



从端侧推理到智能体

# Web AI 的今天和明天

—— Web 进化论 2025 年度大会 · 杭州

# 关于我

前端架构师

Web GDE

掘金社区优秀创作者



钱俊颖 (Jax)

# 目录



01

## Web AI 的概念解读

什么是 Web AI? 有什么用?

02

## 浏览器内置 AI 的能力

任务 API、多模态… 应有尽有

03

## Web AI 中的 Agent

智能体的 Web 化

04

## Web AI 标准化畅想

最佳实践 → 通用标准

# 目录



01

## Web AI 的概念解读

什么是 Web AI？有什么用？

02

## 浏览器内置 AI 的能力

Task API、多模态… 应有尽有

03

## Web AI 中的 Agent

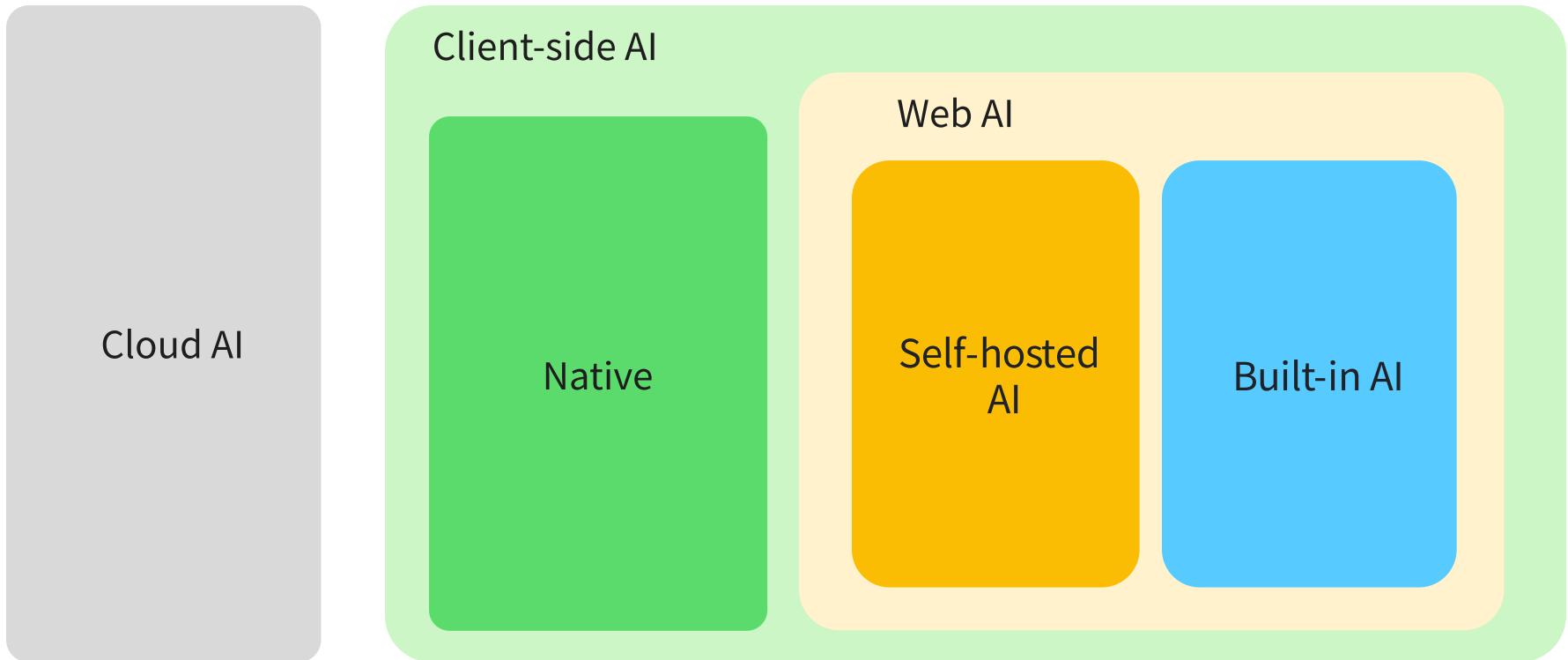
智能体的 Web 化

04

## Web AI 标准化畅想

最佳实践 → 通用标准

# 当我们聊起 Web AI，我们在聊什么？



# Web AI 分类

## Web AI

### 自托管 AI

ML Models

LLMs

自定义运行时：  
Transformers.js / WebLLM / ...

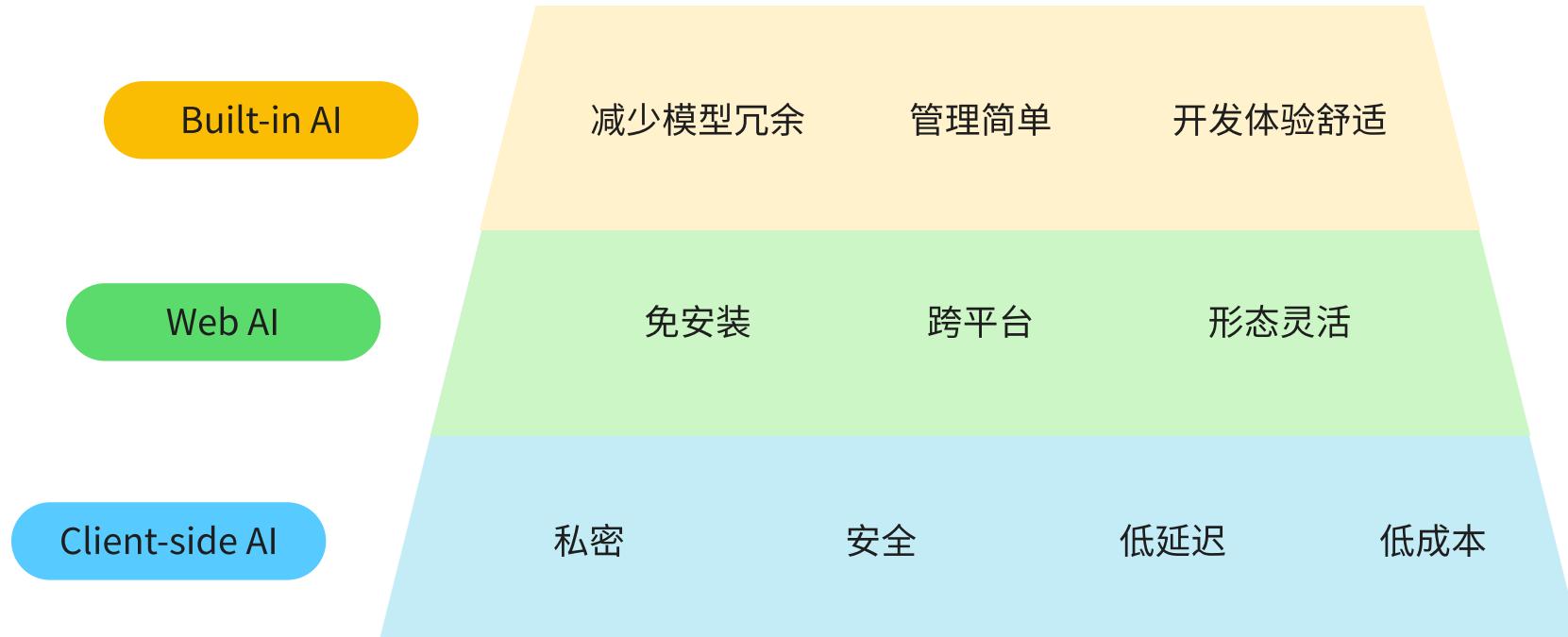
### 内置 AI

ML Models

LLMs

内置运行时

# 优势金字塔



# 局限 —— 端侧非银弹

设备的局限

算力

能耗

...

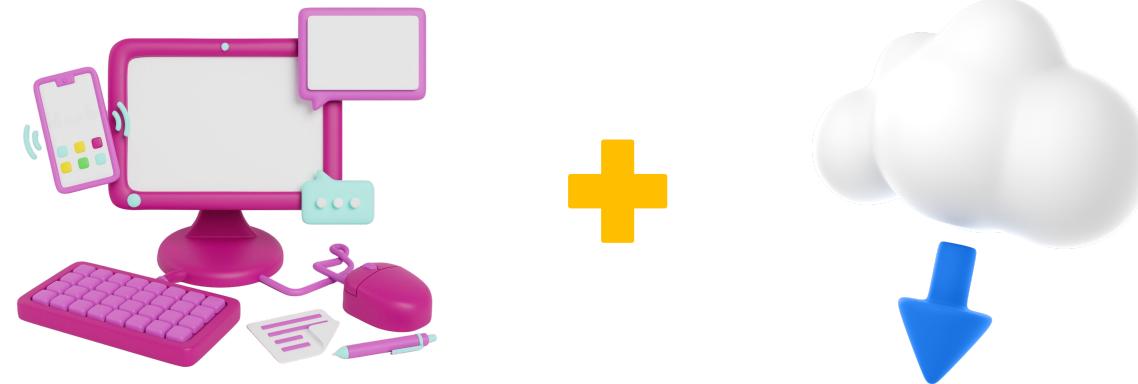
模型的局限

上下文

多模态

...

# Hybrid 模式



“此端基础，彼端就不基础”

# 目录



01

Web AI 的概念解读

什么是 Web AI? 有什么用?

02

浏览器内置 AI 的能力

Task API、多模态… 应有尽有

03

Web AI 中的 Agent

智能体的 Web 化

04

Web AI 标准化畅想

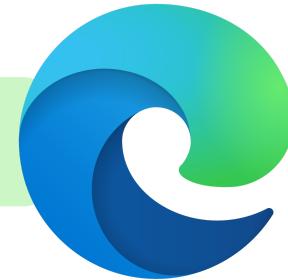
最佳实践 → 通用标准

# 浏览器支持



Gemini Nano

桌面端



Phi-4-mini

?

???

# Built-in AI

Web Apps

Browser Extensions

JavaScript APIs

LLMs

Mixture of Experts

AI Runtime

CPU

GPU

NPU

# JavaScript APIs



EXPERIMENTAL

Language Detector API

Translator API

Proofreader API

Prompt API

Writer API

Summarizer API

Rewriter API

详情参见：[https://developer.chrome.com/docs/ai/built-in-apis#api\\_status](https://developer.chrome.com/docs/ai/built-in-apis#api_status)

# Language Detector API

The screenshot shows a browser developer tools console with the following code and output:

```
> let detector = await LanguageDetector.create()
<- undefined

> await detector.detect('Web 进化论')
<- (27) [{}]
  ▼ 0: {confidence: 0.49303391575813293, detectedLanguage: 'zh'}
  ▶ 1: {confidence: 0.32959455251693726, detectedLanguage: 'en'}
  ▶ 2: {confidence: 0.022281412035226822, detectedLanguage: 'pa'}
  ▶ 3: {confidence: 0.020617268979549408, detectedLanguage: 'ar-Latn'}
  ▶ 4: {confidence: 0.013205839321017265, detectedLanguage: 'pl'}
  ▶ 5: {confidence: 0.012390959076583385, detectedLanguage: 'de'}
  ▶ 6: {confidence: 0.009416790679097176, detectedLanguage: 'ms'}
  ▶ 7: {confidence: 0.00826546922326088, detectedLanguage: 'da'}
  ▶ 8: {confidence: 0.006783562246710062, detectedLanguage: 'mt'}
  ▶ 9: {confidence: 0.005379269365221262, detectedLanguage: 'yi'}
  ▶ 10: {confidence: 0.00478512654080987, detectedLanguage: 'ky'}
  ▶ 11: {confidence: 0.004000000000000001, detectedLanguage: 'ukr'}
  ▶ 12: {confidence: 0.0035000000000000003, detectedLanguage: 'tr'}
  ▶ 13: {confidence: 0.0030000000000000002, detectedLanguage: 'slv'}
  ▶ 14: {confidence: 0.0025000000000000002, detectedLanguage: 'sq'}
  ▶ 15: {confidence: 0.0020000000000000004, detectedLanguage: 'mkd'}
  ▶ 16: {confidence: 0.0015000000000000003, detectedLanguage: 'hrv'}
  ▶ 17: {confidence: 0.0010000000000000002, detectedLanguage: 'hrv'}
  ▶ 18: {confidence: 0.0005000000000000001, detectedLanguage: 'hrv'}
  ▶ 19: {confidence: 0.00020000000000000004, detectedLanguage: 'hrv'}
  ▶ 20: {confidence: 0.00010000000000000002, detectedLanguage: 'hrv'}
  ▶ 21: {confidence: 0.000050000000000000005, detectedLanguage: 'hrv'}
```

# Translator API

```
> let translator = await Translator.create({
  sourceLanguage: 'zh',
  targetLanguage: 'en'
})
< undefined

> await translator.translate('W3C Web中文兴趣组 (Chinese Web Interest Group) 组织本次活动，同时也为W3C AI Agent & the Web技术研讨会的中国区前瞻会议，旨在汇聚来自产业、科研机构、用户代表、开发者和标准化组织的相关利益方，共同探讨智能体 (AI Agent) 在Web生态中的角色与发展，识别潜在的标准化方向，助力构建可持续、开放且互操作的Agentic Web。')
< 'The W3C Web Chinese Interest Group (Chinese Web Interest Group) organizes this event, and also serves as W3C AI Agent & The The forward-looking meeting of the Web Technology Seminar in China, aiming to bring together relevant stakeholders from industries, scientific research institutions, user representatives, developers and standardization organizations to discuss the agents (AI Agent) Role and development in the web ecology, identify potential standardization directions, and help build a sustainable, open and interoperable Agentic Web.'
```

# Proofreader API

The screenshot shows a dark-themed interface for the Proofreader API. At the top, there is a toolbar with icons for play/pause, stop, and other controls, followed by dropdown menus for 'top' and 'Filter'. To the right of the filter is a 'Default levels ▾' button. Below the toolbar, a status bar displays '1 Issue: !1' and a gear icon for settings. The main area is a code editor window containing the following code:

```
> const proofreader = await Proofreader.create();

(await proofreader.proofread("I seen him yesterday at the
store, and he bought two loaf of
bread."))?.correctedInput;

< 'I saw him yesterday at the store, and he bought two loaves
of bread.'
```

Below the code editor, there is a single character '>'.

# Writing Assistance API

```
> const summarizer = await Summarizer.create({
    type: "tldr",
    length: "short",
});
< undefined
> await summarizer.summarize(str)
< 'The Chinese Web Interest Group is holding a meeting to di
scuss how *AI agents* will shape the future of the web, co
vering topics like construction, ecosystem integration, co
mmunication, use cases, and standardization.'
```

# Prompt API

Prompt Engineering

Multimodal

Structured Output

Tool Use

详情参见：<https://github.com/webmachinelearning/prompt-api>

# 目录



01

Web AI 的概念解读

什么是 Web AI? 有什么用?

02

浏览器内置 AI 的能力

Task API、多模态… 应有尽有

03

Web AI 中的 Agent

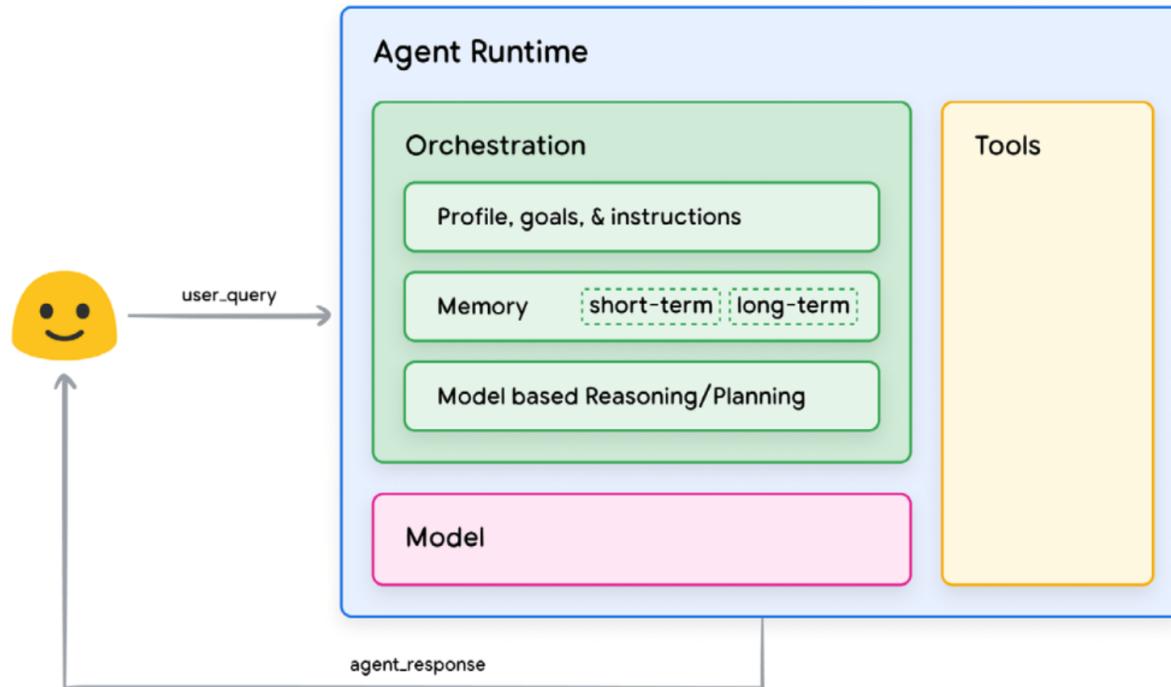
智能体的 Web 化

04

Web AI 标准化畅想

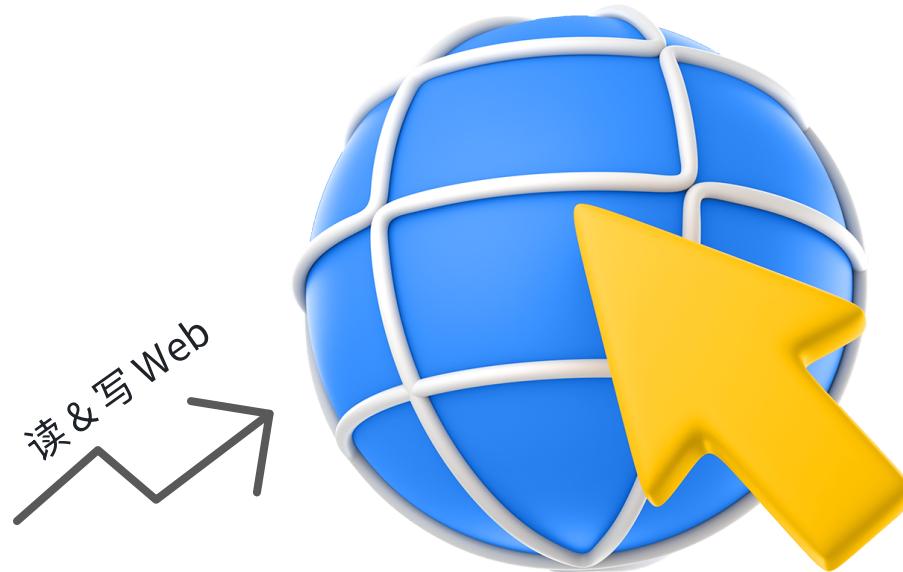
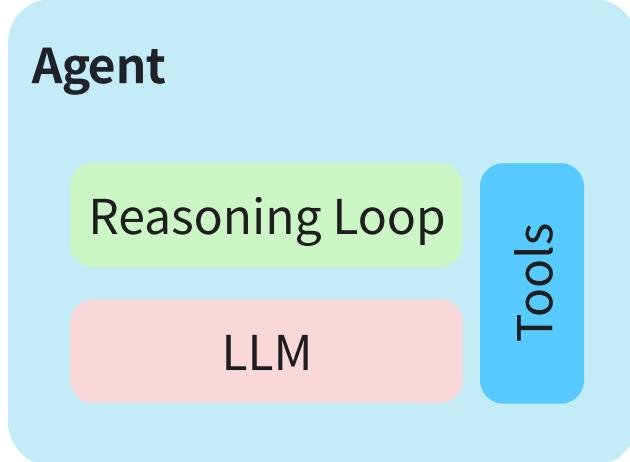
最佳实践 → 通用标准

# Agent 的结构



图片来源: <https://www.kaggle.com/whitepaper-agents>

# Web Agent



读 & 写 Web

# Agent In Web AI

Web Apps

Browser Extensions

Agents

LLMs

AI Runtime

CPU

GPU

NPU

# Web 端实现 AI Agent

## Vanilla

- Self-hosted LLMs
- LLM Runtime
- Orchestration
- Tools

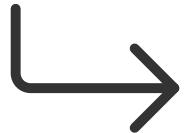
## Built-in AI

- Orchestration
- Tools

## LangGraph.js

- Cloud AI
- Orchestration
- Tools

# 互为工具



# 目录



01

Web AI 的概念解读

什么是 Web AI? 有什么用?

02

浏览器内置 AI 的能力

Task API、多模态… 应有尽有

03

Web AI 中的 Agent

智能体的 Web 化

04

Web AI 标准化畅想

最佳实践 → 通用标准



假如你是秦始皇.....

# 为什么要有标准？

解决通用需求 / 痛点

统一规范

保持开放生态

提炼最佳实践

# 标准是怎么来的？



# 第一步

广义 Web AI / 端侧 Web AI  
读写 Web 的 Agent / 运行在 Web 里的 Agent  
.....



# 正在推进的标准



<https://webmachinelearning.github.io/>

## Community Group Drafts

- [Translator & Language Detector API](#)
- [Writing Assistance APIs](#)

## Community Group Explainers

- [Proofreader API](#)
- [Prompt API \(?\)](#)

# Web AI 基础设施

工具 / API / Agent

数据

模型

运行时环境

算力

# 痛点 $\leftrightarrow$ 标准

多方协作困难

ANP / A2A / ...

数据结构花样百出

MCP / JSON-LD / ...

模型冗余

缓存 / 共享 / 格式 / ...

运行时环境不一致

通用环境 / ..

端侧设备算力弱

端云 Hybrid / ...

# 总结

## 聚焦范围

着眼于端侧 Web 环境的智能化，讨论 Web AI 的结构和价值。

## Agent In Web

不同层级的 Agent，相互之间的关系，在 Web 端的实现方式。



## Built-in AI 能力

对 JavaScript 能力的扩展，全面适应 AI 场景。

## Web AI 标准

标准化进程的现状，可预见的需求和缺口

# 感谢倾听

如果你也对 Web AI 感兴趣，  
热烈欢迎一起交流！

