



t3chfest

Everything You Need to Know About Running LLMs Locally

T3chfest Madrid: 2025

Cedric Clyburn

Senior Developer Advocate

@cedricclyburn

Roberto Carratalá

Principal AI Architect

@rcarrata_

We've got **a lot** to cover today!

We've got **a lot** to cover today!



Wait, so you can
run your own
language models...
completely local?



We've got **a lot** to cover today!



Wait, so you can
run your own
language models...
completely local?



LLM



ramalama



and there are plenty of
open source tools to do so?!

We've got **a lot** to cover today!



Wait, so you can
run your own
language models...
completely local?



and there are plenty of
open source tools to do so?!

A screenshot of the Hugging Face Model Hub website. At the top, it says 'Hugging Face' with a yellow smiley face icon and a search bar. Below that, it shows 'Models 1,492,075' with 'Filter by name' and 'Full-text search' buttons. The main area lists several models:

- Qwen/QwQ-32B** (by deepseek-ai) - Text Generation, Updated about 13 hours ago, 132k interactions, 1.86k stars
- deepseek-ai/DeepSeek-R1** - Text Generation, Updated 15 days ago, 3.43M interactions, 11.1k stars
- microsoft/Phi-4-multimodal-instruct** - Automatic Speech Recognition, Updated 3 days ago, 303k interactions, 1.07k stars

A large black curved arrow points from the text 'But there's over 1.5 million models, which to pick?' at the bottom to this list of models.

But there's over 1.5 million
models, which to pick?

We've got a lot to cover today!



Wait, so you can
run your own
language models...
completely local?

or how can i use my own PDF's
or codebase or API's?

I will give you a couple of documents for T3chFest, do a RAG with them and try to answer.
Give me some session names of the conference.:)

From the provided documents, here are some session names of the conference:

- 01. ¿Están tus datos listos para la era cuántica? La criptografía del futuro
- 02. Superbacterias bajo la lupa de la IA; el futuro de los diagnósticos microbianos
- 03. Surfando el tsunami de desinformación: La contra el caos informativo
- 04. API Catalogue: De una idea a una plataforma escalable
- 05. Cosas que aprendí procesando millones de peticiones por segundo
- 06. Servicio de Ubicación para Emergencias en catástrofes
- 07. La era digital oscura y la biblioteca de internet
- 08. La computación del futuro

These sessions cover a variety of topics, including quantum computing, AI in diagnostics, combating misinformation, API platforms, high-traffic processing, emergency location services, and the future of computing.



and there are plenty of
open source tools to do so?!



Hugging Face

Models 1,492,075

Qwen/QwQ-32B Text Generation • Updated about 13 hours ago • ↓ 132k • ⚡ 1.86k
deepseek-ai/DeepSeek-R1 Text Generation • Updated 15 days ago • ↓ 3.43M • ⚡ 11.1k
microsoft/Phi-4-multimodal-instruct Automatic Speech Recognition • Updated 3 days ago • ↓ 303k • ❤ 1.07k

But there's over 1.5 million
models, which to pick?

Today's Schedule

- ▶ Running **your own AI & LLMs**
- ▶ How to **choose the right model?**
- ▶ Integrating your **data & codebase!**
- ▶ **Demo #1:** Model serving & RAG
- ▶ **Demo #2:** Code assistance & agents
- ▶ **Demo #3:** Adding AI features to apps



Session Slides
red.ht/local-llm

Why's everyone running their **own** AI models?

Why run a model locally?

Take advantage of total AI customization and control



For Developers

Convenience & Simplicity

Familiarity with the Development Environment and adherence of the developers to their "local developer experience" in particular for testing and debugging

Ease of Integration

Simplify the integration of the model with existing systems and applications that are already running locally.

Customization & Control

Easily train or fine-tune your own model, from the convenience of the developer's local machine.

For Organizations

Data Privacy and Security

Data is the fuel for AI, and a differentiator factor (quality, quantity, qualification). Keeping data on-premises ensures sensitive information doesn't leave the local environment → crucial for privacy-sensitive applications

Cost Control

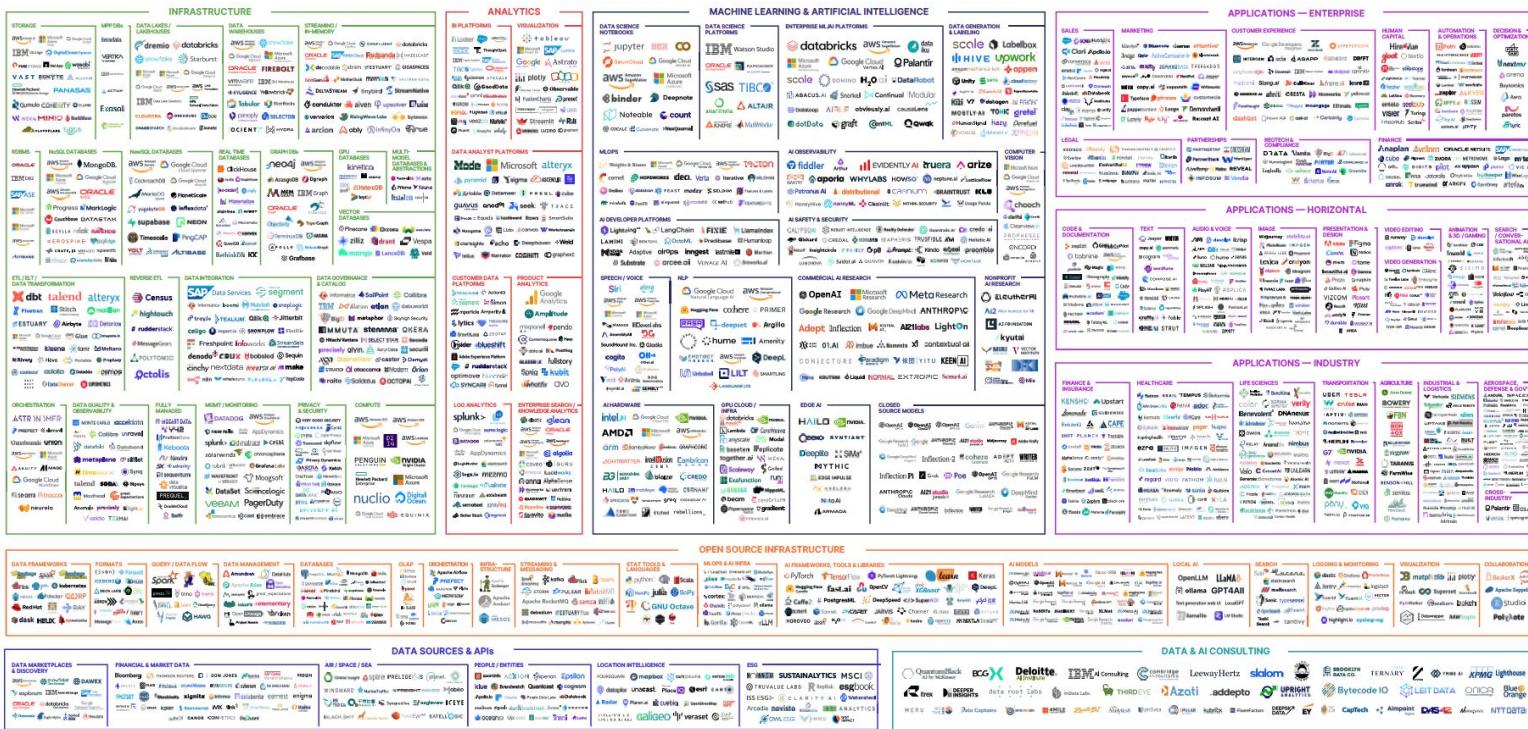
While there is an initial investment in hardware and setup, running locally can potentially reduce ongoing costs of cloud computing services and alleviate the vendor-locking played by Amazon, MSFT, Google

Regulatory Compliance

Some industries have strict regulations about where and how data is processed

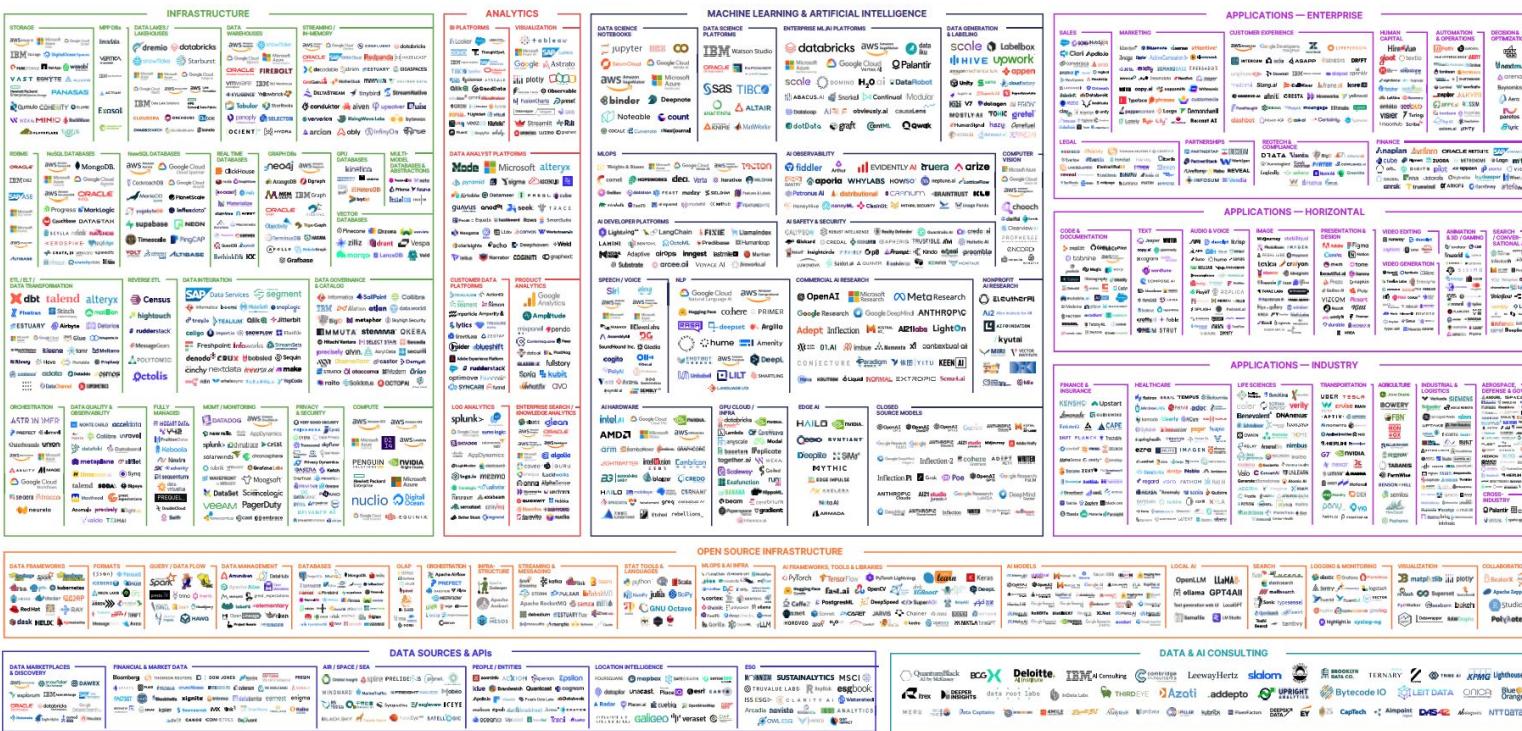


But the stack can be a bit overwhelming!



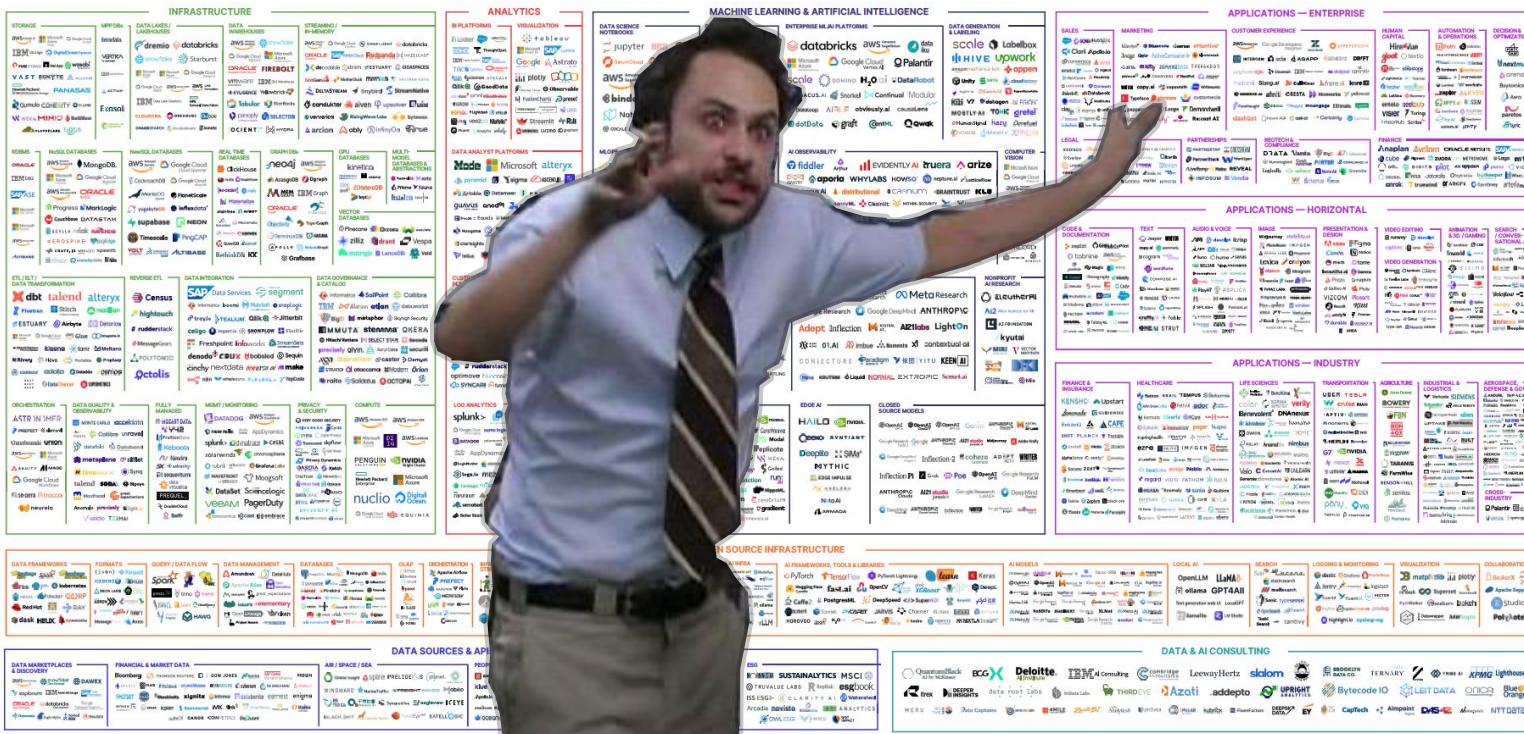
2024 MAD (Machine learning, Artificial Intelligence & Data) Landscape

But the stack can be a **bit** overwhelming!



2024 MAD (Machine learning, Artificial Intelligence & Data) Landscape

But the stack can be a **bit** overwhelming!







Average developer trying to download & manage models, configure serving runtimes, quantize and compress LLMs, ensure correct prompt templates... (Colorized, 2023)

So, what **open source**
tech can help us run AI?

Fortunately, there's a lot... for **every use case!**



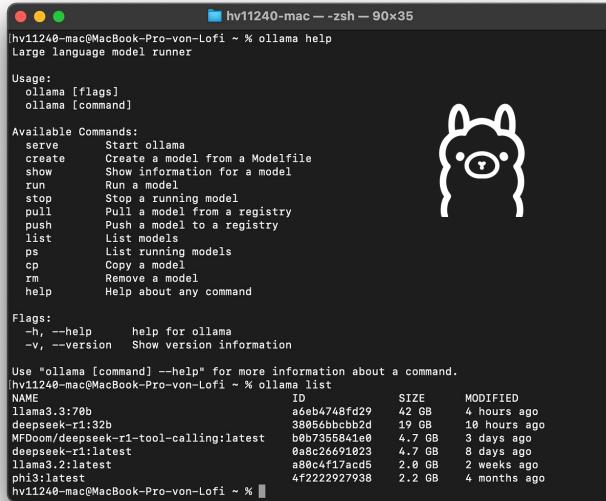
Fortunately, there's a lot... for **every use case!**



Tool #1: Ollama

For simple model downloading & serving

- ▶ **Simple CLI:** “Docker” style tool for running LLMs locally, offline, and privately
- ▶ **Extensible:** Basic model customization (Modelfile) and importing of fine-tuned LLMs
- ▶ **Lightweight:** Efficient and resource-friendly.
- ▶ **Easy API:** API for both inferencing and Ollama itself (ex. download models)



```
[hv11240-mac@MacBook-Pro-von-Lofi ~ % ollama help
Large language model runner

Usage:
  ollama [flags]
  ollama [command]

Available Commands:
  serve      Start ollama
  create     Create a model from a Modelfile
  show       Show information for a model
  run        Run a model
  stop       Stop a running model
  pull       Pull a model from a registry
  push       Push a model to a registry
  list       List models
  ps         List running models
  cp         Copy a model
  rm         Remove a model
  help       Help about any command

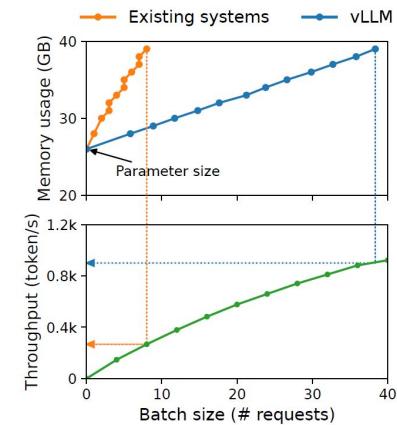
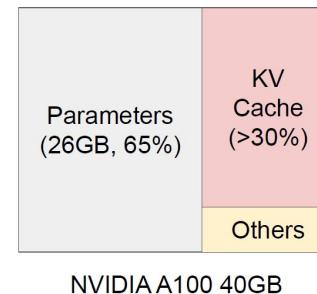
Flags:
  -h, --help   help for ollama
  -v, --version Show version information

Use "ollama [command] --help" for more information about a command.
[hv11240-mac@MacBook-Pro-von-Lofi ~ % ollama list
NAME                           ID          SIZE    MODIFIED
llama3.3:70b                   a6eb4748fd29  42 GB   4 hours ago
deepeekr-r1:32b                38056bbcb2d  19 GB   10 hours ago
MFDoom/deepeekr-r1-tool-calling:latest b0b7355841e0  4.7 GB   3 days ago
deepeekr-r1:latest              0a8c266991023 4.7 GB   8 days ago
llama3.2:latest                 a80c4cf17acd5 2.0 GB   2 weeks ago
phi3:latest                     4f2222927938  2.2 GB   4 months ago
hv11240-mac@MacBook-Pro-von-Lofi ~ % ]
```

Tool #2: vLLM

For scaling things up in production environments

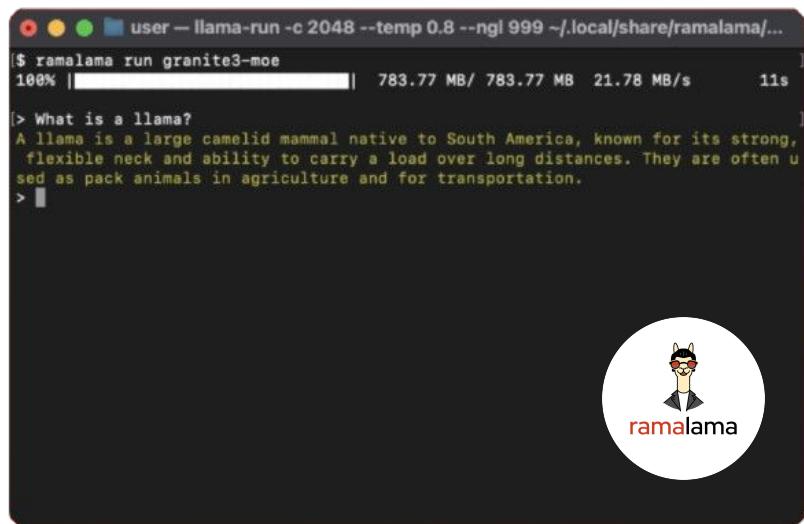
- ▶ **Research-Based:** UC Berkeley project to improve model speeds and GPU consumption
- ▶ **Standardized:** Works with Hugging Face & OpenAI API.
- ▶ **Versatile:** Supports NVIDIA, AMD, Intel, TPUs & more.
- ▶ **Scalable:** Manages multiple requests efficiently, ex. with Kubernetes as an LLM runtime



Tool #3: Ramalama

To make AI boring by using containers

- ▶ **AI in Containers:** Run models with Podman/Docker with no config needed.
- ▶ **Registry Agnostic:** Freedom to pull models from Hugging Face, Ollama, or OCI registries.
- ▶ **GPU Optimized:** Auto-detect & accelerate performance.
- ▶ **Flexible:** Supports llama.cpp, vLLM, whisper.cpp & more.



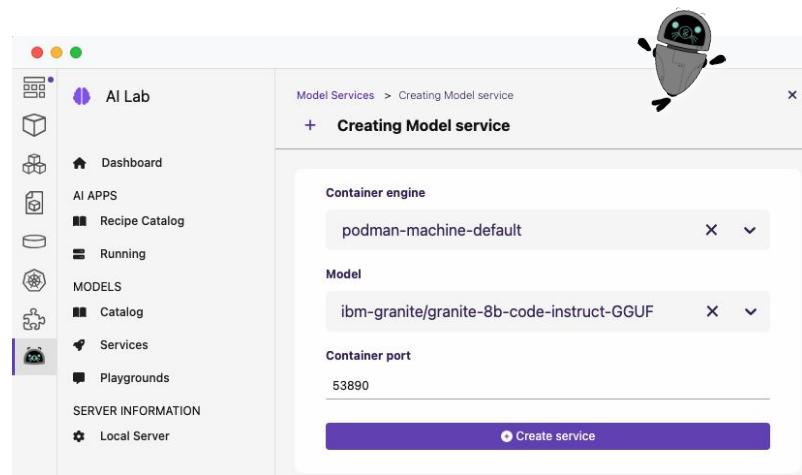
```
$ ramalama run granite3-moe
100% | 783.77 MB/ 783.77 MB  21.78 MB/s  11s
> What is a llama?
A llama is a large camelid mammal native to South America, known for its strong, flexible neck and ability to carry a load over long distances. They are often used as pack animals in agriculture and for transportation.
>
```



Tool #4: Podman AI Lab

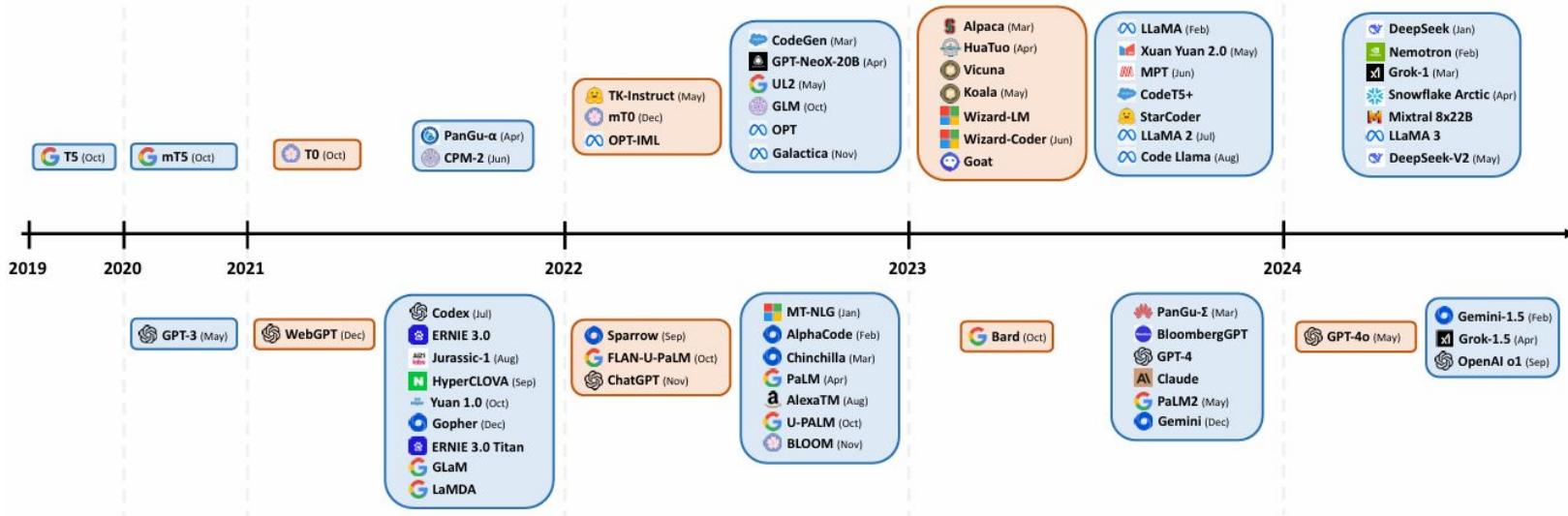
For developers looking to build AI features

- ▶ **For App Builders:** Choose from various recipes like RAG, Agentic, Summarizers
- ▶ **Curated Models:** Easily access Apache 2.0 open-source options.
- ▶ **Container Native:** Easy app integration and movement from local to production.
- ▶ **Interactive Playgrounds:** Test & optimize models with your custom prompts and data.



Cool! But, what specific
model should I be using?

There are plenty of open, and closed model choices!



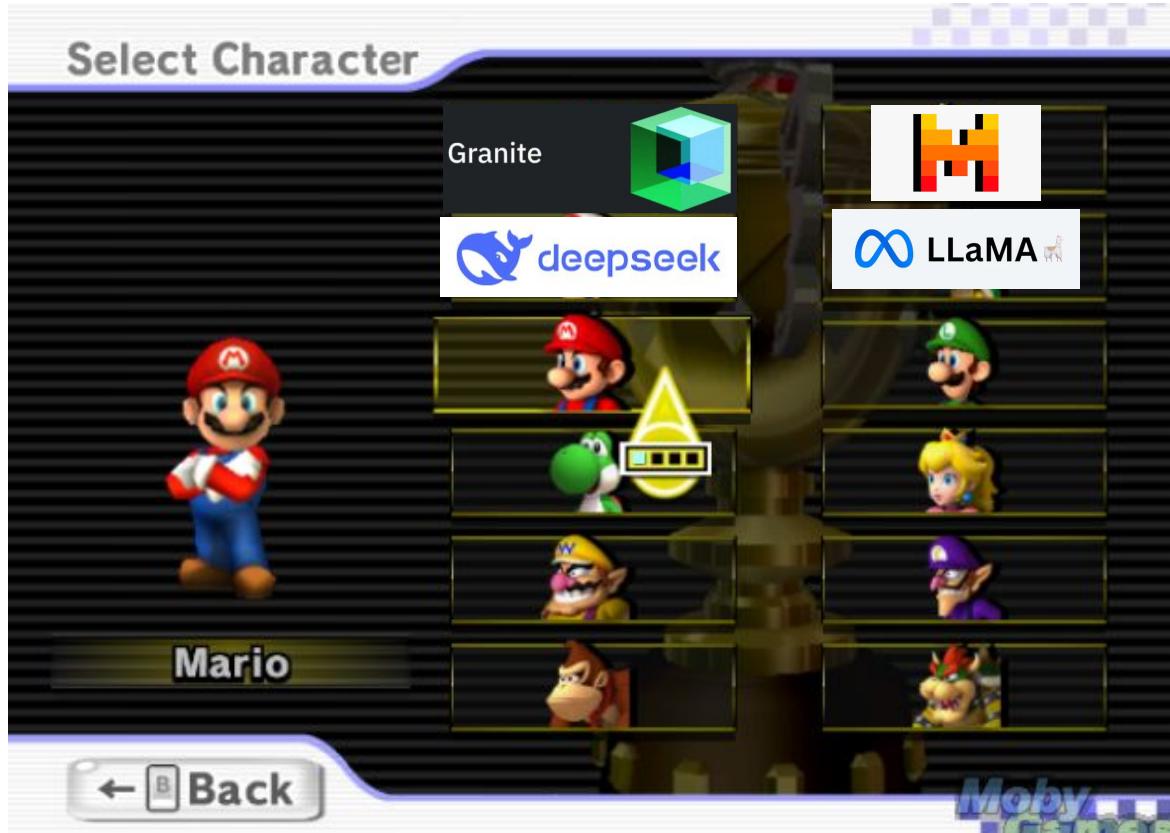
Darn, we're back to here again!



But again, we'll use another video game analogy!



But again, we'll use another video game analogy!

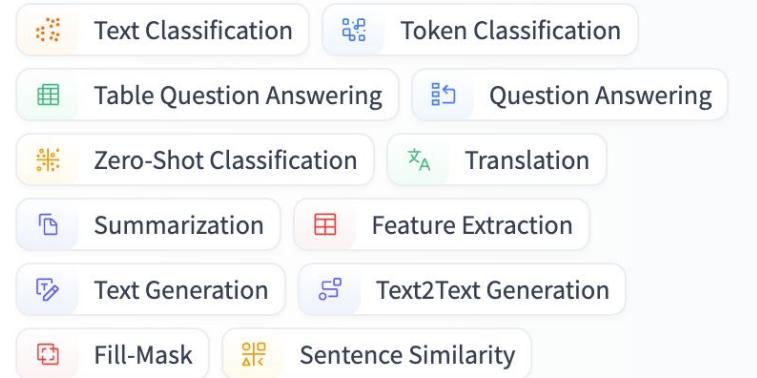


So, which model should you select?

Well... it depends!

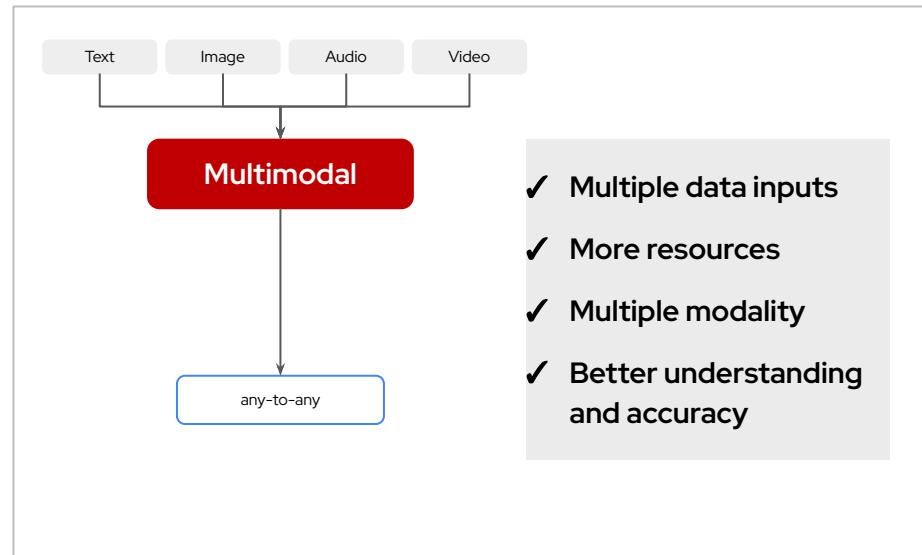
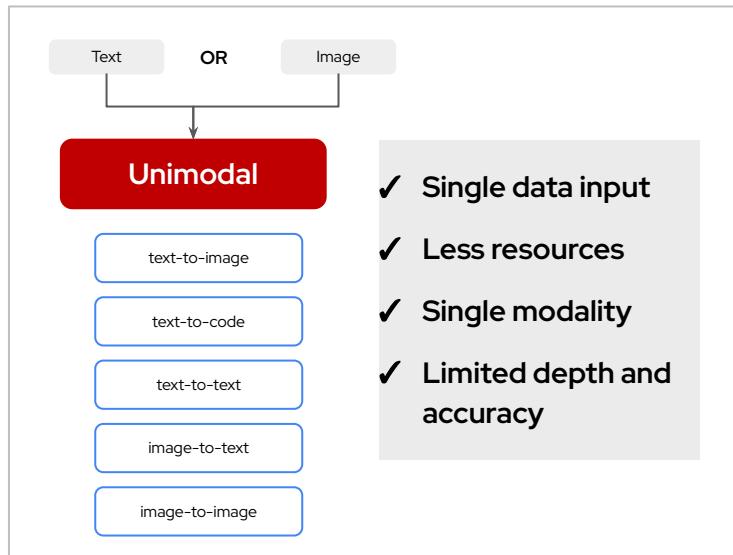
- ▶ It depends on **the use case** that you want to tackle.
- ▶ **DeepSeek** models excel in reasoning tasks and complex problem-solving.
- ▶ **Granite** SLM models perform well in various NLP tasks and multimodal applications.
- ▶ **Mistral** and **LLaMA** are particularly strong in summarization and sentiment analysis.

Natural Language Processing



But not all models are the same!

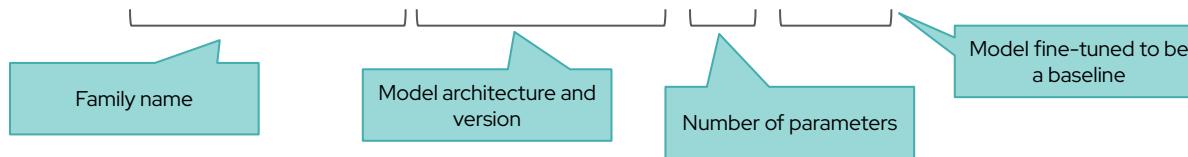
Our data isn't always in one format, it's text, image, audio, etc.



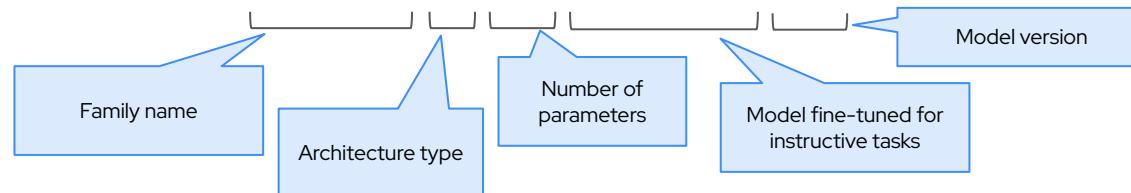
Also! There's a naming convention

Kind of like how our apps are compiled for various architectures!

ibm-granite/granite-3.0-8b-base



Mixtral-8x7B-Instruct-v0.1

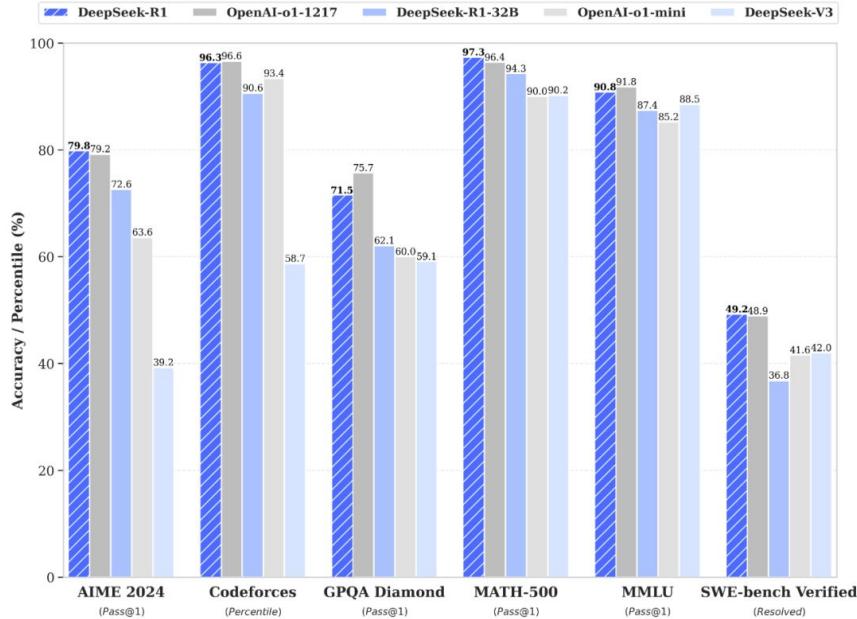


What about model size?



How to deploy a larger model?

For example, DeepSeek-R1 or Llama 3.3-405B



Let's say you
want the best
benchmarks with
a frontier model



How to deploy a larger model?

For example, DeepSeek-R1 or Llama 3.3-405B

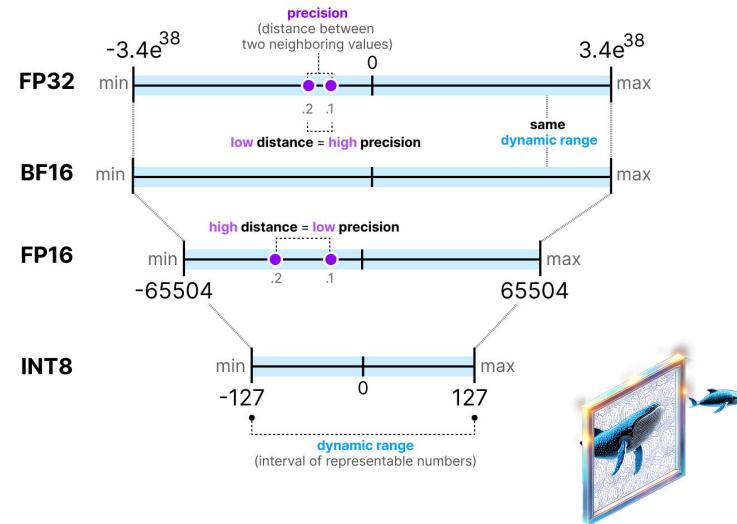


Neither of these situations
is ideal :)

Well, most models for local usage are quantized!

It's a way to compress models, think like a .zip or .tar

- ▶ **Quantization:** A technique to compress LLMs by reducing numerical precision.
- ▶ Converts high-precision weights (FP32) into lower-bit formats (FP16, INT8, INT4).
- ▶ **Reduces memory footprint**, making models easier to deploy.



Well, most models for local usage are quantized!

It's a way to compress models, think like a .zip or .tar

- ▶ **The Benefit?** Run LLMs on “any” device, not just your local machine but IoT & Edge too
- ▶ Results in **faster and lighter models** that still maintain reasonable accuracy
 - Testing with Llama 3.1, for W4A16-INT resulted in **2.4x performance speedup** and **3.5x model size compression**
- ▶ Works on **GPUs & CPUs!**





& there's a open repository of Quantized Models

Check it out on Hugging Face, & save resources on LLM serving!

Broad Collection



Llama



Qwen



Gemma



Mistral



DeepSeek



Phi



Molmo



Granite



Nemotron

Comprehensive Validation

Open LLM Leaderboard evaluation scores

Benchmark	Meta-Llama-3.1-70B-Instruct	Meta-Llama-3.1-70B-Instruct-FP8(this model)	Recovery
MMLU (5-shot)	83.83	83.73	99.88%
MMLU-cot (0-shot)	86.01	85.44	99.34%
ARC Challenge (0-shot)	93.26	92.92	99.64%
GSM-8K-cot (8-shot, strict-match)	94.92	94.54	99.60%
Hellaswag (10-shot)	86.75	86.64	99.87%
Winogrande (5-shot)	85.32	85.95	100.7%
TruthfulQA (0-shot, mc2)	60.68	60.84	100.2%
Average	84.40	84.29	99.88%

Extensive Selection

Formats

- W4/8A16
- W8A8-INT8
- W8A8-FP8
- 2:4 sparse

Algorithms

- GPTQ / AWQ
- SmoothQuant
- SparseGPT
- RTN

Hardware



GPUs



TPUs



Instinct



CPUs

Cut GPU costs in half ready-to-deploy
inference-optimized checkpoints

AI Engine? **Check ✓**
AI Model? **Check ✓**
What about your data?

How can you integrate AI with your unique data?

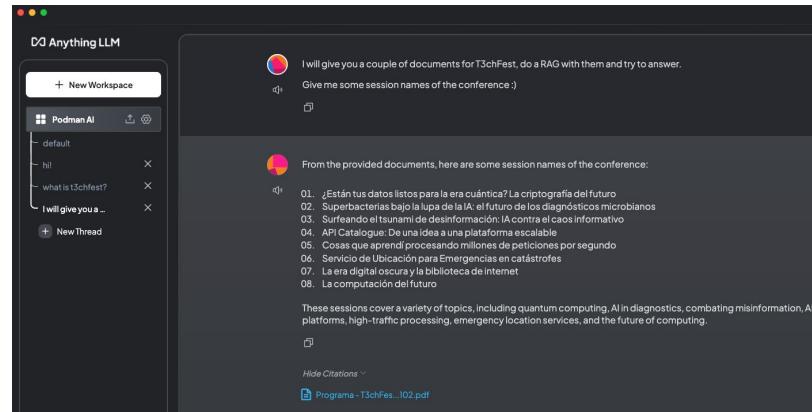
Fortunately, many tools exist for this too!

|O|O
|I|O|I

Data Interfaces

Pull in documents (PDF), web results, and agents together.

Tools: AnythingLLM,
OpenWebUI, LM Studio



Ask a question to a PDF & receive citations!



How can you integrate AI with your unique data?

Fortunately, many tools exist for this too!



Code Assistance

Use a model as a pair programmer, to generate and explain your codebase.

Tools: Continue, Cody, Cursor, Windsurf

The screenshot shows the Continue AI IDE interface. On the left, there's a sidebar with various icons. In the center, a floating window titled 'CONTINUE' displays a snippet of code from 'data.py' (lines 4-7):

```
def get_ds_dl(transform,
               batch_size,
               train_folder='Rock-Paper-
Scissors/train/',
               test_folder='Rock-Paper-
```

Below this, a message says 'explain me these lines of code'. A tooltip at the bottom of the window says 'Sure, I'd be happy to explain these lines of code.' To the right, the full 'data.py' file is shown in a code editor:

```
roshambo-notebooks > data.py ...
1 from torch.utils.data import DataLoader
2 from torchvision.datasets import ImageFolder
3
4 def get_ds_dl(transform,
5               batch_size,
6               train_folder='Rock-Paper-Scissors/train/',
7               test_folder='Rock-Paper-Scissors/test'):
8
9     ds_train = ImageFolder(train_folder, transform=transform)
10    ds_test = ImageFolder(test_folder, transform=transform)
11
12    dl_train = DataLoader(ds_train, batch_size=batch_size)
13    dl_test = DataLoader(ds_test, batch_size=batch_size)
14
15    return ds_train, ds_test, dl_train, dl_test
```

No more copy/pasting, it's part of the IDE!



How can you integrate AI with your unique data?

Fortunately, many tools exist for this too!



Prompting & Building Apps

Experiment with data, build
Proof of Concepts, and
integrate AI into apps.

Tools: Podman AI Lab,
Docker Gen AI Stack

Recipe Catalog
Natural Language Processing

Chatbot This recipe provides a blueprint for developers to create their own AI-powered chat applications using Streamlit. v1.5.0	Chatbot PydanticAI This recipe provides a blueprint for developers to create their own AI-powered chat applications with the pydantic framework using Streamlit. v1.5.0	Summarizer This recipe guides into creating custom LLM-powered summarization applications using Streamlit. v1.5.0
Code Generation This recipes showcases how to leverage LLM to build your own custom code generation application. v1.5.0	RAG Chatbot This application illustrates how to integrate RAG (Retrieval Augmented Generation) into LLM applications enabling to interact with your own documents. v1.5.0	Node.js RAG Chatbot This application illustrates how to integrate RAG (Retrieval Augmented Generation) into LLM applications written in Node.js enabling to interact with your own documents. v1.5.0
Java-based ChatBot (Quarkus) This is a Java Quarkus-based recipe demonstrating how to create an AI-powered chat applications. v1.5.0	Node.js based ChatBot This is a NodeJS based recipe demonstrating how to create an AI-powered chat applications. v1.5.0	Function calling This recipes guides into multiple function calling use cases, showing the ability to structure data and chain multiple tasks, using Streamlit. v1.5.0
Audio		
Computer Vision		

Starting points
for common AI
apps



How can you integrate AI with your unique data?

Fortunately, many tools exist for this too!



Data Interfaces

Pull in documents (PDF), web results, and agents together.

Tools: AnythingLLM, OpenWebUI, LM Studio



Code Assistance

Use a model as a pair programmer, to generate and explain your codebase.

Tools: Continue, Cody, Cursor, Windsurf



Prompting & Building Apps

Experiment with data, build Proof of Concepts, and integrate AI into apps.

Tools: Podman AI Lab, Docker Gen AI Stack





Demo Time!

Model serving & RAG

Code Repository URL
red.ht/t3chfest-demo



Demo Time!

Code assistants & agents

Code Repository URL
red.ht/t3chfest-demo



Code Repository URL
red.ht/t3chfest-demo

Demo Time!

Adding AI features to apps

Thank you! You're awesome!

- ▶ Running **your own AI & LLMs**
- ▶ How to **choose the right model?**
- ▶ Integrating **your data & codebase!**



Session Slides

red.ht/local-l1m



Connect on LinkedIn!

Thank you

Join the DevNation

Red Hat Developer serves the builders. The problem solvers who create careers with code. Let's keep in touch!

- Join Red Hat Developer at developers.redhat.com/register
- Follow us on any of our social channels
- Visit dn.dev/upcoming for a schedule of our upcoming events

Red Hat Developer

Build here. Go anywhere.



linkedin.com/company/red-hat



youtube.com/user/RedHatVideos



facebook.com/redhatinc



twitter.com/RedHat