BKK.JS creators garten Microsoft Learn Student Ambassadors

Developing JavaScripts

For Open-Source Generative-AI Developers in Nowadays

PRELUDE of JavaScript Bangkok 2.0.0

Created By

Charunthon Limseelo - Microsoft Learn Student Ambassador
Pancheva Niruttikul - Google Developer Student
and Tatta Tameeyonk - Head of Academic, EBA CU

Collaborating with Poonyada Phanitpotchamarn

BKK.JS #21 UNLEASHED - September 14th, 2024

Charunthon Limseelo (Boat)

Beta Microsoft Learn Student Ambassadors at KMUTT
+ Microsoft Office Specialist (Excel)
+ Open-sourced AI and ML Interest, with Data Science Applications
+ Applied Skills (AI Field)

- Charunthon Limseelo
- @boatchrnthn
- Charunthon Limseelo
- Boat Charunthon (boatchrnthn)

Two types of Technological Ownerships
Proprietary and Open-Source

Proprietary

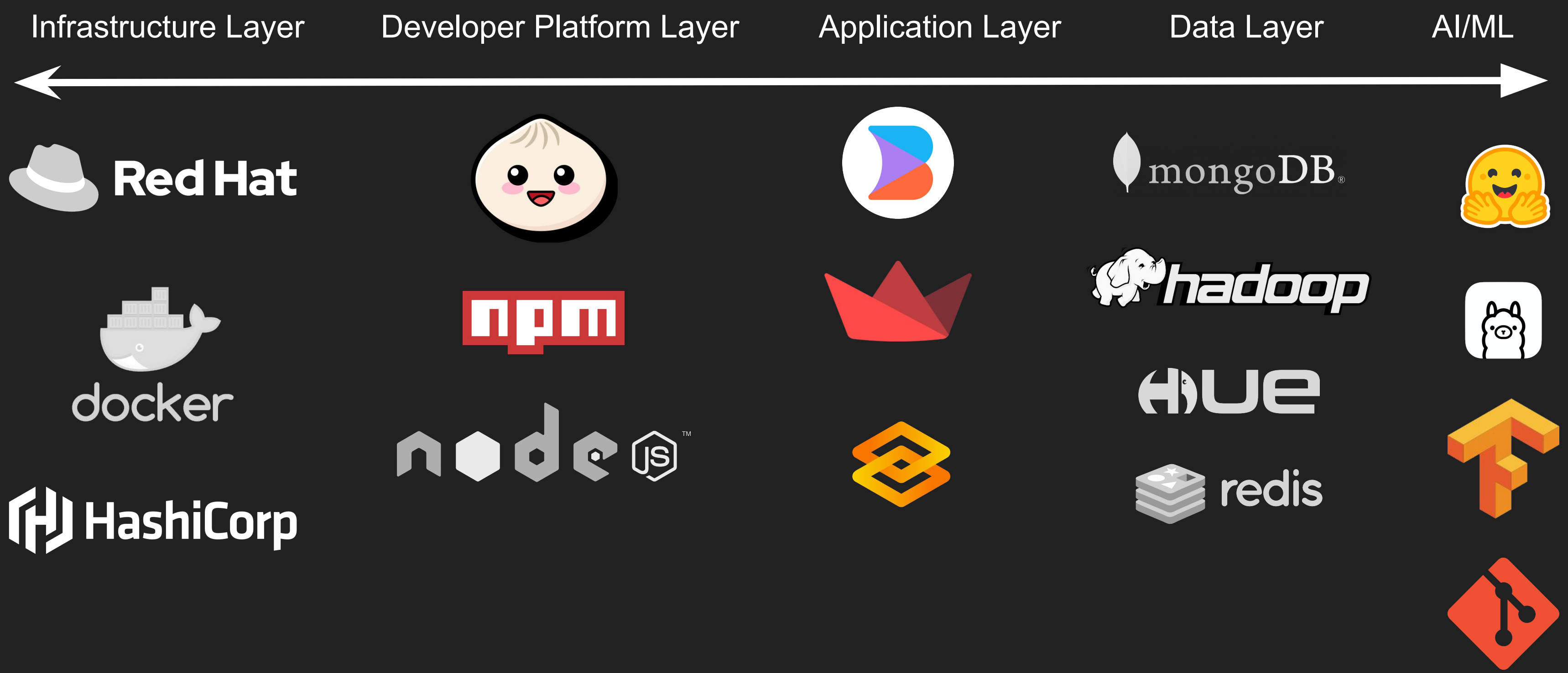
(adj.) used, made, or marketed by one having the exclusive legal right, privately owned and managed and run as a profit-making organization

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

For you, what do you know about Open-Source?

Some Open-Source tools in Tech Stack



Infrastructure Layer

Developer Platform Layer

Application Layer

Data Layer

AI/ML

Everything that I’m going to demonstrate…will be based on only Small Language Models (SLMs)🌿

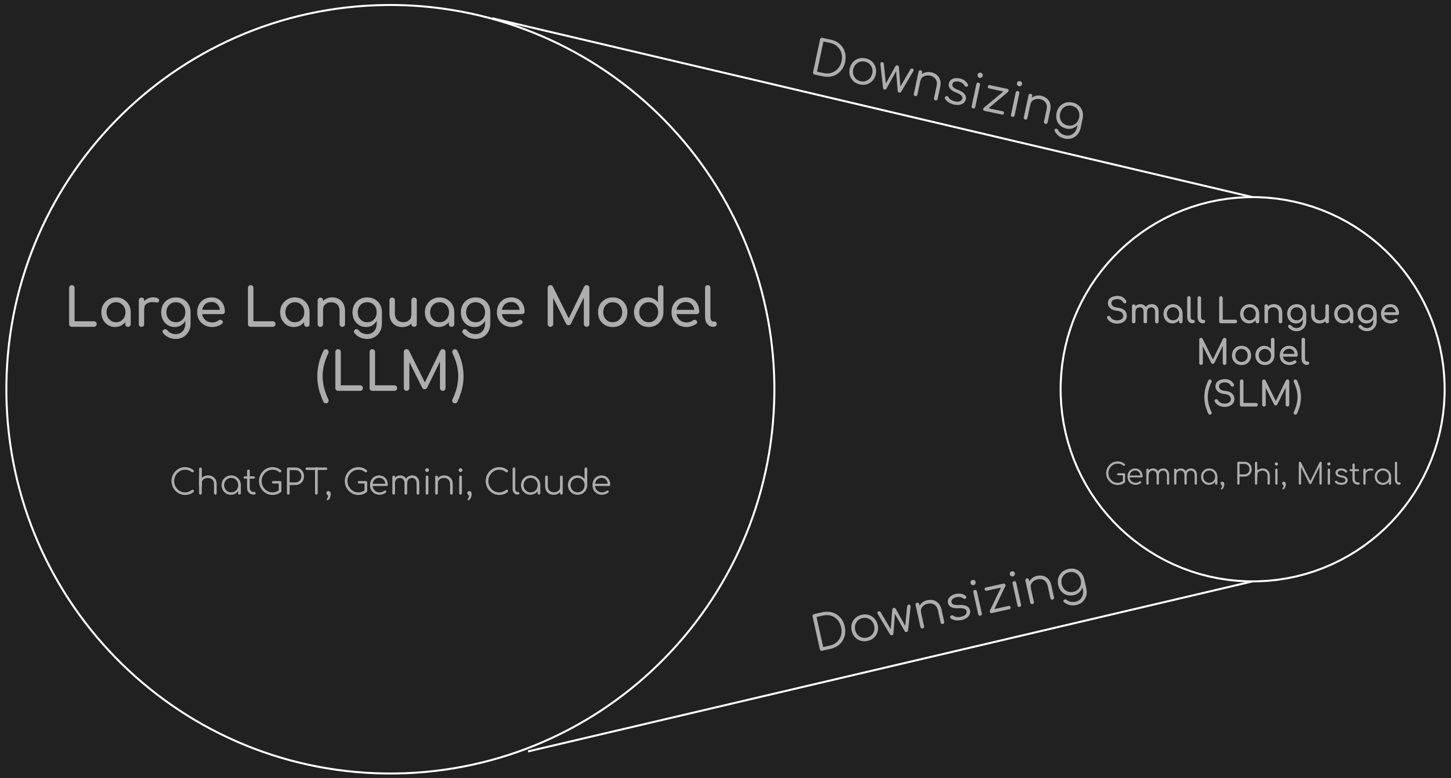
Fundamental AI Tools To Learn for JavaScript Developers

Hugging Face
(Transformer.js)

Ollama.js
(Newest)

GitHub Models
(Newest-Proprietary)

GitHub Repository
(Open-Source)



Large Language Model
(LLM)

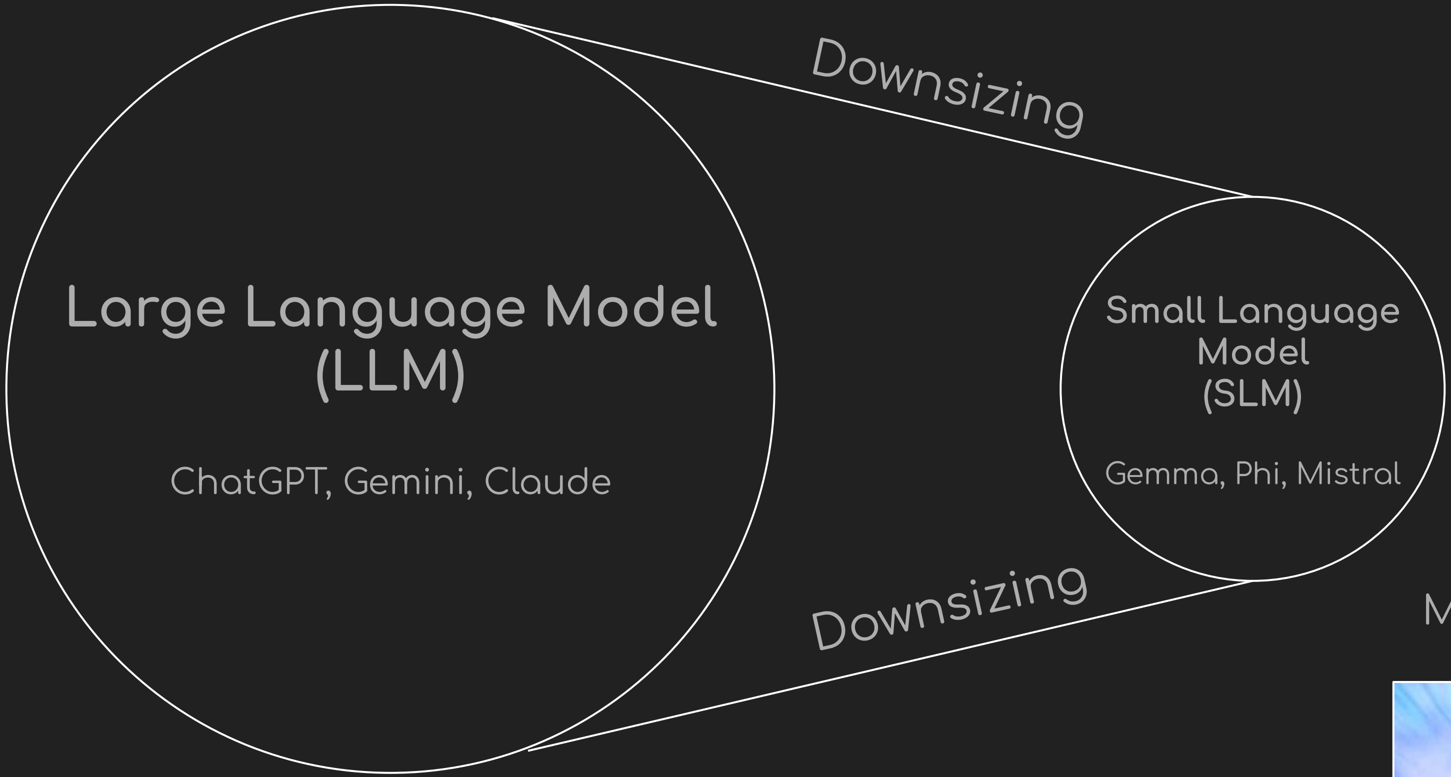
ChatGPT, Gemini, Claude

Downsizing

Downsizing

Small Language Model
(SLM)

Gemma, Phi, Mistral



Large Language Model (LLM)

ChatGPT, Gemini, Claude

Downsizing

Downsizing

Small Language Model (SLM)

Gemma, Phi, Mistral

Minimizer

Introducing

GitHub Models

Hugging Face Killer?

What you need to know about GitHub Models..?

GitHub Models aimed at enabling developers to become AI engineers by providing access to industry-leading AI models directly on GitHub.

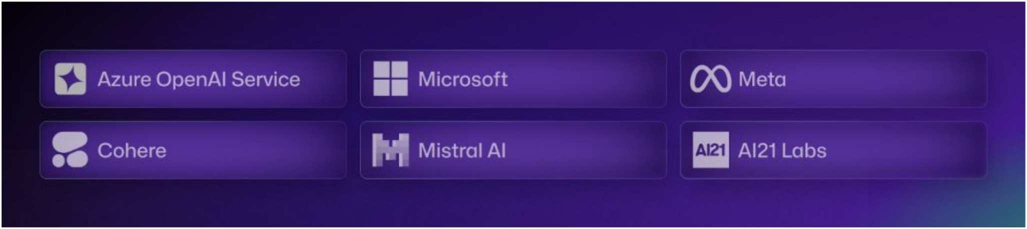
- **Access to AI Models**: experiment with models like Llama 3.1, GPT-4o, Phi 3, and Mistral Large 2 through a built-in playground.

- **Integration with Tools**: The models can be integrated into Codespaces, VS Code, and deployed via Azure AI.

- **Interactive Learning**: The platform offers an interactive playground for students, hobbyists, and startups to explore and test AI models.

- **Privacy and Security**: GitHub ensures that no prompts or outputs are shared with model providers or used to train the models.

List of available models →



Some Syntaxes

On using Github Models with JavaScript

```
import ModelClient from "@azure-rest/ai-inference";
import { AzureKeyCredential } from "@azure/core-auth";

const token = process.env["GITHUB_TOKEN"];
const endpoint = "https://models.inference.ai.azure.com";
const modelName = "Phi-3.5-mini-instruct";
```

```
export async function main() {

  const client = new ModelClient(endpoint, new AzureKeyCredential(token));

  const response = await client.path("/chat/completions").post({
    body: {
      messages: [
        { role:"system", content: "You are a helpful assistant." },
        { role:"user", content: "What is the capital of France?" }
      ],
      model: modelName,
      temperature: 1.,
      max_tokens: 1000,
      top_p: 1.
    }
  });
```

Learn more from here:

Azure Inference REST client library for JavaScript | Microsoft Learn

```
  if (response.status !== "200") {
    throw response.body.error;
  }
  console.log(response.body.choices[0].message.content);
}

main().catch((err) => {
  console.error("The sample encountered an error:", err);
});
```

Learn more from here:

[Azure Inference REST client library for JavaScript | Microsoft Learn]

Trip to Memory Lane

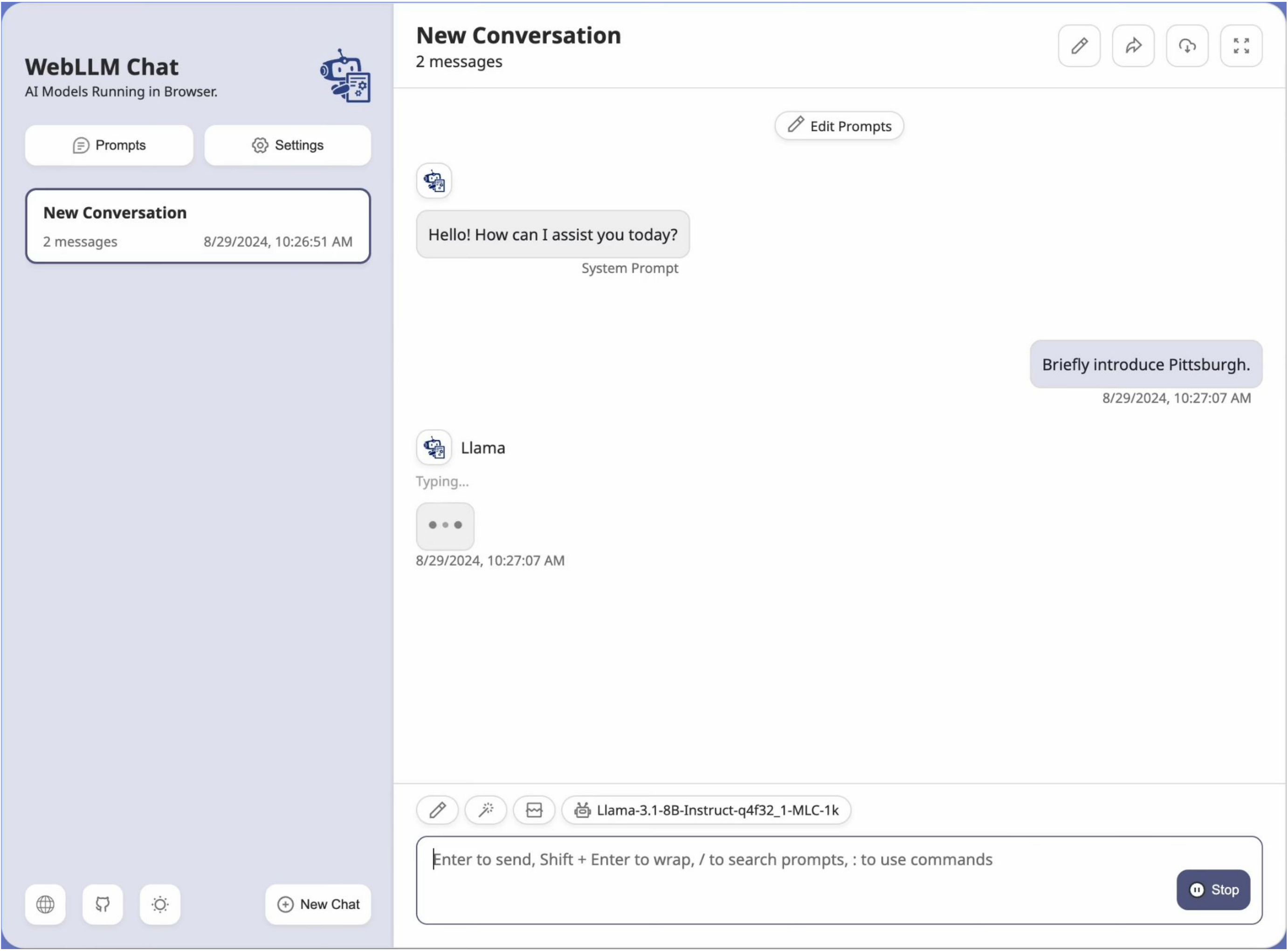
Transformer.js

In Hugging Face

MACHINE LEARNING COMPILATION

Experimenting **WebLLM Chat** in your browser + including model cache in the device?

For what?



WebLLM Chat
AI Models Running in Browser.

New Conversation
2 messages 8/29/2024, 10:26:51 AM

New Conversation
2 messages

Hello! How can I assist you today?
System Prompt

Briefly introduce Pittsburgh.
8/29/2024, 10:27:07 AM

Llama
Typing...
8/29/2024, 10:27:07 AM

Llama-3.1-8B-Instruct-q4f32_1-MLC-1k

Enter to send, Shift + Enter to wrap, / to search prompts, : to use commands

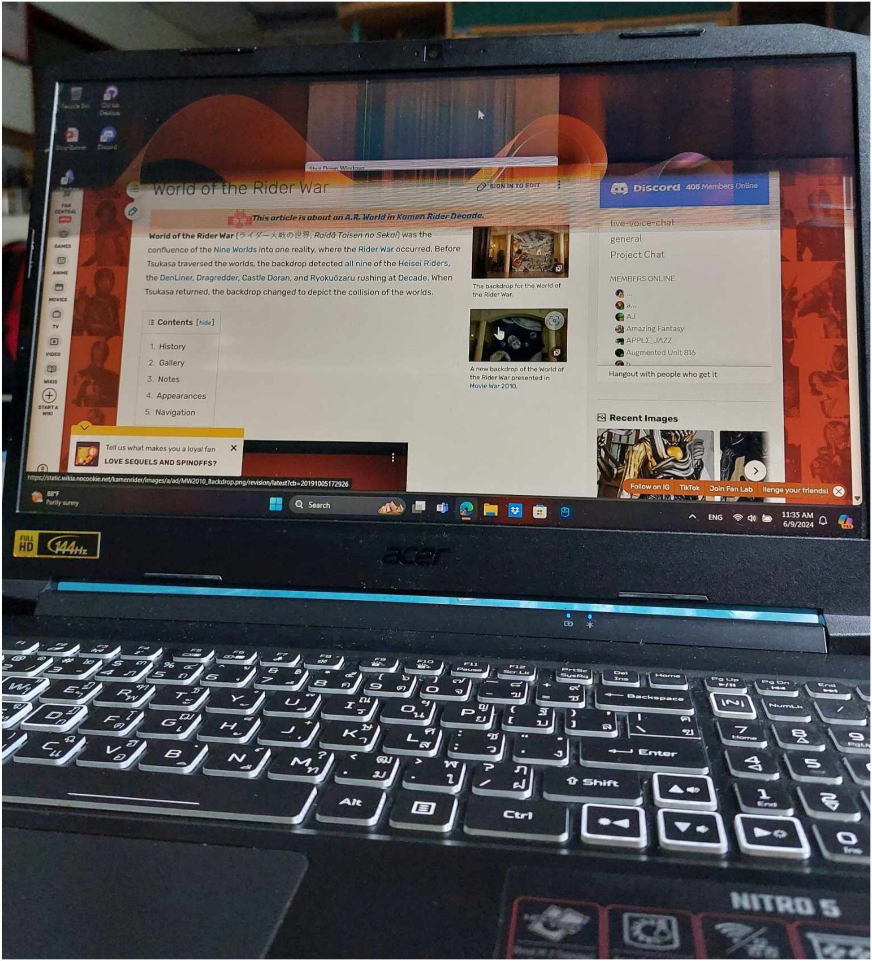
Phi-3-mini

WebGPU

A private and powerful AI chatbot that runs locally in your browser.

Powered by

Transformers.js



World of the Rider War

This article is about an A.R. World in Kamen Rider Decade.

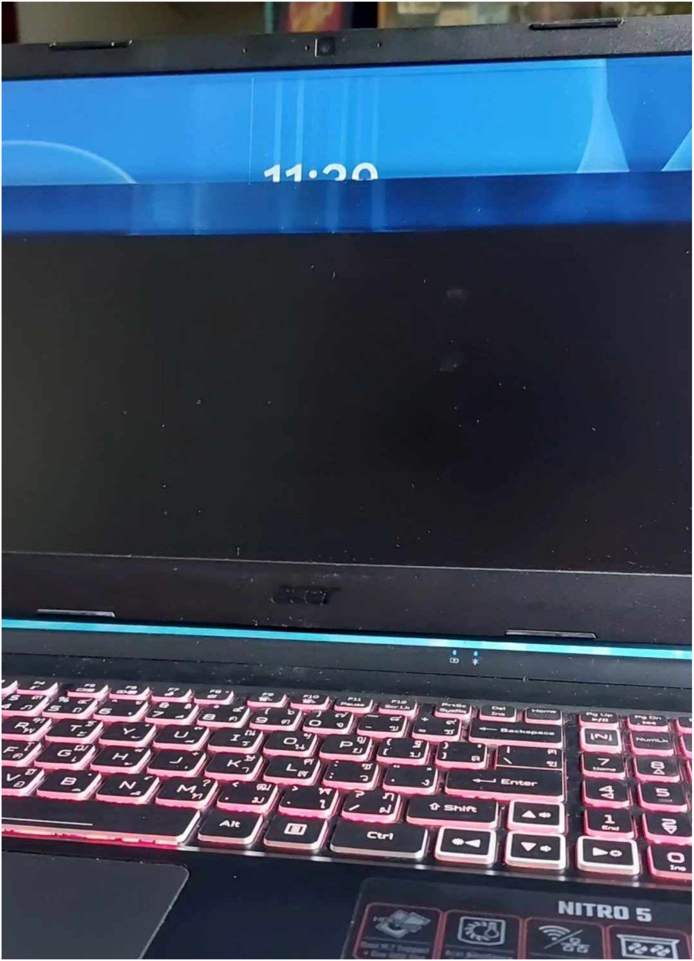
World of the Rider War (ライダー大戦の世界, Raidā Taisen no Sekai) was the confluence of the Nine Worlds into one reality, where the Rider War occurred. Before Tsukasa traversed the worlds, the backdrop detected all nine of the Heisei Riders, the DenLiner, Dragredder, Castle Doran, and Ryukōzaru rushing at Decade. When Tsukasa returned, the backdrop changed to depict the collision of the worlds.

The backdrop for the World of the Rider War.

A new backdrop of the World of the Rider War presented in Movie War 2010.

Contents [hide]

1. History
2. Gallery
3. Notes
4. Appearances
5. Navigation



Some Syntax

Transformer.js in Hugging Face

app.js

```javascript
import { useEffect, useState, useRef } from 'react';

import Chat from './components/Chat';
import ArrowRightIcon from
'./components/icons/ArrowRightIcon';
import StopIcon from './components/icons/StopIcon';
import Progress from './components/Progress';

const IS_WEBGPU_AVAILABLE = !!navigator.gpu;
const STICKY_SCROLL_THRESHOLD = 120;
```

app.js

```
// Create a reference to the worker object.
  const worker = useRef(null);

  const textareaRef = useRef(null);
  const chatContainerRef = useRef(null);

  // Model loading and progress
  const [status, setStatus] = useState(null);
  const [loadingMessage, setLoadingMessage] = useState('');
  const [progressItems, setProgressItems] = useState([]);
  const [isRunning, setIsRunning] = useState(false);

  // Inputs and outputs
  const [input, setInput] = useState('');
  const [messages, setMessages] = useState([]);
  const [tps, setTps] = useState(null);
  const [numTokens, setNumTokens] = useState(null);
```

app.js

```javascript
function onEnter(message) {
  setMessages(prev => [
    ...prev,
    { "role": "user", "content": message },
  ]);
  setTps(null);
  setIsRunning(true);
  setInput('');
}

useEffect(() => {
  resizeInput();
}, [input]);
```

app.js

```
  // We use the `useEffect` hook to setup the worker as
soon as the `App` component is mounted.
  useEffect(() => {
    if (!worker.current) {
      // Create the worker if it does not yet exist.
      worker.current = new Worker(new URL('./worker.js',
import.meta.url), {
        type: 'module'
      });
    }
```

worker.js

```
import {
    AutoTokenizer,
    AutoModelForCausalLM,
    TextStreamer,
    StoppingCriteria,
} from '@xenova/transformers';
```

worker.js

```
class TextGenerationPipeline {
    static model_id = null;
    static model = null;
    static tokenizer = null;
    static streamer = null;

    static async getInstance(progress_callback = null) {
        // Choose the model based on whether fp16 is available
        this.model_id ??= (await hasFp16())
            ? 'Xenova/Phi-3-mini-4k-instruct_fp16'
            : 'Xenova/Phi-3-mini-4k-instruct';

        this.tokenizer ??= AutoTokenizer.from_pretrained(this.model_id, {
            legacy: true,
            progress_callback,
        });
```

worker.js

```
this.model ??= AutoModelForCausalLM.from_pretrained(this.model_id, {
            dtype: 'q4',
            device: 'webgpu',
            use_external_data_format: true,
            progress_callback,
        });

        return Promise.all([this.tokenizer, this.model]);
    }
}
```

Quantization

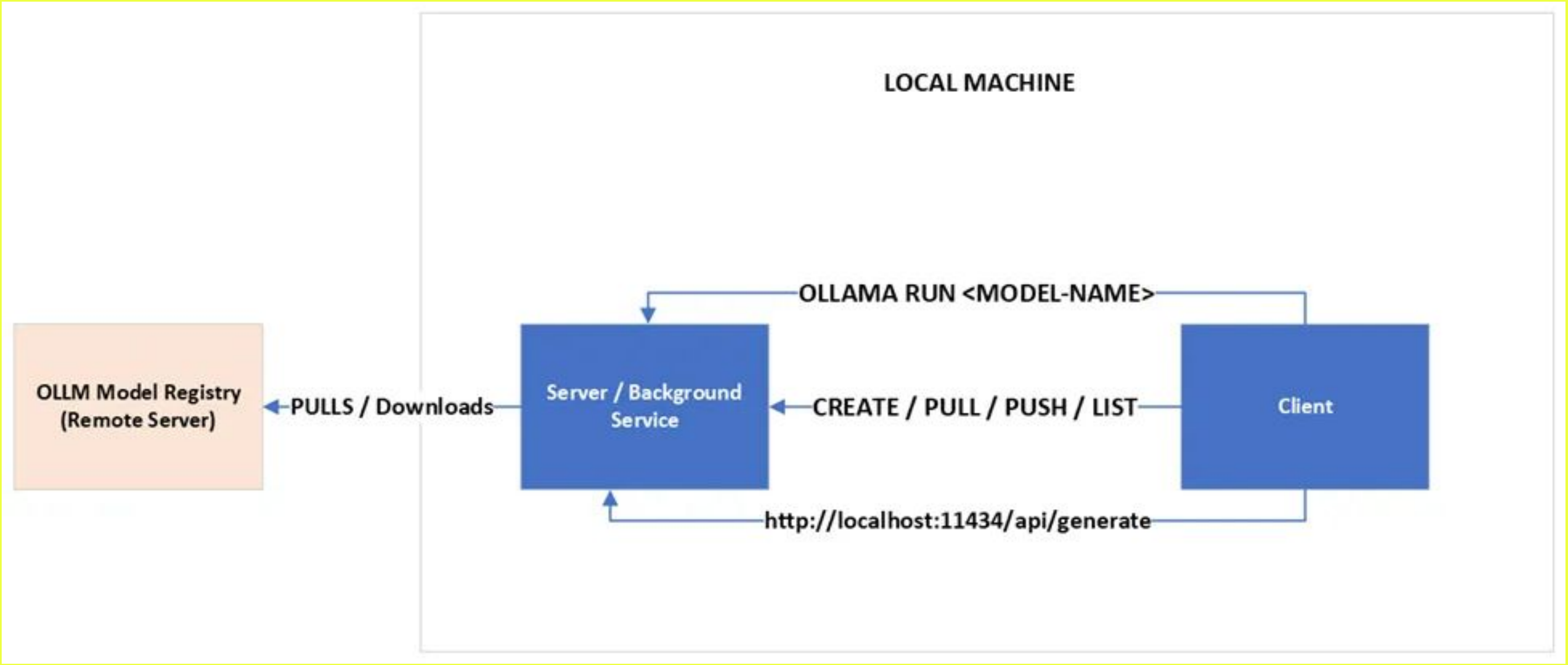
Why Quantization?

- Reduce their size and computational cost. This is especially important for deploying models on devices with limited resources, such as mobile phones or embedded systems.
- By converting high-precision floating-point numbers to lower-precision integers, we can significantly reduce the memory footprint and accelerate inference time.
- While quantization may introduce some loss of accuracy, it often provides a good trade-off between performance and model size, making it a valuable technique for real-world applications.

Introducing

Ollama.js

In Ollama - the local AI platform in your device



LOCAL MACHINE

OLLM Model Registry
(Remote Server)

PULLS / Downloads

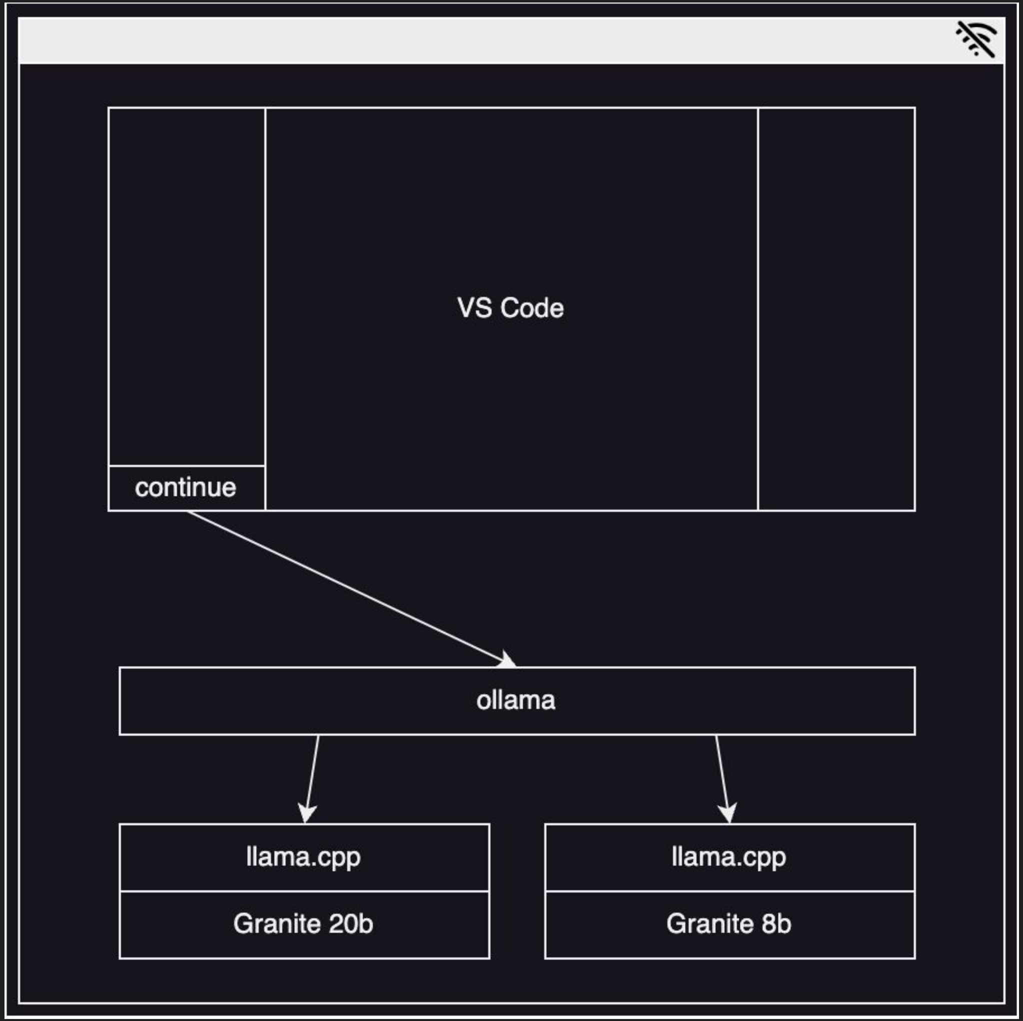
Server / Background
Service

OLLAMA RUN <MODEL-NAME>

CREATE / PULL / PUSH / LIST

http://localhost:11434/api/generate

Client


VS Code
continue
ollama
llama.cpp
Granite 20b
llama.cpp
Granite 8b

```
import { OpenAI } from "openai";

const openai = new OpenAI({
    baseURL: "http://localhost:11434/v1",
    apiKey: "__not_needed_by_ollama__",
});

const chunks = await openai.chat.completions.create({
    model: "phi3.5",
    messages: [{ role: "user", content: }]
})
```

Short Demo

For Ollama.js in Visual Studio Code

https://github.com/chrnthnkmutt/phi3.5-js-experiment

Is it unacceptable to use JavaScript on AI Development?

It isn't...but you might need to consider something...

Comparing Python and JavaScript in developing AI

Feature	Python	JavaScript
Popularity	Widely used for AI, machine learning, and data science	Increasingly popular, especially for web-based AI applications
Ecosystem	Extensive libraries and frameworks (NumPy, TensorFlow, PyTorch, Scikit-learn)	Growing ecosystem with libraries like TensorFlow.js and Keras.js
Performance	Generally faster for computationally intensive tasks	Can be optimized for performance but may lag behind Python for certain applications
Ease of Use	Readable syntax, making it easier for beginners	Can be more complex for beginners due to asynchronous programming
Web Integration	Requires additional tools (e.g., Flask, Django)	Built-in web capabilities, making it more suitable for web-based AI
Mobile Development	Can be used with frameworks like Kivy	Directly compatible with mobile platforms (iOS, Android)
Community Support	Large and active community	Growing community, especially for web-based AI

The Possibilities of Running AI on JavaScript/TypeScript with Open-Source Models

Three-minute blog.

Both Thai and English version available at Medium

Microsoft Learn
Student Ambassadors

X @boatchrnthn

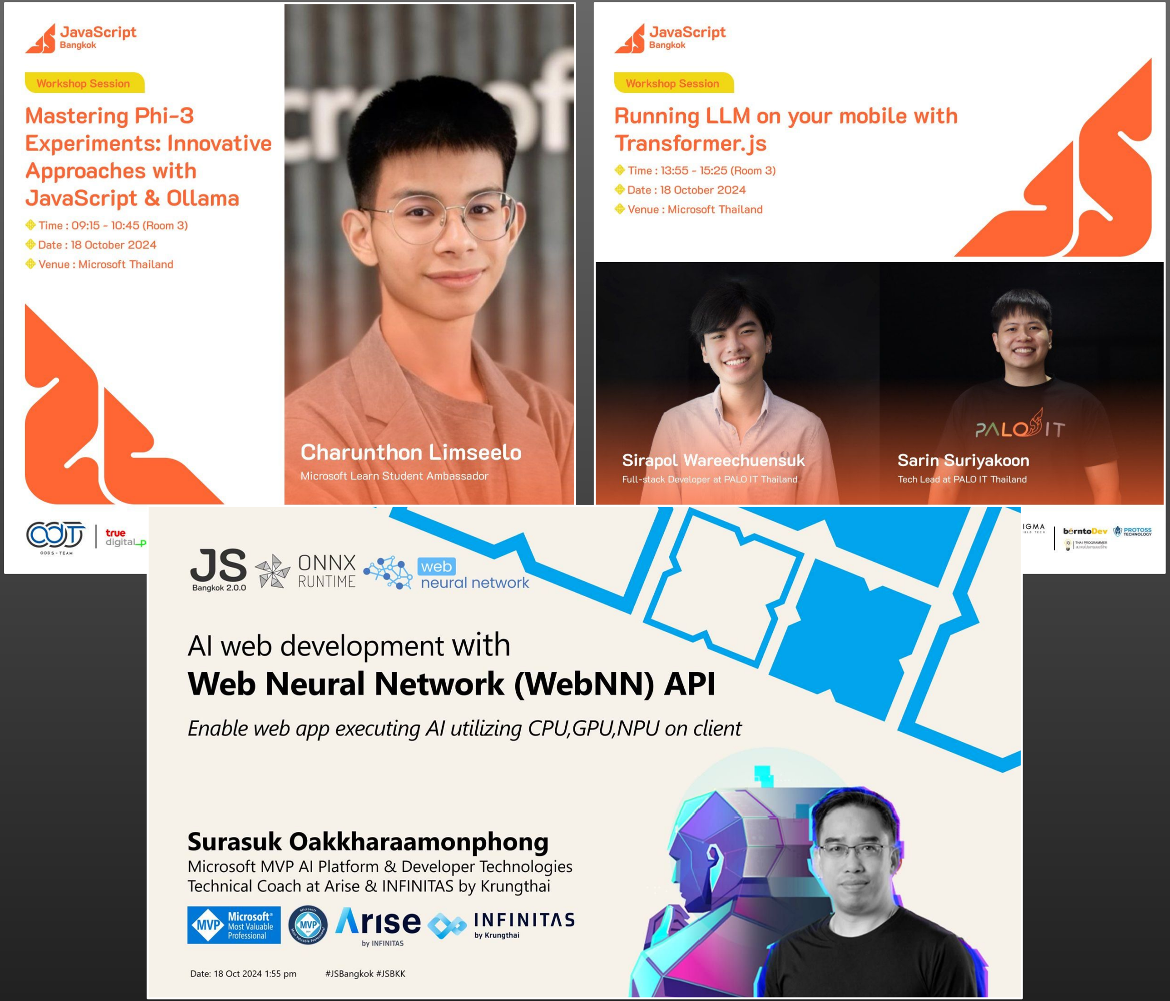
Sessions Schedule

October 19th (Next)	JavaScript Bangkok 2.0.0 : Mastering Phi3.5 Experiment (Workshop Session)
November 8th	Microsoft: Season of AI Episode 2 – Copilots Topic: GitHub Codespace/GitHub Copilot (including Ollama Environment)
(TBC) November 30th	National Coding Day (Conference Day - 30-min section)

*Apart from this, everyone is feel free to contact or invite me as a guest speaker...

The field would be around fundamental open-source AI development for students, educators, open-source developers, office workers, along with engaging on Small Language Models for environmental sustainability and cost optimization, with trends of AI in nowadays :)

AI Workshop Sessions at Microsoft in JavaScript Bangkok 2.0.0



JavaScript Bangkok

Workshop Session

Mastering Phi-3 Experiments: Innovative Approaches with JavaScript & Ollama

- Time : 09:15 - 10:45 (Room 3)
- Date : 18 October 2024
- Venue : Microsoft Thailand

Charunthon Limseelo

Microsoft Learn Student Ambassador

JavaScript Bangkok

Workshop Session

Running LLM on your mobile with Transformer.js

- Time : 13:55 - 15:25 (Room 3)
- Date : 18 October 2024
- Venue : Microsoft Thailand

Sirapol Wareechuensuk

Full-stack Developer at PALO IT Thailand

Sarin Suriyakoon

Tech Lead at PALO IT Thailand

JS Bangkok 2.0.0 ONNX RUNTIME web neural network

AI web development with

Web Neural Network (WebNN) API

Enable web app executing AI utilizing CPU,GPU,NPU on client

Surasuk Oakkharaamonphong

Microsoft MVP AI Platform & Developer Technologies

Technical Coach at Arise & INFINITAS by Krungthai

Date: 18 Oct 2024 1:55 pm #JSBangkok #JSBKK

Microsoft

Season of AI

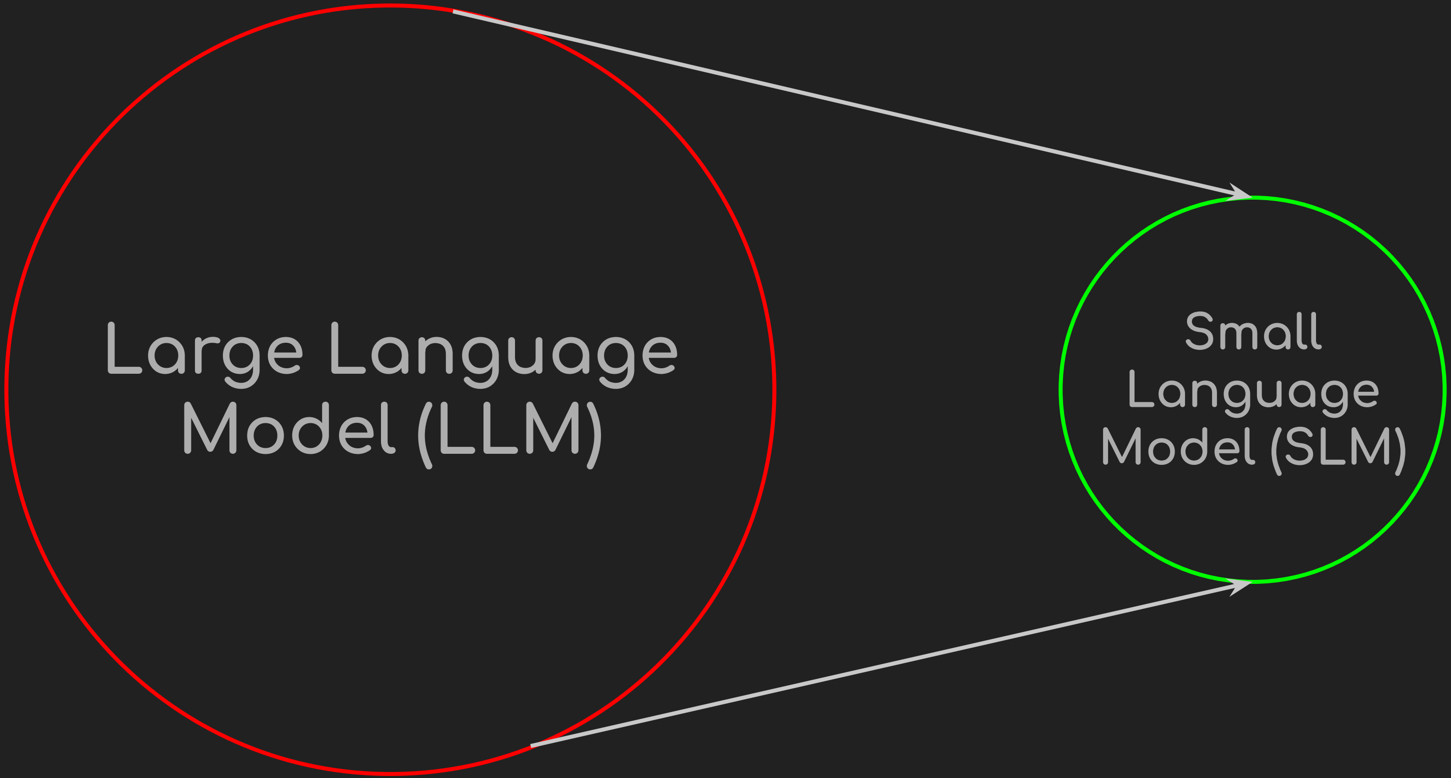
Microsoft Community Pilot

Discussion Session at Microsoft in Season of AI Ep.2: Copilot

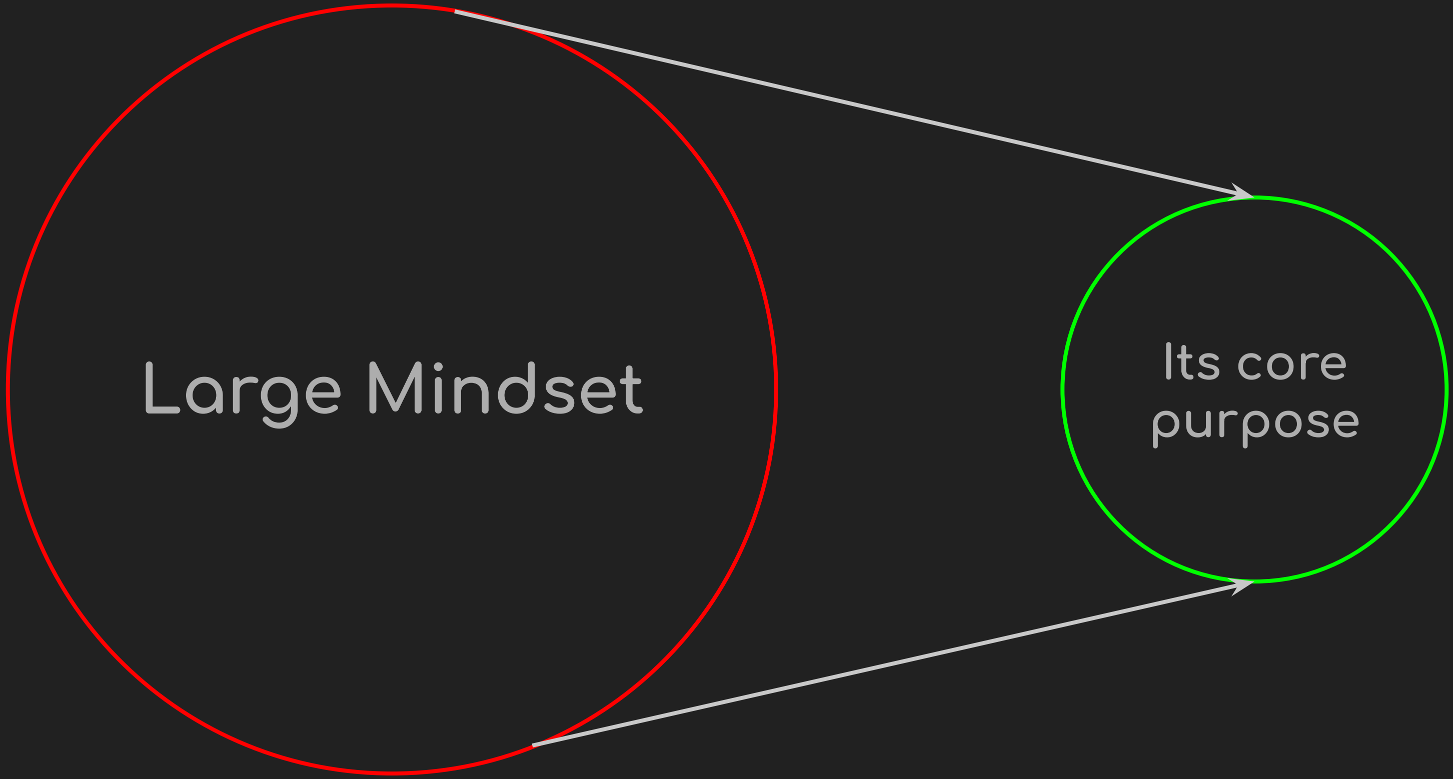
Along with

Kittikorn Prasertsak (P.Prem) From

BORNTO DEV



We all know that Large Language Model almost know everything and have more performance to get information from human…but sometimes they might lose some specific concepts of the content to specialize on…



Just like our life, we might have very large mindset to do lots of things you've dreamed. However, we might lose the thing really important or the main purpose of it…Try to do small first and grow with something to be better one…Like you make fine tuning or RAG thing

Every little thing you do leads up to a bigger thing.

Happy Developing Small Open-source AI with JavaScript!

Brendan Eich