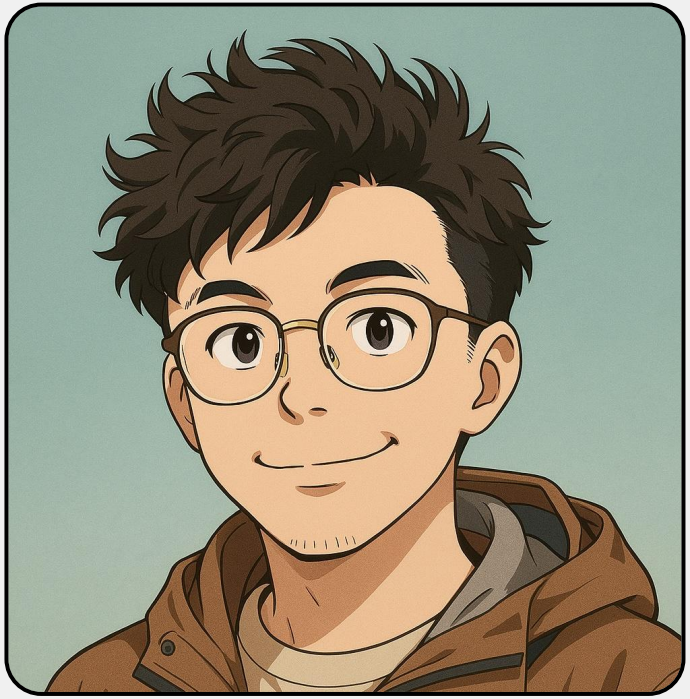
Google Developer Group
Guangzhou

Agents on Android

On-device LLMs and Intelligent Agents



EI Zhang (2BAB)

Android GDE / Tech Lead

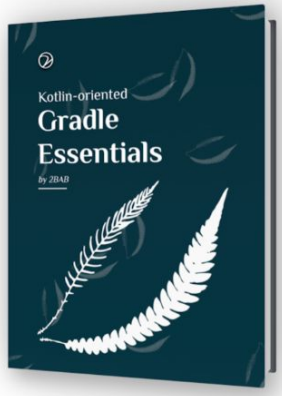
Build with AI

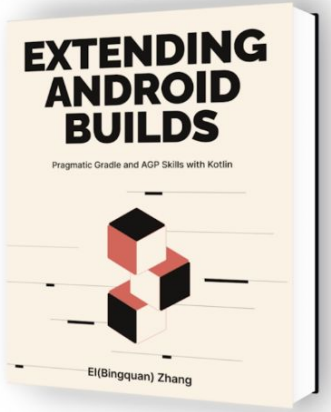
EI

- ✦ Google Developer Expert (GDE) for Android
- ✦ 9+ YOE on Mobile Infrastructures
- ✦ Enthusiast of AI, Build Tools (Gradle), Compiler (Kotlin & KSP), Kotlin Multiplatform
- ✦ Pursuer of Elegant UI/UX Effect
- ✦ Residing in Singapore 🇸🇬
- ✦ E-Mail: xx2bab@gmail.com

My Thoughts

Posts / Conference Talks / Podcasts

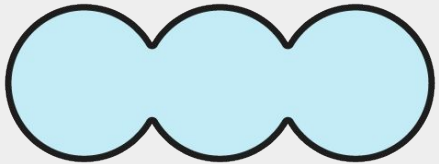


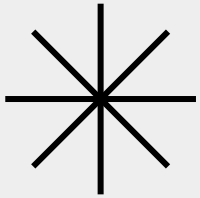


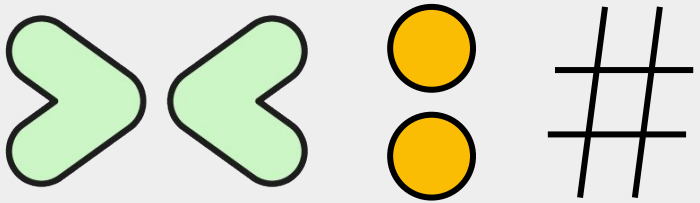
ToC

1. Recent Advances in Google's On-Device Models

简述小模型发展近况、推理框架、设备硬件的限制、模型 实测结果等





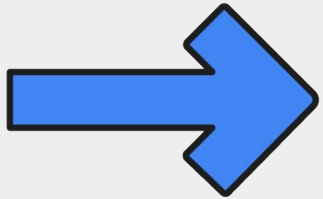


2. A Real-Estate Agent demo

Mini Codelab: 使用 On-Device Model 的 0 基础 0 框架 Agent 编写，包括 Prompts 优化、在线和离线的信息获取与 RAG 集成等。（需要电脑和 Android 设备）

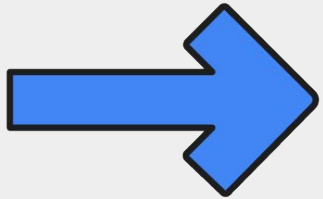
Notice

TL;DR
LLM 小模型重点是了解短处、发挥长处，本次分享内容偏前瞻探索。

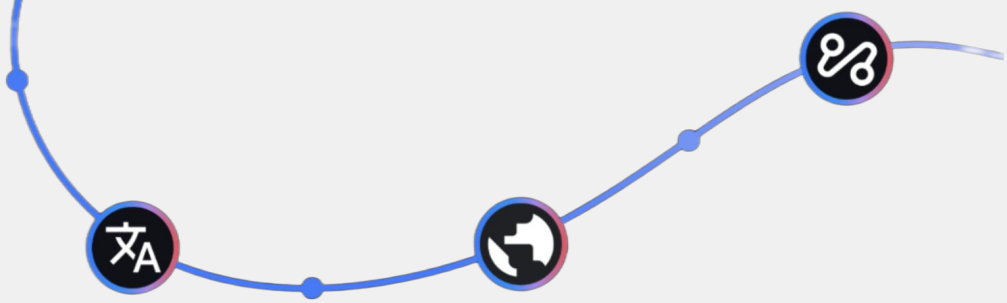


Notice

云端生产环境中的 LLM 小模型(<7B):
应用于文本分类、感情分析、质量打分、信息检索增强 RAG 等等;低功耗、低时延。



Gemma3 1B

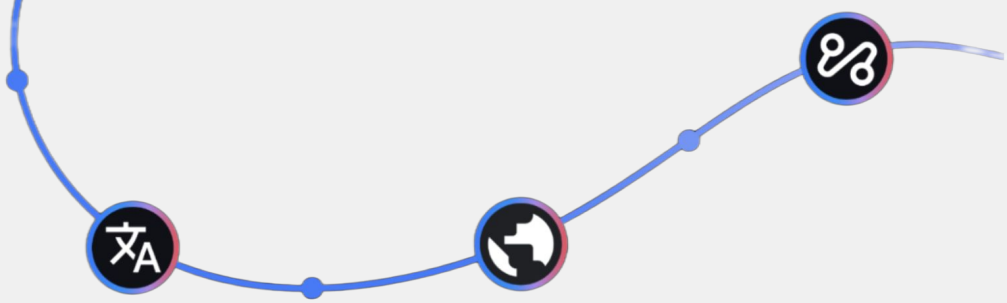


Small 687M memory footprint for q4_0

Optimized for strong english and multilingual performance

32k context length

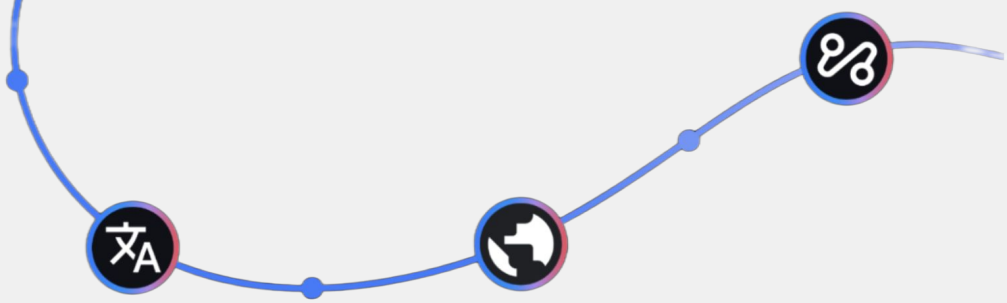
Gemma3 1B



Note that all benchmark stats are from a Samsung S24 Ultra with 2048 KV cache size, 1024 tokens prefill, 256 tokens decode.

Weight Quantization	Backend	Prefill (tokens/sec)	Decode (tokens/sec)	Time to first token (sec)	Model size (MB)	Peak RSS Memory (MB)	GPU Memory (MB)
dynamic_int4	CPU	322.5	47.4	3.1s	529	1138.31	-
dynamic_int4	GPU	2585.9	56.4	4.5s	529	1205.28	585.66

Gemini Nano 1.8B



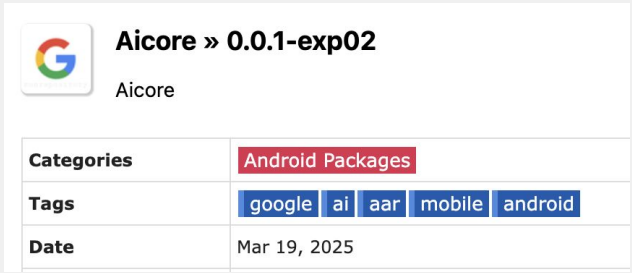
Approximately 1.2 GB memory usage

TPU/NPU acceleration

Approximately 4096 context length

Gemini Nano 1.8B

Based on 0.0.1-exp01
- Decode ~45t/s
- Max Tokens in Decoding ~250 tokens

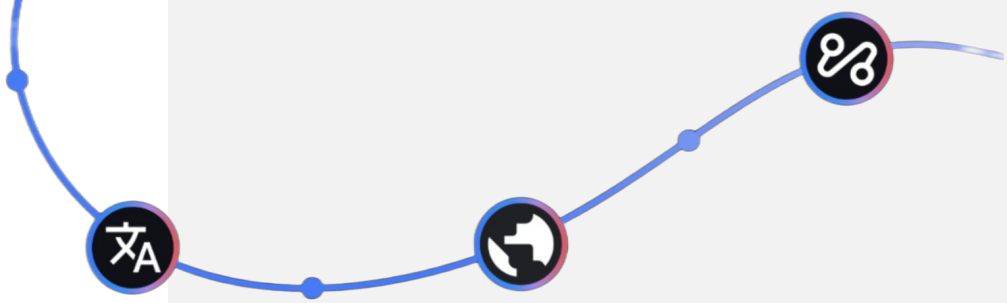


Aicore » 0.0.1-exp02

Aicore

Categories	Android Packages
Tags	google ai aar mobile android
Date	Mar 19, 2025

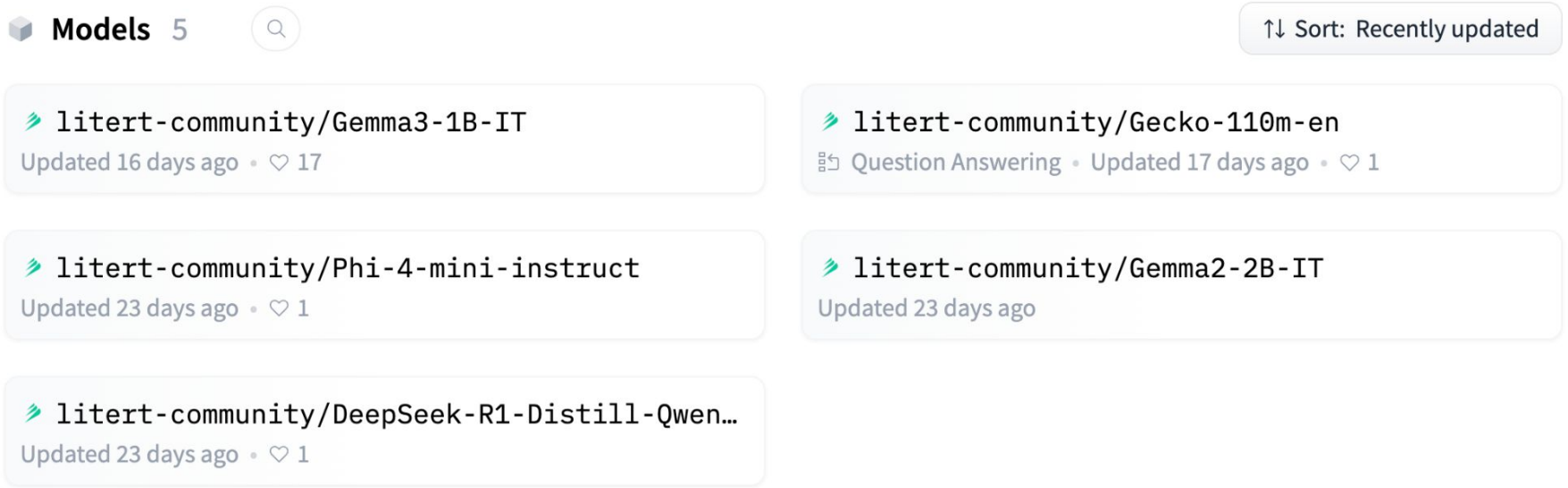
Google x On-Device Model



（On Mobile Devices）

Model	Inference Framework	Backend	OpenSource
Gemma (1/2/3) 1B/2B	**MediaPipe（LiteRT + XNNPack)**	CPU/GPU	Y
Gemini Nano	**AICore**	TPU/NPU	N

Actually, MediaPipe LLM Task supports more than one models...



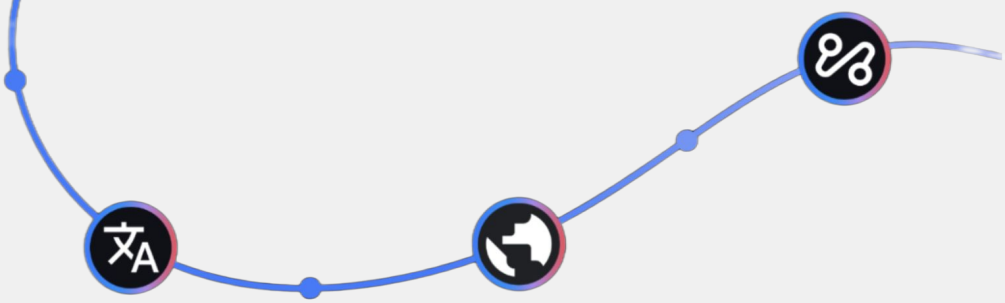
huggingface.co/litert-community

Deepseek-R1 1.5B

(distill-qwen)

Note that all benchmark stats are from a Samsung S24 Ultra with 1280 KV cache size, 512 tokens prefill, 128 tokens decode.

	Backend	Prefill (tokens/sec)	Decode (tokens/sec)	Time-to-first-token (sec)	Memory (RSS in MB)	Model size (MB)
fp32 (baseline)	CPU	45	6	8	6,213	7,124
dynamic_int8		261	23	2	1,936	1,861



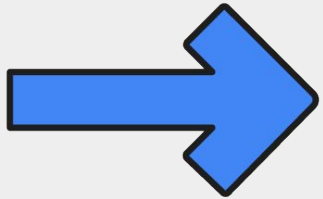
Gecko 110M en

(Embedder Model)

	Backend	Max sequence length	Init time (ms)	Inference time (ms)	Memory (RSS in MB)	Model size (MB)
dynamic_int8	CPU	512	30.04	353.1	225.6	120
dynamic_int8	CPU	1024	79.17	954	619.5	145

Notice

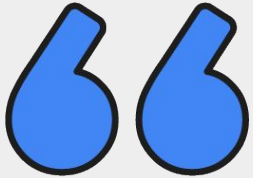
Simple Demos

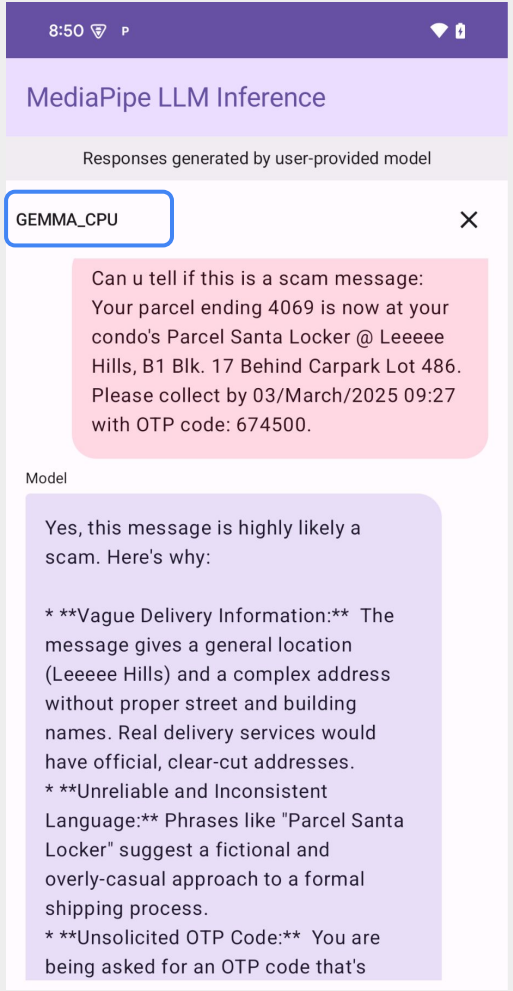


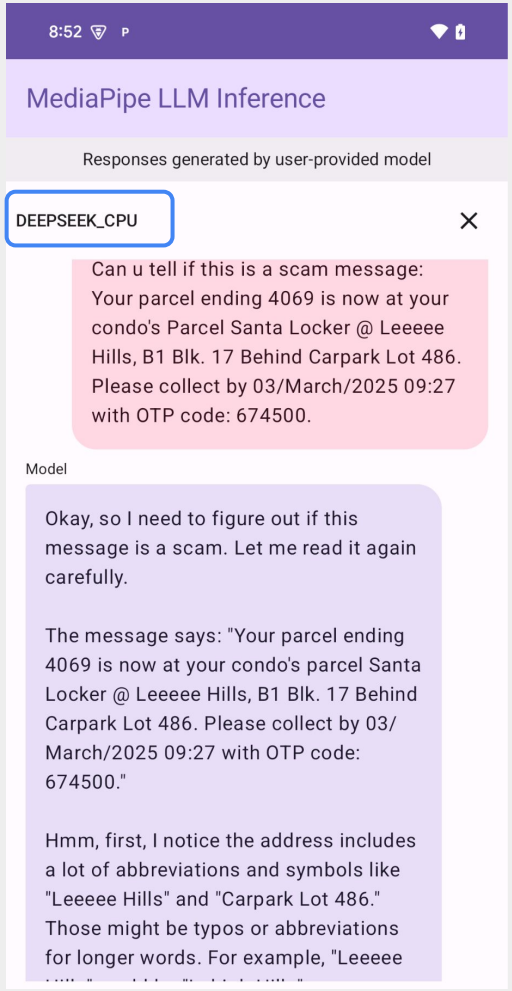
Gemma3 1B - Best practice: Text generation

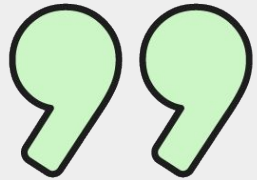
A 1B parameter is great at app context -> short text:

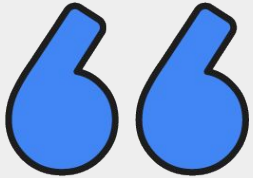
- Data Captioning
- Summarization
- Smart Reply/Detection
- Simple In-Game Dialog

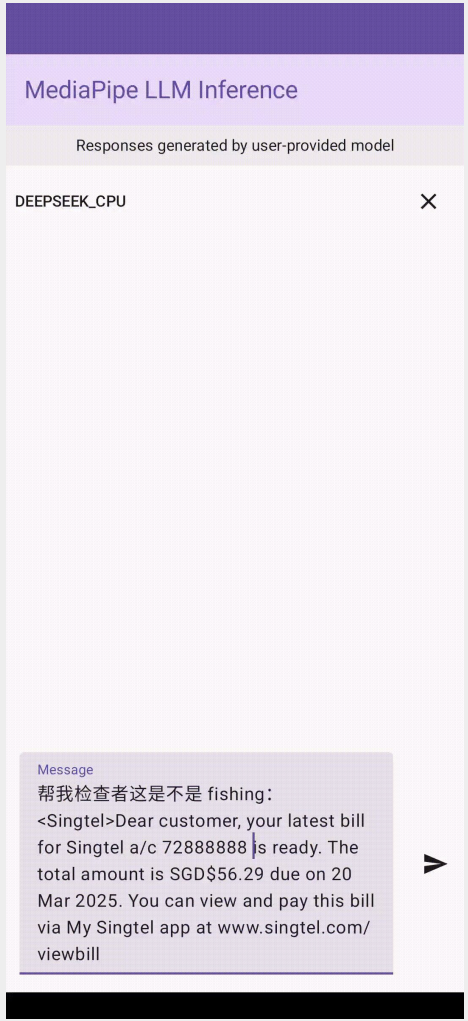


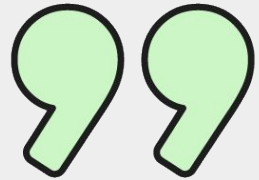

8:50
MediaPipe LLM Inference
Responses generated by user-provided model
GEMMA_CPU
Can u tell if this is a scam message: Your parcel ending 4069 is now at your condo's Parcel Santa Locker @ Leeeee Hills, B1 Blk. 17 Behind Carpark Lot 486. Please collect by 03/March/2025 09:27 with OTP code: 674500.
Model
Yes, this message is highly likely a scam. Here's why:
* **Vague Delivery Information:** The message gives a general location (Leeeee Hills) and a complex address without proper street and building names. Real delivery services would have official, clear-cut addresses.
* **Unreliable and Inconsistent Language:** Phrases like "Parcel Santa Locker" suggest a fictional and overly-casual approach to a formal shipping process.
* **Unsolicited OTP Code:** You are being asked for an OTP code that's


8:52
MediaPipe LLM Inference
Responses generated by user-provided model
DEEPSEEK_CPU
Can u tell if this is a scam message: Your parcel ending 4069 is now at your condo's Parcel Santa Locker @ Leeeee Hills, B1 Blk. 17 Behind Carpark Lot 486. Please collect by 03/March/2025 09:27 with OTP code: 674500.
Model
Okay, so I need to figure out if this message is a scam. Let me read it again carefully.
The message says: "Your parcel ending 4069 is now at your condo's parcel Santa Locker @ Leeeee Hills, B1 Blk. 17 Behind Carpark Lot 486. Please collect by 03/March/2025 09:27 with OTP code: 674500."
Hmm, first, I notice the address includes a lot of abbreviations and symbols like "Leeeee Hills" and "Carpark Lot 486." Those might be typos or abbreviations for longer words. For example, "Leeeee



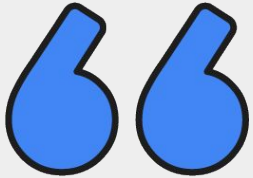


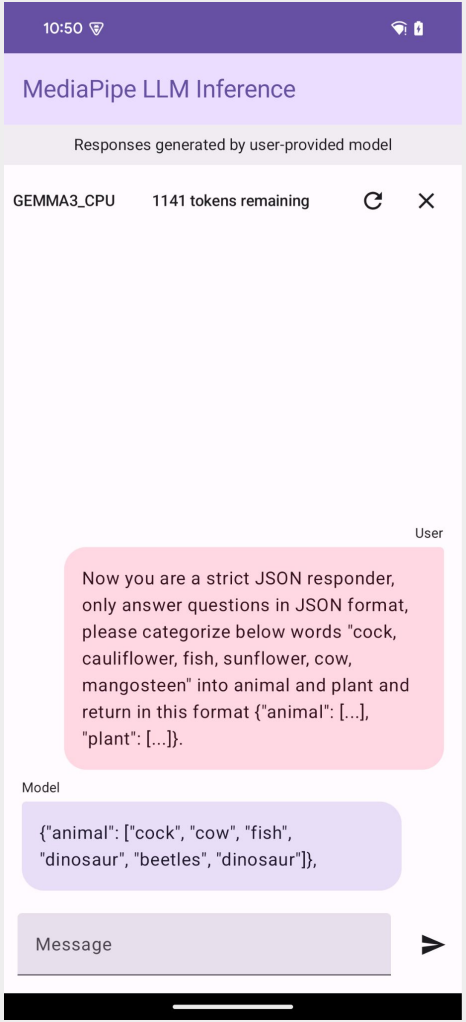

MediaPipe LLM Inference
Responses generated by user-provided model
DEEPSEEK_CPU
Message
帮我检查者这是不是 fishing：
<Singtel>Dear customer, your latest bill for Singtel a/c 72888888 is ready. The total amount is SGD$56.29 due on 20 Mar 2025. You can view and pay this bill via My Singtel app at www.singtel.com/viewbill

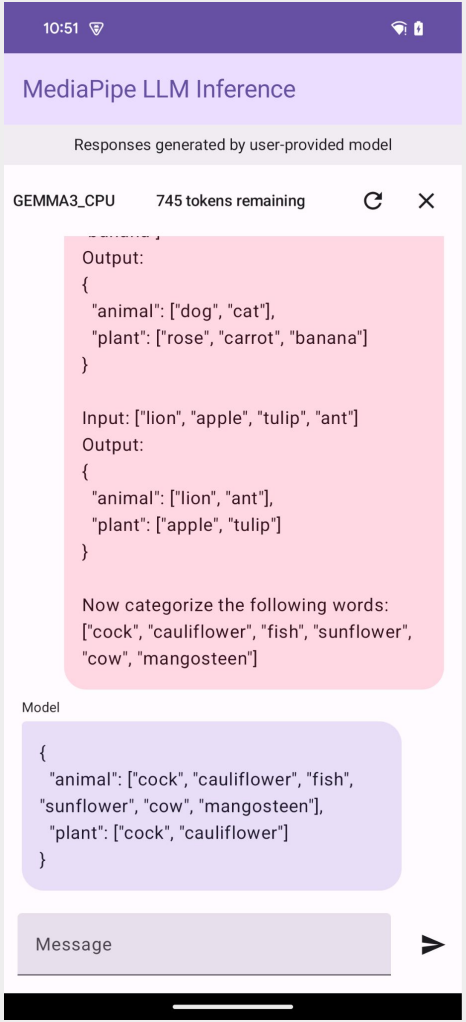


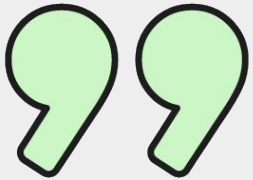
Best practice: Customize Gemma3 1B

- Few shot examples
- Fine-tune (LoRA or SFT)
- Use Retrieval Augmented Generation (RAG) to consume more content

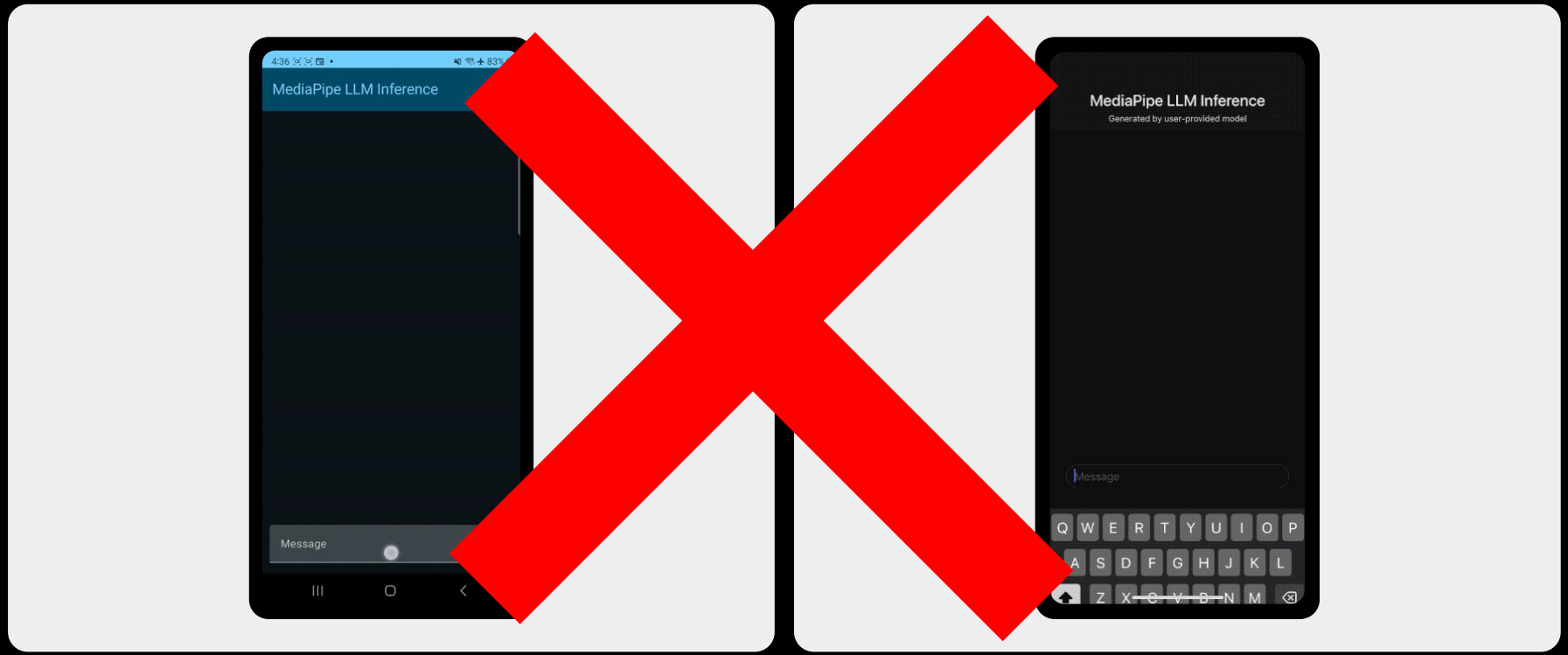



10:50
MediaPipe LLM Inference
Responses generated by user-provided model
GEMMA3_CPU
1141 tokens remaining
User
Now you are a strict JSON responder, only answer questions in JSON format, please categorize below words "cock, cauliflower, fish, sunflower, cow, mangosteen" into animal and plant and return in this format {"animal": [...], "plant": [...]}.
Model
{"animal": ["cock", "cow", "fish", "dinosaur", "beetles", "dinosaur"]},
Message

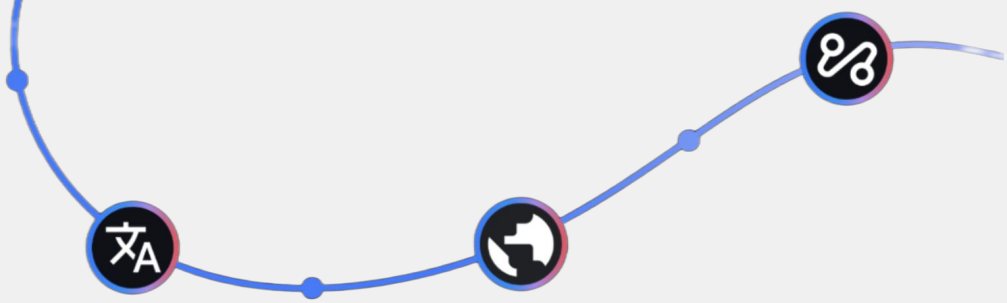

10:51
MediaPipe LLM Inference
Responses generated by user-provided model
GEMMA3_CPU
745 tokens remaining
Output:
{
"animal": ["dog", "cat"],
"plant": ["rose", "carrot", "banana"]
}
Input: ["lion", "apple", "tulip", "ant"]
Output:
{
"animal": ["lion", "ant"],
"plant": ["apple", "tulip"]
}
Now categorize the following words: ["cock", "cauliflower", "fish", "sunflower", "cow", "mangosteen"]
Model
{
"animal": ["cock", "cauliflower", "fish", "sunflower", "cow", "mangosteen"],
"plant": ["cock", "cauliflower"]
}
Message



Worst practice: Gemma3 1B as chatbot



Results for illustrative purpose



当前的限制

(基于多次实际测试)

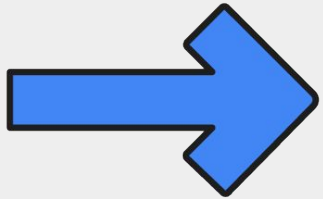
- 用 MediaPipe LLM Task 推理时 Context 大小在 **1500~2000 Tokens** 左右；
- 小模型的**注意力范围小**，适当运用 Prompt Engineering 时也要注意 Prompt 编写不可过长；
- 在未 Fine Tune 时，纯 Prompt 提示时 ReAct 的思考方式在 Gemma3 1B 上难以实现。

Note that all benchmark stats are from a Samsung S24 Ultra with 2048 KV cache size, 1024 tokens prefill, 256 tokens decode.

Note that all benchmark stats are from a Samsung S24 Ultra with 1280 KV cache size, 512 tokens prefill, 128 tokens decode.

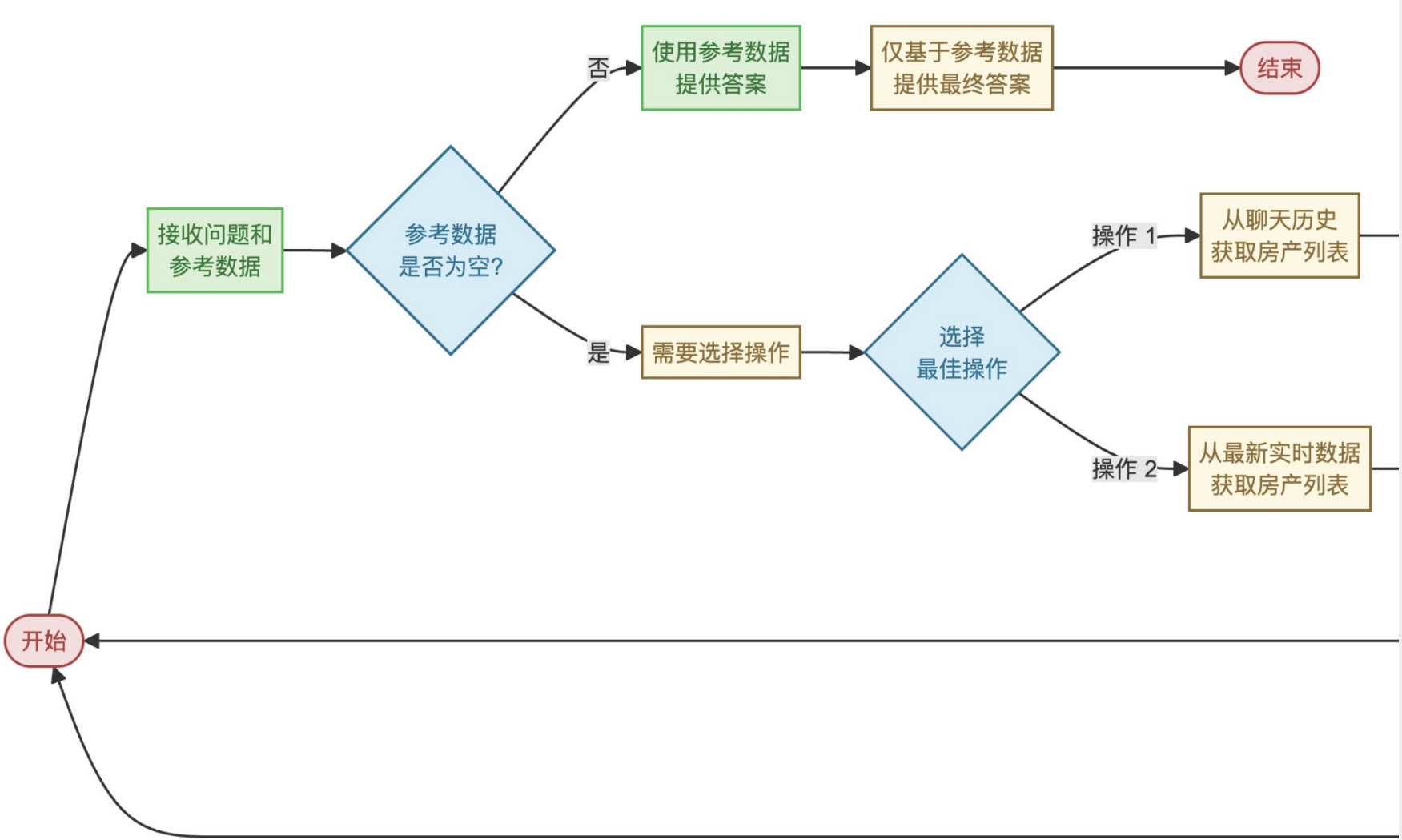
Notice

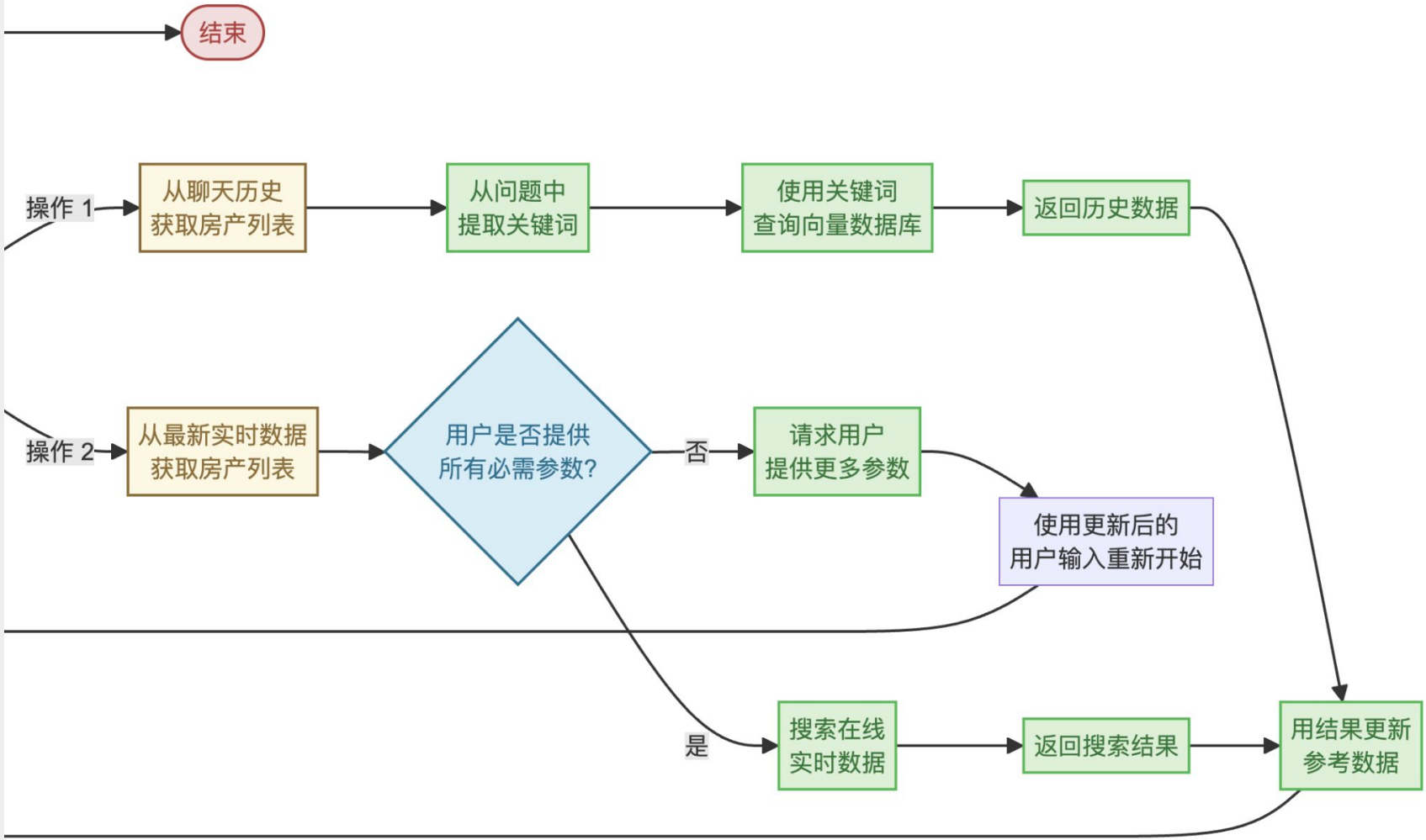
Mini Codelab



你是位房产经纪人。你有一个工具可以调用，需要参数：位置(必)，类型(必)，价格(必)，卧室(选)。

1. 用户输入：[question]
2. 缺参数？告知用户，结束。
3. 参数全？生成API调用：`{"action": "search", ...`
4. 等待API结果：`[{"name": "propertyA", "subway": "0.5", "price": "600k" ..]`
5. 分析：是否地铁近(subway)、便宜(price)、高层(floor)。
6. 评分：1-5，生成报告。
7. 如果有任何告知用户的操作，获取新的输入后回到开头。


开始
接收问题和
参考数据
参考数据
是否为空?
否
使用参考数据
提供答案
仅基于参考数据
提供最终答案
结束
是
需要选择操作
选择
最佳操作
操作 1
从聊天历史
获取房产列表
操作 2
从最新实时数据
获取房产列表


结束
操作 1
从聊天历史
获取房产列表
从问题中
提取关键词
使用关键词
查询向量数据库
返回历史数据
操作 2
从最新实时数据
获取房产列表
用户是否提供
所有必需参数?
否
请求用户
提供更多参数
使用更新后的
用户输入重新开始
是
搜索在线
实时数据
返回搜索结果
用结果更新
参考数据

1. **角色设定 (Role prompting)**：“You are a professional real estate agent…”
2. **显式指令 (Explicit instructions):** “Carefully extract … without including ANY numerical details.”
3. **强调和重复 (Instruction reinforcement):** “(NO numbers at all!)”
4. **示范式提示 (Few-shot prompting):** “here are 3 examples”
5. **分步拆解 (Task decomposition)**
6. **强调输出格式和约束 (Output constraints)**
7. **…**

README

OnDevice Agent Example

A prototype Android application demonstrating an on-device LLM agent for property search. The app uses MediaPipe for LLM inference and implements a chat-like interface for user interactions.

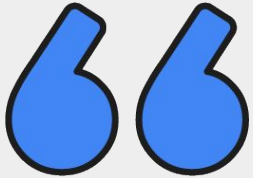
Features

- On-device LLM inference using MediaPipe
- Property search functionality
- Chat-like interface with:
  - Message types (System, User, Function, Result)
  - Message animations
  - Auto-scrolling to latest messages
  - Expandable message content
  - Clear conversation history
- Material 3 design with custom styling

Tech Stack

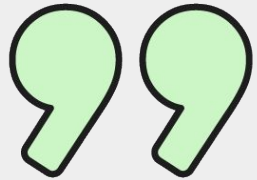
- Kotlin
- Jetpack Compose
- MediaPipe Tasks for LLM inference （Gemma3 1B/DS 1.5B)
- Koin for dependency injection
- Ktor for network requests
- Kotlin Coroutines & Flow

https://github.com/2BAB/OnDeviceAgent



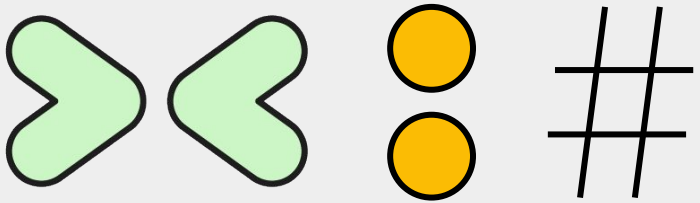
与 Android API 的结合：

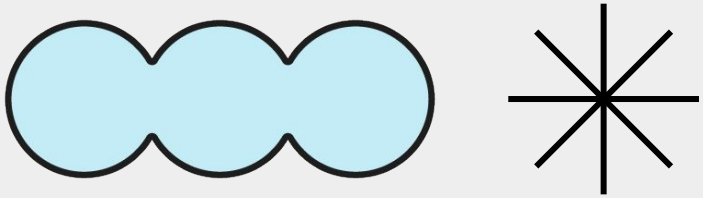
1. WorkManager (Schedule Task ~10 mins)
2. Foreground Service (Keep Alive)
3. Widgets (Keep Alive)



小结

1. 应用层基础能力均已具备，模型和硬件的升级、系统级的模型共享以及更开放的数据协议等都会带来新机会





2. 而系统级别的机会则包括 OEM 和 IoT 的方向都在起步期，或许可以更早得运用在部分垂直领域。

Google Developer Group
Editable Location

Thank you!

↓

Q&A

Build with AI