

基于 LLM 构建 Chrome Extension 效率工具

谢亚东 / Lepton AI



⚡ Elmo is your AI companion to create summaries and insights



Install from Chrome Web Store

Free / No Account Required / Supports Multiple Languages



Featured on
Chrome Web Store



Developed by
Established Publisher



Work seamlessly across HTML, Youtube, PDF, and Google Docs.



Guillermo Rauch ✓

This looks really handy



Bing Xu ✓

Elmo is the co-pilot for Chrome. Super helpful when reading new API documents. Thanks @LeptonAI for



Diogo Santos ✓

Elmo is your AI extension 🧠 for Chrome to create summaries, insights, and extended knowledge. What does Elmo offer? 🤔 ✓ Summaries and highlights; ✓ Keep asking questions; ✓ Dive deep into keywords; ✓ Chat with PDFs; ✓ Talk with YouTube.



Alvin-GenAI ✓

I've downloaded <https://elmo.chat> an AI Chrome extension to create summaries, insights and extended knowledge. It does - summarizes your websites right next to the page, - summarizes PDF files too, - summarizes YouTube videos.



Tulsi Prasad

I tried out this AI chrome extension yesterday! 1. No product hunt launches 2. Featured on Chrome Web Store like an OG 3. No signup needed, just plug and play! Works pretty accurate on blogs and textual content, but videos are promising as well!

本站預設外觀已更換為Vector 2022。如有任何疑問可前往討論頁提出。

[关闭]

荷塘月色

文A 添加语言 ▾

作品 讨论 不转换 ▾

阅读 编辑 查看历史 工具 ▾

下載

荷塘月色

作者：朱自清 1927年

1927年7月，北京清華園

 姊妹计划: 百科·数据项

這幾天心裡頗不寧靜。今晚在院子裡坐著乘涼，忽然想起日日走過的荷塘，在這滿月的光裡，總該另有一番樣子吧。月亮漸漸地升高了，牆外馬路上孩子們的歡笑，已經聽不見了；裏在屋裡拍著閨兒，迷迷糊糊地哼著眠歌。我悄悄地披了大衫，帶上門出去。

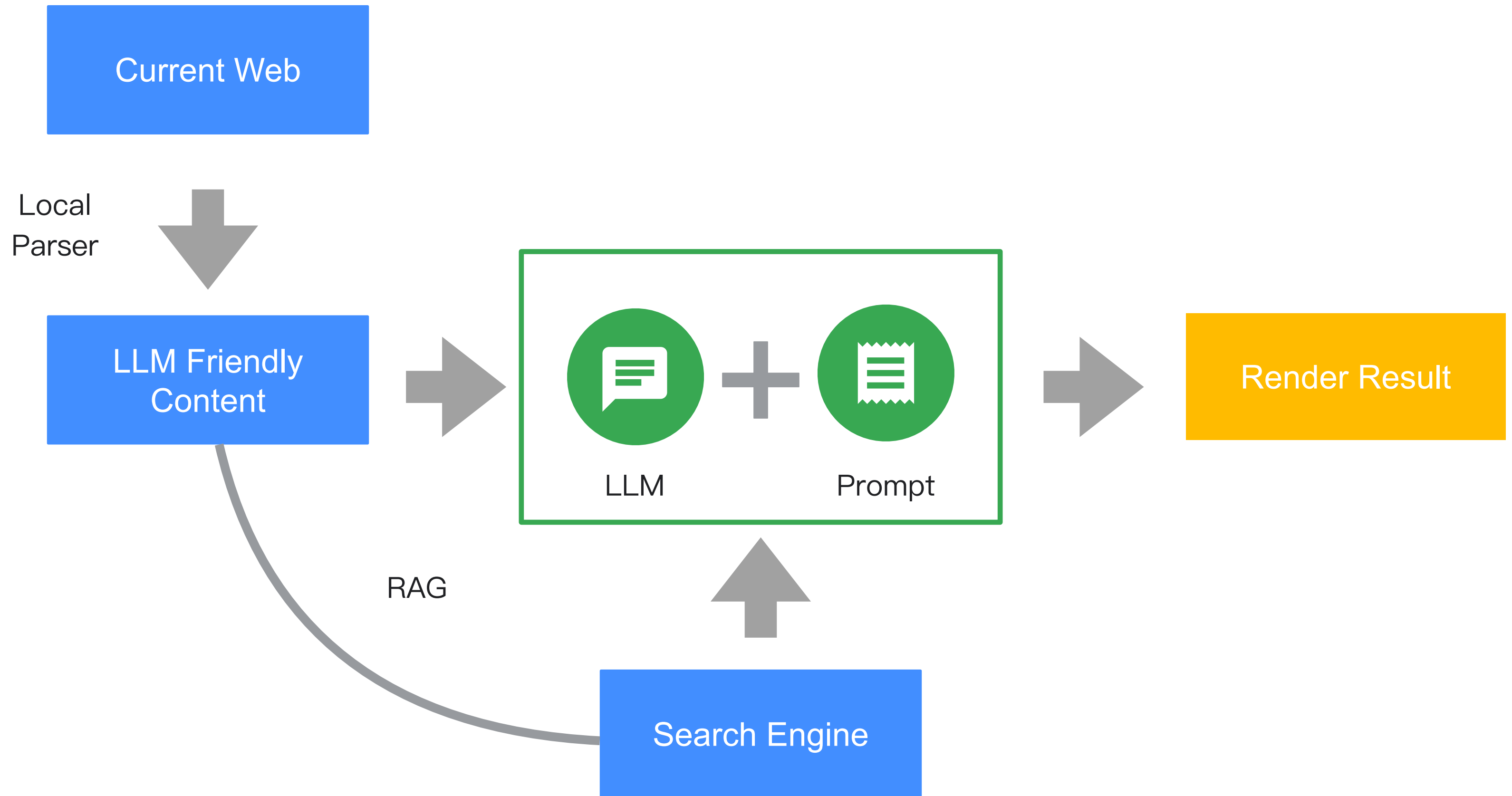
沿著荷塘，是一條曲折的小煤屑路。這是一條幽僻的路；白天也少人走，夜晚更加寂寞。荷塘四面，長著許多樹，蓊蓊鬱鬱的。路的一旁，是些楊柳，和一些不知道名字的樹。沒有月光的晚上，這路上陰森森的，有些怕人。今晚卻很好，雖然月光也還是淡淡的。

路上只我一個人，背著手踱著。這一片天地好像是我的；我也像超出了平常的自己，到了另一世界裡。我愛熱鬧，也愛冷靜；愛群居，也愛獨處。像今晚上，一個人在這蒼茫的月下，什麼都可以想，什麼都可以不想，便覺是個自由的人。白天裡一定要做的事，一定要說的話，現在都可不理。這是獨處的妙處，我且受用這無邊的荷香月色好了。

曲曲折折的荷塘上面，彌望的是田田的葉子。葉子出水很高，像亭亭的舞女的裙。層層的葉子中間，零星地點綴著些白花，有裊娜地開著的，有羞澀地打著朵兒的；正如一粒粒的明珠，又如碧天裡的星星，又如剛出浴的美人。微風過處，送來縷縷清香，彷彿遠處高樓上渺茫的歌聲似的。這時候葉子與花也有一絲的顫動，像閃電般，霎時傳過荷塘的那邊去了。葉子本是肩並肩密密地挨著，這便宛然有了一道凝碧的波痕。葉子底下是脈脈的流水，遮住了，不能見一些顏色；而葉子卻更見風致了。

月光如流水一般，靜靜地瀉在這一片葉子和花上。薄薄的青霧浮起在荷塘裡。葉子和花彷彿在牛乳中洗過一樣；又像籠著輕紗的夢。雖然是滿月，天上卻有一層淡淡的雲，所以不能朗照；但我以為這恰是到了好處——酣眠固不可少，小睡也別有風味的。月光是隔了樹照過來的，高處叢生的灌木，落下參差的斑駁的黑影，峭楞楞如鬼一般；彎彎的楊柳的稀疏的情影，卻又像是畫在荷葉上。塘中的月色並不均勻；但光與影有著和諧的旋律，如梵婀玲上奏著的名曲。

荷塘的四面，遠遠近近，高高低低都是樹，而楊柳最多。這些樹將一片荷塘重重圍住；只在小路一旁，漏著幾段空隙，像是特為月光留下的。樹色一例是陰陰的，乍看像一團煙霧；但楊柳的丰姿，便在煙霧裡也辨得出。樹梢上隱隱約約的是一帶遠山，只有些大意罷了。樹縫裡也漏著一兩點路燈光，沒精打采的，是渴睡人的眼。這時候最熱鬧的，要數樹上的蟬聲與水裡的蛙



Section 1

LLM

Section 1

LLM

WIKIPEDIA

The Free Encyclopedia

Search Wikipedia

Search

Contents

hide

(Top)

> Background

> Events

> Other confrontations

Casualties

Humanitarian crisis

Scale of destruction

War crimes

Diplomatic impact

> Reactions

> Regional effects

> Media coverage

See also

Notes

References

External links

Israel– Hamas war

ArticleTalk

From Wikipedia, the free encyclopedia

LMSYS Chatbot Arena Leaderboard

https://chat.lmsys.org/

Rank* (UB)	Model	Arena Score	95% CI	Votes	Organization	License	Knowledge Cutoff
1	GPT-4o-2024-05-13	1287	+2/-3	49124	OpenAI	Proprietary	2023/10
2	Claude 3.5 Sonnet	1271	+5/-6	14543	Anthropic	Proprietary	2024/4
2	Gemini-Advanced-0514	1267	+4/-4	38444	Google	Proprietary	Online
2	Gemini-1.5-Pro-API-0514	1263	+3/-4	42152	Google	Proprietary	2023/11
4	Gemini-1.5-Pro-API-0409-Preview	1257	+3/-3	55731	Google	Proprietary	2023/11
5	GPT-4-Turbo-2024-04-09	1257	+2/-3	69244	OpenAI	Proprietary	2023/12
7	GPT-4-1106-preview	1251	+3/-3	85076	OpenAI	Proprietary	2023/4
7	Claude 3 Opus	1248	+3/-3	138524	Anthropic	Proprietary	2023/8
7	GPT-4-0125-preview	1246	+3/-2	78333	OpenAI	Proprietary	2023/12
10	Yi-Large-preview	1240	+3/-3	44403	01 AI	Proprietary	Unknown

Section 1

Open LLM Leaderboard

https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard

T ▲	Model ▲	Average 📈 ▲	IFEval ▲	BBH ▲	MATH Lv1 5 ▲	GPQA ▲	MUSR ▲	MMLU-PRO ▲
🔹	Qwen/Qwen2-72B-Instruct 📄	43.02	79.89	57.48	35.12	16.33	17.17	48.92
💬	meta-llama/Meta-Llama-3-70B-Instruct 📄	36.67	80.99	50.19	23.34	4.92	10.92	46.74
🔹	Qwen/Qwen2-72B 📄	35.59	38.24	51.86	29.15	19.24	19.73	52.56
🟢	mistralai/Mixtral-8x22B-Instruct-v0.1 📄	34.35	71.84	44.11	18.73	16.44	13.49	38.7
💬	HuggingFaceH4/zephyr-orpo-141b-A35b-v0.1 📄	34.23	65.11	47.5	18.35	17.11	14.72	39.85
💬	microsoft/Phi-3-medium-4k-instruct 📄	33.12	64.23	49.38	16.99	11.52	13.05	40.84
💬	01-ai/Yi-1.5-34B-Chat 📄	33.08	60.67	44.26	23.34	15.32	13.06	39.12
💬	CohereForAI/c4ai-command-r-plus 📄	31.3	76.64	39.92	7.55	7.38	20.42	33.24
🔹	abacusai/Smaug-72B-v0.1 📄	29.98	51.7	42.42	17.75	9.62	15.39	40.46
🟢	Qwen/Qwen1.5-110B 📄	29.98	34.22	44.28	23.04	13.65	13.71	48.45
💬	Qwen/Qwen1.5-110B-Chat 📄	29.64	59.39	44.98	0	12.19	16.29	42.5
💬	microsoft/Phi-3-small-128k-instruct 📄	29.16	63.44	45.57	0	9.84	14.7	38.94

“如何解决使用模型时
碰到的问题？”

“等用上最新的模型，
问题就不用解决了”

Section 2

Prompt

Section 2

Use prompt to support multiple language

default:

return `Go step by step, read the content I provide, and perform the following actions:

Step 1: Summarize this context in one sentence;

Step 2: Expand the summarization and write an abstract of a few sentences;

Step 3: Additional bullet points highlighting the opinions conveyed in the context.

Please return the detailed notes you carefully wrote after reading the text repeatedly, answer in the following format:

Summary

<one sentence summary here>

Abstract

<abstract here>

Opinions

<additional bullet points highlighting the opinions here>

Your answer:

{answer}`;

case "ja":

case "ja-JP":

return `一度に一步步ずつ進めてみましょう。私が提供する内容を読んで、次の作業を慎重に実行してください。

ステップ 1: このコンテキストを 1 つの文に要約します。

ステップ 2: コンテキストの内容を要約し、要約を作成します。

ステップ 3: コンテキストの要点。

本文を繰り返し読んで丁寧に書いたメモを返却し、以下のマークダウン形式と日本語で解答してください。

一文でまとめると

<一文でまとめるとはこちら>

要約

<ここに要約>

要点

<要点はここにあります>

あなたの答えは:

{答え}`;

Section 2

Use prompt to create index

You are a large language AI assistant built by Lepton AI. You are given a user question, and please write clean, concise and accurate answer to the question. You will be given a set of related contexts to the question, each starting with a reference number like [[citation:x]], where x is a number. Please use the context and cite the context at the end of each sentence if applicable.

Your answer must be correct, accurate and written by an expert using an unbiased and professional tone. Please limit to 1024 tokens. Do not give any information that is not related to the question, and do not repeat. Say "information is missing on" followed by the related topic, if the given context do not provide sufficient information.

Please cite the contexts with the reference numbers, in the format [citation:x]. If a sentence comes from multiple contexts, please list all applicable citations, like [citation:3][citation:5]. Other than code and specific names and citations, your answer must be written in the same language as the question.

Here are the set of contexts:

{context}

Remember, don't blindly repeat the contexts verbatim. And here is the user question:

Who said live long and prosper?

[Rewrite](#)

Answer

The phrase "live long and prosper" is most famously associated with the character Mr. Spock, played by Jewish actor Leonard Nimoy, in the Star Trek TV series. ¹ Nimoy learned the hand gesture that accompanies this phrase, known as the Vulcan salute, from a Jewish religious phrase written by Theodore Sturgeon and is frequently used in the Star Trek universe. ⁵ Today, "live long and prosper" has transcended its original meaning and become a symbol of peace and goodwill. ⁶ ⁷ Rabbi Morrison, an Orthodox Jew, confirmed the origin of the salute from a Jewish religious perspective. He said "live long and prosper" in the context of the Vulcan salute.


Live Long And Prosper - Meaning & Origin ...




'Live long and prosper' is an abbreviated version of a traditional Jewish religious blessing. It came to a wider...

<https://www.phrases.org.uk/meanings/l...>


Sources


Live Long And Prosper ...
1 - www.phrases.or... 

The Jewish roots of Le...
2 - www.washingto... 

Vulcan salute - Wikipedia
3 - en.wikipedia.org 

The Origins of 11 F...
4 - www.mentalflo...

Live Long & Prosper: St...
5 - screenrant.com 

Live Long and Prosper!...
6 - usdictionary.com 

How Leonard Nimoy's J...
7 - www.startrek.com 


This is Why We 'Liv...
8 - www.startrek.c...

Related

Who first saw the Vulcan salute as a child?

What is the version of the blessing in the Priestly Blessing?

Ask Lepton AI anything ...

 Lepton AI Browser Extension



Math Tutor

Teach me a lesson on quadratic equations.



Math Worksheet Generator

Create a set of elementary math worksheets for math educators and parents.



Scavenger Hunt

Create a curated list of scavenger hunt concepts.



Unit Testing

Add unit tests for a Python function.



Geometry problem solving

Solve for X in an image.



Trip recommendations

Convert unorganized text into structured tables.



Time complexity

Identify the time complexity of a function and optimize it.



Opossum Search

Create a webpage based on the users specifications.



Recipe creator

Generate a custom recipe from a photo of what you want to eat.



Object identifier

Get a description of an object and its uses from a photo.



Marketing writer

Get catchy advertising copy tailored to your product and target audience.



List items from image

Get a list of objects in a photo.



Hurricane charting



Blog post creator



Barista Bot

- Developer Console
- Developer Discord
- Support

Get started

- Overview
- Quickstart
- Intro to Claude

Learn about Claude

- Use cases >
- Models
- Security and compliance ↗

Build with Claude

- Define success criteria
- Develop test cases
- Prompt engineering ▾

Overview

- Prompt generator

PROMPT ENGINEERING

Prompt engineering overview

Before prompt engineering

This guide assumes that you have:

1. A clear definition of the success criteria for your use case
2. Some ways to empirically test against those criteria
3. A first draft prompt you want to improve

If not, we highly suggest you spend time establishing that first. Check out [Define your success criteria](#) and [Create strong empirical evaluations](#) for tips and guidance.



Prompt generator

Don't have a first draft prompt? Try the prompt generator in the Anthropic Console!

Before prompt engineering

When to prompt engineer

How to prompt engineer

Prompt engineering tutorial

When to prompt engineer

This guide focuses on success criteria that are controllable through prompt engineering. Not every success criterion or failing eval is best solved by prompt engineering. For example, latency

Prompt Engineering

Introduction

LLM Settings

Basics of Prompting

Prompt Elements

General Tips for Designing Prompts

Examples of Prompts

Techniques

Zero-shot Prompting

Few-shot Prompting

Chain-of-Thought Prompting

Self-Consistency

Generate Knowledge Prompting

Prompt Chaining

Tree of Thoughts

Retrieval Augmented Generation

Automatic Reasoning and Tool-use

Prompt Engineering

Prompt Engineering Guide

Prompt engineering is a relatively new discipline for developing and optimizing prompts to efficiently use language models (LMs) for a wide variety of applications and research topics. Prompt engineering skills help to better understand the capabilities and limitations of large language models (LLMs).

Researchers use prompt engineering to improve the capacity of LLMs on a wide range of common and complex tasks such as question answering and arithmetic reasoning. Developers use prompt engineering to design robust and effective prompting techniques that interface with LLMs and other tools.

Prompt engineering is not just about designing and developing prompts. It encompasses a wide range of skills and techniques that are useful for interacting and developing with LLMs. It's an important skill to interface, build with, and understand capabilities of LLMs. You can use prompt engineering to improve safety of LLMs and build new capabilities like augmenting LLMs with domain knowledge and external tools.

Motivated by the high interest in developing with LLMs, we have created this new prompt engineering guide that contains all the latest papers, advanced prompting techniques, learning guides, model-specific prompting guides, lectures, references, new LLM capabilities, and tools related to prompt engineering.

Want to learn more?



We've partnered with Maven to deliver the following live cohort-based courses on prompt engineering:

- [LLMs for Everyone](#) (Beginner) - learn about the latest prompt engineering techniques and how to effectively apply them to real world use cases

On This Page

Want to learn more?

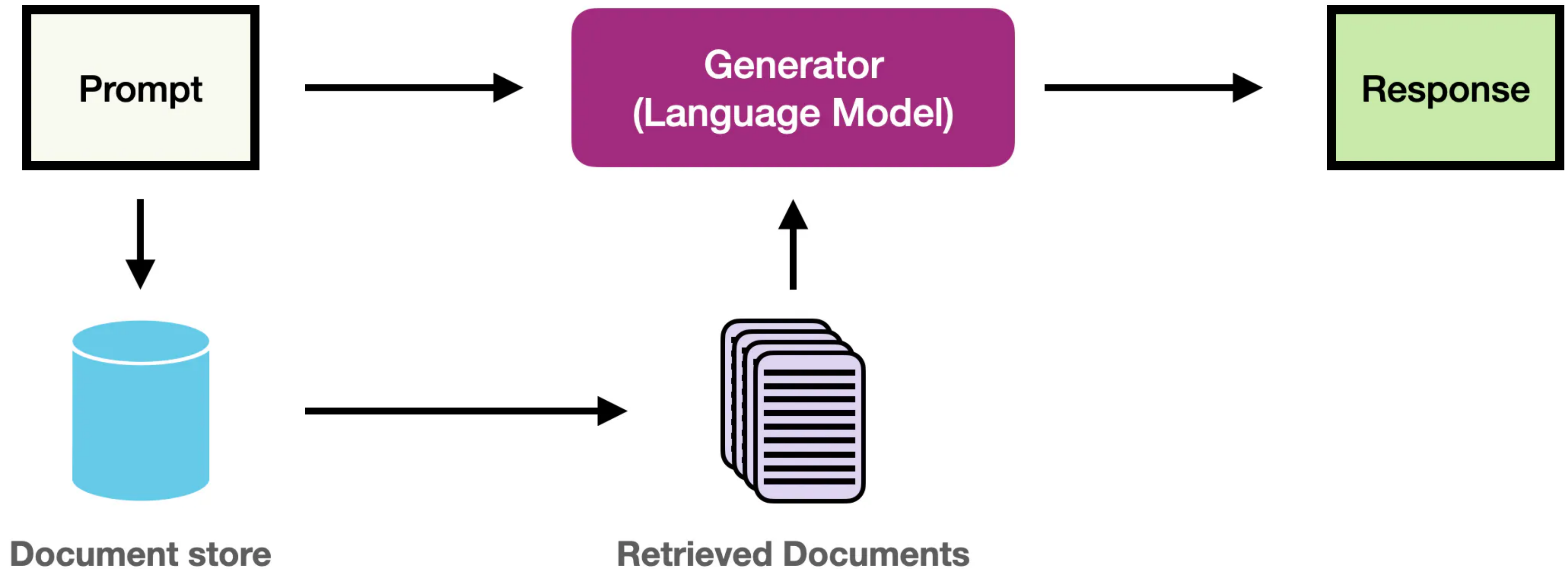
Question? Give us feedback →

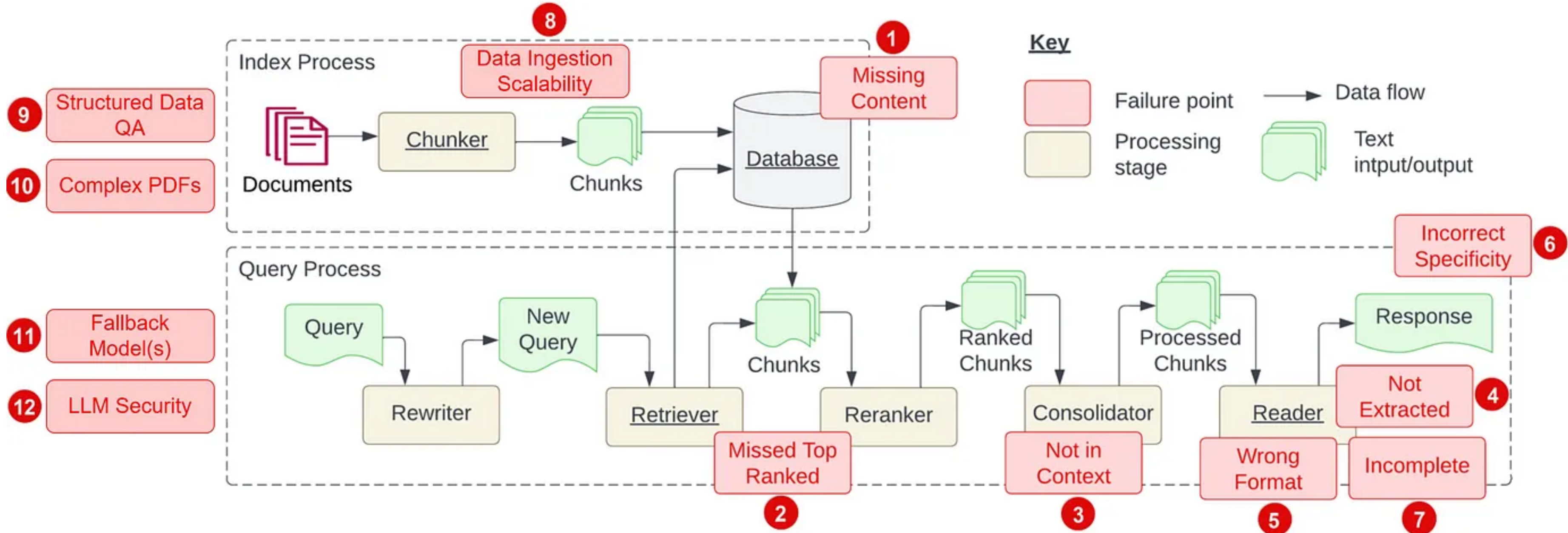
Edit this page

Section 3

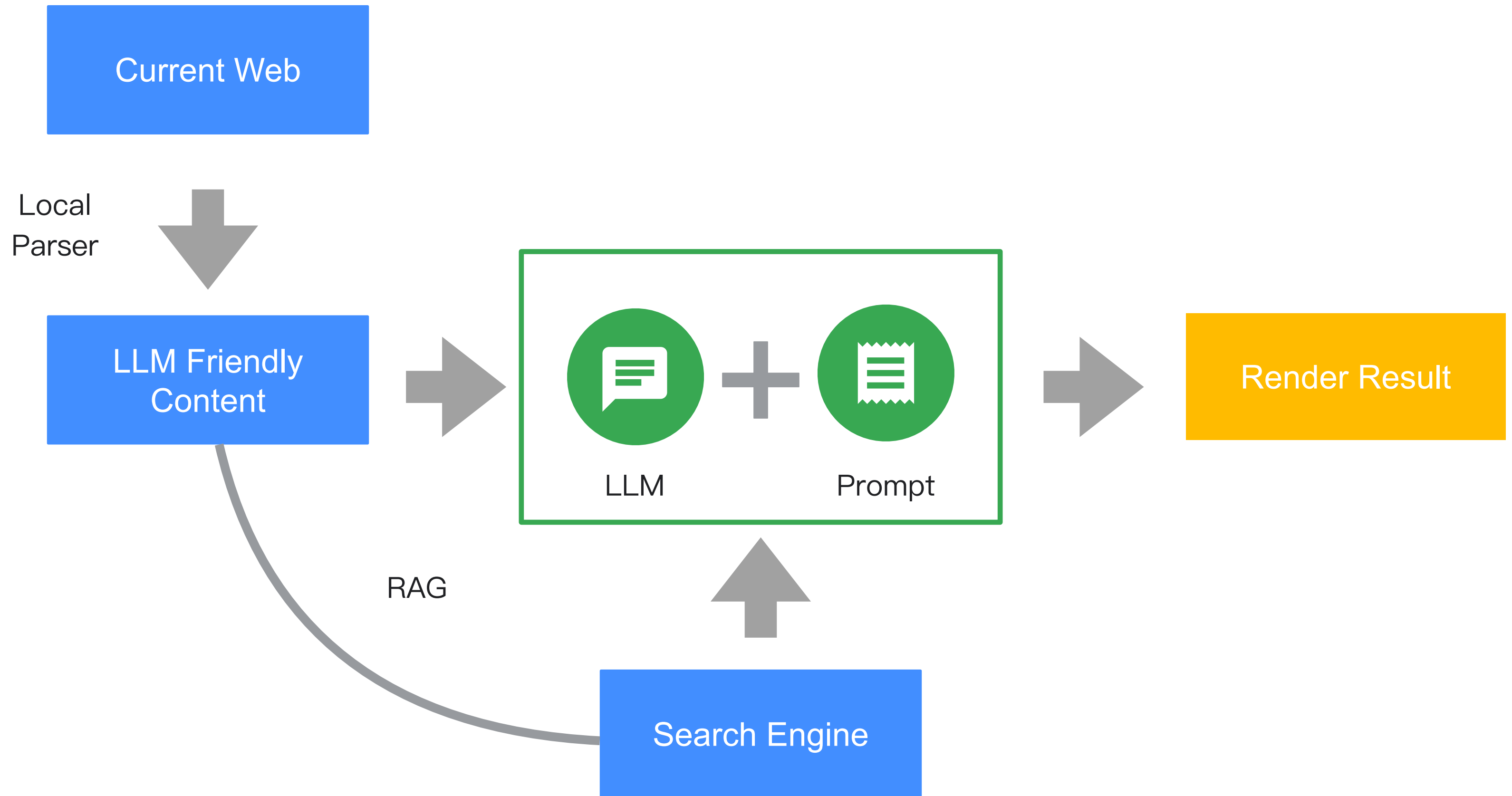
RAG

Retrieval Augmented Generation





Proposed Solutions		
1 Clean your data & Better prompting	5 Better prompting, output parsing, pydantic programs, & OpenAI JSON mode	9 Chain-of-table pack & Mix-self-consistency pack
2 Hyperparameter tuning & Reranking	6 Advanced retrieval strategies	10 Embedded table retrieval
3 Tweak retrieval strategies & Finetune embeddings	7 Query transformations	11 Neutrino router & OpenRouter
4 Clean your data, prompt compression, & LongContextReorder	8 Parallelizing ingestion pipeline	12 NeMo Guardrails & Llama Guard



Section3

Missing content

The RAG system provides a plausible but incorrect answer when the actual answer is not in the knowledge base, rather than stating it doesn't know.

Super long context

The correct answer is overlooked, causing the system to fail to deliver accurate responses.

Wrong source format

When an instruction to extract information in a specific format, like a image, table or list.

Wrong output format

The output format is not as expected

Unstable

What if your model runs into issues, such as rate limit errors with models.

Security

Prompt injection, handle insecure outputs, and prevent sensitive information disclosure

Build-in AI

```
const session = await window.ai.createTextSession();

const result = await session.prompt("Write me a poem");

console.log(result);
```

CHROME

Built-in AI Early Preview Program

Welcome and heads-up about the Prompt API

Authors

Kenji Bah
Thomas S
Alexandra

Contact

See [this se](#)

Last-updated


Jun 29, 20
See [change](#)

Intro

Welcome, and thank you for participating in our Early Preview Program for built-in AI capabilities ([Google I/O 2024](#)). Your involvement is invaluable as we explore opportunities to improve or augment experiences with AI!

The Built-in AI Early Preview Program has several goals:

- **Listening:** We're eager to hear your feedback on early-stage APIs. Help us better understand problems you're trying to solve. This will ensure that we design the APIs you need, in the right way.
- **Exploring:** We want to facilitate the discovery of promising applications for built-in AI, which will directly inform our roadmap and prioritization discussions. To this extent, we'll provide access to exploratory APIs such as the Prompt API.
- **Supporting:** Your insights and interest will inform discussions with other browser vendors towards common standards for AI integration in web browsers.



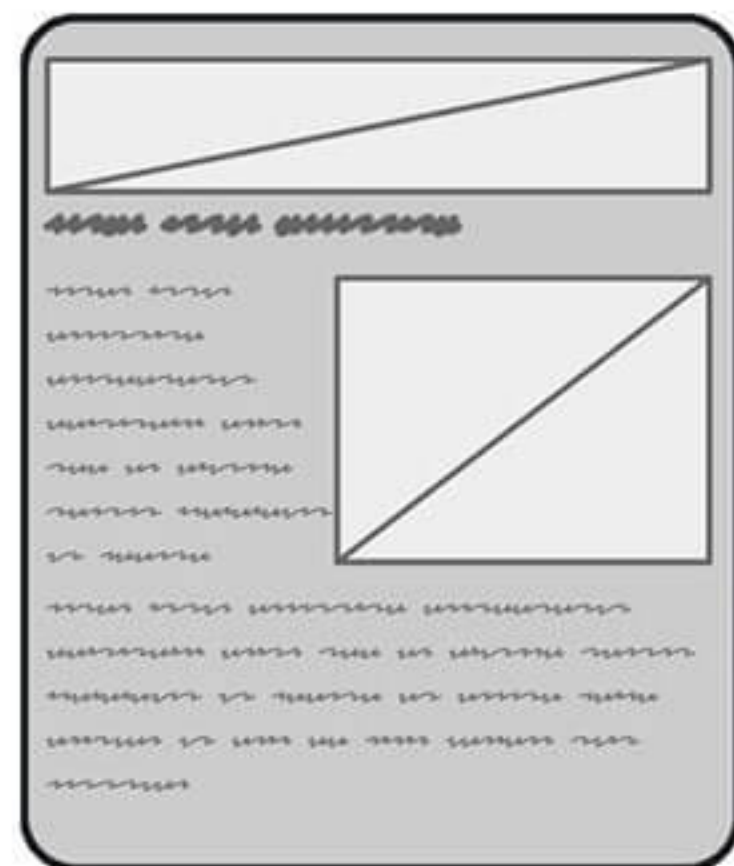
Know of other folks who would love to join this program? Perhaps you got access to this document from a friend?

Sign up to get the latest updates directly in your inbox.

In this first update, we're excited to provide details about the upcoming exploratory Prompt API, which will facilitate the discovery of AI use cases through local prototyping. More concretely, this API will let you use Gemini Nano on-device directly in your development environment.

 Your site or web app

 Chrome



Exploratory APIs
For local experiments and use case determination.

Gemini Nano

Task APIs
Purpose specific, pre-trained.

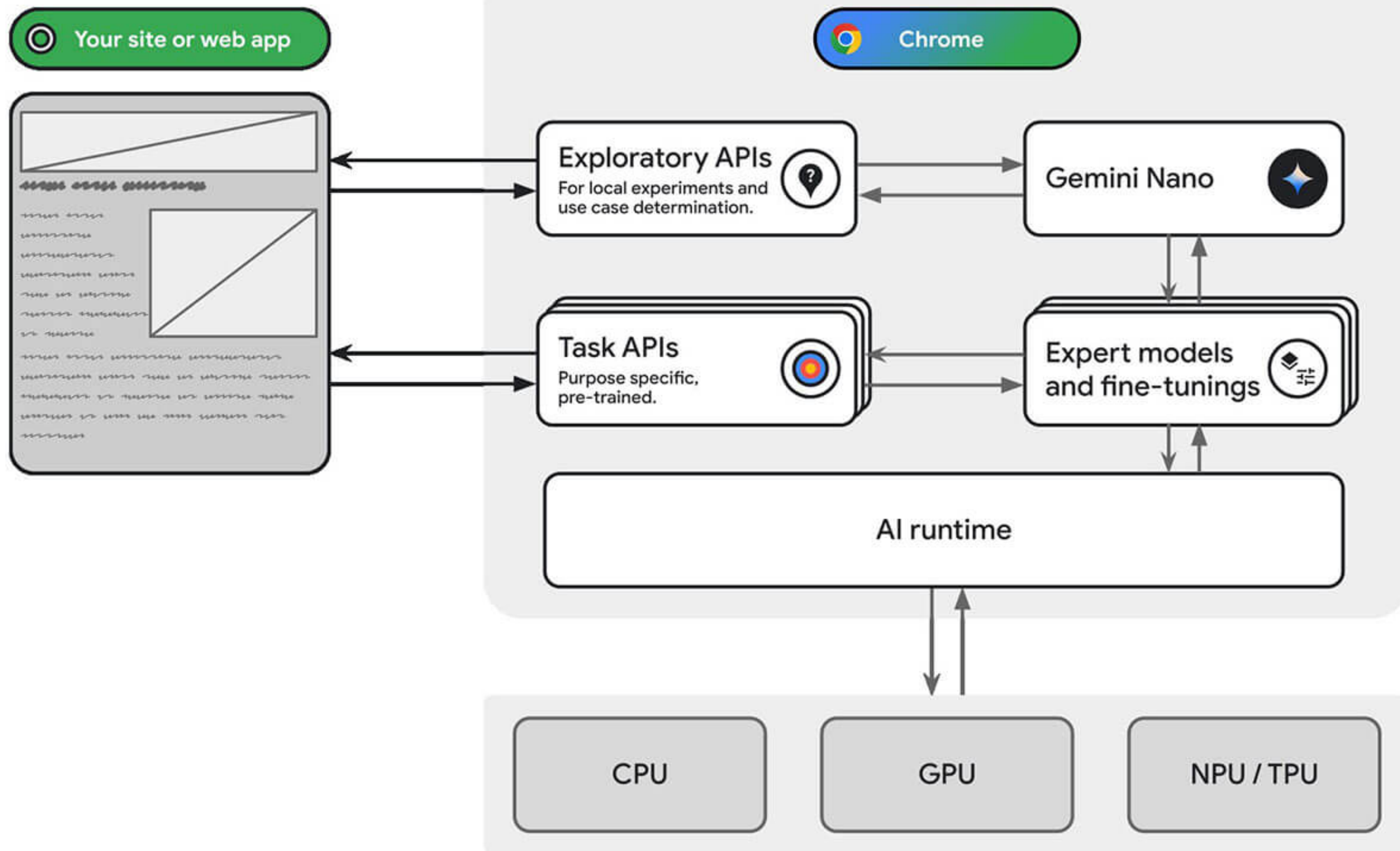
Expert models and fine-tunings

AI runtime

CPU

GPU

NPU / TPU



Web Translation API

For a known source language

If the source language is known, using the API looks like so:

```
const canTranslate = await translation.canTranslate({
  sourceLanguage: "en",
  targetLanguage: "ja"
});

if (canTranslate !== "no") {
  const translator = await translation.createTranslator({
    sourceLanguage: "en",
    targetLanguage: "ja"
  });

  console.assert(translator.sourceLanguage === "en");
  console.assert(translator.targetLanguage === "ja");

  const text = await translator.translate("Hello, world!");
  const readableStreamOfText = await translator.translateStreaming(`
    Four score and seven years ago our fathers brought forth, upon this...
  `);
} else {
  // Use alternate methods
}
```

Built-in AI

```
const session = await
window.ai.createTextSession();
```

```
const result = await session.prompt(`Translate
the following sentence into Chinese: "This
proposal is an early design sketch by the Chrome
built-in AI team to describe the problem below
and solicit feedback on the proposed solution. It
has not been approved to ship in Chrome."`);
console.log(result);
```

“该提议是由 Chrome 内部的人工智能组员早期设计 [VM125:4](#) 的草图，旨在描述以下问题并征求对提议解决方案的反馈。该提议尚未获得 Chrome 的批准以供发货。”

Thank You