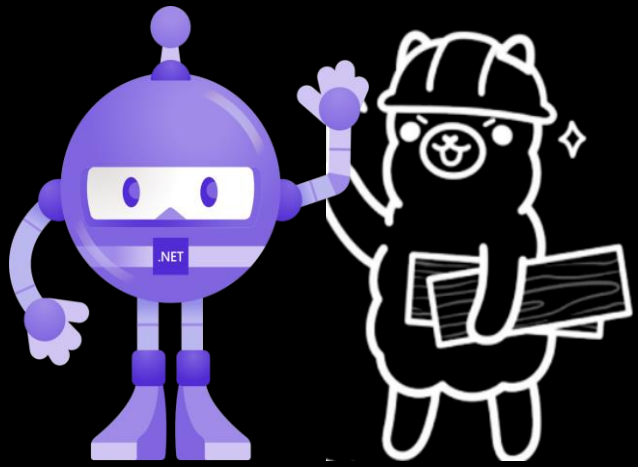
Microsoft Learn Student Ambassadors

Microsoft Research

Developing Local AI Models with .NET and AutoGen 🤖

Charunthon Limseelo (@boatchrnthn)
Microsoft Learn Student Ambassador | Thai Student Tech Community Lead

.NET Conf Thailand at Seven Peaks Software (November 23rd)



Charunthon Limseelo (Boat)

Beta Microsoft Learn Student Ambassadors (Thai Student Tech Lead) and Google Developer Group (GDG Cloud Bangkok) Member

+ Microsoft Office Specialist (Excel)
+ Open-source small AI and NLP Interest, with BDE Applications for business aspect
+ Applied Skills Challenger (Azure AI Document Intelligence and NLP)

Charunthon Limseelo

@boatchrnthn

Charunthon Limseelo

Boat Charunthon (boatchrnthn)

Acknowledgement To All AI Leads and Specialists

Engineering, Research-Based, and Practical Leads

Business, Financial, Commercial-based, and Daily Users

Proprietary

(adj.) used, made, or marketed by one having the exclusive legal right, privately owned and managed and run as a **profit-making** organization, Close-Source

Open-Source

(adj.) having the source code **freely available** for possible modification and redistribution, along with publicly available for use by the community at large

Bad gateway

The web server reported a bad gateway error.

OpenAI

SUBSCRIBE TO UPDATES

ChatGPT is unavailable for some users. Subscribe

Investigating - We are currently investigating this issue.

Jun 04, 2024 - 07:33 PDT

Uptime over the past 90 days. View historical uptime.

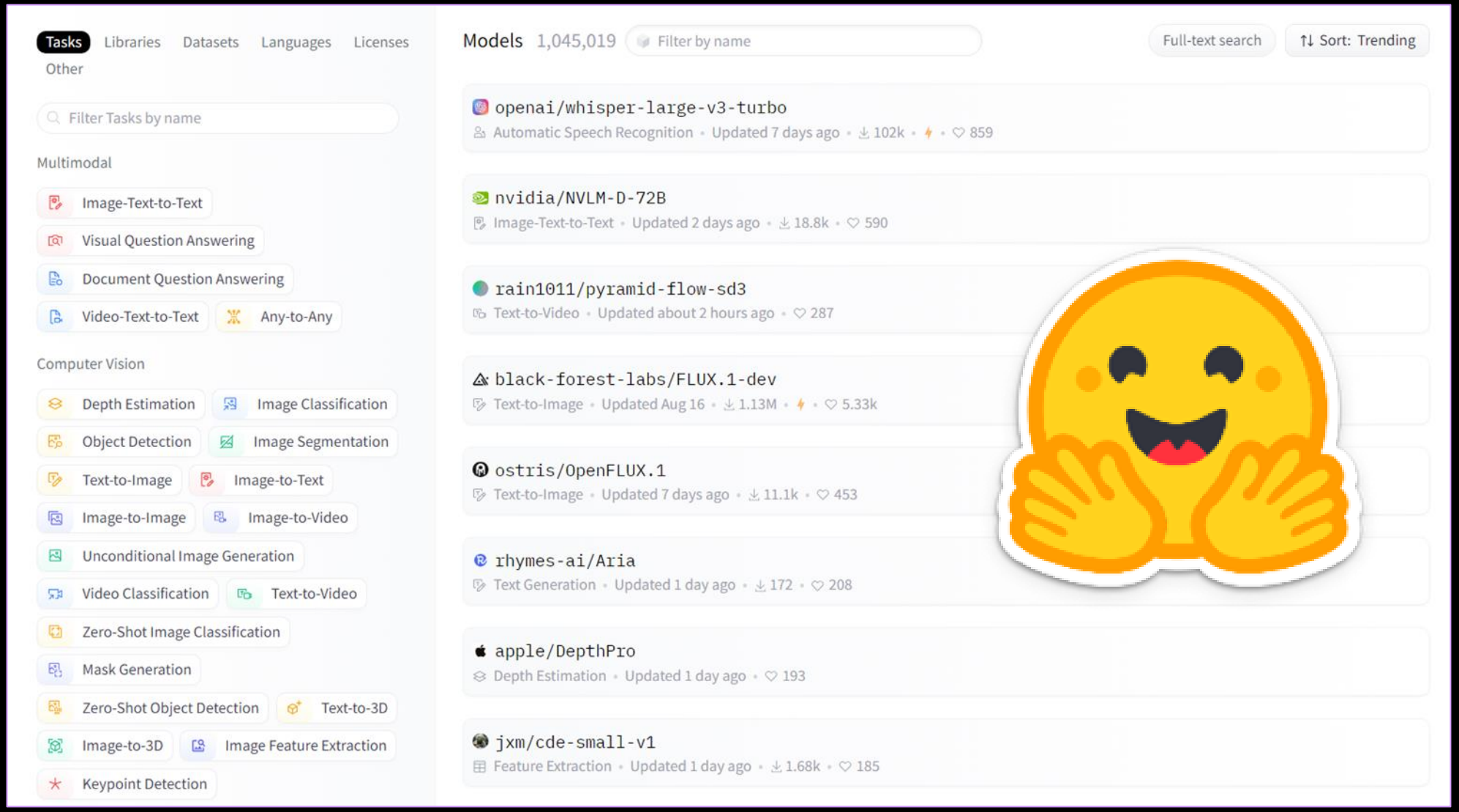
API ? Operational

90 days ago — 99.93 % uptime — Today

ChatGPT ? Major Outage

90 days ago — 99.42 % uptime — Today

Labs ? Operational



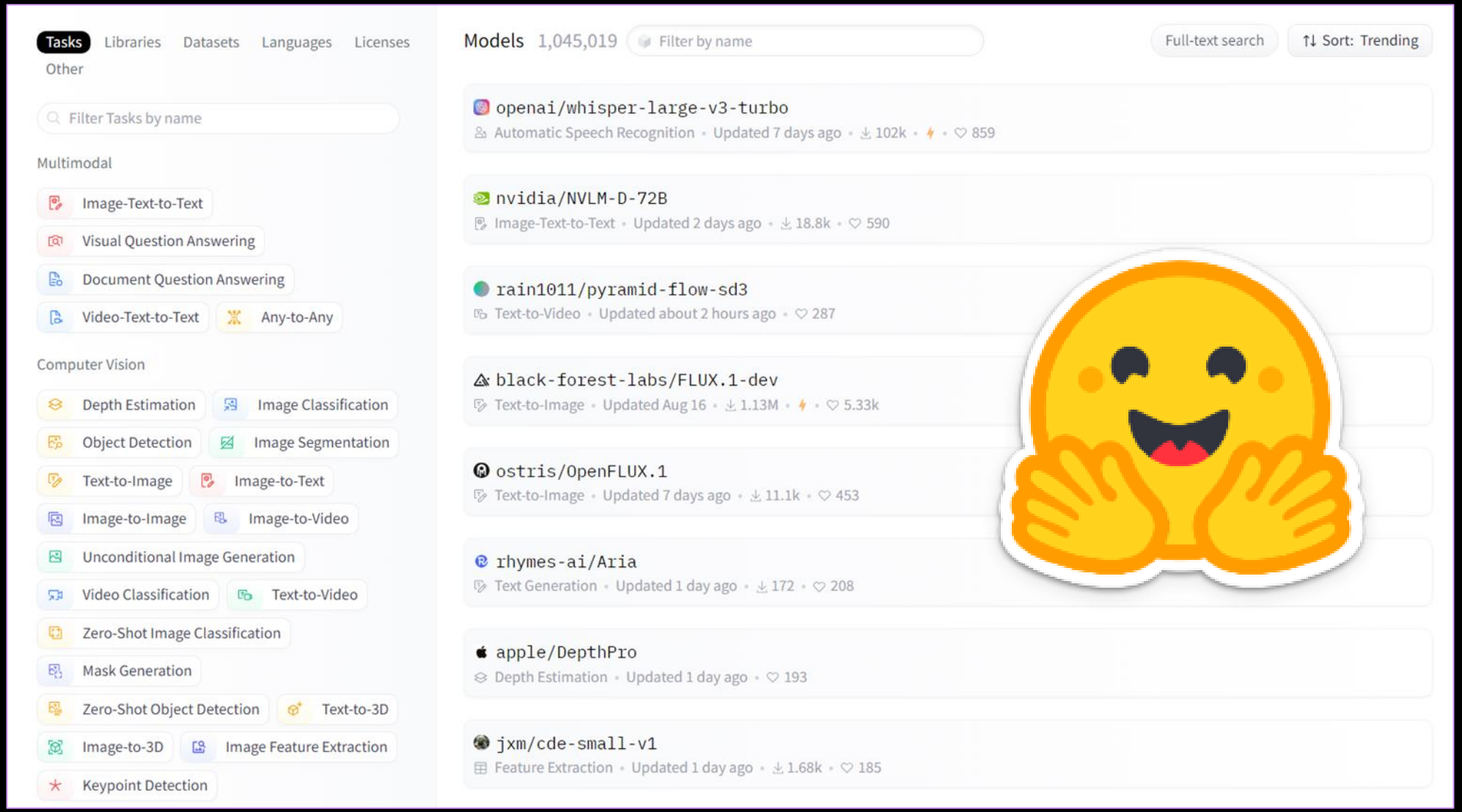
Data Leak

Server's Down

No Internet Connection

Data Centers' Power Consumption

Prevent from Paying Premium Services



Found out some LLMs are too big to download, need to use API to connect (Require internet)

I have an interesting way to prevent all these problems for using each device offline!

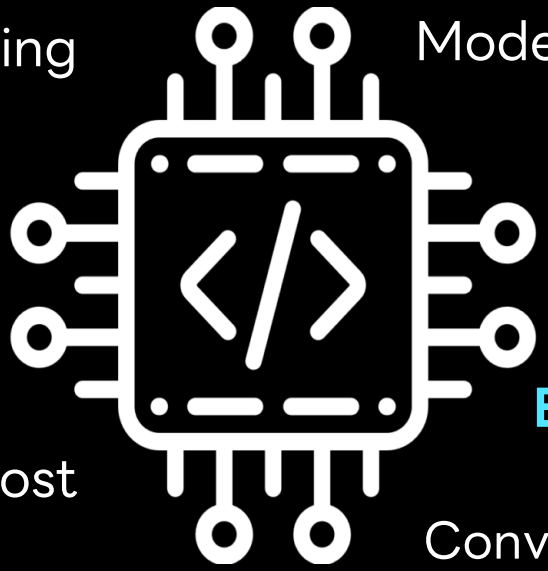
Small Language Models **(SLMs)**

What is Small Language Model?

Use in offline mode

Local Hosting

Model Drift



Inference Time

Rate Limits

Data Privacy and Protection

Lower Resource Requirements

Ease of Development

Token Usage and Cost

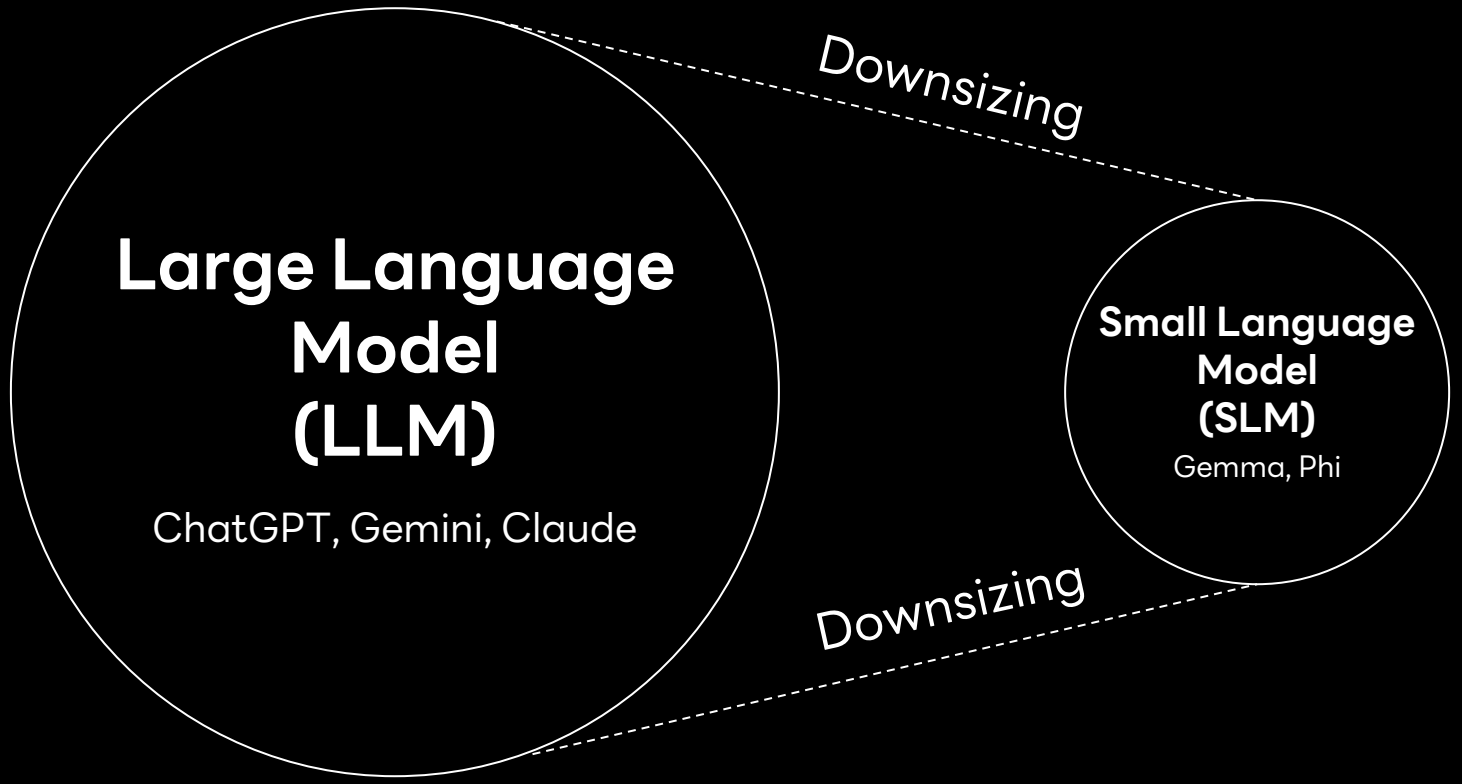
Conversation & Context Management

Natural Language Generation

Knowledge Intensive NPU

And Environmental Preservation 🌿

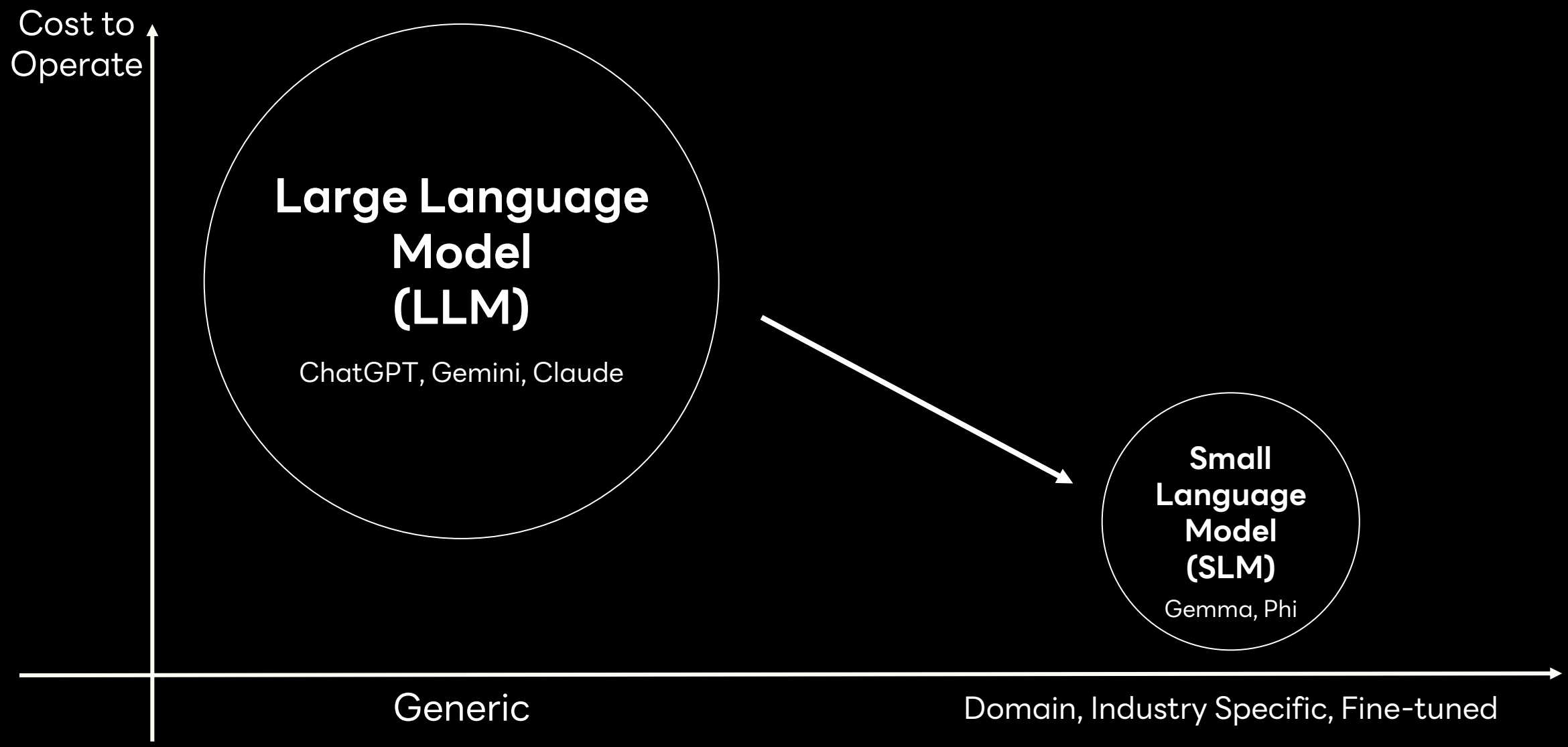
Language Model Comparison



- Ten of billions in parameters
- Requires substantial computational power for training and development
- Higher performance in broader and more complex tasks

- Millions to few billions in parameters **(should be lower than 7B)**
- Capable of being trained with consumer GPUs and lower budgets
- Effective for specific and narrow tasks

Language Model Comparison



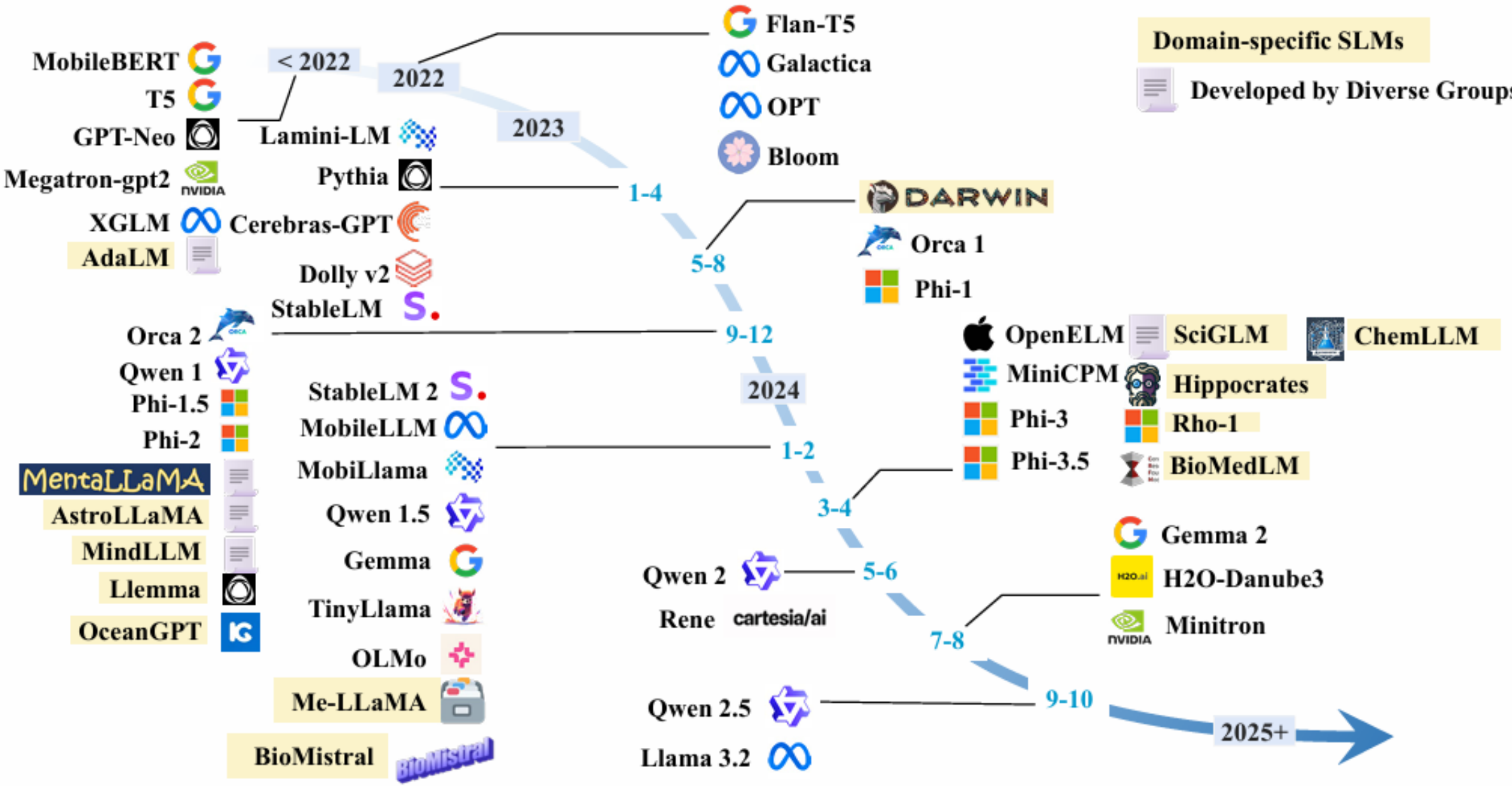
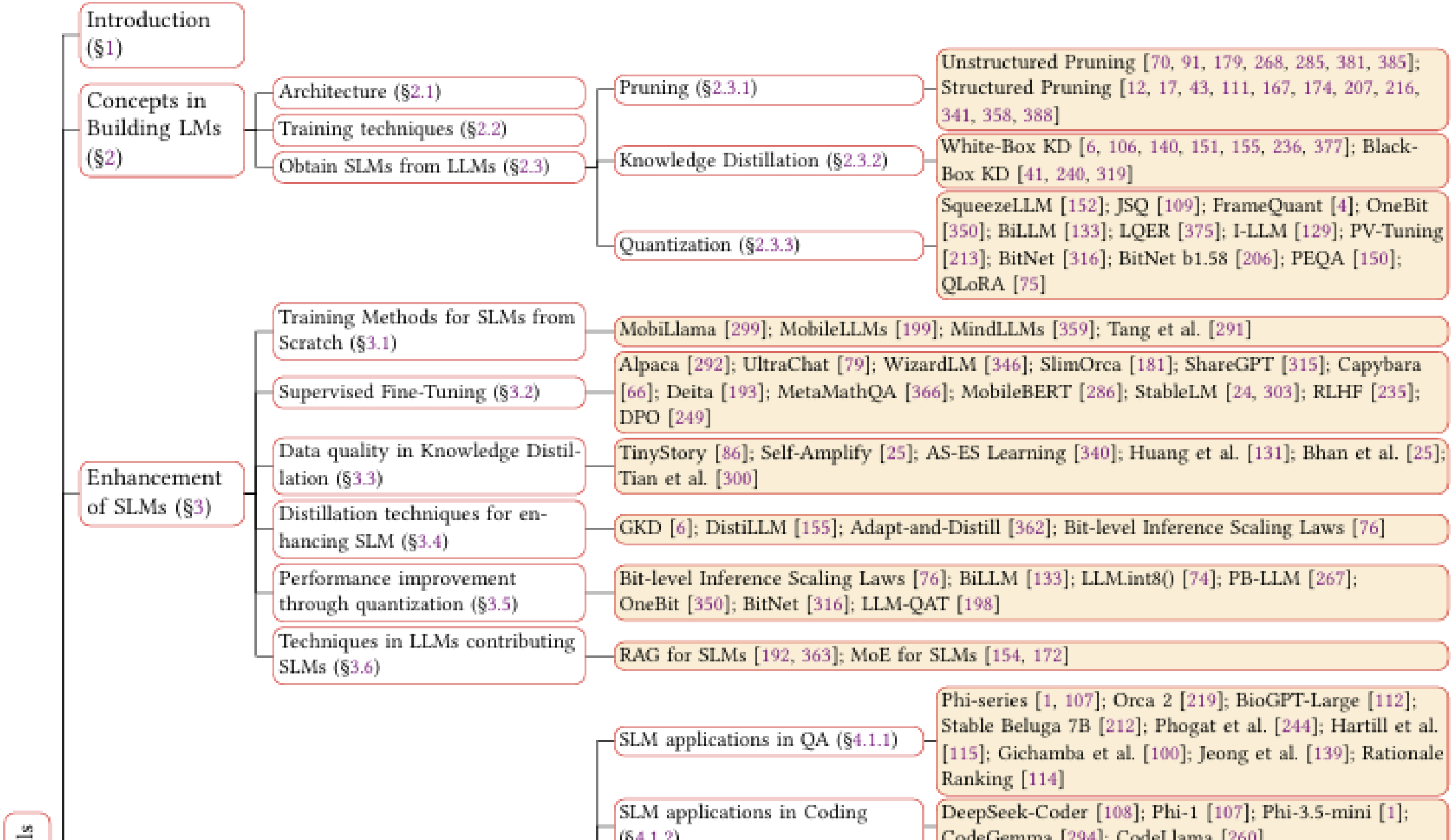
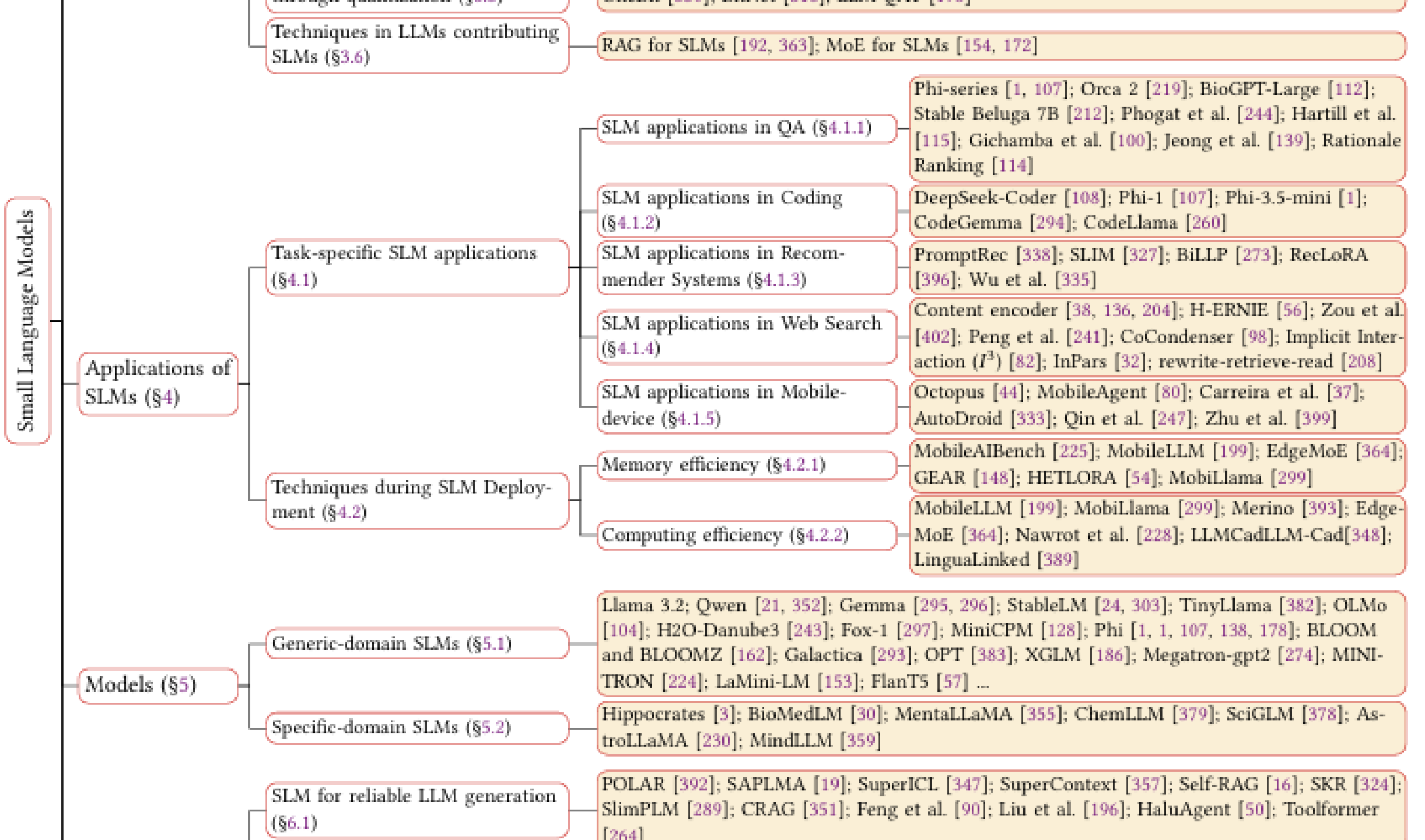


Fig. 3. A timeline of existing small language models.


Introduction (§1)
Concepts in Building LMs (§2)
Architecture (§2.1)
Training techniques (§2.2)
Obtain SLMs from LLMs (§2.3)
Pruning (§2.3.1)
Unstructured Pruning [70, 91, 179, 268, 285, 381, 385]; Structured Pruning [12, 17, 43, 111, 167, 174, 207, 216, 341, 358, 388]
Knowledge Distillation (§2.3.2)
White-Box KD [6, 106, 140, 151, 155, 236, 377]; Black-Box KD [41, 240, 319]
Quantization (§2.3.3)
SqueezeLLM [152]; JSQ [109]; FrameQuant [4]; OneBit [350]; BiLLM [133]; LQER [375]; I-LLM [129]; PV-Tuning [213]; BitNet [316]; BitNet b1.58 [206]; PEQA [150]; QLoRA [75]
Enhancement of SLMs (§3)
Training Methods for SLMs from Scratch (§3.1)
MobiLlama [299]; MobileLLMs [199]; MindLLMs [359]; Tang et al. [291]
Supervised Fine-Tuning (§3.2)
Alpaca [292]; UltraChat [79]; WizardLM [346]; SlimOrca [181]; ShareGPT [315]; Capybara [66]; Deita [193]; MetaMathQA [366]; MobileBERT [286]; StableLM [24, 303]; RLHF [235]; DPO [249]
Data quality in Knowledge Distillation (§3.3)
TinyStory [86]; Self-Amplify [25]; AS-ES Learning [340]; Huang et al. [131]; Bhan et al. [25]; Tian et al. [300]
Distillation techniques for enhancing SLM (§3.4)
GKD [6]; DistiLLM [155]; Adapt-and-Distill [362]; Bit-level Inference Scaling Laws [76]
Performance improvement through quantization (§3.5)
Bit-level Inference Scaling Laws [76]; BiLLM [133]; LLM.int8() [74]; PB-LLM [267]; OneBit [350]; BitNet [316]; LLM-QAT [198]
Techniques in LLMs contributing SLMs (§3.6)
RAG for SLMs [192, 363]; MoE for SLMs [154, 172]
SLM applications in QA (§4.1.1)
Phi-series [1, 107]; Orca 2 [219]; BioGPT-Large [112]; Stable Beluga 7B [212]; Phogat et al. [244]; Hartill et al. [115]; Gichamba et al. [100]; Jeong et al. [139]; Rationale Ranking [114]
SLM applications in Coding
DeepSeek-Coder [108]; Phi-1 [107]; Phi-3.5-mini [1];


Small Language Models
Applications of SLMs (§4)
Task-specific SLM applications (§4.1)
SLM applications in QA (§4.1.1)
Phi-series [1, 107]; Orca 2 [219]; BioGPT-Large [112]; Stable Beluga 7B [212]; Phogat et al. [244]; Hartill et al. [115]; Gichamba et al. [100]; Jeong et al. [139]; Rationale Ranking [114]
SLM applications in Coding (§4.1.2)
DeepSeek-Coder [108]; Phi-1 [107]; Phi-3.5-mini [1]; CodeGemma [294]; CodeLlama [260]
SLM applications in Recommender Systems (§4.1.3)
PromptRec [338]; SLIM [327]; BiLLP [273]; RecLoRA [396]; Wu et al. [335]
SLM applications in Web Search (§4.1.4)
Content encoder [38, 136, 204]; H-ERNIE [56]; Zou et al. [402]; Peng et al. [241]; CoCondenser [98]; Implicit Interaction (I^3) [82]; InPars [32]; rewrite-retrieve-read [208]
SLM applications in Mobile-device (§4.1.5)
Octopus [44]; MobileAgent [80]; Carreira et al. [37]; AutoDroid [333]; Qin et al. [247]; Zhu et al. [399]
Techniques during SLM Deployment (§4.2)
Memory efficiency (§4.2.1)
MobileAIBench [225]; MobileLLM [199]; EdgeMoE [364]; GEAR [148]; HETLORA [54]; MobiLlama [299]
Computing efficiency (§4.2.2)
MobileLLM [199]; MobiLlama [299]; Merino [393]; EdgeMoE [364]; Nawrot et al. [228]; LLMCadLLM-Cad[348]; LinguaLinked [389]
Techniques in LLMs contributing SLMs (§3.6)
RAG for SLMs [192, 363]; MoE for SLMs [154, 172]
Models (§5)
Generic-domain SLMs (§5.1)
Llama 3.2; Qwen [21, 352]; Gemma [295, 296]; StableLM [24, 303]; TinyLlama [382]; OLMo [104]; H2O-Danube3 [243]; Fox-1 [297]; MiniCPM [128]; Phi [1, 1, 107, 138, 178]; BLOOM and BLOOMZ [162]; Galactica [293]; OPT [383]; XGLM [186]; Megatron-gpt2 [274]; MINITRON [224]; LaMini-LM [153]; FlanT5 [57] ...
Specific-domain SLMs (§5.2)
Hippocrates [3]; BioMedLM [30]; MentaLLaMA [355]; ChemLLM [379]; SciGLM [378]; AstroLLaMA [230]; MindLLM [359]
SLM for reliable LLM generation (§6.1)
POLAR [392]; SAPLMA [19]; SuperICL [347]; SuperContext [357]; Self-RAG [16]; SKR [324]; SlimPLM [289]; CRAG [351]; Feng et al. [90]; Liu et al. [196]; HaluAgent [50]; Toolformer [264]

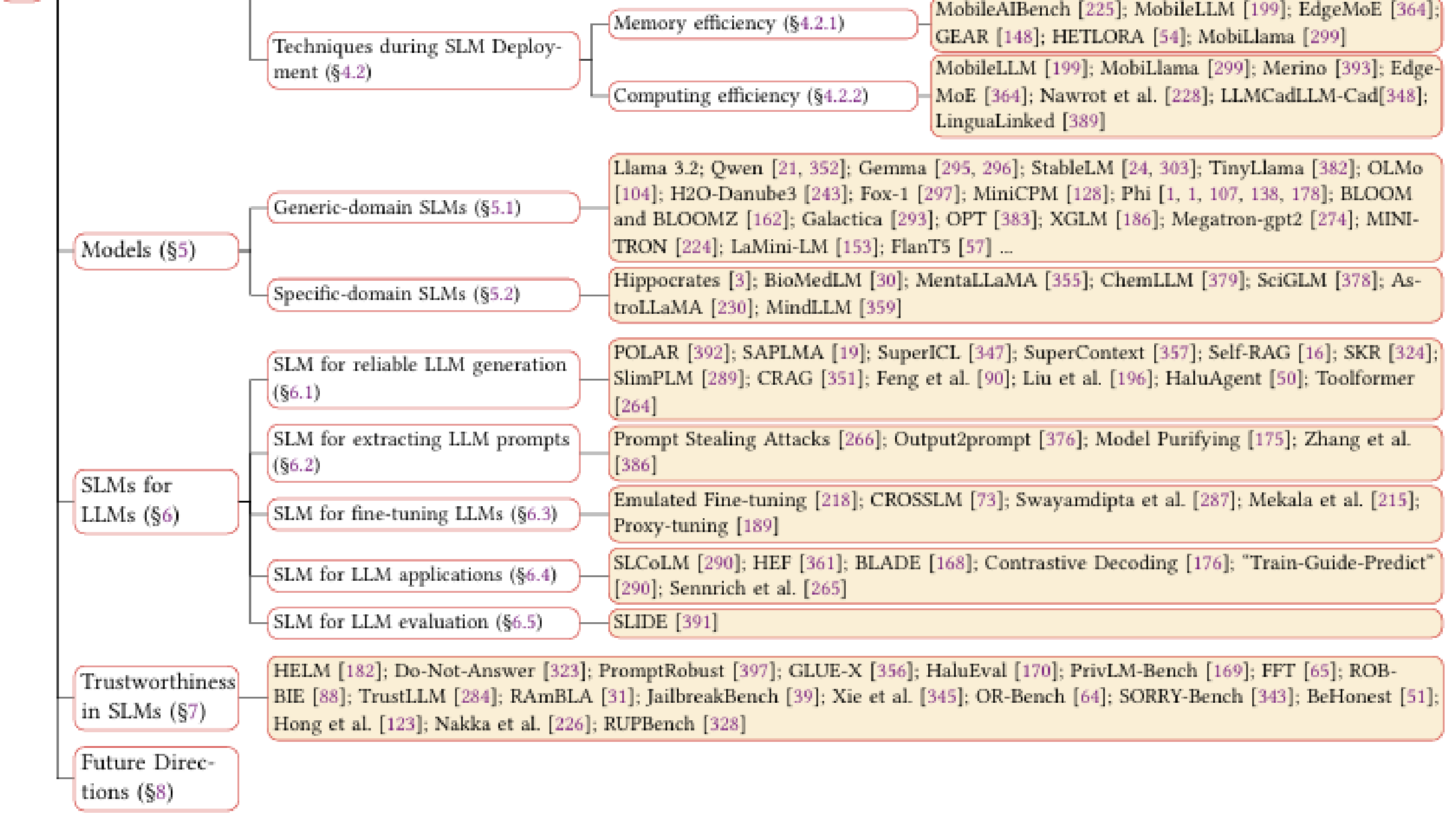


Fig. 1. Overview of Small Language Models.

Less-Parameters Language Models Arena

Microsoft

Phi-1 : 1.3B
Phi-2 : 2.7B
Phi-3.5: 3.8B (≈2.2 GB)

Meta

LLaMA 3.2 : 3B (≈2.02 GB)
Thai-Lang Supported

Deepmind's Gemma 2
2B: ≈1.6 GB
9B: ≈5.4 GB (LLM)

Alibaba Cloud

Qwen2.5 3B (≈1.93 GB)
Thai-Lang Supported

SCB X

Typhoon 8B (≈4.7-4.9 GB)
(Consider as LLM)
Thai-Lang Supported

KBTG

THaLLE 7B (≈4.4 GB-5.4 GB)
(Consider as LLM)
Thai-Lang Supported

Size Converted with llama.cpp/GGUF/Safetensor Size

Case Examples as Using Small Language Model

Anuchit Sapanpong (O) - Actor
Biography Trivia with Small Language Model (using RAG Technique)

Wichayada Chamnansilp (Namin) - Programmer
Offline AI Interaction Research

CK Cheong (Fastwork) - Influencer
Research and Local ESL Practice for his hobby

Jirachai Chansivanon (P' Job) - Developer
TinyLM for Internet of Things, OfflineGPT on Airplane

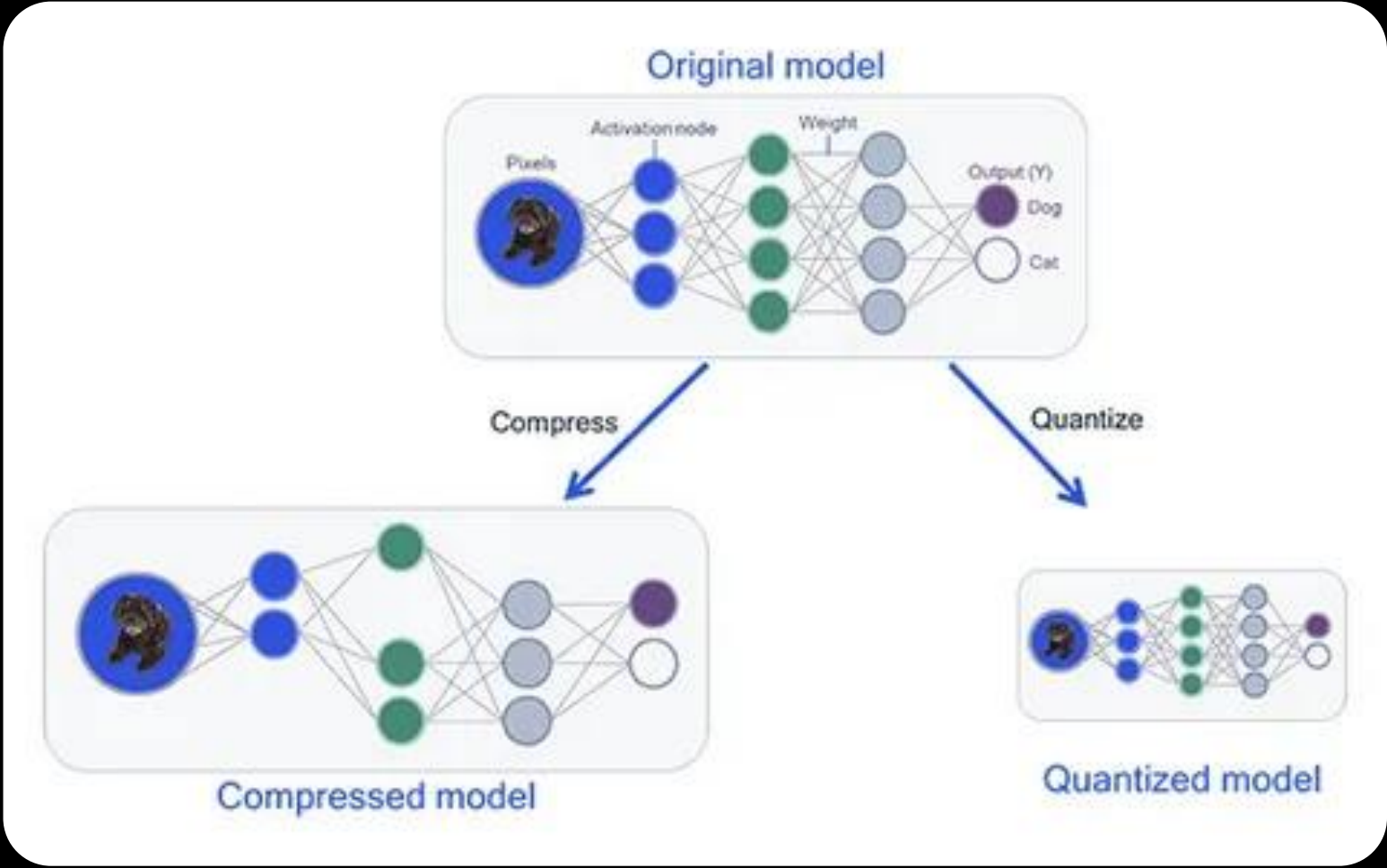
Wit Sitthivaekin (The Standard) - Influencer
History Trivia on Local Device

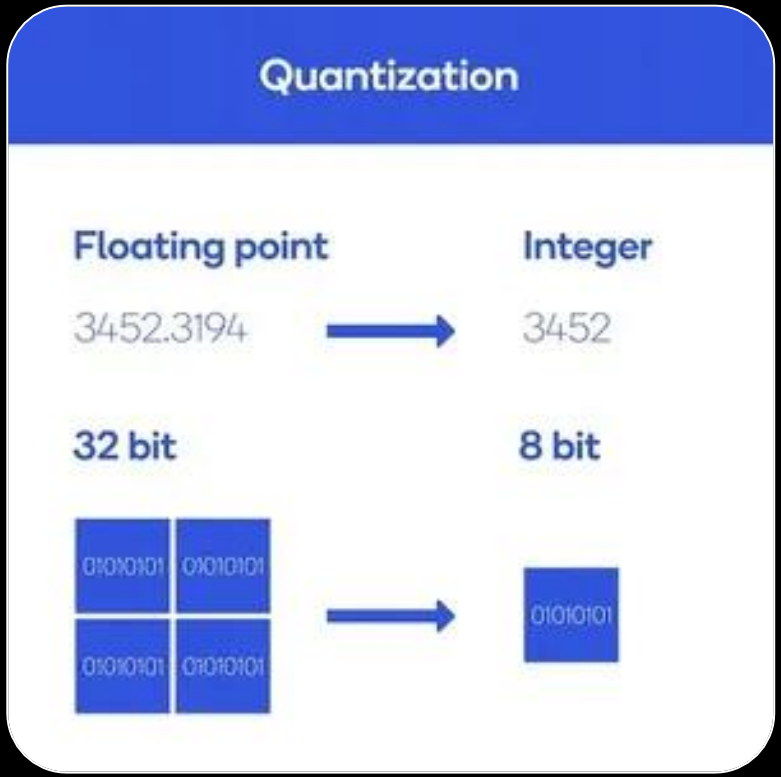
How are you going to use

Small Language Models

on your device at the first step?

SLMs Quantization





Reduce size and computation cost

Deploy model with limited device resources

Reduce Memory Footprint + Accelerate inference time

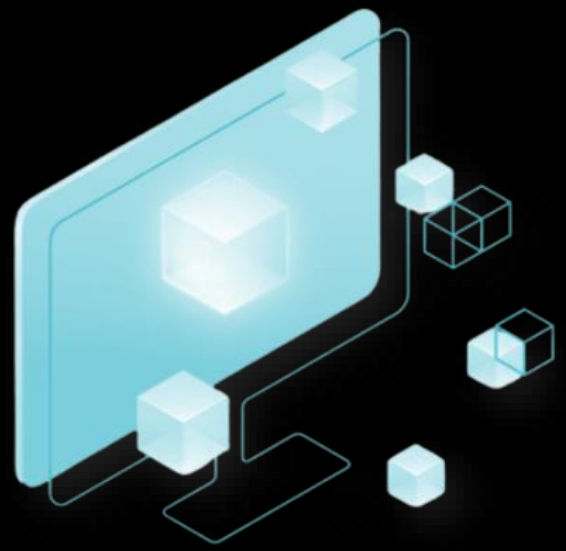
May loss some accuracy (but could improve by fine tuning)

Generally Available

Microsoft’s Phi-3.5

microsoft/Phi-3.5-mini-instruct · Hugging Face

Microsoft's Phi-3.5 Family

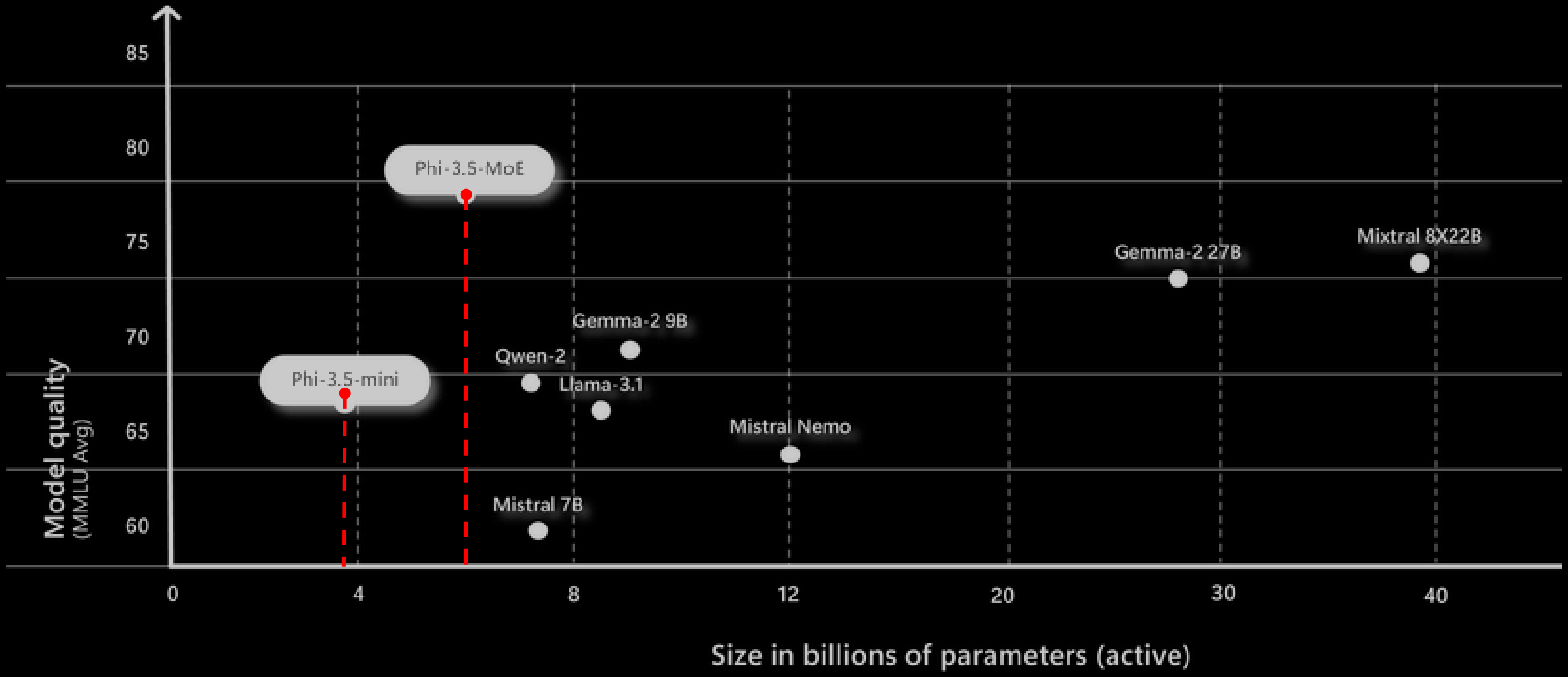


Model For Today's Experiment

Phi-3.5 mini-4k instruct (SLM)

- **Architecture**: 3.8B parameters and is a dense decoder-only Transformer model. The model is fine-tuned with Supervised fine-tuning (SFT) and Direct Preference Optimization (DPO) to ensure alignment with human preferences and safety guidelines.
- **Inputs**: Text. It is best suited for prompts using chat format.
- **Context Window Length**: 4K tokens
- **GPUs**: 512 H100-80G
- **Training time**: 7 days
- **Training data**: 3.3T tokens
- **Outputs**: Generated text in response to the input
- **Dates**: Our models were trained between February and April 2024
- **Status**: This is a static model trained on an offline dataset with cutoff date October 2023. Future versions of the tuned models may be released as we improve models.

Phi-3.5 Performance

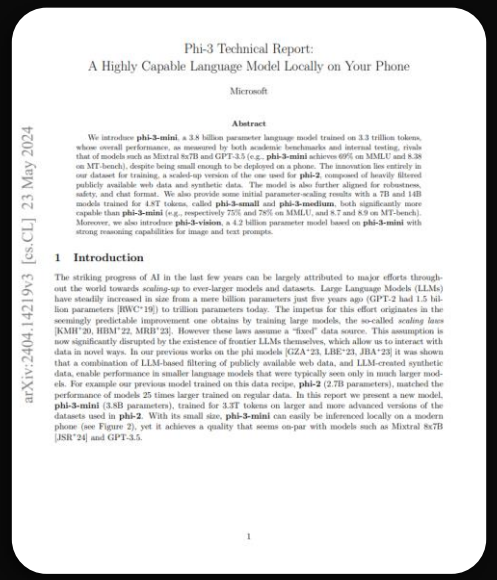


Textbooks Are All You Need

Evolution of Phi family

Phi-1 and Phi-1.5 (First Development)

Phi-2 (First Public Released) – Ignite 2023



Phi-3 (Latest & Stable) – Before #MSBuild 2024

Source: Microsoft Research (ArXiv)

Phi-1 and Phi-1.5 (First Development)

- 1.3 Billion Parameters
- Focusing on Python Coding and Some Commonsense reasoning
- First state-of-the-art performance on Python coding tasks (test with HumanEval and MBPP)
- Comparable performance to model 5x larger in language understanding

Phi-2 (First Public Released) – Ignite 2023

- 2.7 Billion Parameters
- More efficient on reasoning and language
- Matches or outperforms models up to 25x larger on complex benchmarks. Ideal for research and exploration

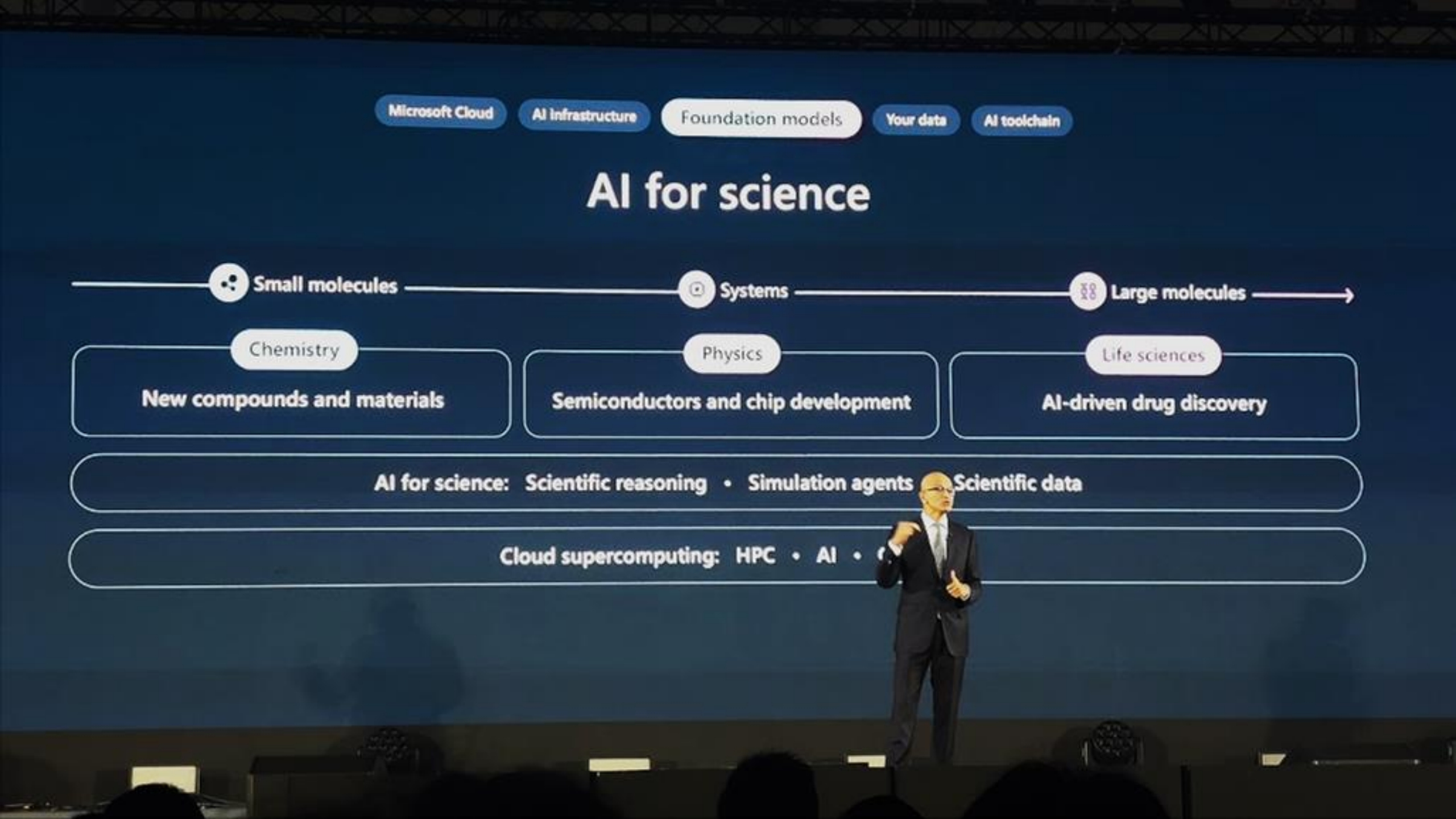
Phi-3 (Stable) – Before #MSBuild 2024

- At least 3.8B parameters
- More efficient on scientific reasoning and outperform on mathematical problems
- Providing multimodal features on Phi-3-vision-128k-instruct

Phi-3.5 - Fine Tuning from Phi-3 (Latest - Released on August 2024)

- Supporting Multilingual Context (including Thai Language) - MoE model
- More Logical Thinking for specific tasks

Source: Microsoft Research (ArXiv)


Microsoft Cloud
AI infrastructure
Foundation models
Your data
AI toolchain
AI for science
Small molecules
Systems
Large molecules
Chemistry
New compounds and materials
Physics
Semiconductors and chip development
Life sciences
AI-driven drug discovery
AI for science: Scientific reasoning • Simulation agents • Scientific data
Cloud supercomputing: HPC • AI •

When you cannot purely use Multilingual Model

The **phi-3.5-MoE** adopts an Mixture-of-Experts (MoE) architecture to selectively activate parts of modules on specific inputs to improve the model efficiency. It incorporates MoE layer as its feedforward models, employing the top2 routing among 16 expert networks. Particularly, each expert network is a separate GLU network and the routing module will selectively activate 2 expert networks out of the 16 expert networks for each token, leaving 16×3.8B model to have 6.6B activated parameters with 42B

model-00001-of-00017.safetensors Safe	4.99 GB LFS	upload initial files	3 months ago
model-00002-of-00017.safetensors Safe	4.99 GB LFS	upload initial files	3 months ago
model-00003-of-00017.safetensors Safe	4.99 GB LFS	upload initial files	3 months ago
model-00004-of-00017.safetensors Safe	4.99 GB LFS	upload initial files	3 months ago
model-00005-of-00017.safetensors Safe	4.99 GB LFS	upload initial files	3 months ago
model-00006-of-00017.safetensors Safe	4.99 GB LFS	upload initial files	3 months ago
model-00007-of-00017.safetensors Safe	4.99 GB LFS	upload initial files	3 months ago
model-00008-of-00017.safetensors Safe	4.99 GB LFS	upload initial files	3 months ago
model-00009-of-00017.safetensors Safe	4.99 GB LFS	upload initial files	3 months ago
model-00010-of-00017.safetensors Safe	4.99 GB LFS	upload initial files	3 months ago
model-00011-of-00017.safetensors Safe	4.99 GB LFS	upload initial files	3 months ago
model-00012-of-00017.safetensors Safe	4.96 GB LFS	upload initial files	3 months ago
model-00013-of-00017.safetensors Safe	4.99 GB LFS	upload initial files	3 months ago
model-00014-of-00017.safetensors Safe	4.99 GB LFS	upload initial files	3 months ago
model-00015-of-00017.safetensors Safe	4.99 GB LFS	upload initial files	3 months ago
model-00016-of-00017.safetensors Safe	4.99 GB LFS	upload initial files	3 months ago
model-00017-of-00017.safetensors Safe	3.91 GB LFS	upload initial files	3 months ago

We don’t know which one is Thai Language

Phi-3.5 family of open models

Now also in MaaS for Azure AI Studio	New	New	New
Phi-3.5-mini	Phi-3.5-small	Phi-3.5-medium	Phi-3.5-vision

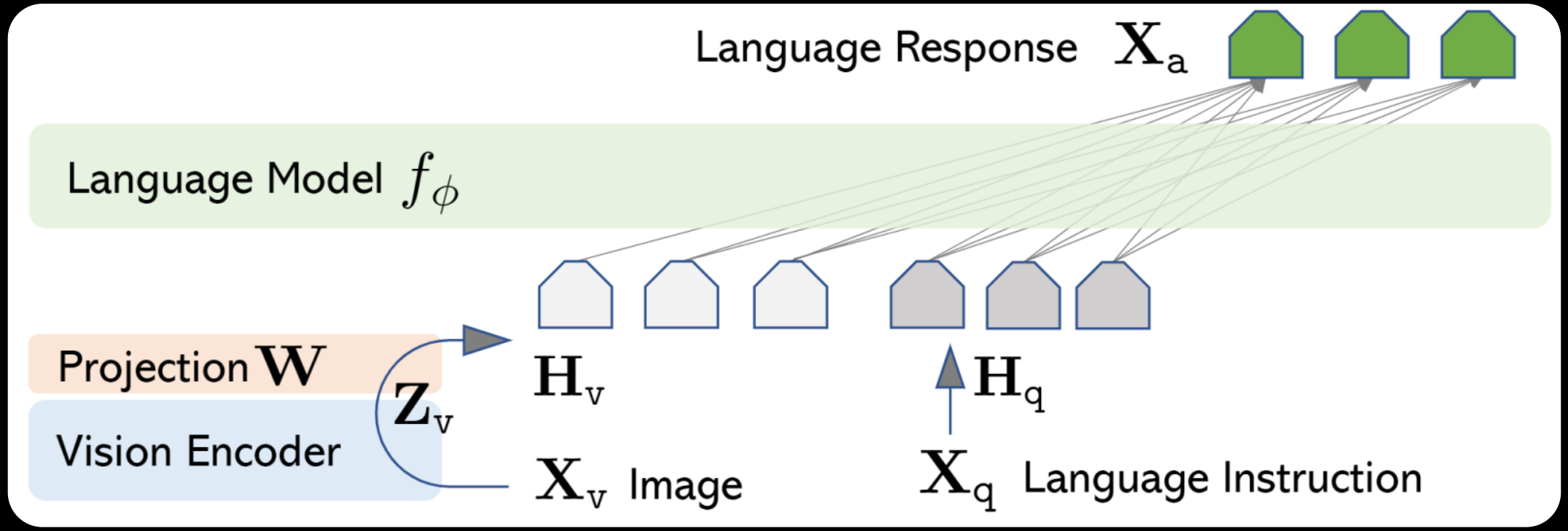
Generally Available At

Hugging Face

LlaVA - Large Language-and-Vision Assistant

- An end-to-end trained large multimodal model that connects a vision encoder and LLM for general-purpose visual and language understanding.

- **Open-source.** Making GPT-4 generated visual instruction tuning data, our model and code base publicly available.



Microsoft's Responsible AI principles

Fairness

Reliability & Safety

Privacy & Security

Inclusiveness

Transparency

Accountability

P.Y.D. - Simple Responsible AI principles for SLMs

P stands for **Privacy**:

- **Data Minimization:** Collect only necessary data.
- **Consent:** Obtain informed consent for data use.
- **Anonymization:** Protect personal information.

Y stands for **Your Responsibility:**

- **Bias Mitigation:** Address biases in data and algorithms.
- **Transparency:** Explain AI decisions and processes.
- **Accountability:** Take responsibility for AI outcomes.

D stands for **Diversity:**

- **Inclusivity:** Ensure AI development and use benefits diverse populations.
- **Fairness:** Avoid discrimination in AI systems.
- **Equity:** Address disparities in AI access and outcomes.

Introducing

Ollama

Running AI on Local Devices

What is Ollama?

- Ollama is a platform that makes local development with open-source language models a breeze.
- With Ollama, everything you need to run an LLM—model weights and all of the config—is packaged into a single Modelfile. Think it is Docker for LLMs. Act like GitHub at some points
- Run Llama 3.1, Phi 3.5, Mistral, Gemma 2, and other models from Hugging Face. Customize and create your own.

Microsoft Meta

Introducing

OpenAI API Capability

Ollama now has built-in compatibility with the OpenAI Chat Completions API, making it possible to use more tooling and applications with Ollama locally.

```python
from openai import OpenAI

client = OpenAI(
    base_url = 'http://localhost:11434/v1',
    api_key='ollama', # required, but unused
)

response = client.chat.completions.create(
  model="llama2",
  messages=[
    {"role": "system", "content": "You are a helpful assistant."}
  ]
)
print(response.choices[0].message.content)
```

```javascript
import OpenAI from 'openai'

const openai = new OpenAI({
  baseURL: 'http://localhost:11434/v1',
  apiKey: 'ollama', // required but unused
})

const completion = await openai.chat.completions.create({
  model: 'llama2',
  messages: [{ role: 'user', content: 'Why is the sky blue?' }],
})

console.log(completion.choices[0].message.content)
```

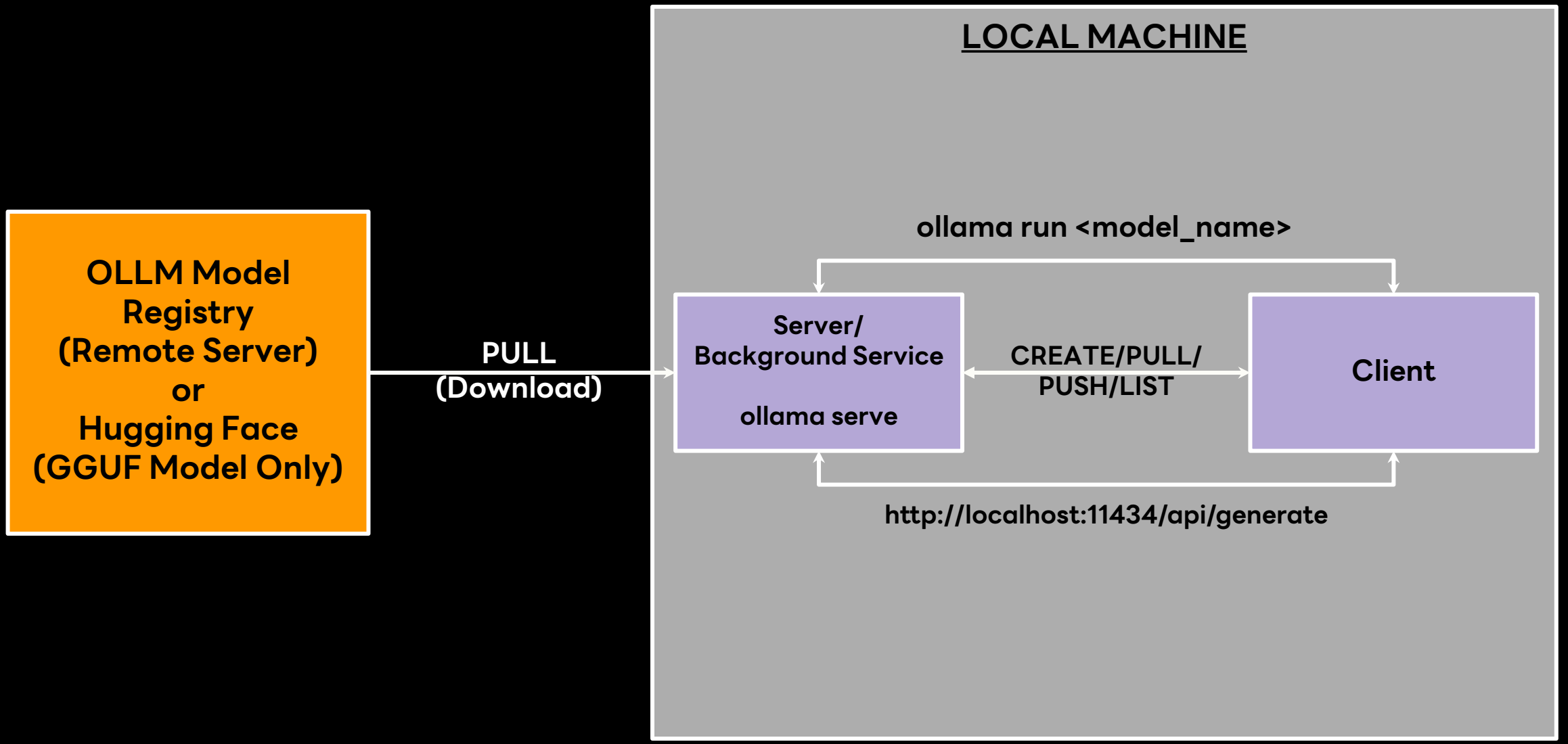
Introducing

Download from Hugging Face

You can use **any GGUF quants** created by the community on Hugging Face directly with Ollama, without creating a new Modelfile. You can run any of them with a single **ollama run** command. We also provide customizations like choosing quantization type to improve your overall experience.

Hugging Face

```
ollama run hf.co/username/repository
```


OLLM Model
Registry
(Remote Server)
or
Hugging Face
(GGUF Model Only)
PULL
(Download)
LOCAL MACHINE
ollama run <model_name>
Server/
Background Service
ollama serve
CREATE/PULL/
PUSH/LIST
Client
http://localhost:11434/api/generate

First Demo

Ollama Implementation on .NET with **Semantic Kernel**

Old
Logo

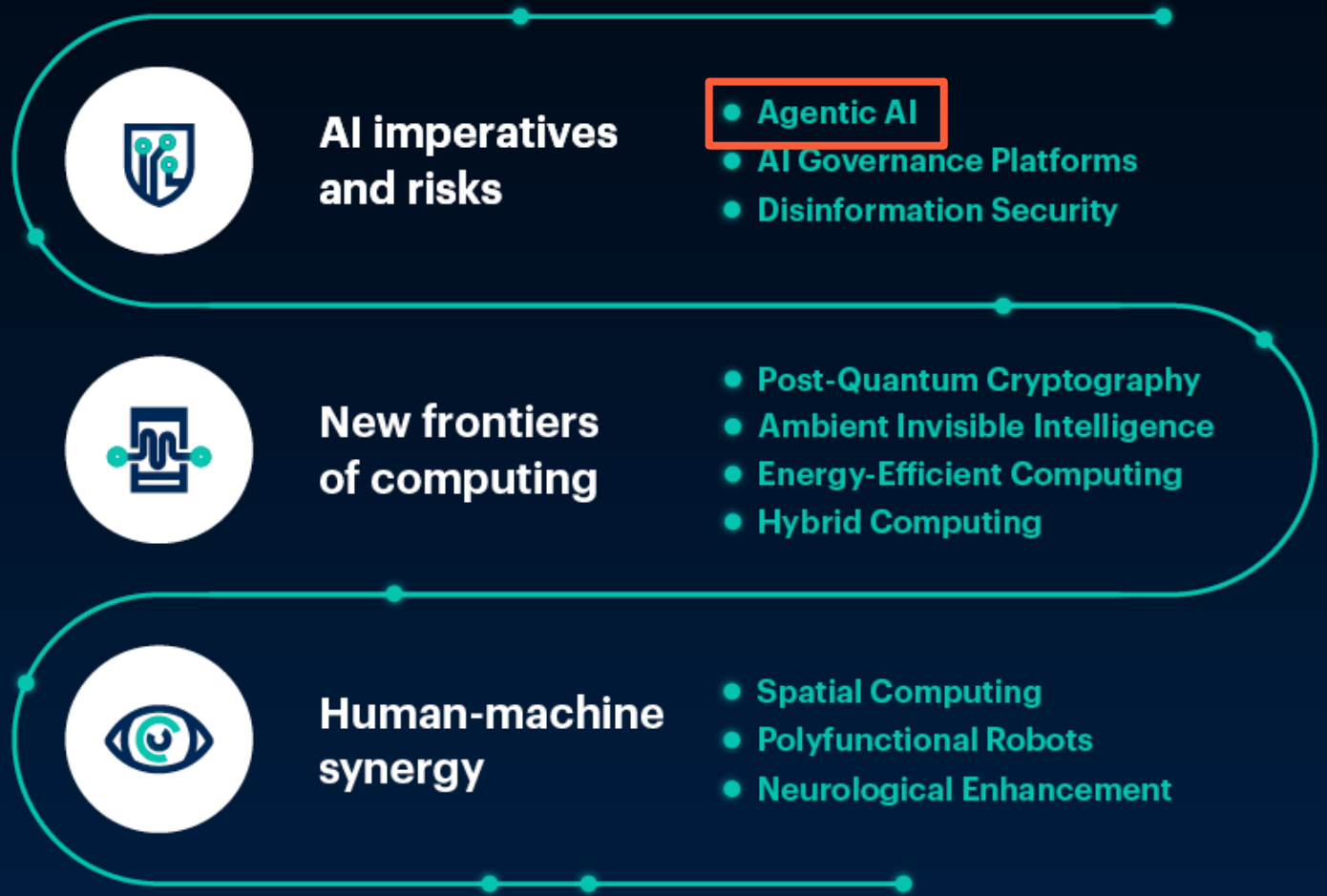
New
Logo

Why

Semantic Kernel <u>important</u>

for .NET developers along to the year 2025**?**

2025 Top 10 Strategic Technology Trends



Source: Gartner

Gartner

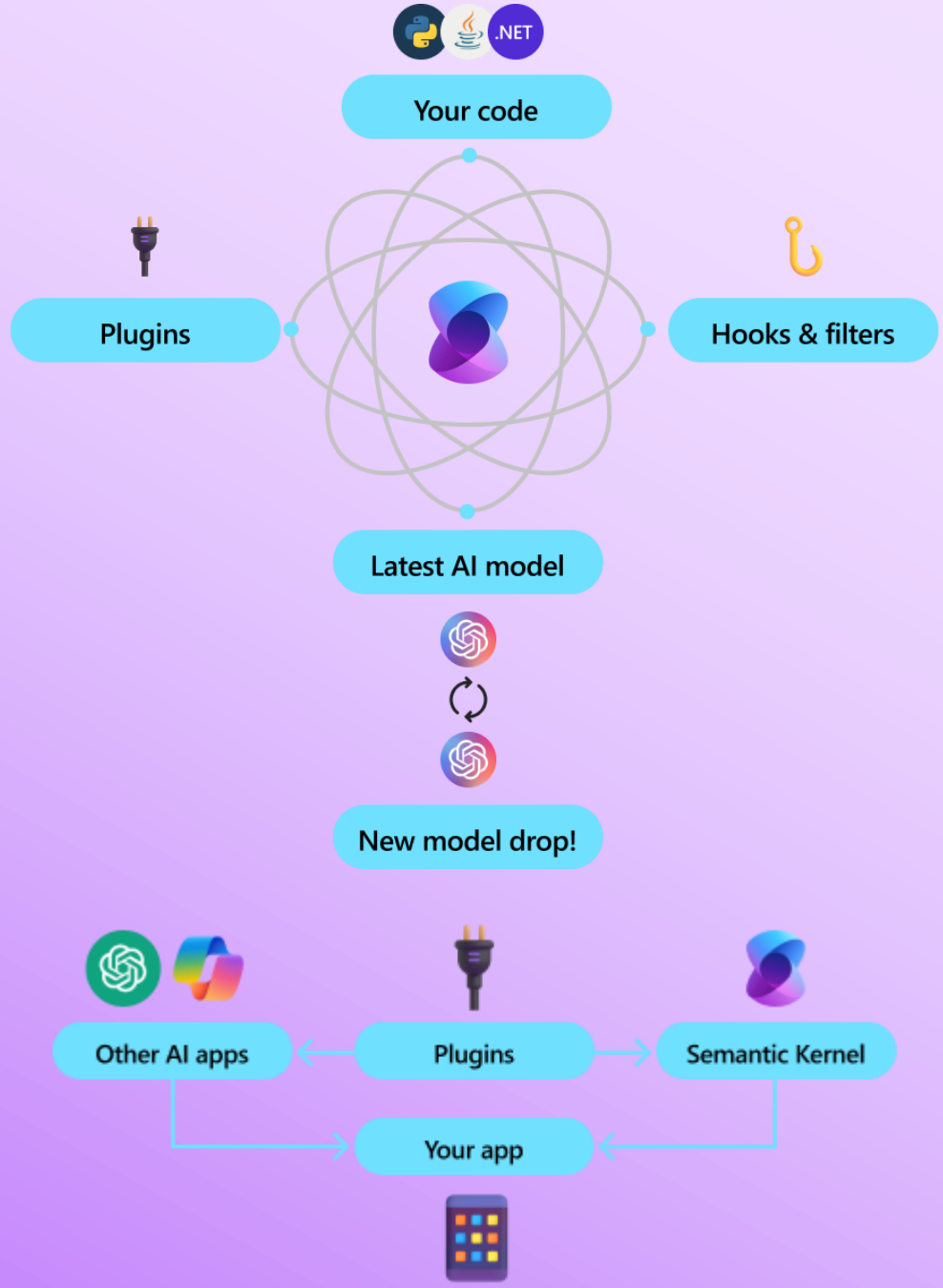
What is AI Agent and Multi-Agent AI System?

- An **AI agent** is an autonomous entity that can perceive its environment, reason about it, and take actions to achieve its goals. It functions through perception, reasoning, and action components.

- A **Multi-Agent AI System** is a system composed of multiple AI agents that interact and collaborate to achieve shared or individual goals. It exhibits decentralization, emergent behavior, collaboration, and competition. Applications include simulation, robotics, game AI, and e-commerce.

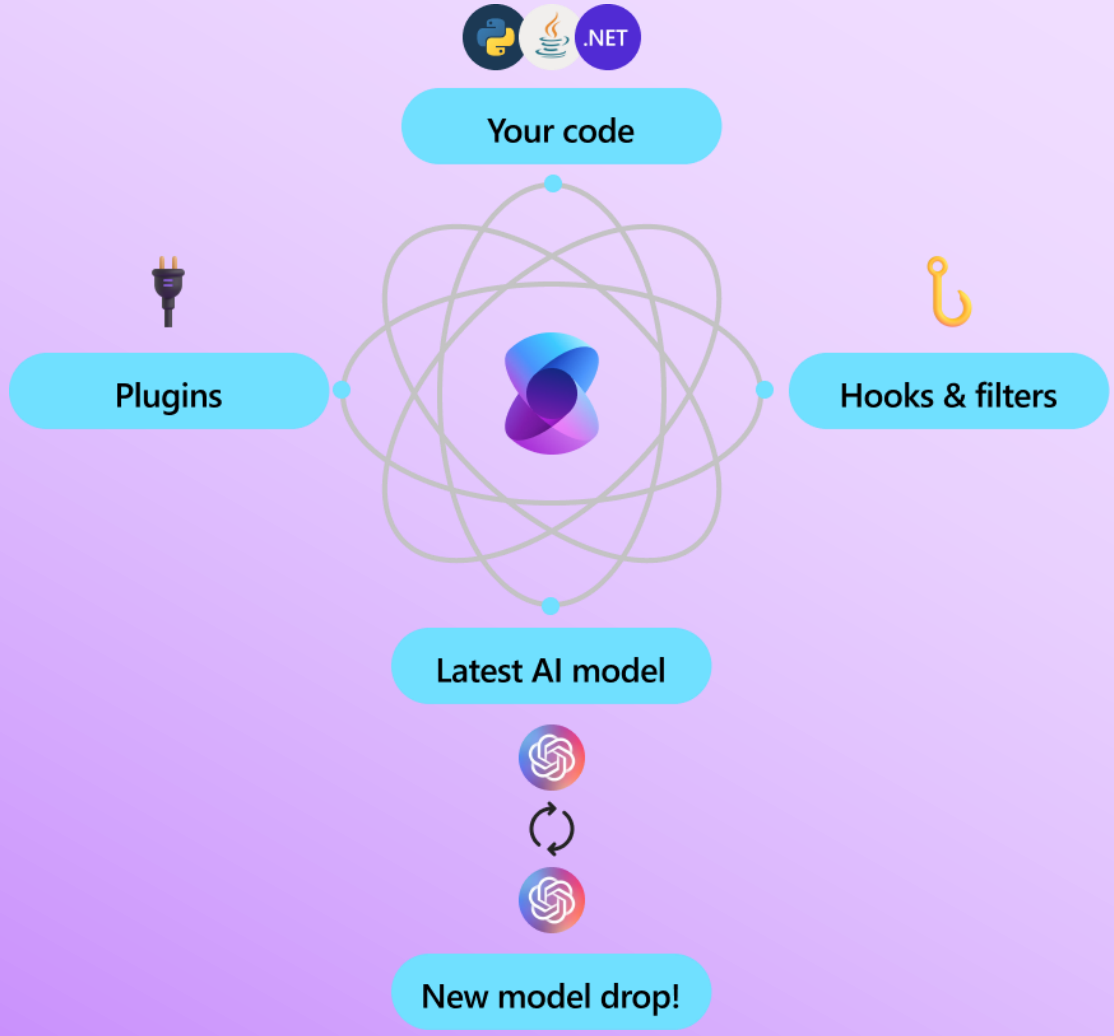
Semantic Kernel

Semantic Kernel is a lightweight, open-source development kit that lets you easily build **AI agents** and integrate the latest AI models into your **C# (.NET)**, Python, or Java codebase. It serves as an efficient middleware that enables rapid delivery of enterprise-grade solutions.



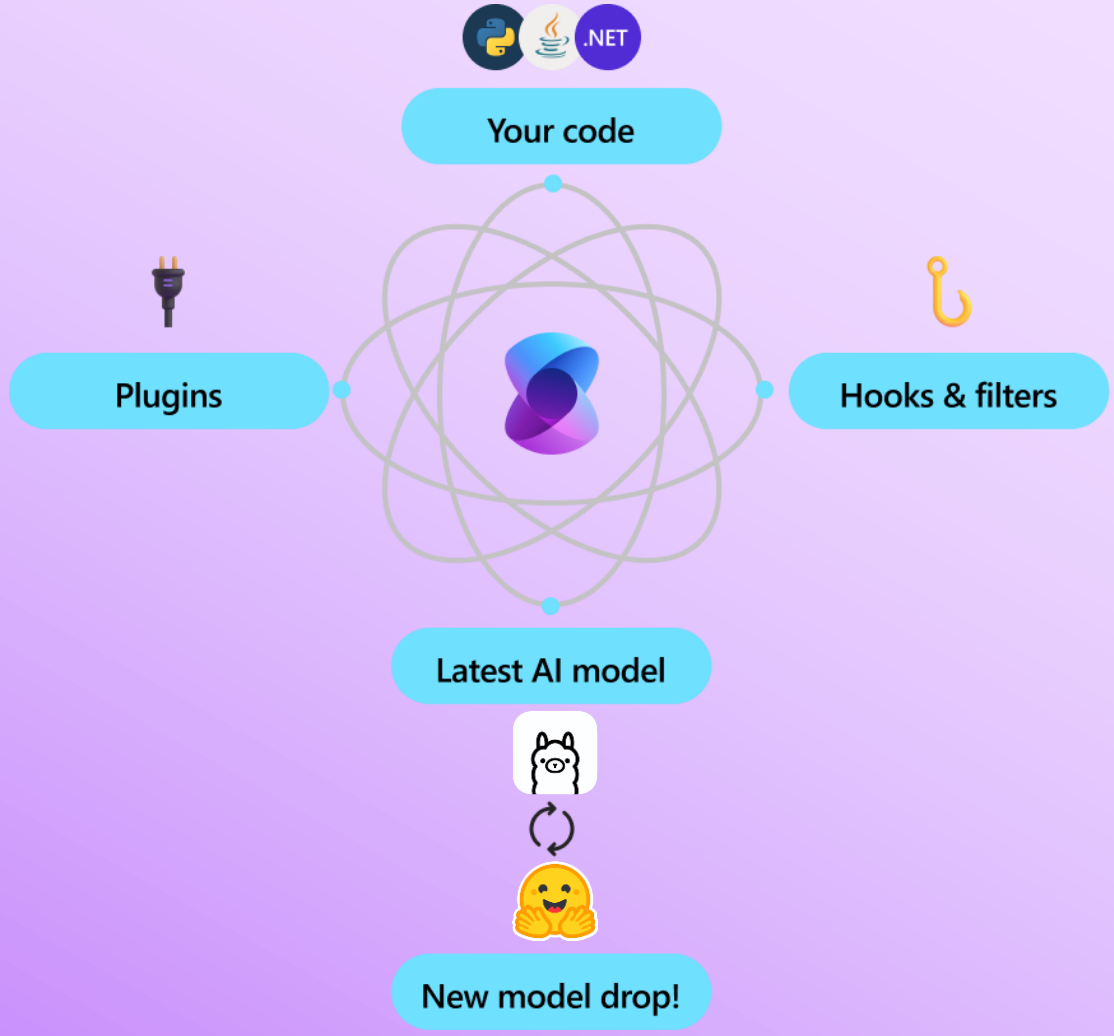
Semantic Kernel

Semantic Kernel is a lightweight, open-source development kit that lets you easily build **AI agents** and integrate the latest AI models into your **C# (.NET)**, Python, or Java codebase. It serves as an efficient middleware that enables rapid delivery of enterprise-grade solutions.



Semantic Kernel

Semantic Kernel is a lightweight, open-source development kit that lets you easily build **AI agents** and integrate the latest AI models into your **C# (.NET)**, Python, or Java codebase. It serves as an efficient middleware that enables rapid delivery of enterprise-grade solutions.



Installing all dependencies in Terminal

```
> dotnet new console

> dotnet add package Microsoft.SemanticKernel
> dotnet add package Microsoft.SemanticKernel.Connectors.Ollama

> code .       # entering inside Visual Studio Code
```

Declaring Libraries, Models from Ollama (Plug-in)

```
using Microsoft.SemanticKernel;
using Microsoft.SemanticKernel.ChatCompletion;

// Create a kernel with OpenAI chat completion API
// Warning due to the experimental state of some Semantic Kernel SDK
features.
#pragma warning disable SKEXP0070
Kernel kernel = Kernel.CreateBuilder()
                      .AddOllamaChatCompletion(
                          modelId: “your ollama model name",
                          endpoint: new Uri("http://localhost:11434"))
                      .Build();
```

Chat Completion + Storing Chat History

```
var aiChatService = kernel.GetRequiredService<IChatCompletionService>();
var chatHistory = new ChatHistory();

while (true)
{
    // Get user prompt and add to chat history
    Console.WriteLine("Your prompt:");
    var userPrompt = Console.ReadLine();
    chatHistory.Add(new ChatMessageContent(AuthorRole.User, userPrompt));
```

Chat Completion + Storing Chat History

```
    // Stream the AI response and add to chat history
    Console.WriteLine("AI Response:");
    var response = "";
    await foreach(var item in
        aiChatService.GetStreamingChatMessageContentsAsync(chatHistory))
    {
        Console.Write(item.Content);
        response += item.Content;
    }
    chatHistory.Add(new ChatMessageContent(AuthorRole.Assistant, response));
    Console.WriteLine();
}
```

Ollama Implementation with AutoGen for .NET for making the AI agents

What is AutoGen?

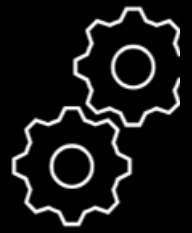
AutoGen An Open-Source Programming Framework for Agentic AI

Multi-Agent Conversation Framework

AutoGen provides multi-agent conversation framework as a high-level abstraction. With this framework, one can conveniently build LLM workflows.

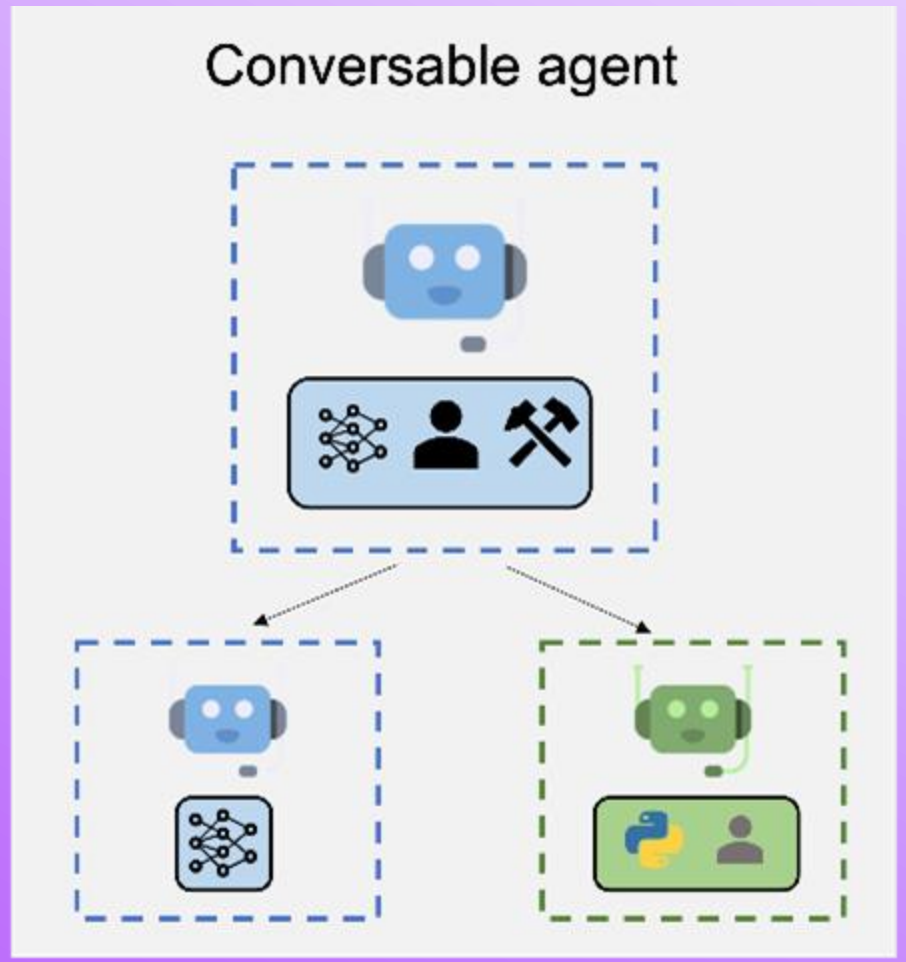
Easily Build Diverse Applications

AutoGen offers a collection of working systems spanning a wide range of applications from various domains and complexities.

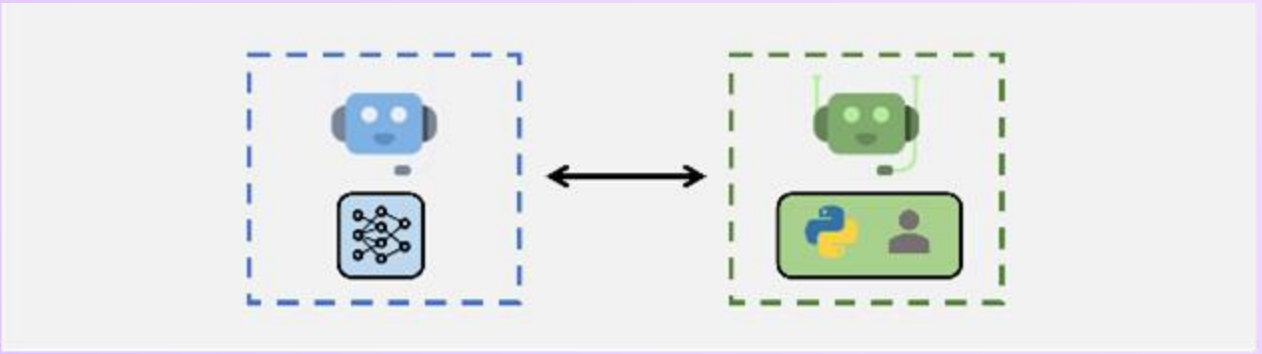


Enhanced LLM Inference & Optimization

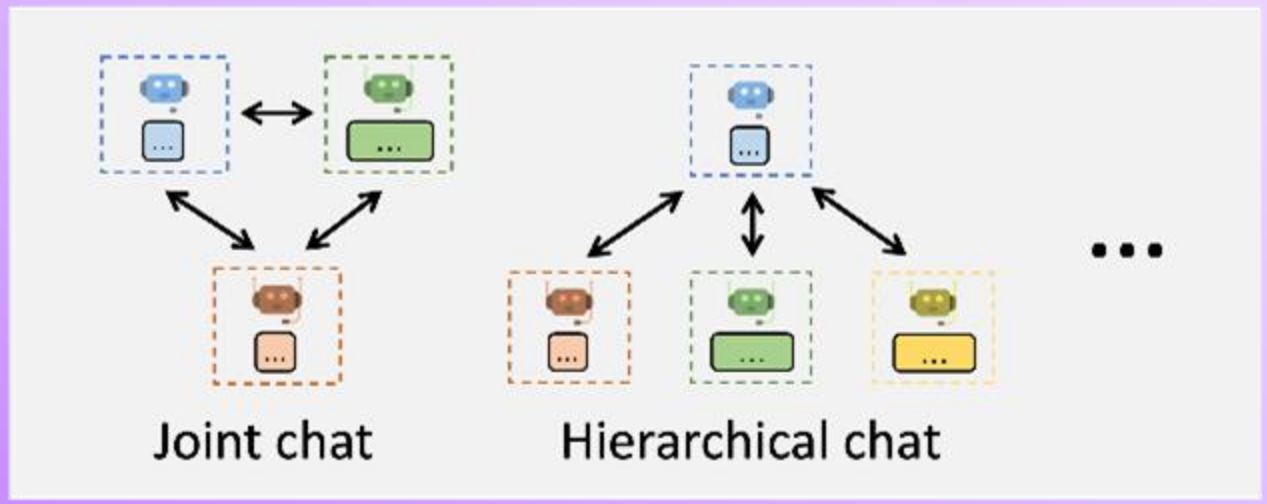
AutoGen supports enhanced LLM inference APIs, which can be used to improve inference performance and reduce cost.



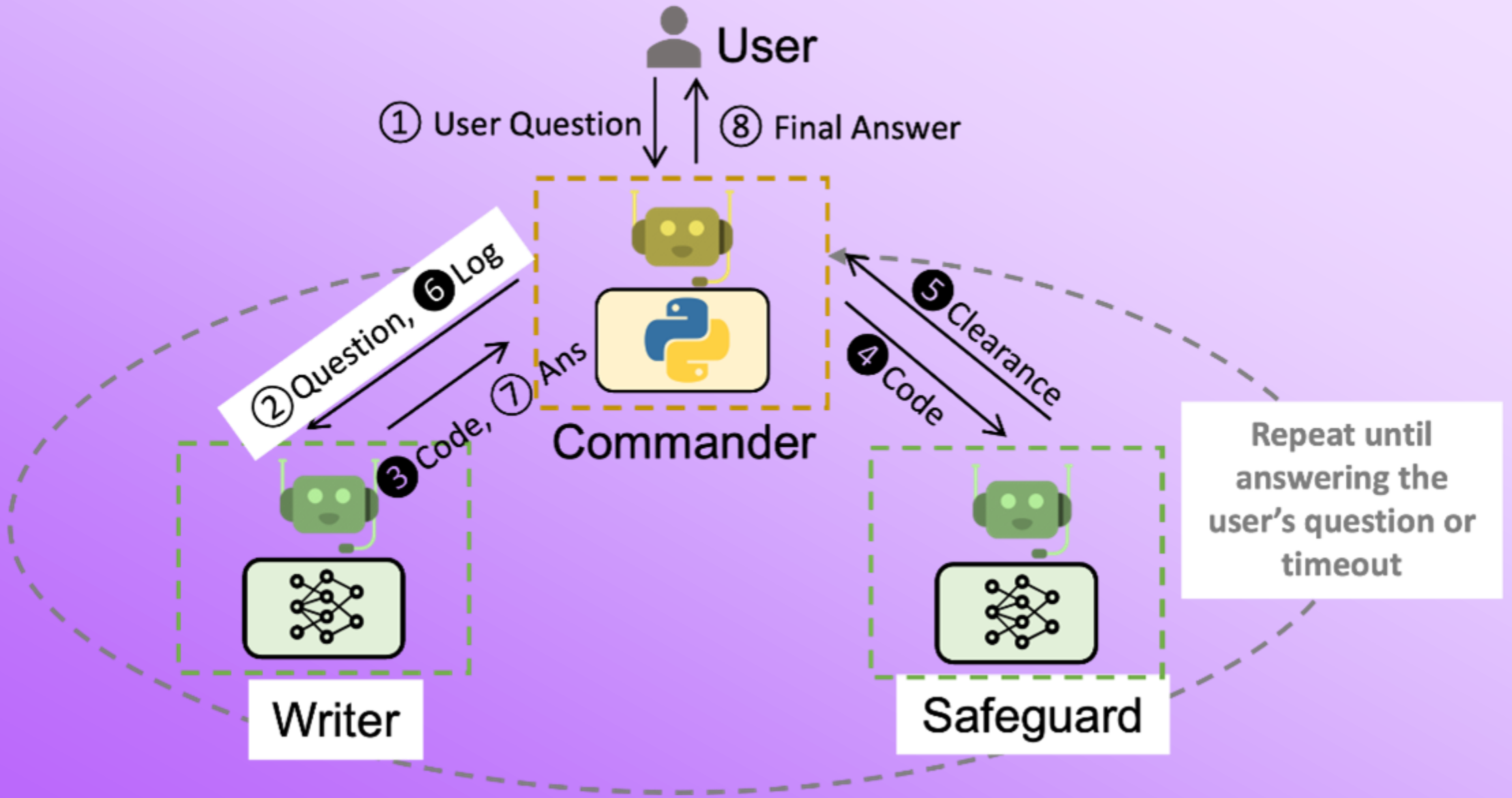
Agent Customization



Multi-Agent Conversations



Flexible Conversation Patterns


User
① User Question
⑧ Final Answer
②Question, ⑥Log
③Code, ⑦Ans
Commander
⑤Clearance
④Code
Repeat until answering the user's question or timeout
Writer
Safeguard

Some Syntaxes

Running Ollama Model with AutoGen for .NET and Semantic Kernel

Pre-requisite

First, install the AutoGen.Ollama package using the following command:

```
dotnet add package AutoGen.Ollama
```

Step 2: Add using statement

```
using AutoGen.Core; using AutoGen.Ollama.Extension;
```

Create OllamaAgent: Text Based

```
using var httpClient = new HttpClient()
{
    BaseAddress = new Uri("http://localhost:11434"),
};

var ollamaAgent = new OllamaAgent(
    httpClient: httpClient,
    name: "ollama",
    modelName: "llama3:latest",
    systemMessage: "You are a helpful AI assistant")
    .RegisterMessageConnector()
    .RegisterPrintMessage();

var reply = await ollamaAgent.SendAsync("Can you write a piece of C# code to
calculate 100th of fibonacci?");
```

Create OllamaAgent: Multimodal Based (VLM)

```
using var httpClient = new HttpClient()
{
    BaseAddress = new Uri("http://localhost:11434"),
};

var ollamaAgent = new OllamaAgent(
    httpClient: httpClient,
    name: "ollama",
    modelName: "llava-phi3",
    systemMessage: "You are a helpful AI assistant")
    .RegisterMessageConnector()
    .RegisterPrintMessage();

var image = Path.Combine("resource", "images", "background.png");
var binaryData = BinaryData.FromBytes(File.ReadAllBytes(image), "image/png");
var imageMessage = new ImageMessage(Role.User, binaryData);
var textMessage = new TextMessage(Role.User, "what's in this image?");
var reply = await ollamaAgent.SendAsync(chatHistory: [textMessage, imageMessage]);
```

Code and Slides are available on my GitHub

chrnthnkmutt/
BoatSLM_cs_experiment

This repository is use to make the experiment of using ollama, along with custom small language model, in .NET framework

1 Contributor
0 Issues
0 Stars
0 Forks

Key Takeaways

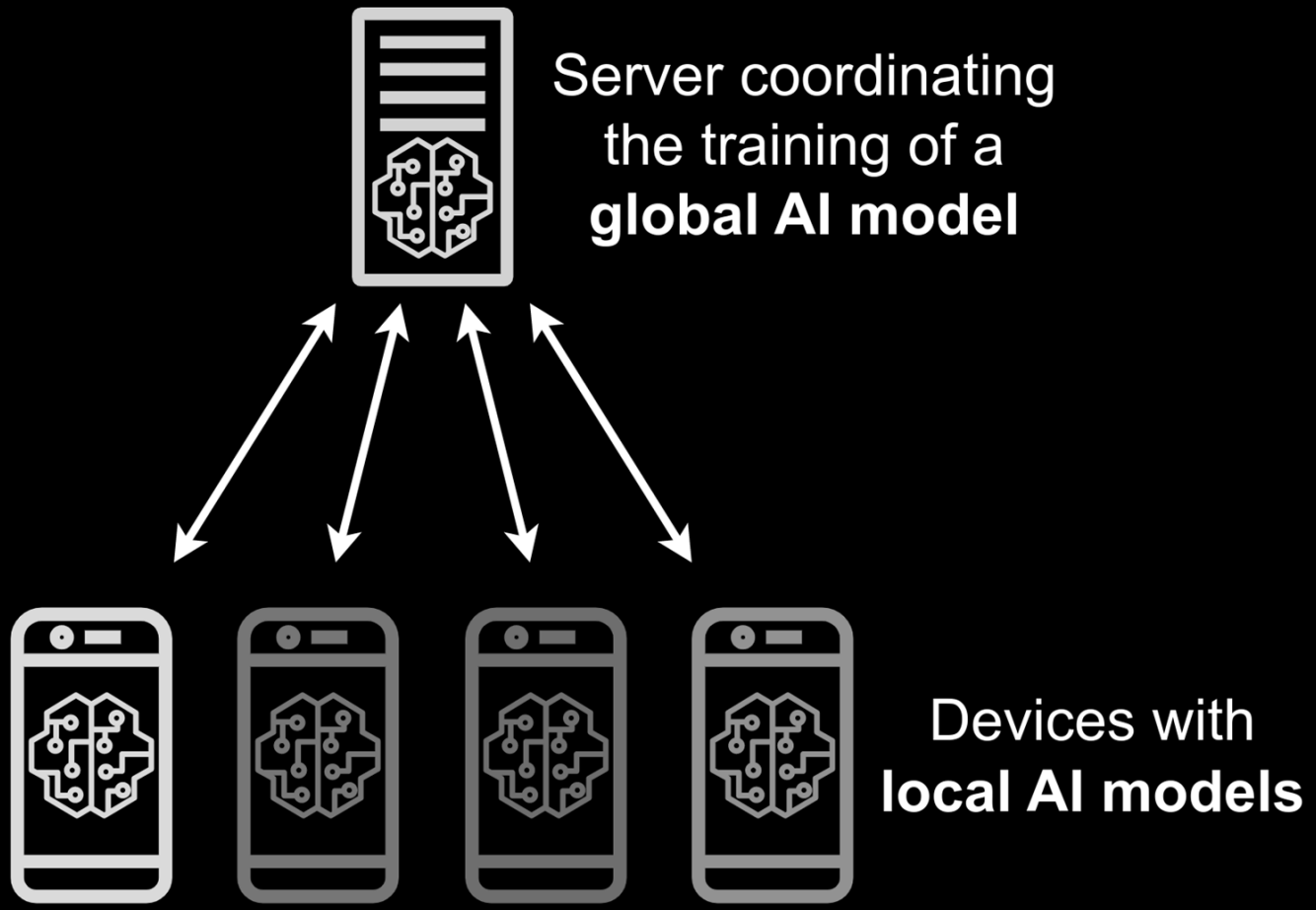
Upcoming Trends of Generative AI and NLPs Industry

- Shift to **multimodality**: Text → text, images, pictures, speech

- **Explainability** (Reason/Evidence for why the output is produce)
  Interpretability (understanding how the model works)
  Trustworthiness (safety and transparency of model usage)

- **Model Miniaturization**: separation between **language fluency** and **knowledge base** for AI on small devices (Laptop, Smartphones) with Multi Agents Compatibility

Federated Learning



The question is...

Why Thai People Still Doesn't Use Small Language Models?

- Hallucination & Performance
- Limited Device Resources
- Languages Barrier
- Low Engagement?

Let me be there for you ❤️

to learn more on new innovations on the world

Our New Finetuned **Small Language Model**
For Thai People




will be launched and available on **National Coding Day** next week in **Hugging Face**


Hundreds of Thai Data & AI **Community Leaders** are in 2024
We hope to see you more than a **million** leaders and trainers in 2025 as a part of us :)

Escaping Yourself from “**Know-it-all**” to “**Learn-it-all**”

Thank You!

Charunthon Limseelo

@boatchrnthn

Charunthon Limseelo

Boat Charunthon (boatchrnthn)