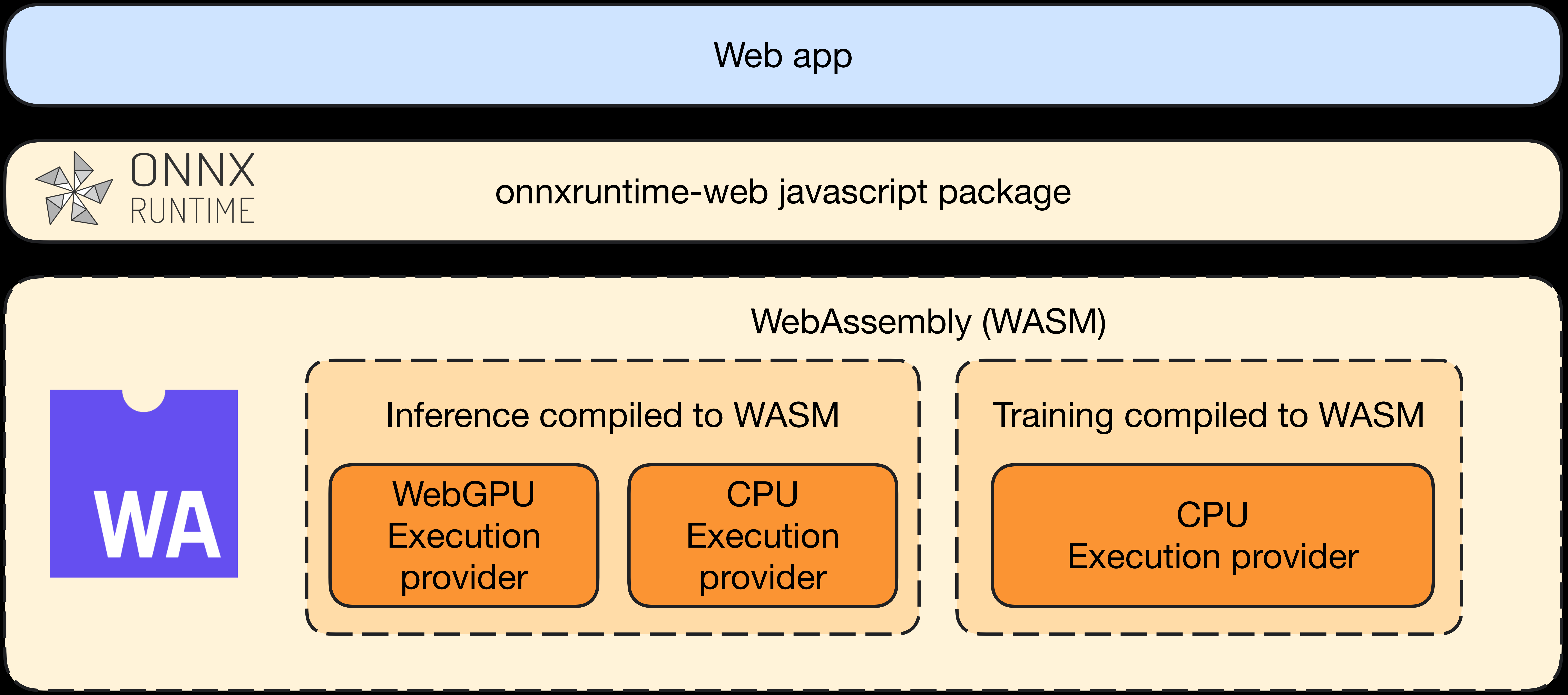
https://github.com/snakers4/silero-vad



AI Recorder

Here is the demo of client-side AI work with cloud-side AI. Try speak something, the transcription will appear here.

https://github.com/vthinkxie/ai-recorder



Web app

onnxruntime-web javascript package

WebAssembly (WASM)

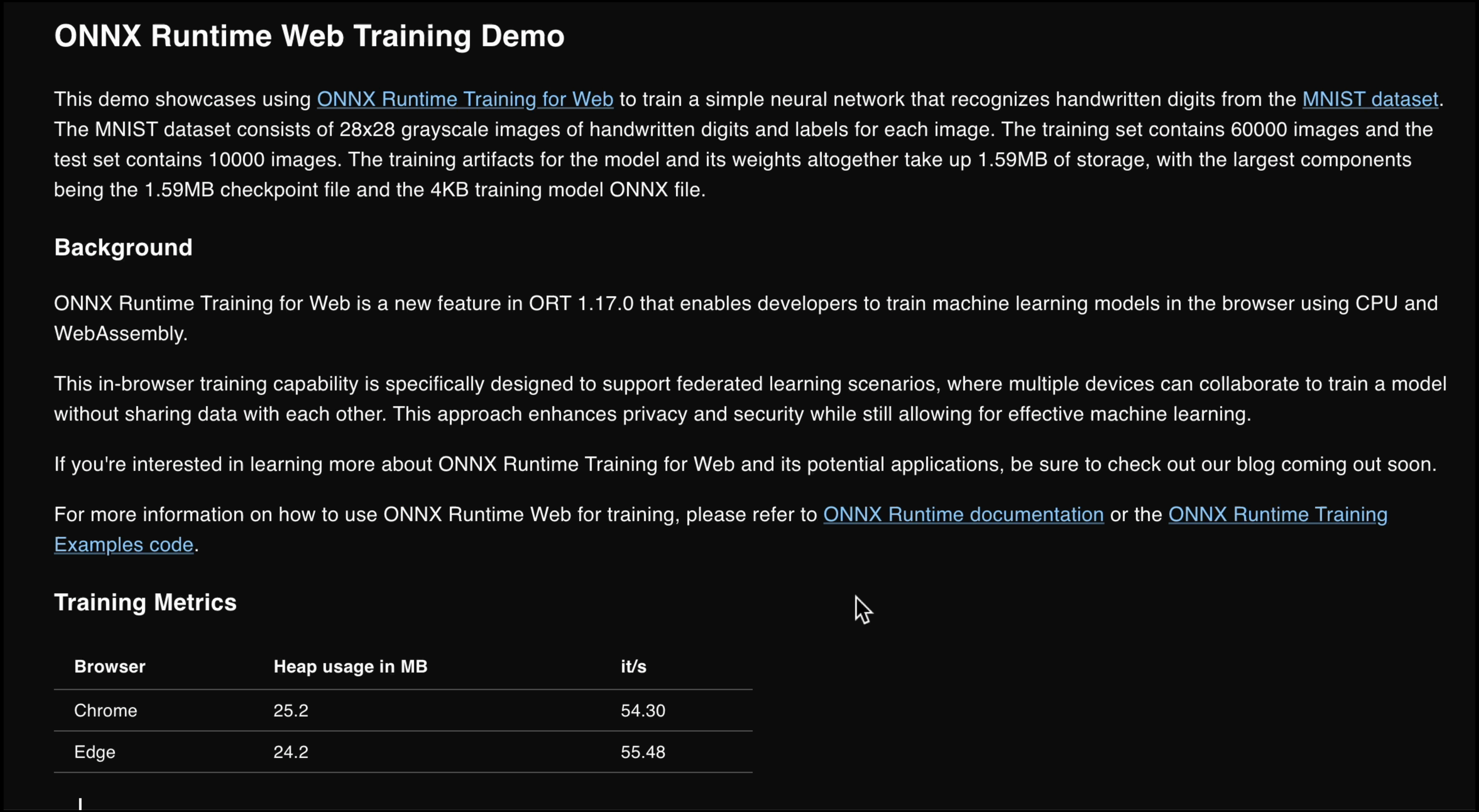
Inference compiled to WASM

WebGPU Execution provider

CPU Execution provider

Training compiled to WASM

CPU Execution provider



ONNX Runtime Web Training Demo

This demo showcases using ONNX Runtime Training for Web to train a simple neural network that recognizes handwritten digits from the MNIST dataset. The MNIST dataset consists of 28x28 grayscale images of handwritten digits and labels for each image. The training set contains 60000 images and the test set contains 10000 images. The training artifacts for the model and its weights altogether take up 1.59MB of storage, with the largest components being the 1.59MB checkpoint file and the 4KB training model ONNX file.

Background

ONNX Runtime Training for Web is a new feature in ORT 1.17.0 that enables developers to train machine learning models in the browser using CPU and WebAssembly.

This in-browser training capability is specifically designed to support federated learning scenarios, where multiple devices can collaborate to train a model without sharing data with each other. This approach enhances privacy and security while still allowing for effective machine learning.

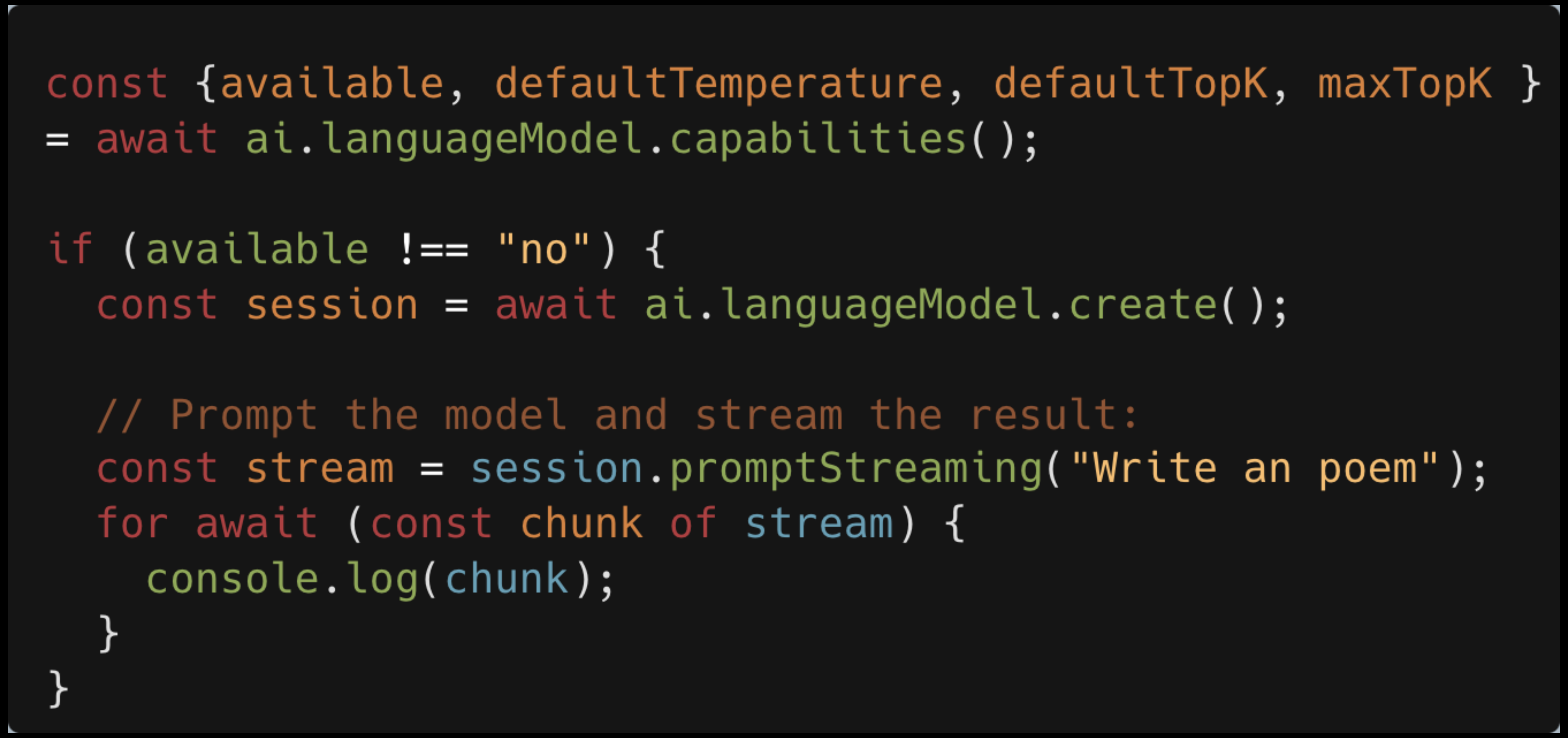
If you're interested in learning more about ONNX Runtime Training for Web and its potential applications, be sure to check out our blog coming out soon.

For more information on how to use ONNX Runtime Web for training, please refer to ONNX Runtime documentation or the ONNX Runtime Training Examples code.

Training Metrics

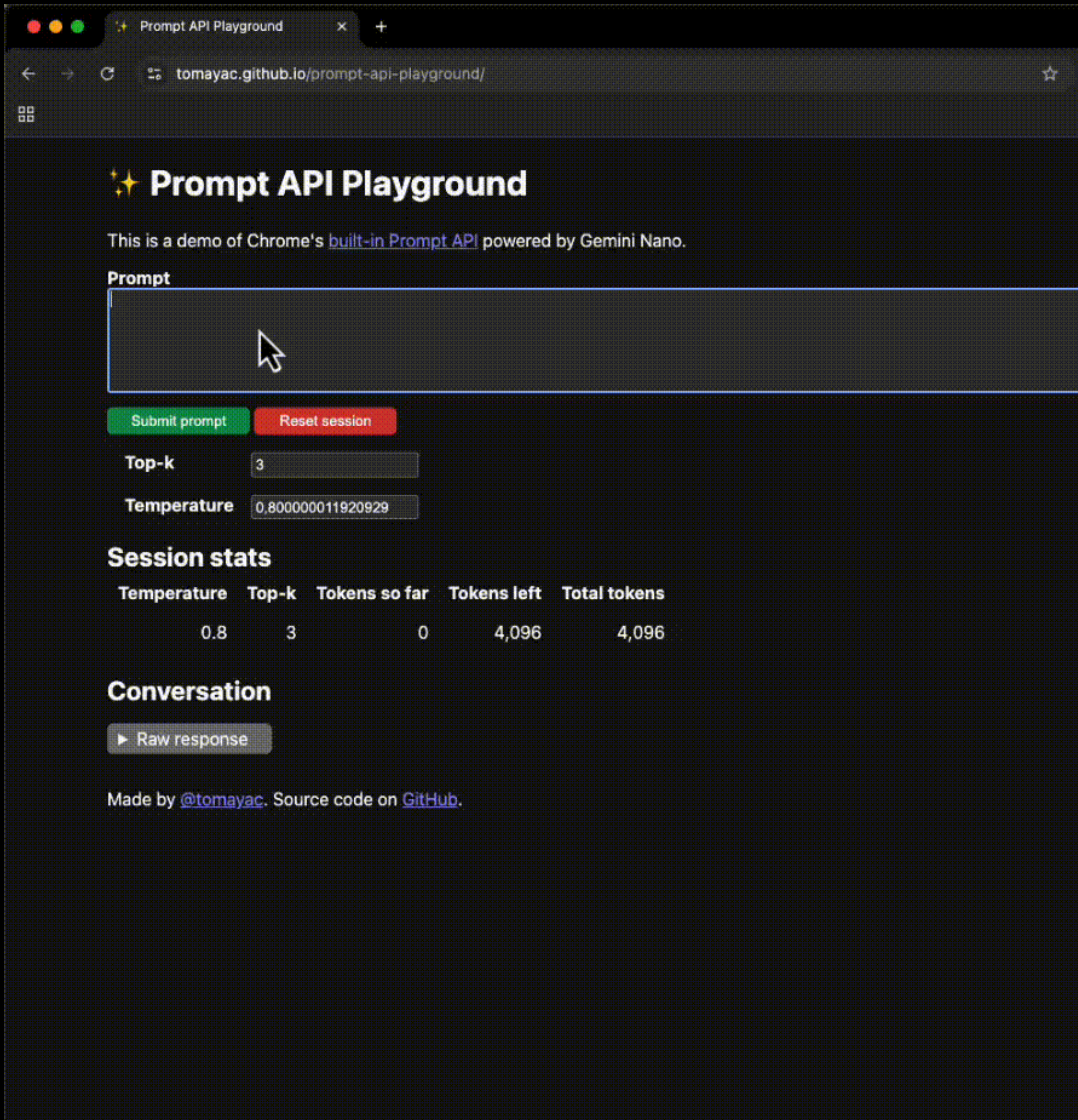
Browser	Heap usage in MB	it/s
Chrome	25.2	54.30
Edge	24.2	55.48

Chrome Built-in AI



```
const {available, defaultTemperature, defaultTopK, maxTopK }
= await ai.languageModel.capabilities();

if (available !== "no") {
  const session = await ai.languageModel.create();

  // Prompt the model and stream the result:
  const stream = session.promptStreaming("Write an poem");
  for await (const chunk of stream) {
    console.log(chunk);
  }
}
```

https://docs.google.com/document/d/1VG8HIyz361zGduWgNG7R_R8Xkv0OOJ8b5C9QKeCjU0c



✨ Prompt API Playground

This is a demo of Chrome's built-in Prompt API powered by Gemini Nano.

Prompt

Top-k: 3

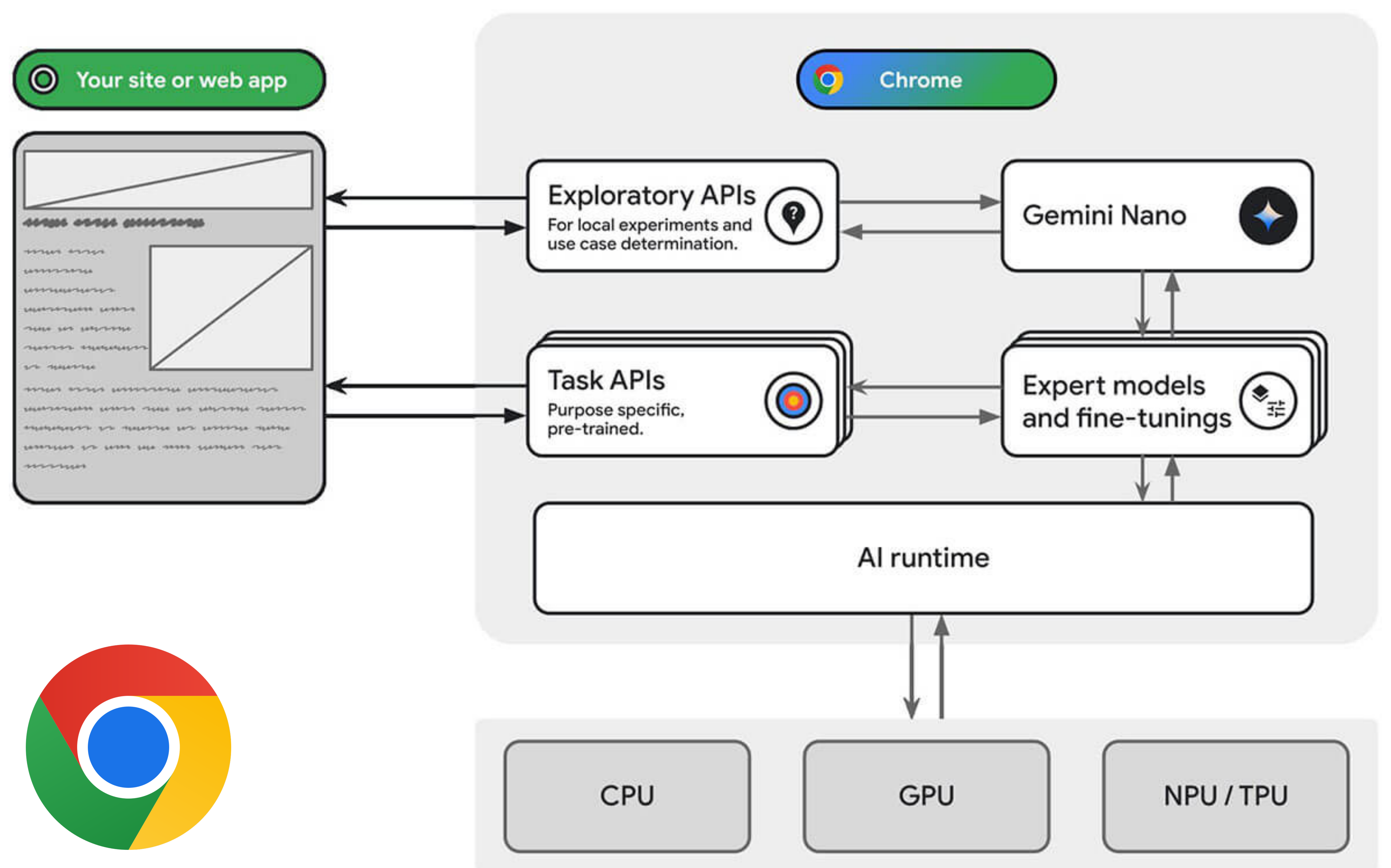
Temperature: 0.800000011920929

Session stats

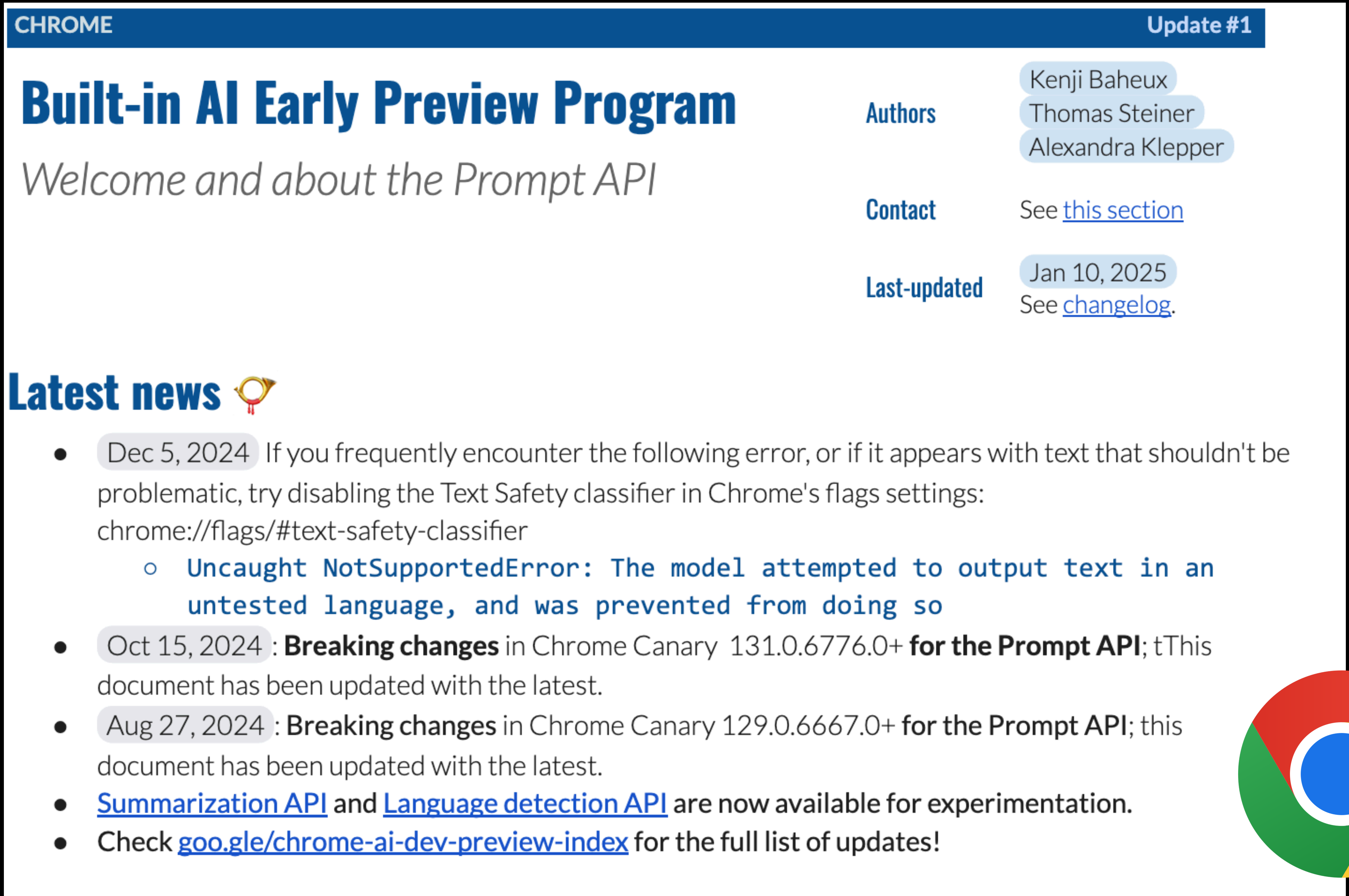
Temperature	Top-k	Tokens so far	Tokens left	Total tokens
0.8	3	0	4,096	4,096

Conversation

Made by @tomayac. Source code on GitHub.



Your site or web app
Chrome
Exploratory APIs
For local experiments and use case determination.
Gemini Nano
Task APIs
Purpose specific, pre-trained.
Expert models and fine-tunings
AI runtime
CPU
GPU
NPU / TPU



CHROME Update #1

Built-in AI Early Preview Program

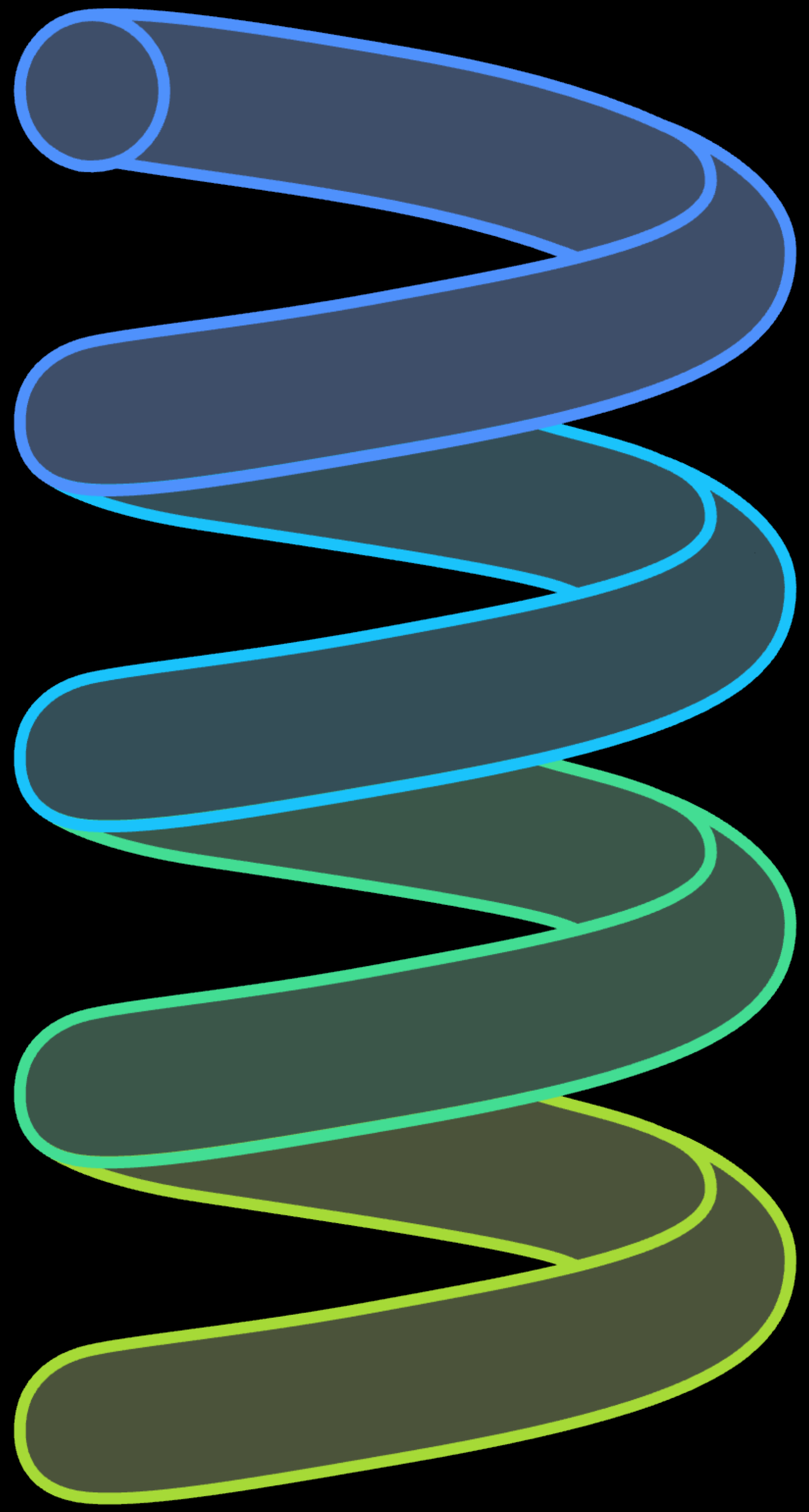
Welcome and about the Prompt API

Authors	Kenji Baheux Thomas Steiner Alexandra Klepper
Contact	See this section
Last-updated	Jan 10, 2025 See changelog.

Latest news 📯

- Dec 5, 2024 If you frequently encounter the following error, or if it appears with text that shouldn't be problematic, try disabling the Text Safety classifier in Chrome's flags settings: chrome://flags/#text-safety-classifier
    - `Uncaught NotSupportedError: The model attempted to output text in an untested language, and was prevented from doing so`
- Oct 15, 2024 : **Breaking changes** in Chrome Canary 131.0.6776.0+ **for the Prompt API**; tThis document has been updated with the latest.
- Aug 27, 2024 : **Breaking changes** in Chrome Canary 129.0.6667.0+ **for the Prompt API**; this document has been updated with the latest.
- Summarization API and Language detection API are now available for experimentation.
- Check goo.gle/chrome-ai-dev-preview-index for the full list of updates!

2.2. 新的渲染方案



传统模式

前端工程师直接编写代码,控制所有渲染细节

AI 辅助编程

前端工程师编写提示词，LLM 生成代码，工程师审核后应用

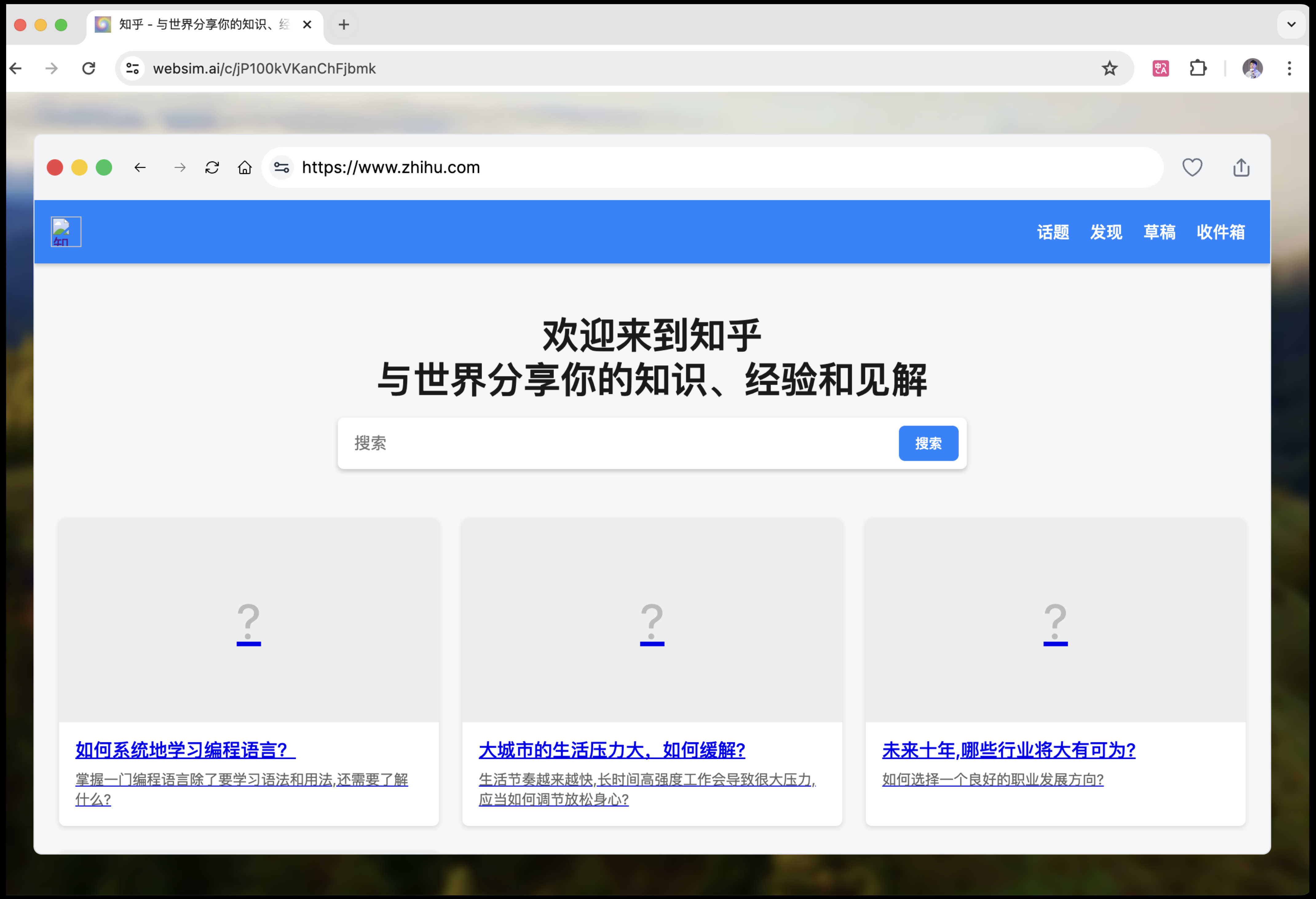
AI 实时生成

LLM 实时生成并执行代码，能够根据用户需求动态生成前端界面

AI 接管渲染

跳过生成代码阶段，生成内容作为界面的一部分直接渲染

AI 实时生成

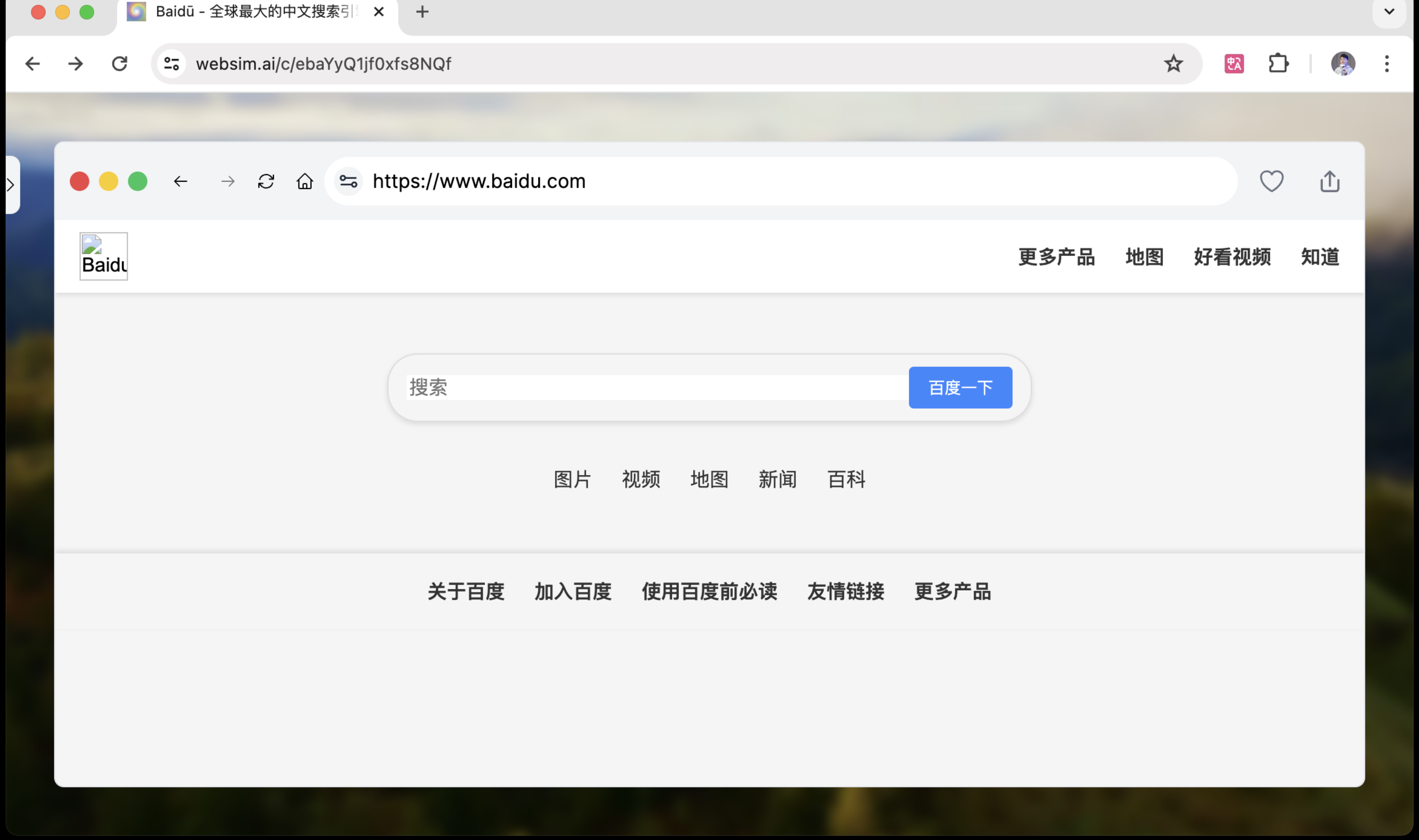


欢迎来到知乎
与世界分享你的知识、经验和见解

如何系统地学习编程语言?
掌握一门编程语言除了要学习语法和用法,还需要了解什么?

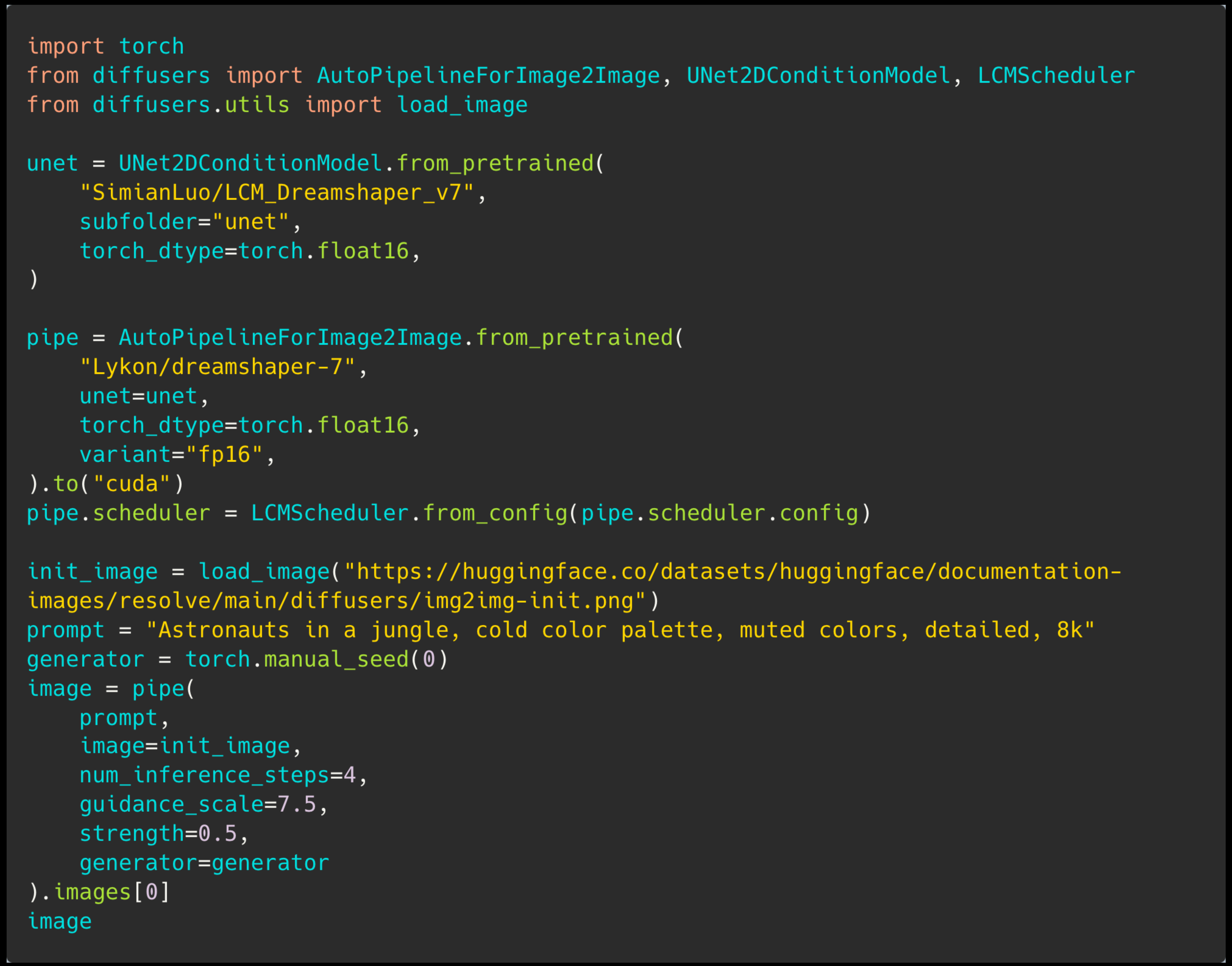
大城市的生活压力大，如何缓解?
生活节奏越来越快,长时间高强度工作会导致很大压力,应当如何调节放松身心?

未来十年,哪些行业将大有可为?
如何选择一个良好的职业发展方向?

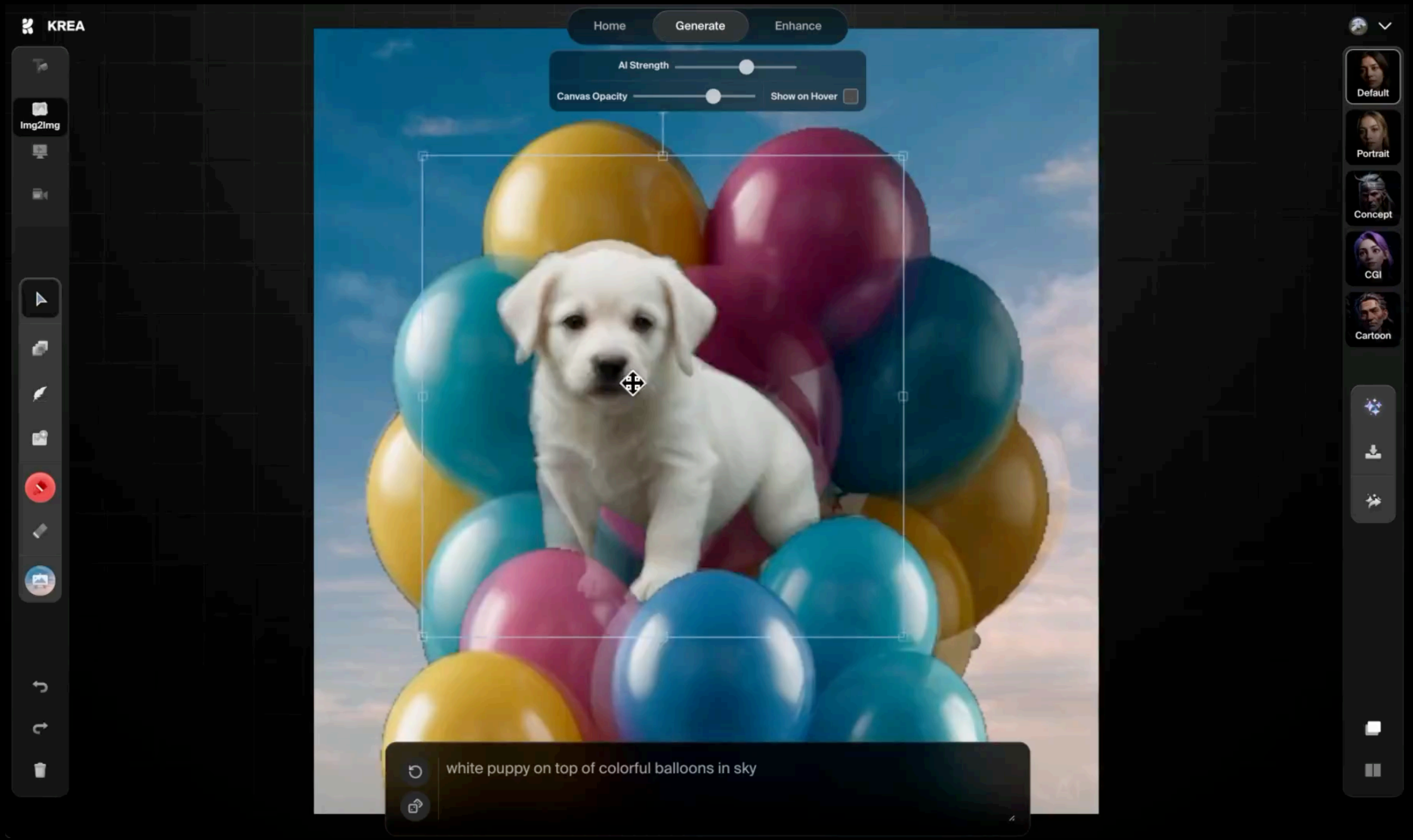


https://websim.ai/

AI 接管渲染



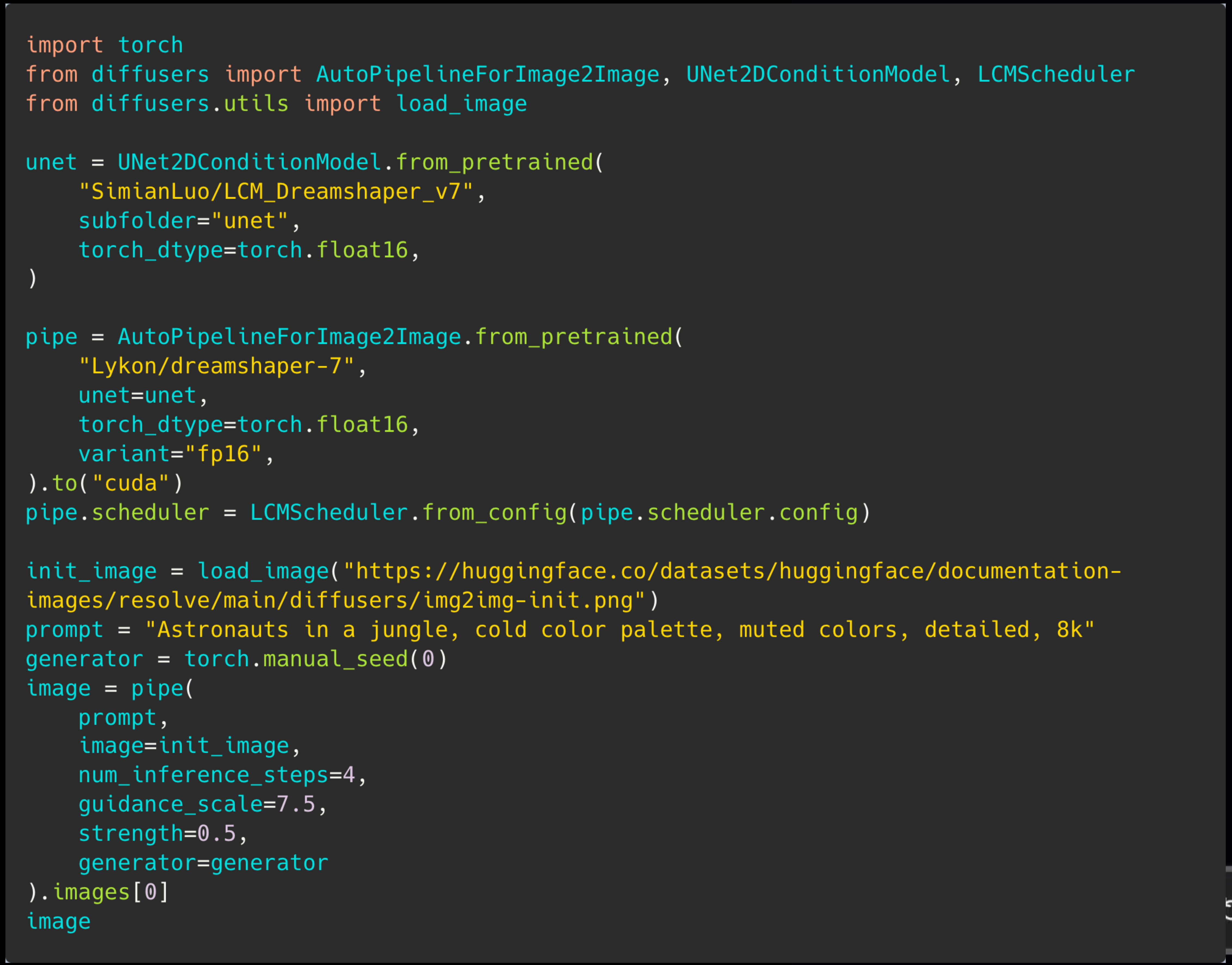
```
import torch
from diffusers import AutoPipelineForImage2Image, UNet2DConditionModel, LCMScheduler
from diffusers.utils import load_image

unet = UNet2DConditionModel.from_pretrained(
    "SimianLuo/LCM_Dreamshaper_v7",
    subfolder="unet",
    torch_dtype=torch.float16,
)

pipe = AutoPipelineForImage2Image.from_pretrained(
    "Lykon/dreamshaper-7",
    unet=unet,
    torch_dtype=torch.float16,
    variant="fp16",
).to("cuda")
pipe.scheduler = LCMScheduler.from_config(pipe.scheduler.config)

init_image = load_image("https://huggingface.co/datasets/huggingface/documentation-
images/resolve/main/diffusers/img2img-init.png")
prompt = "Astronauts in a jungle, cold color palette, muted colors, detailed, 8k"
generator = torch.manual_seed(0)
image = pipe(
    prompt,
    image=init_image,
    num_inference_steps=4,
    guidance_scale=7.5,
    strength=0.5,
    generator=generator
).images[0]
image
```



white puppy on top of colorful balloons in sky

https://www.krea.ai/

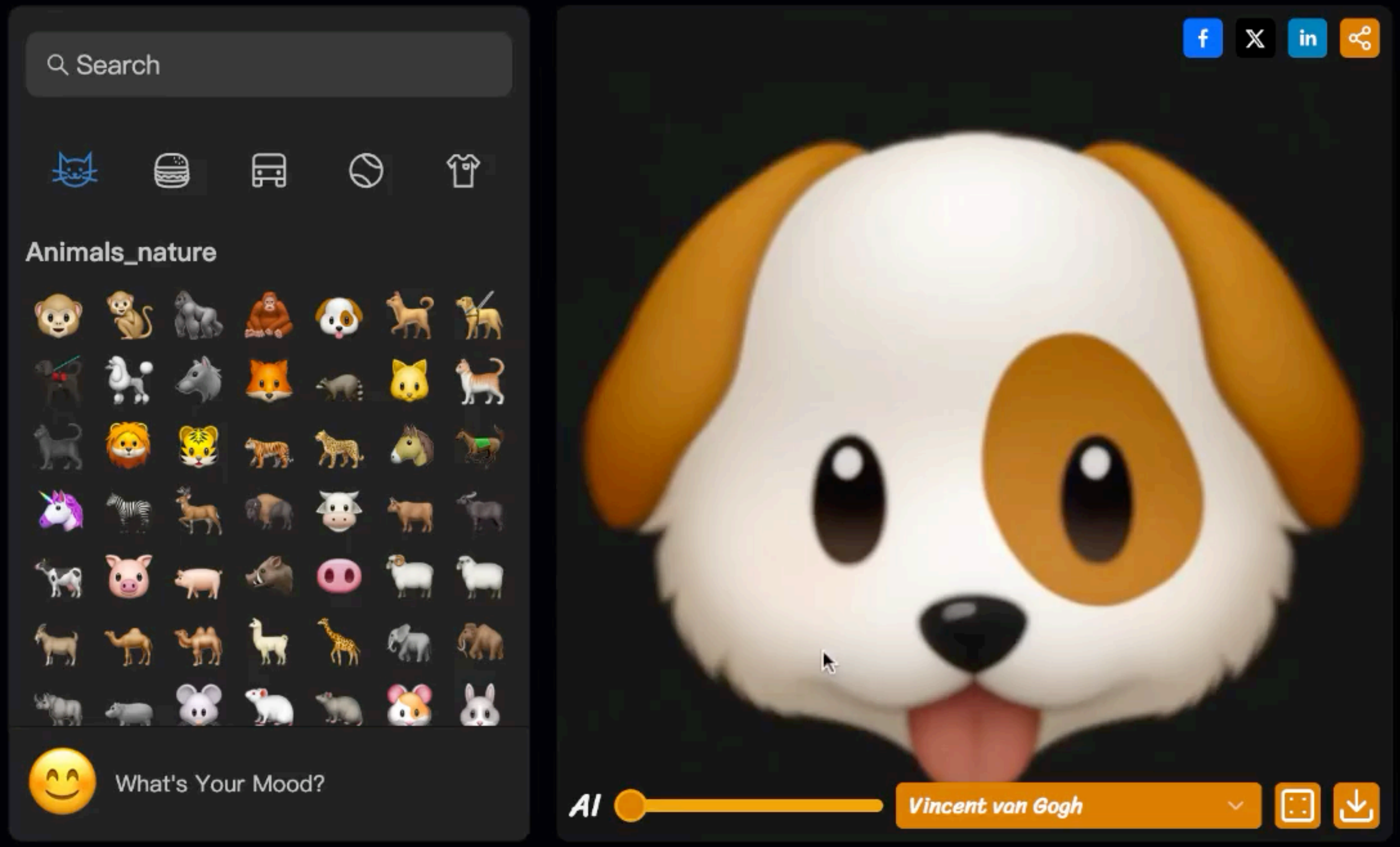
AI 接管渲染



```
import torch
from diffusers import AutoPipelineForImage2Image, UNet2DConditionModel, LCMScheduler
from diffusers.utils import load_image

unet = UNet2DConditionModel.from_pretrained(
    "SimianLuo/LCM_Dreamshaper_v7",
    subfolder="unet",
    torch_dtype=torch.float16,
)

pipe = AutoPipelineForImage2Image.from_pretrained(
    "Lykon/dreamshaper-7",
    unet=unet,
    torch_dtype=torch.float16,
    variant="fp16",
).to("cuda")
pipe.scheduler = LCMScheduler.from_config(pipe.scheduler.config)

init_image = load_image("https://huggingface.co/datasets/huggingface/documentation-
images/resolve/main/diffusers/img2img-init.png")
prompt = "Astronauts in a jungle, cold color palette, muted colors, detailed, 8k"
generator = torch.manual_seed(0)
image = pipe(
    prompt,
    image=init_image,
    num_inference_steps=4,
    guidance_scale=7.5,
    strength=0.5,
    generator=generator
).images[0]
image
```

tryEmoji

Turn emoji into amazing artwork via AI



https://github.com/leptonai/tryemoji

在 AI 时代构建前端技术壁垒

面向 AI 做技术选型

- 使用 AI 友好的技术栈
- 适应新的项目组织方式

掌握 AI 数据处理与渲染方案

- AI 模型是前端技术栈的一部分
- 关注端侧模型的发展和落地

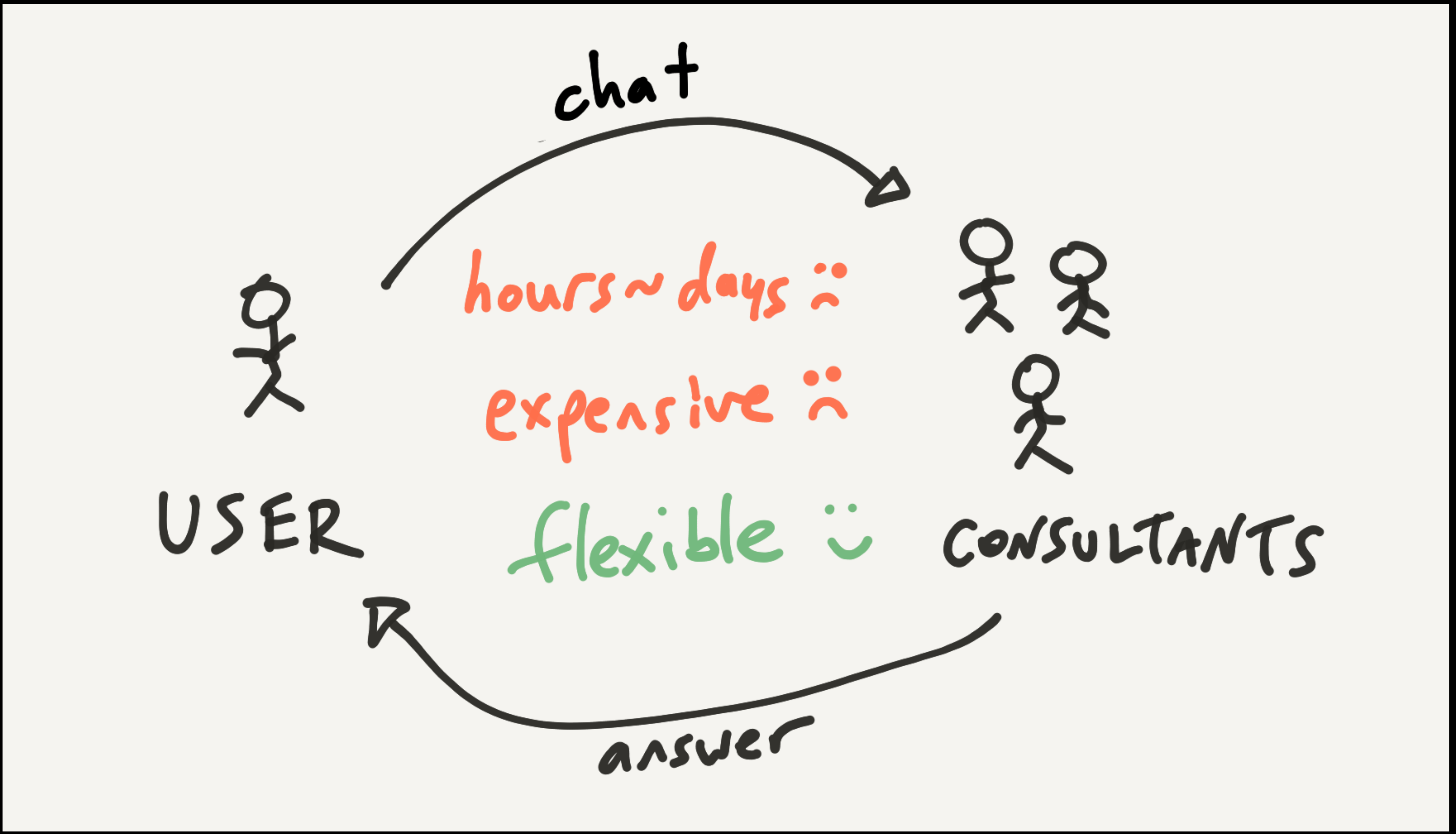
二、在 AI 时代拥有更好的竞争力

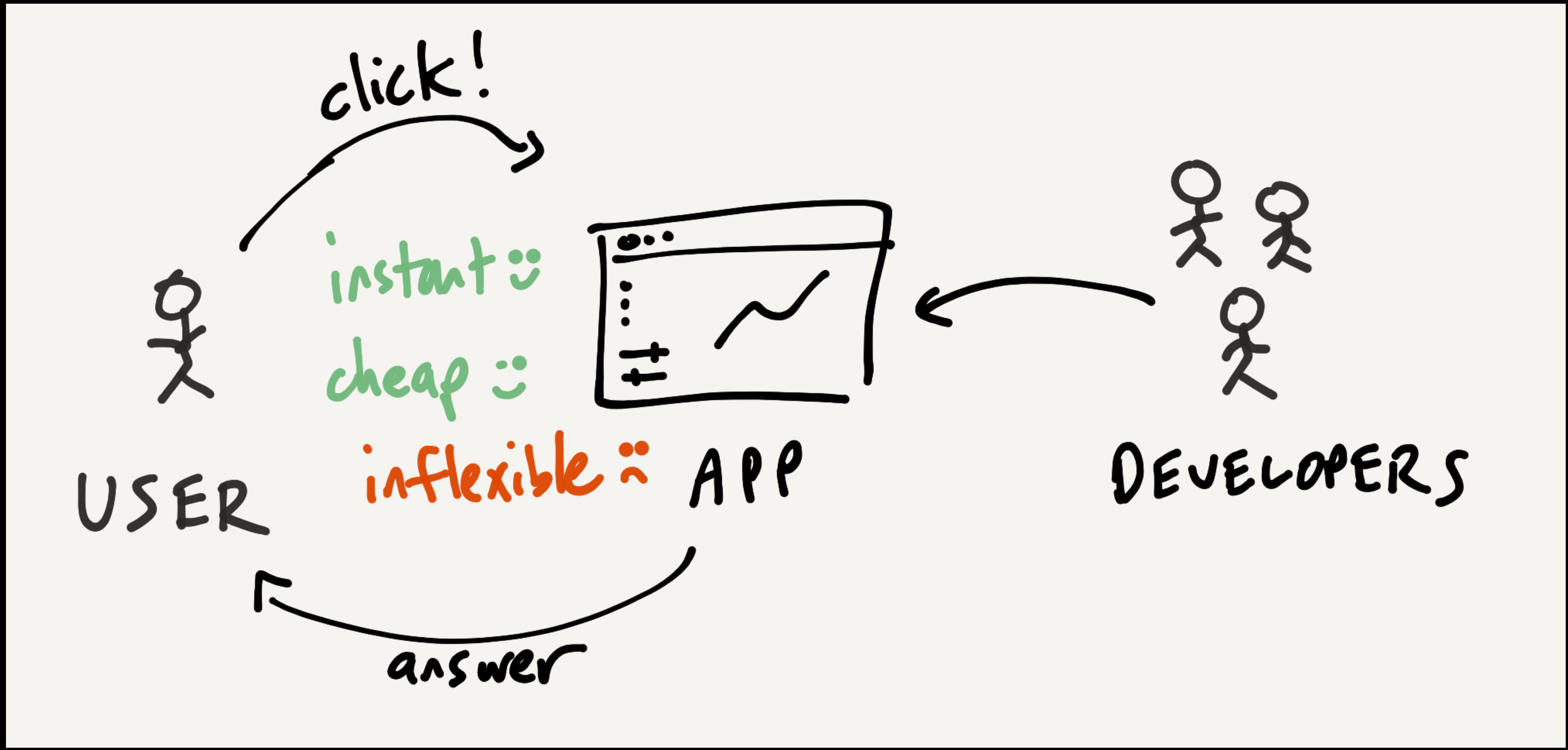
在 AI 时代拥有更好的竞争力

1. 跟随软件发展

2. 跟随生产力进步

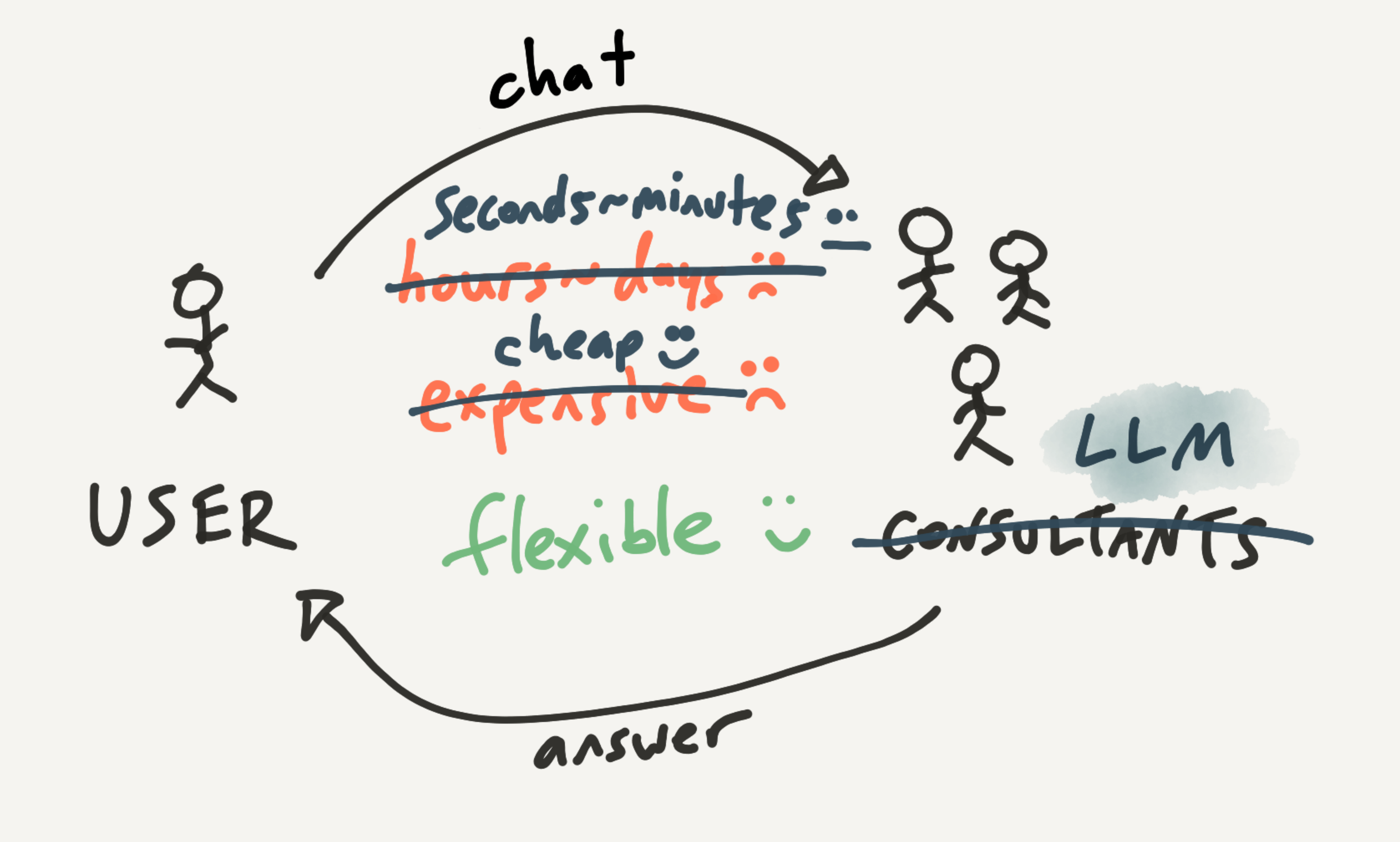
1. 跟随软件发展

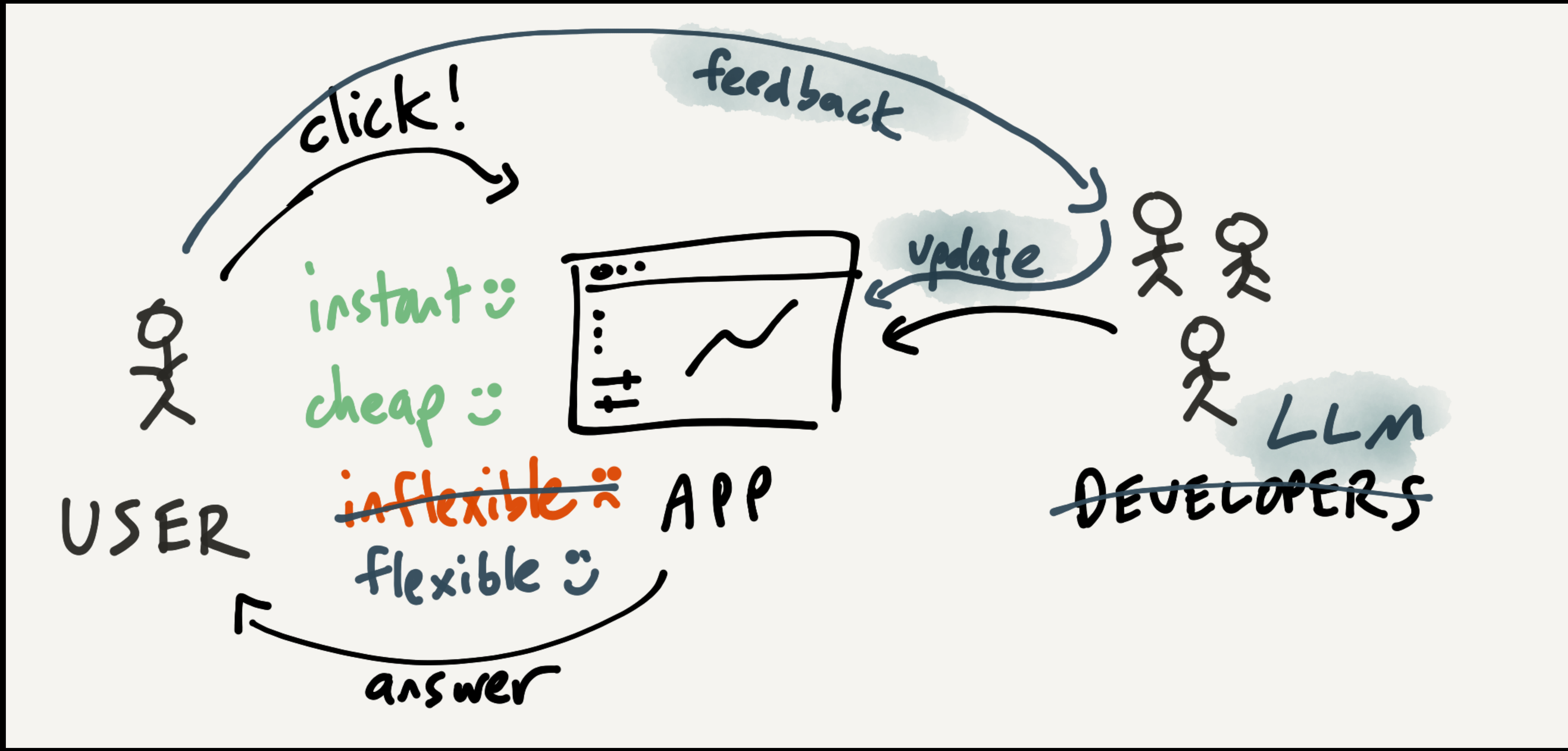




https://www.geoffreylitt.com/2023/03/25/llm-end-user-programming.html

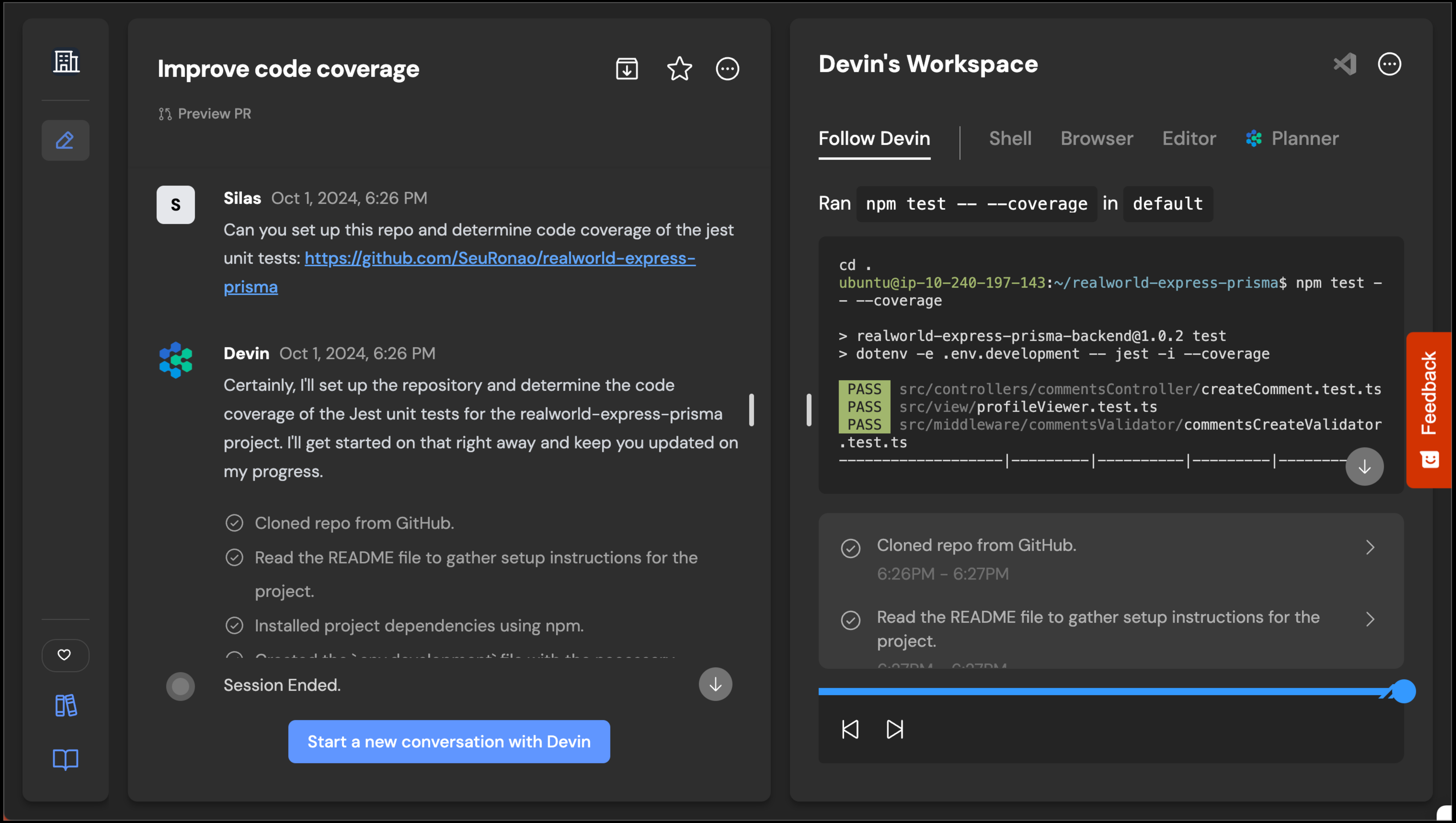
Before





https://www.geoffreylitt.com/2023/03/25/llm-end-user-programming.html

Now



Silas Oct 1, 2024, 6:26 PM

Can you set up this repo and determine code coverage of the jest unit tests: https://github.com/SeuRonao/realworld-express-prisma

Devin Oct 1, 2024, 6:26 PM

Certainly, I'll set up the repository and determine the code coverage of the Jest unit tests for the realworld-express-prisma project. I'll get started on that right away and keep you updated on my progress.

- Cloned repo from GitHub.
- Read the README file to gather setup instructions for the project.
- Installed project dependencies using npm.

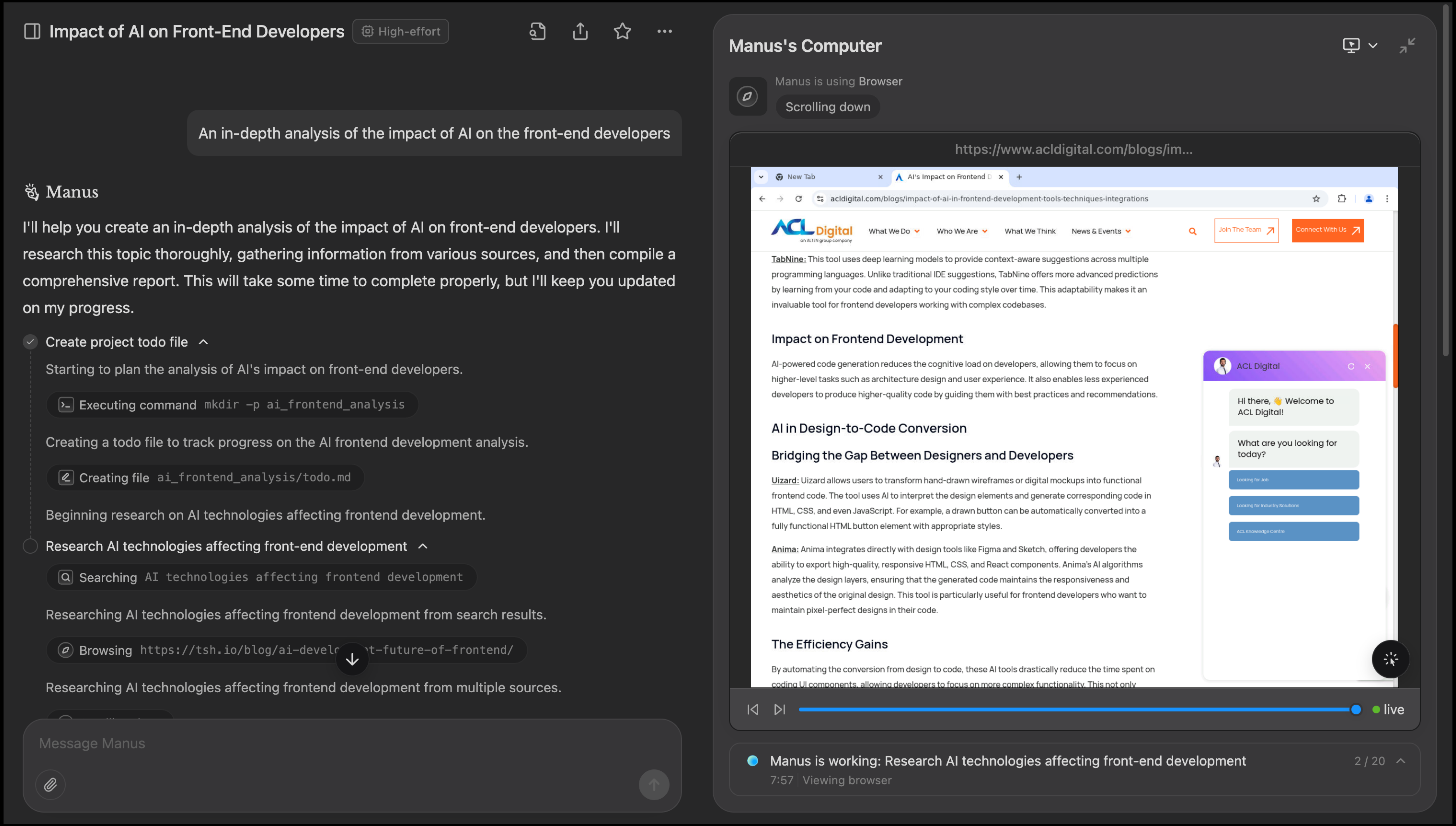
Session Ended.

Ran `npm test -- --coverage` in `default`

```
cd .
ubuntu@ip-10-240-197-143:~/realworld-express-prisma$ npm test -
- --coverage

> realworld-express-prisma-backend@1.0.2 test
> dotenv -e .env.development -- jest -i --coverage

PASS  src/controllers/commentsController/createComment.test.ts
PASS  src/view/profileViewer.test.ts
PASS  src/middleware/commentsValidator/commentsCreateValidator
.test.ts
------------------|---------|----------|---------|--------
```

- Cloned repo from GitHub.
  6:26PM - 6:27PM
- Read the README file to gather setup instructions for the project.

https://devin.ai/



Impact of AI on Front-End Developers

An in-depth analysis of the impact of AI on the front-end developers

Manus

I'll help you create an in-depth analysis of the impact of AI on front-end developers. I'll research this topic thoroughly, gathering information from various sources, and then compile a comprehensive report. This will take some time to complete properly, but I'll keep you updated on my progress.

- ✓ Create project todo file
  - Starting to plan the analysis of AI's impact on front-end developers.
  - Executing command `mkdir -p ai_frontend_analysis`
  - Creating a todo file to track progress on the AI frontend development analysis.
  - Creating file `ai_frontend_analysis/todo.md`
  - Beginning research on AI technologies affecting frontend development.
- ○ Research AI technologies affecting front-end development
  - Searching `AI technologies affecting frontend development`
  - Researching AI technologies affecting frontend development from search results.
  - Browsing `https://tsh.io/blog/ai-develo t-future-of-frontend/`
  - Researching AI technologies affecting frontend development from multiple sources.

TabNine: This tool uses deep learning models to provide context-aware suggestions across multiple programming languages. Unlike traditional IDE suggestions, TabNine offers more advanced predictions by learning from your code and adapting to your coding style over time. This adaptability makes it an invaluable tool for frontend developers working with complex codebases.

Impact on Frontend Development

AI-powered code generation reduces the cognitive load on developers, allowing them to focus on higher-level tasks such as architecture design and user experience. It also enables less experienced developers to produce higher-quality code by guiding them with best practices and recommendations.

AI in Design-to-Code Conversion

Bridging the Gap Between Designers and Developers

Uizard: Uizard allows users to transform hand-drawn wireframes or digital mockups into functional frontend code. The tool uses AI to interpret the design elements and generate corresponding code in HTML, CSS, and even JavaScript. For example, a drawn button can be automatically converted into a fully functional HTML button element with appropriate styles.

Anima: Anima integrates directly with design tools like Figma and Sketch, offering developers the ability to export high-quality, responsive HTML, CSS, and React components. Anima's AI algorithms analyze the design layers, ensuring that the generated code maintains the responsiveness and aesthetics of the original design. This tool is particularly useful for frontend developers who want to maintain pixel-perfect designs in their code.

The Efficiency Gains

By automating the conversion from design to code, these AI tools drastically reduce the time spent on coding UI components, allowing developers to focus on more complex functionality. This not only

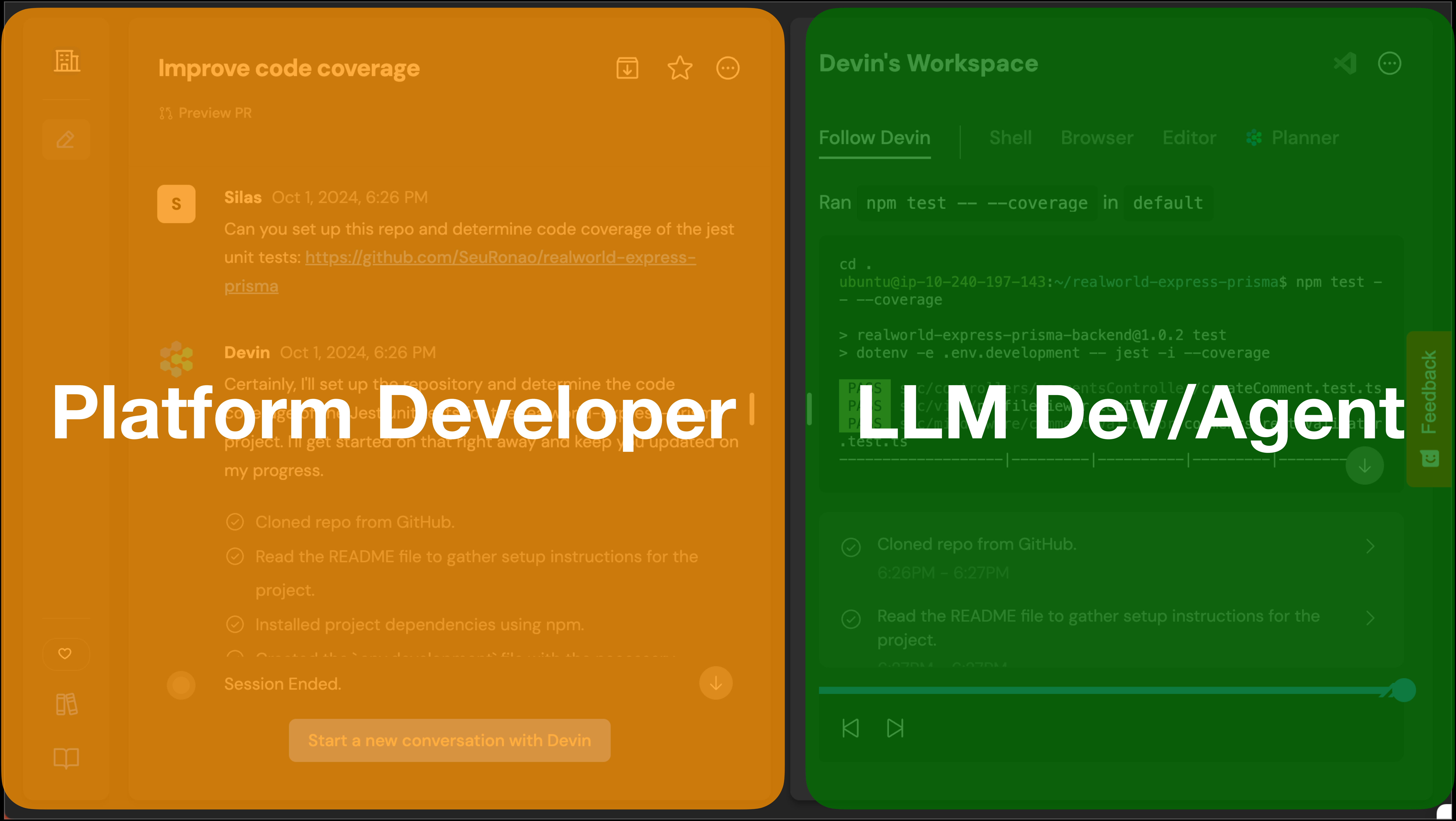
Hi there, 👋 Welcome to ACL Digital!

What are you looking for today?

Manus is working: Research AI technologies affecting front-end development

7:57 Viewing browser

https://manus.im/



Platform Developer

LLM Dev/Agent

Improve code coverage

Preview PR

Silas Oct 1, 2024, 6:26 PM

Can you set up this repo and determine code coverage of the jest unit tests: https://github.com/SeuRonao/realworld-express-prisma

Devin Oct 1, 2024, 6:26 PM

Certainly, I'll set up the repository and determine the code coverage of the Jest unit tests for the realworld-express-prisma project. I'll get started on that right away and keep you updated on my progress.

- Cloned repo from GitHub.
- Read the README file to gather setup instructions for the project.
- Installed project dependencies using npm.

Session Ended.

Start a new conversation with Devin

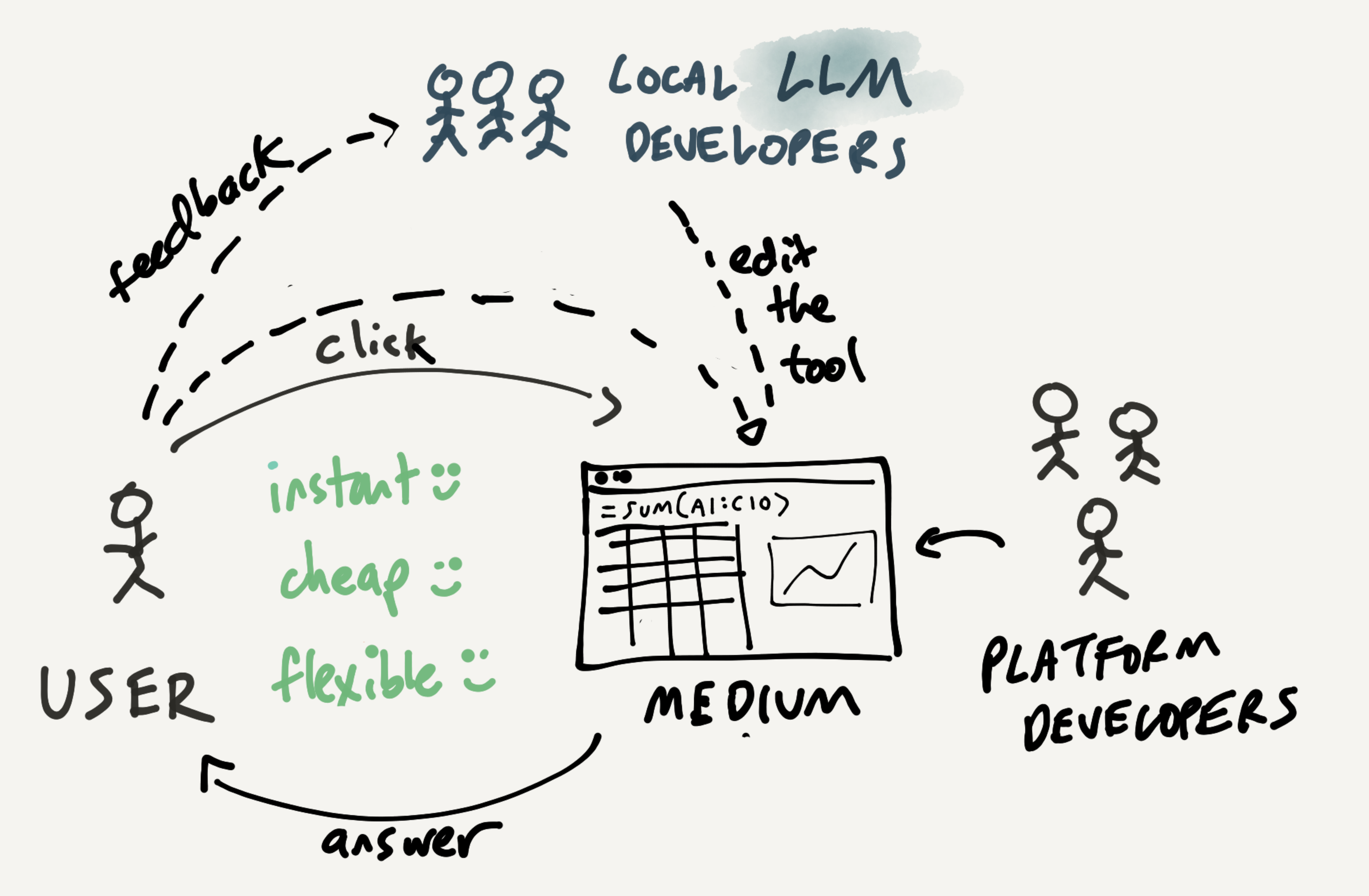
Devin's Workspace

Ran `npm test -- --coverage` in `default`

```
cd .
ubuntu@ip-10-240-197-143:~/realworld-express-prisma$ npm test -
- --coverage

> realworld-express-prisma-backend@1.0.2 test
> dotenv -e .env.development -- jest -i --coverage

.test.ts
-------------------|----------|-----------|----------|---------
```

Cloned repo from GitHub.
6:26PM - 6:27PM

Read the README file to gather setup instructions for the project.

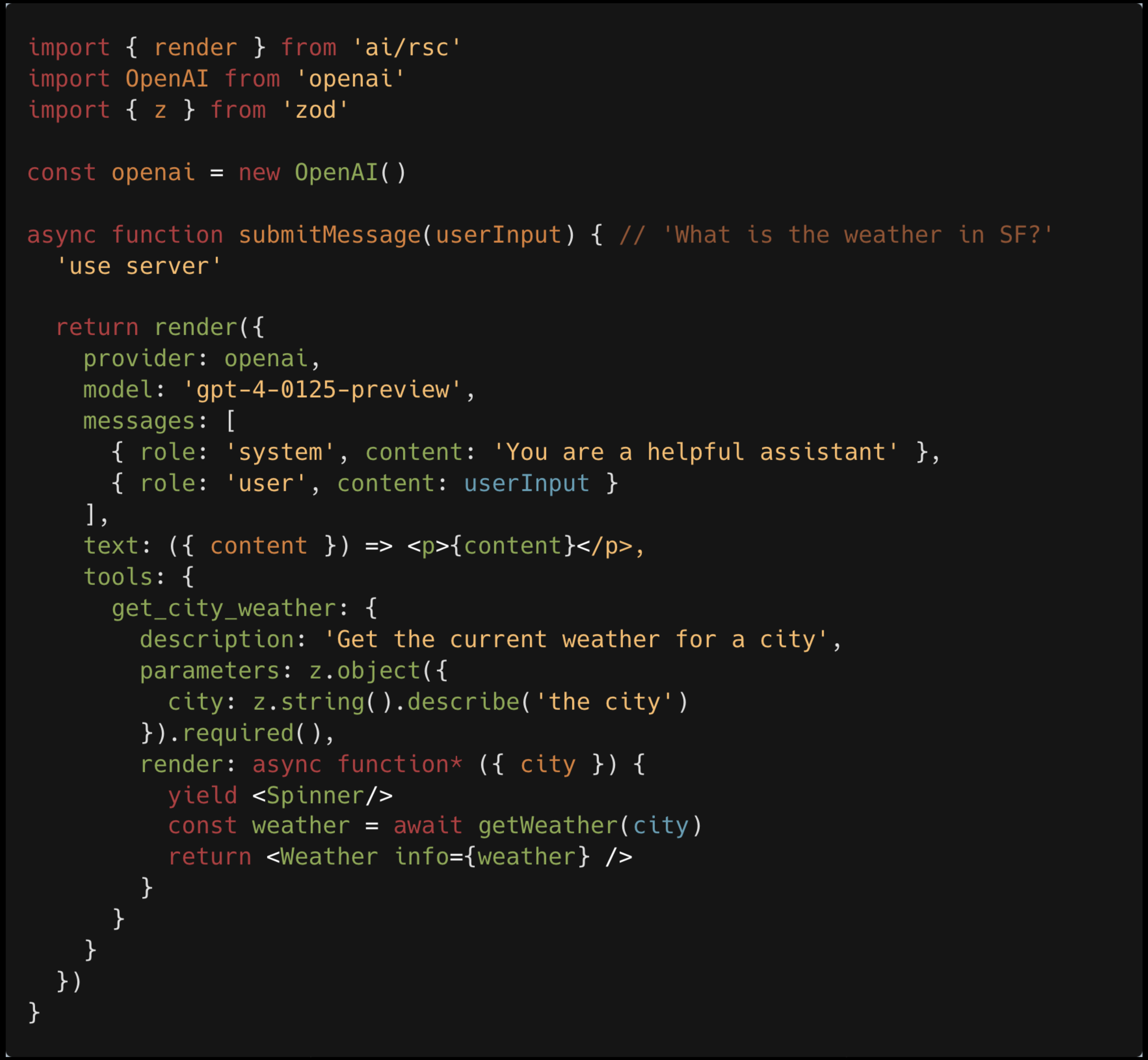


Future: Malleable software in the age of LLMs

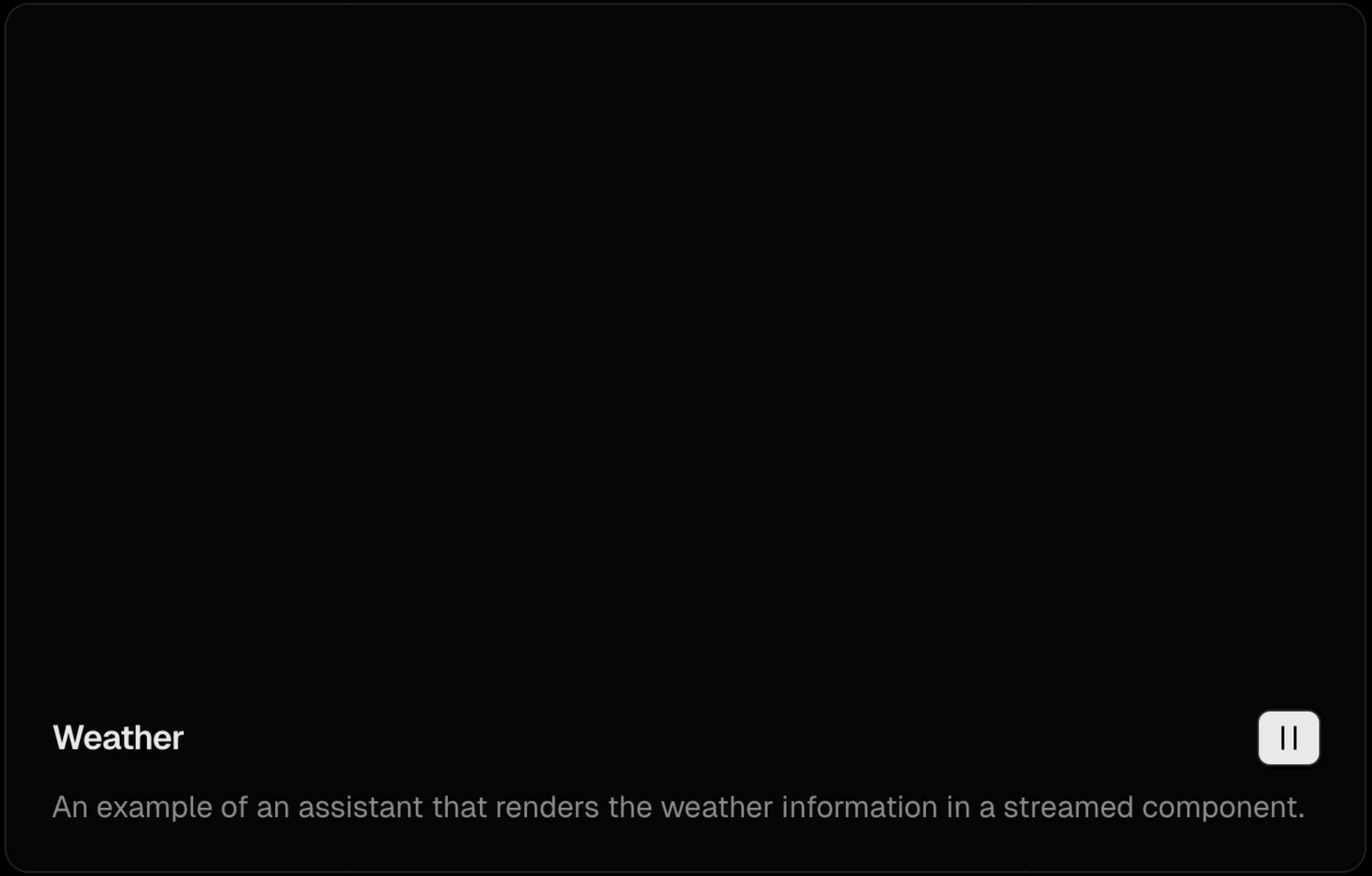
https://www.geoffreylitt.com/2023/03/25/llm-end-user-programming.html

可塑性软件
Malleable Software
与 Local LLM Agent
的开发与构建者

npm install ai



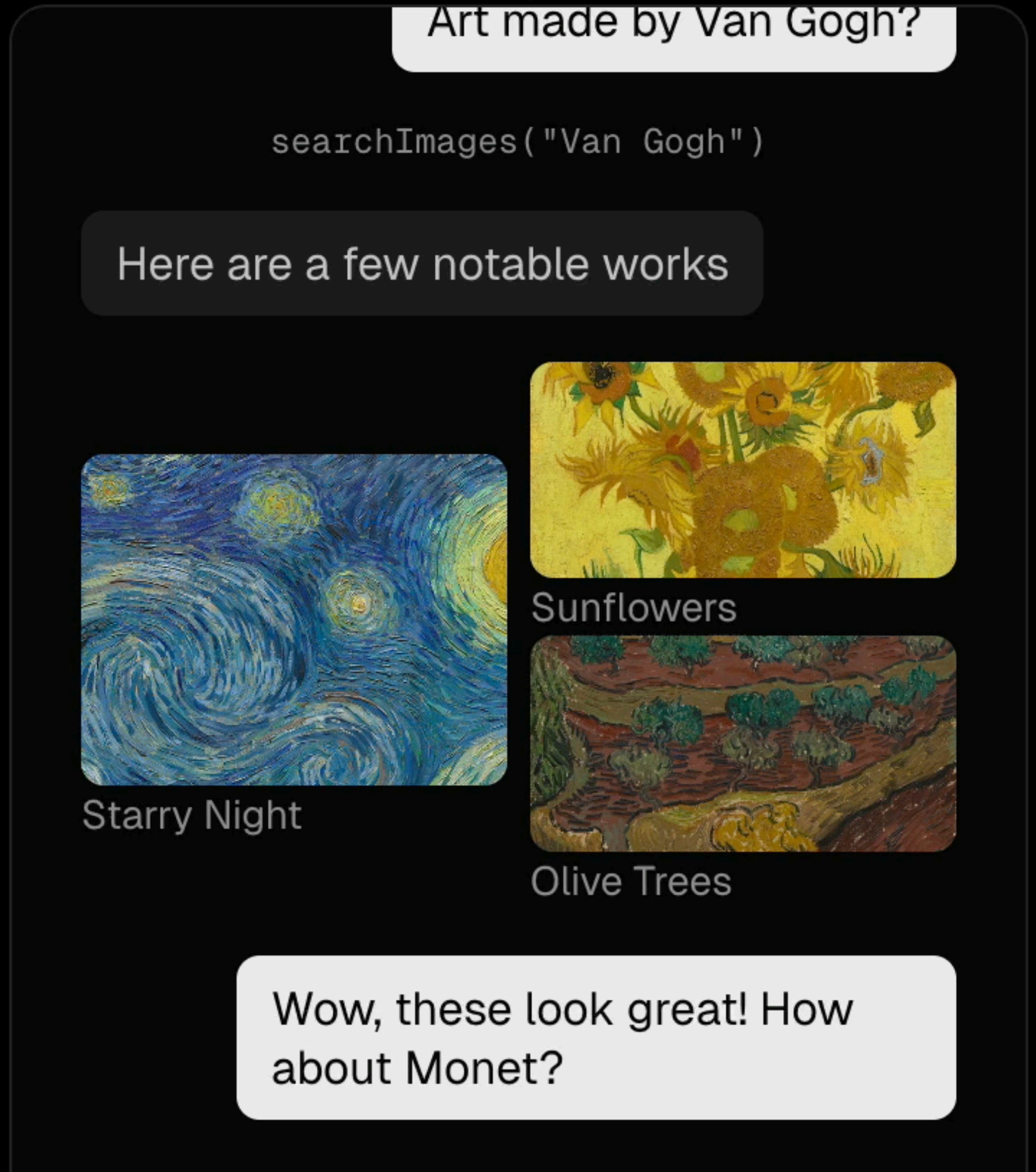
```
import { render } from 'ai/rsc'
import OpenAI from 'openai'
import { z } from 'zod'

const openai = new OpenAI()

async function submitMessage(userInput) { // 'What is the weather in SF?'
  'use server'

  return render({
    provider: openai,
    model: 'gpt-4-0125-preview',
    messages: [
      { role: 'system', content: 'You are a helpful assistant' },
      { role: 'user', content: userInput }
    ],
    text: ({ content }) => <p>{content}</p>,
    tools: {
      get_city_weather: {
        description: 'Get the current weather for a city',
        parameters: z.object({
          city: z.string().describe('the city')
        }).required(),
        render: async function* ({ city }) {
          yield <Spinner/>
          const weather = await getWeather(city)
          return <Weather info={weather} />
        }
      }
    }
  })
}
```



Weather

An example of an assistant that renders the weather information in a streamed component.

npm install ai



Art made by Van Gogh?

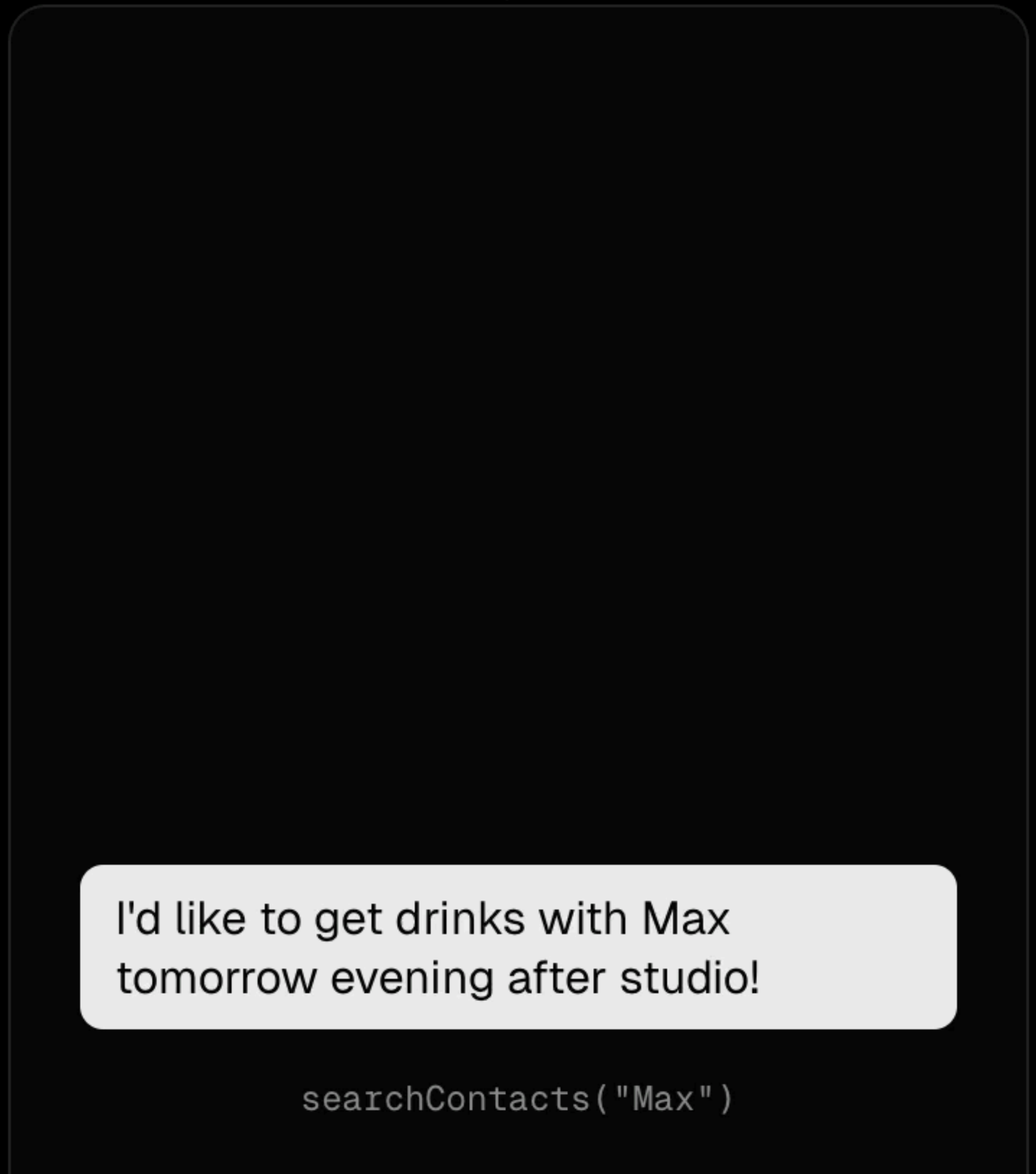
`searchImages("Van Gogh")`

Here are a few notable works

Wow, these look great! How about Monet?

Search

Let your users see more than words can say by rendering components directly within your search experience.



I'd like to get drinks with Max tomorrow evening after studio!

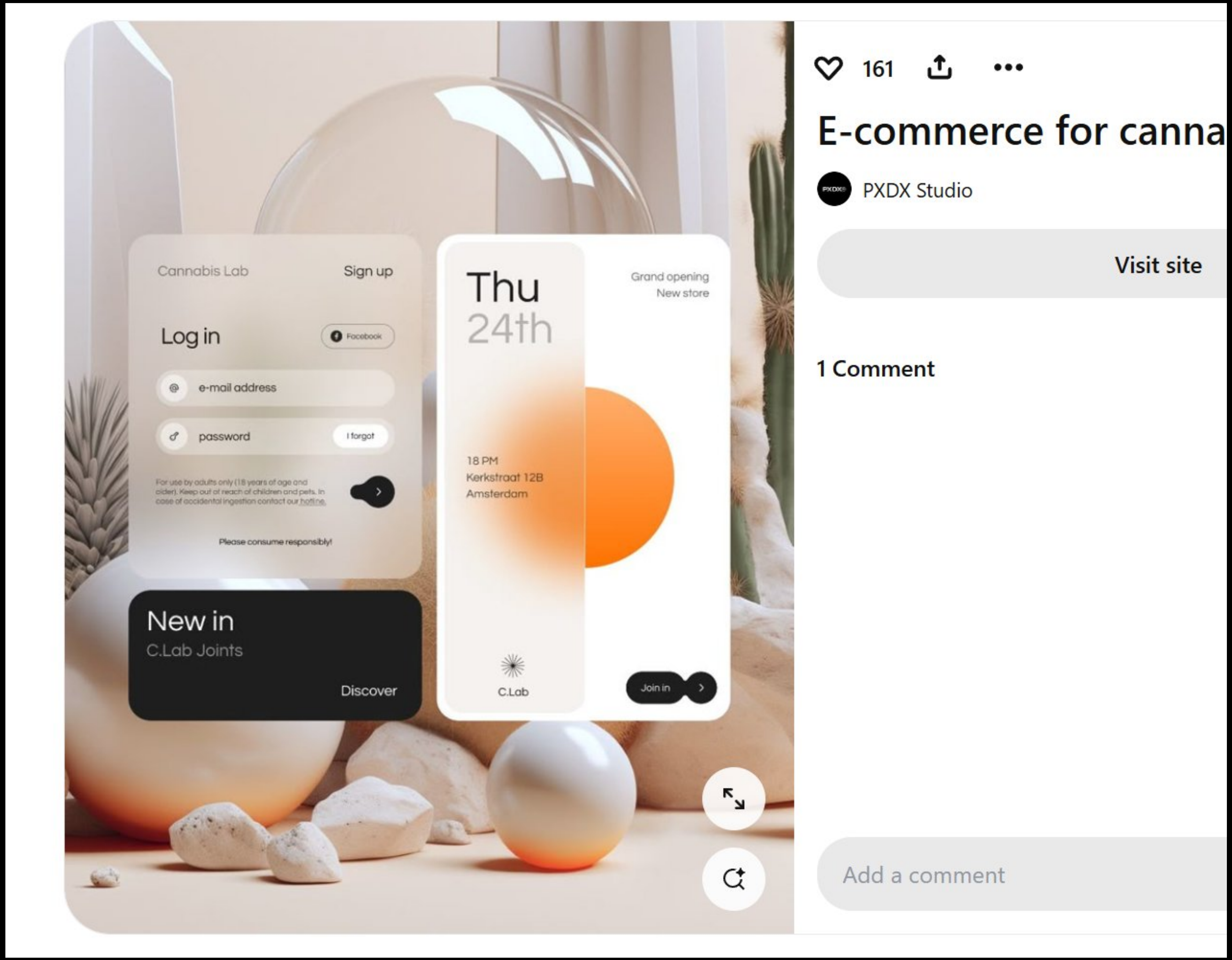
`searchContacts("Max")`

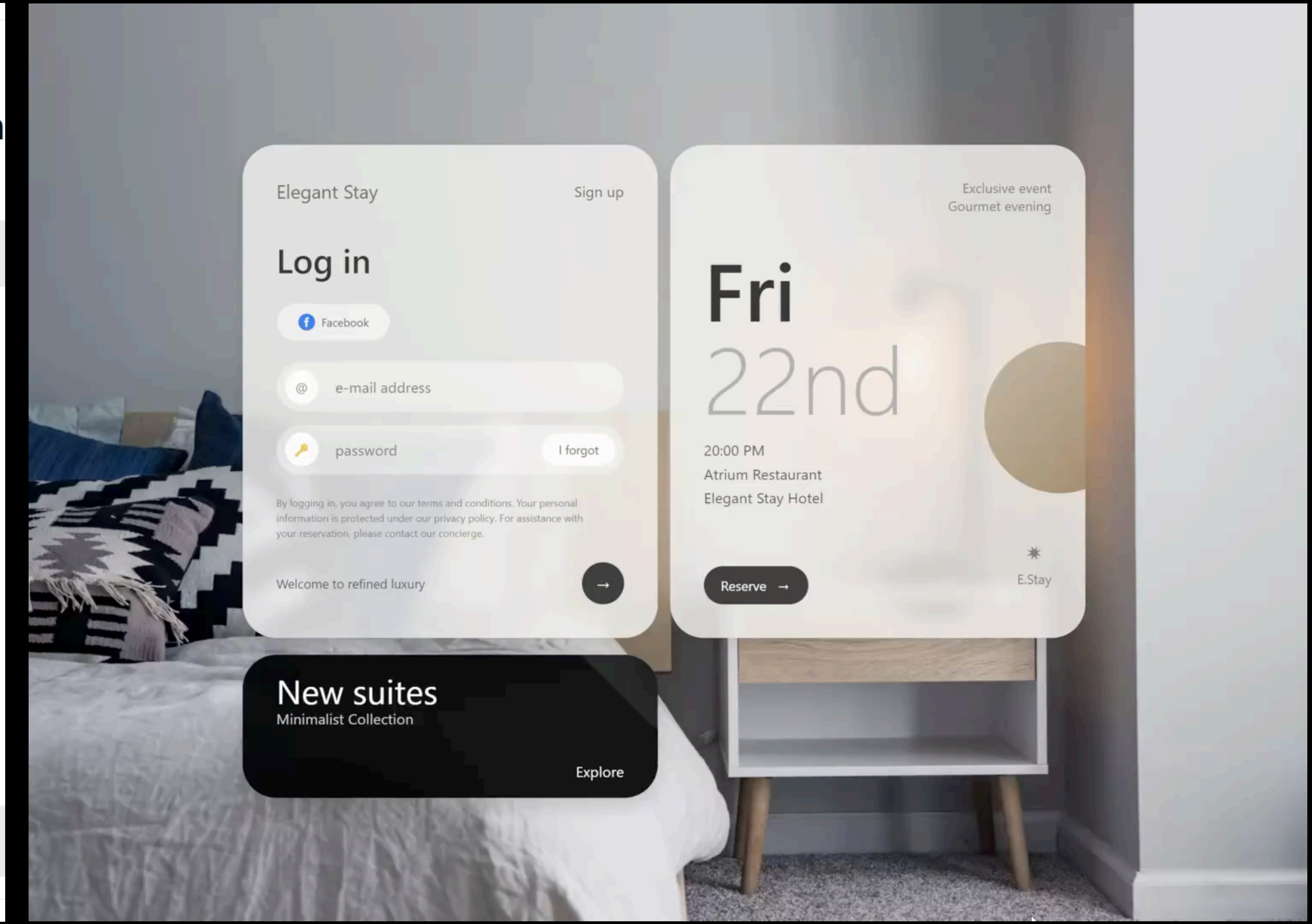
Task Planning

Make it easier for your users to interpret agent execution so they can stay in the loop with the magic behind the scenes.

```
import { render } from 'ai/rsc'
import OpenAI from 'openai'
import { z } from 'zod'

const openai = new OpenAI()

async function submitMessage(userInput) { // 'What is the weather in SF?'
  'use server'

  return render({
    provider: openai,
    model: 'gpt-4-0125-preview',
    messages: [
      { role: 'system', content: 'You are a helpful assistant' },
      { role: 'user', content: userInput }
    ],
    text: ({ content }) => <p>{content}</p>,
    tools: {
      get_city_weather: {
        description: 'Get the current weather for a city',
        parameters: z.object({
          city: z.string().describe('the city')
        }).required(),
        render: async function* ({ city }) {
          yield <Spinner/>
          const weather = await getWeather(city)
          return <Weather info={weather} />
        }
      }
    }
  })
}
```

2. 跟随生产力进步



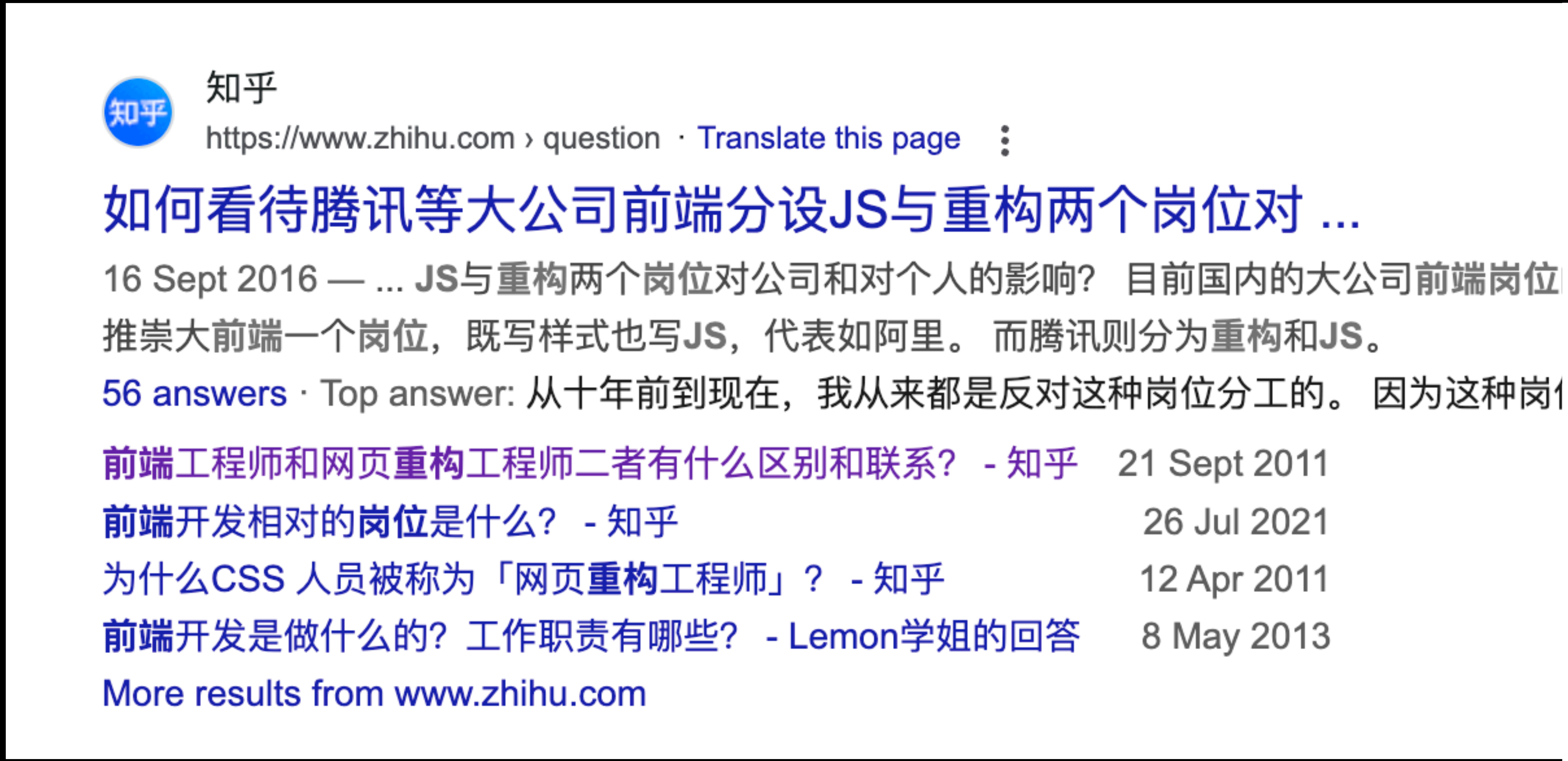


By logging in, you agree to our terms and conditions. Your personal information is protected under our privacy policy. For assistance with your reservation, please contact our concierge.

https://x.com/lepadphone/status/1896212860013031615

谁会被 AI 替代

程序员
产品经理
设计师



知乎
https://www.zhihu.com › question · Translate this page

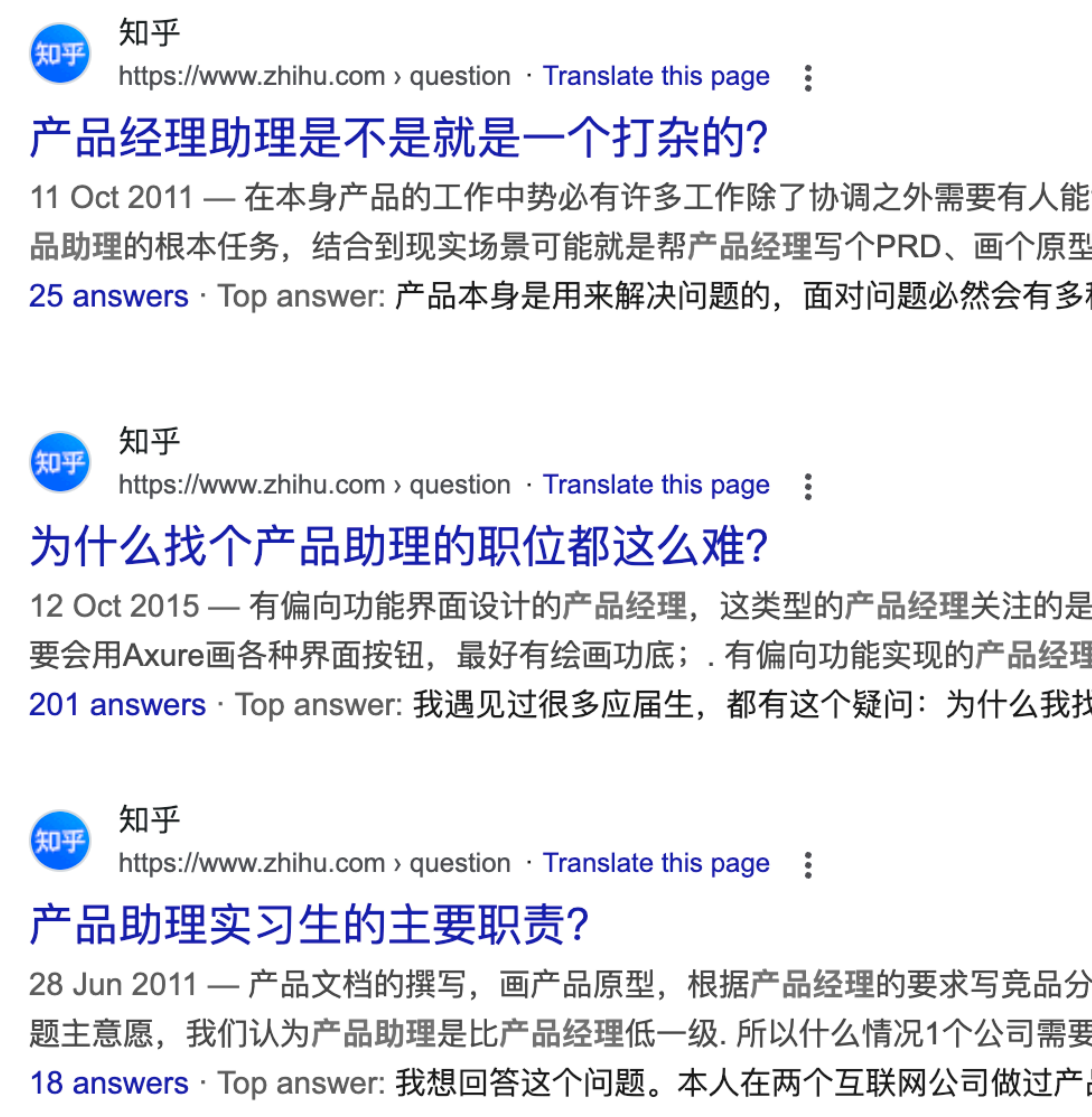
如何看待腾讯等大公司前端分设JS与重构两个岗位对 ...

16 Sept 2016 — ... **JS**与**重构**两个**岗位**对公司和对个人的影响？ 目前国内的大公司**前端岗位**
推崇大**前端**一个**岗位**，既写样式也写**JS**，代表如阿里。 而腾讯则分为**重构**和**JS**。

56 answers · Top answer: 从十年前到现在，我从来都是反对这种岗位分工的。 因为这种岗

前端工程师和网页**重构**工程师二者有什么区别和联系？ - 知乎　21 Sept 2011
前端开发相对的**岗位**是什么？ - 知乎　26 Jul 2021
为什么CSS 人员被称为「网页**重构**工程师」？ - 知乎　12 Apr 2011
前端开发是做什么的？工作职责有哪些？ - Lemon学姐的回答　8 May 2013
More results from www.zhihu.com

生产力的提升带来角色的合并



知乎
https://www.zhihu.com › question · Translate this page

产品经理助理是不是就是一个打杂的?

11 Oct 2011 — 在本身产品的工作中势必有许多工作除了协调之外需要有人能
品助理的根本任务，结合到现实场景可能就是帮**产品经理**写个PRD、画个原

25 answers · Top answer: 产品本身是用来解决问题的，面对问题必然会有多

知乎
https://www.zhihu.com › question · Translate this page

为什么找个产品助理的职位都这么难?

12 Oct 2015 — 有偏向功能界面设计的**产品经理**，这类型的**产品经理**关注的是
要会用Axure画各种界面按钮，最好有绘画功底；. 有偏向功能实现的**产品经**

201 answers · Top answer: 我遇见过很多应届生，都有这个疑问：为什么我

知乎
https://www.zhihu.com › question · Translate this page

产品助理实习生的主要职责?

28 Jun 2011 — 产品文档的撰写，画产品原型，根据**产品经理**的要求写竞品分
题主意愿，我们认为**产品助理**是比**产品经理**低一级. 所以什么情况1个公司需要

18 answers · Top answer: 我想回答这个问题。本人在两个互联网公司做过产

从“交付前端”到“交付价值”

传统开发技能

专注于编码和技术专长

VS

产品思维技能

强调解决方案和用户洞察

- AI工具降低了编码门槛,使更多人能参与产品构建
- 开发者角色从”编写代码"向"定义问题与解决方案"转变
- “有用"的判断和"好用"的设计能力将变得更加关键
- 技术壁垒降低后,产品思维和用户洞察将成为核心竞争力

做一个好的 “AI”

Chain of Thought

分步骤、有条理地思考问题，
而不是直接跳到结论

Context Length

信息记忆与处理能力,能够保持较长的”上下文记忆"，记住项目背景、重要细节和历史决策

Instruction Following

准确理解并执行指示，找准工作重点，适应工作要求

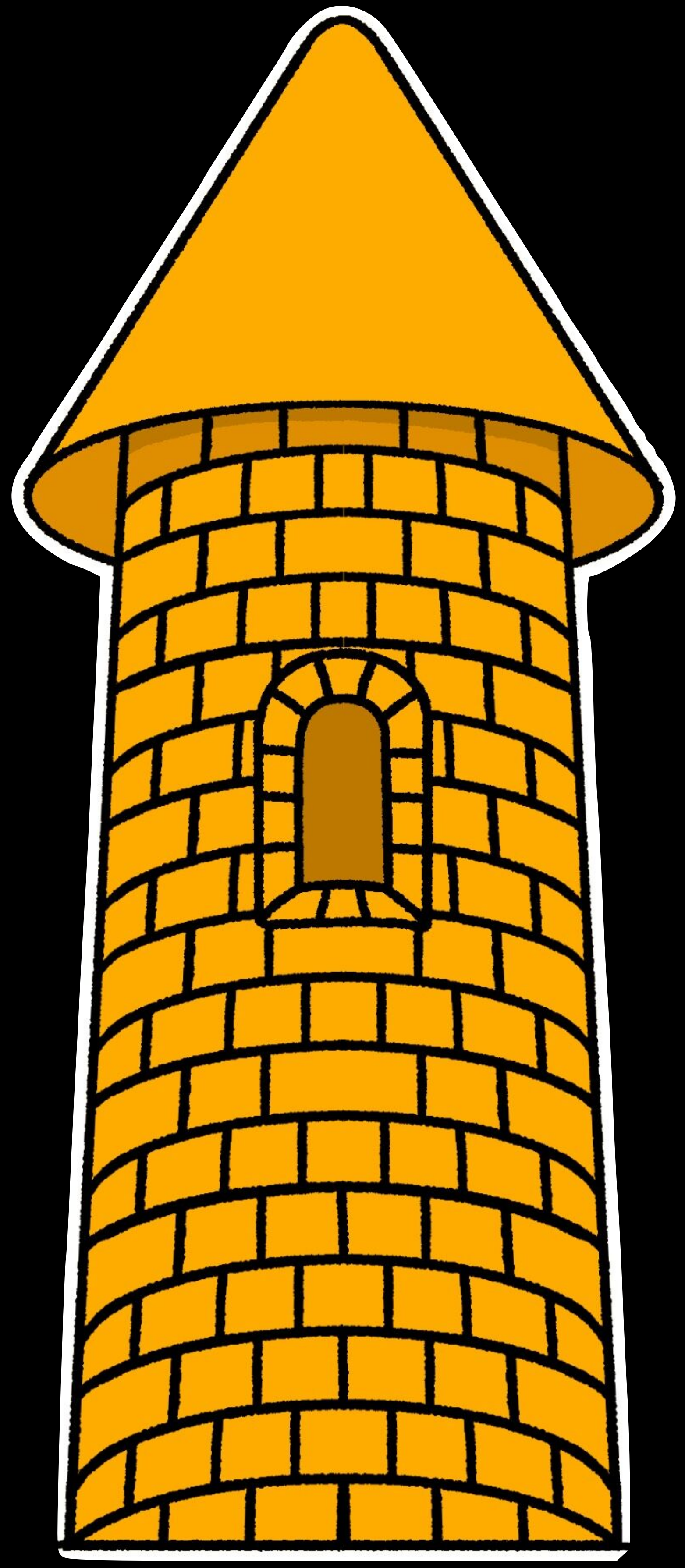
Generalization

卓越的理解力，根据已有知识融汇贯通，能够快速消化新信息并应用到工作中

Hallucination

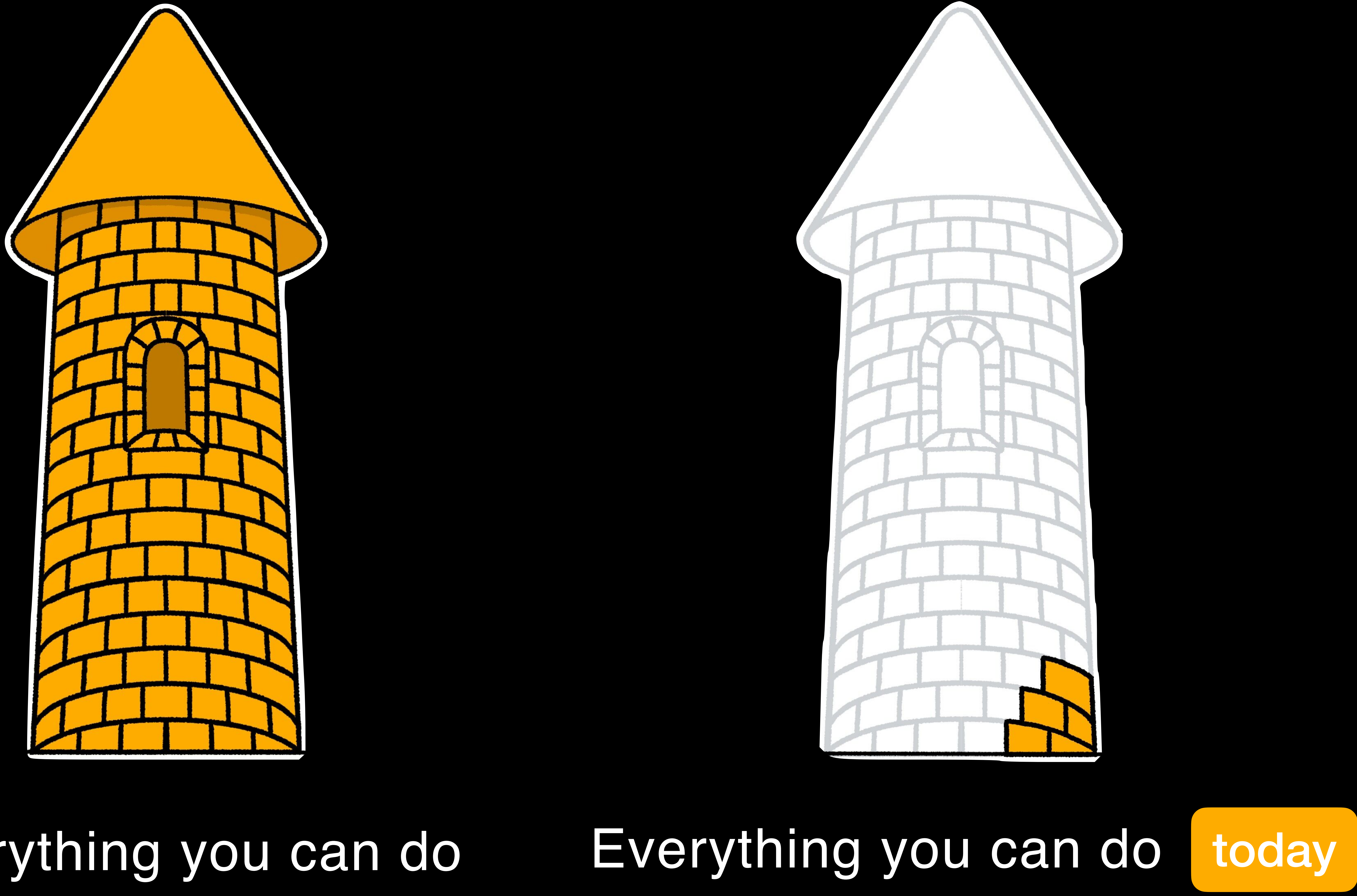
提供准确、真实的信息,不会凭空捏造或过度推断

从“做点什么”开始



Everything you can do

从“做点什么”开始



Everything you can do

Everything you can do **today**

在 AI 时代拥有更好的竞争力

跟随软件发展趋势

- 可塑性软件 Malleable Software
- Local LLM Agent

跟随生产力进步

- 从交付前端到交付价值
- 做一个好的 AI
- 从先做点什么开始

THANK YOU