Red Hat AI

Building with Open Source AI

A Crash Course: World Summit AI

Cedric Clyburn
Senior Developer Advocate
Red Hat AI
@cedricclyburn

Legare Kerrison
Developer Advocate
Red Hat AI
@legarekerrison

Red Hat

- open source
- cloud computing and automation
- **artificial intelligence** and **machine learning**

Open models haven't always been on-par with closed

For 2023 and the start of 2024, closed models dramatically outpaced open

Slide from Neural Magic Board Meeting in March 2023

Problem: Open-Source LLMs Lagging

State-of-the-art human-aligned models from OpenAI, Google, Meta are no longer open source.

Why ChatGPT Succeeds

- Big Models (=$$$)
- Lots of Data (=$$$)
- Human Alignment (quality data)

Open-source models are lagging significantly in quality.

# shots	BBH 3	MMLU 0/5	RAFT 5
OPT 1.3B	27.9	23.5/25.9	49.1†
OPT 30B	28.4	24.2/26.1	59.1†
OPT 175B	30.2	27.3/34.2	63.2†
T5 11B	29.5	-/25.9	–
PaLM 62B	37.4	-/55.1	–
PaLM 540B	49.1	-/71.3	–
OpenAI davinci	33.6	-/32.3	64.5
OPT-IML-Max 1.3B	26.5	34.9/29.5	55.9†
OPT-IML-Max 30B	30.9	46.3/43.2	69.3†
OPT-IML-Max 175B	35.7	49.1/47.1	79.3†
T0pp 11B	13.0	46.7/33.7	56.8†
FLAN-T5 11B	45.3	53.7/54.9	79.5†
FLAN-PaLM 62B	47.5	-/59.6	–
FLAN-PaLM 540B	57.9	-/73.5	–
OpenAI text-davinci-002	48.6	-/64.5	72.1
OpenAI text-davinci-003	50.9	-/74.2	–
OpenAI code-davinci-002	52.8	-/77.4	–

Open source

Closed source

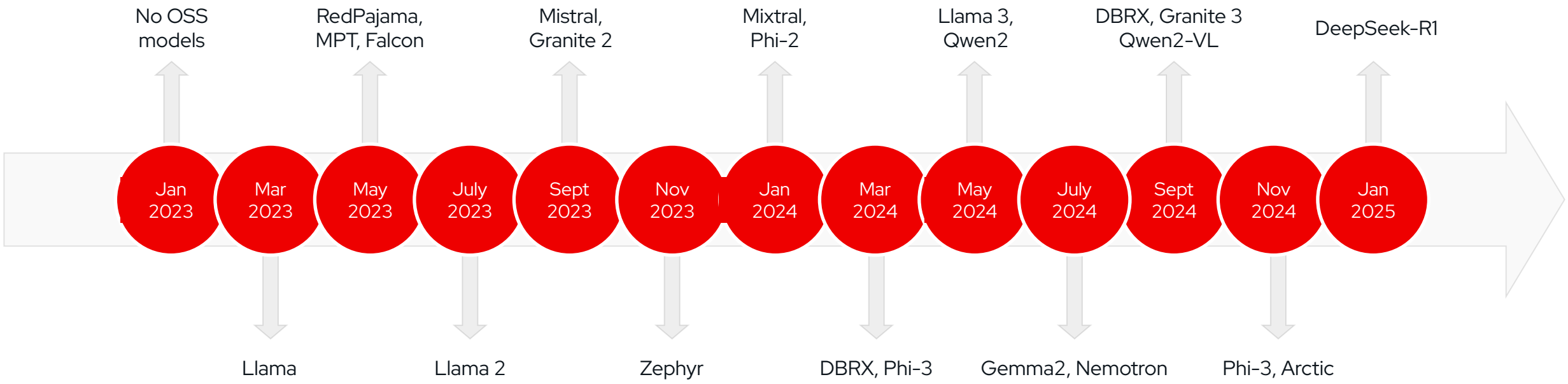
Table 14: Test-set performance of OPT-IML-Max, trained on all tasks in our benchmark, on Big-Bench Hard, MMLU, and RAFT.

Neural Magic | Confidential

8

The Power of Open

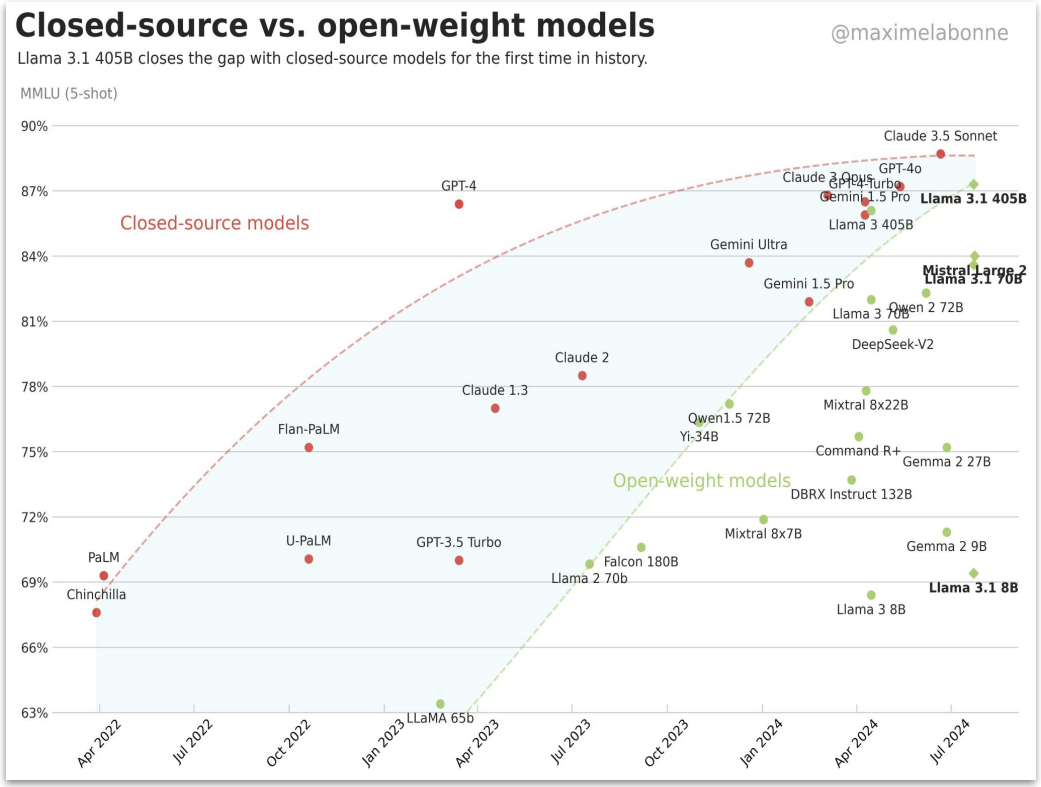
There has been an explosion of capability from open-source over the last 2 years.



Meta · Ai2 · NVIDIA · MISTRAL AI_ · Hugging Face · databricks · snowflake · IBM · Google · Microsoft · deepseek

The Power of Open

Open models are deployment targets **today**. And the trend is not slowing down.



Headlines

Meta

Llama

- 650M downloads in 2024
- 85,000 Llama derivative models
- 1B, 3B, 8B, 70B, 405B variants
- Multilingual, Multimodal, Mobile

deepseek

R1

- First reasoning model on par in quality with OpenAI O1
- 1-70B parameter distilled versions
- Global market pandemonium?

Models are commoditizing → many options for diverse enterprise needs.

Advantages of Open Source Models

Open-source models play an important role in the **Enterprise AI landscape**.

Cost

- Self managed infrastructure
- 1B-405B size - match task difficulty to model

Customization

- Improve accuracy and costs with task specific tuning

Control

- Model lifecycle (no changes to the model in place)
- Resources (no rate limits / API downtime)

Security

- Complete data privacy (no 3rd party APIs)

Open source is great driver for **Innovation**

Open source is about more than developing software.
It's how we built Red Hat.

And it's completely revolutionizing the AI ecosystem too.

Open Source Principles

There are various organizations that define open source, but generally...

Transparency

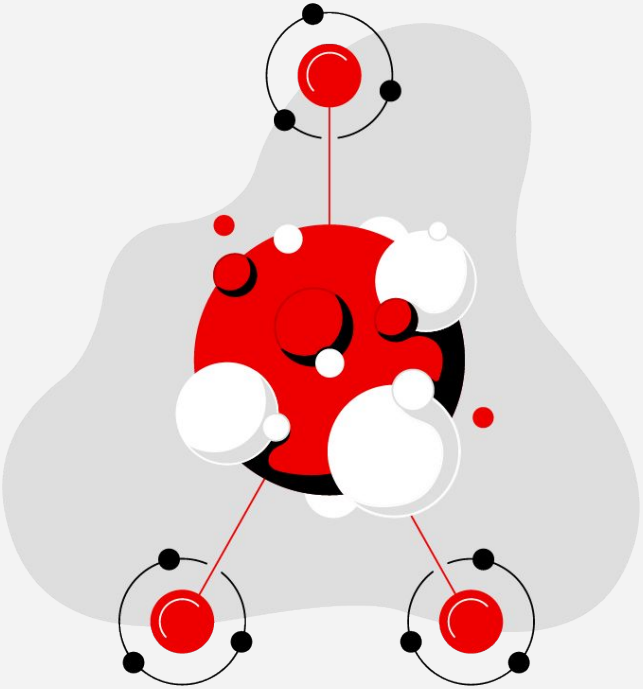
Open source ensures that code and processes are visible and accessible to everyone, fostering trust and accountability.

Community Collaboration

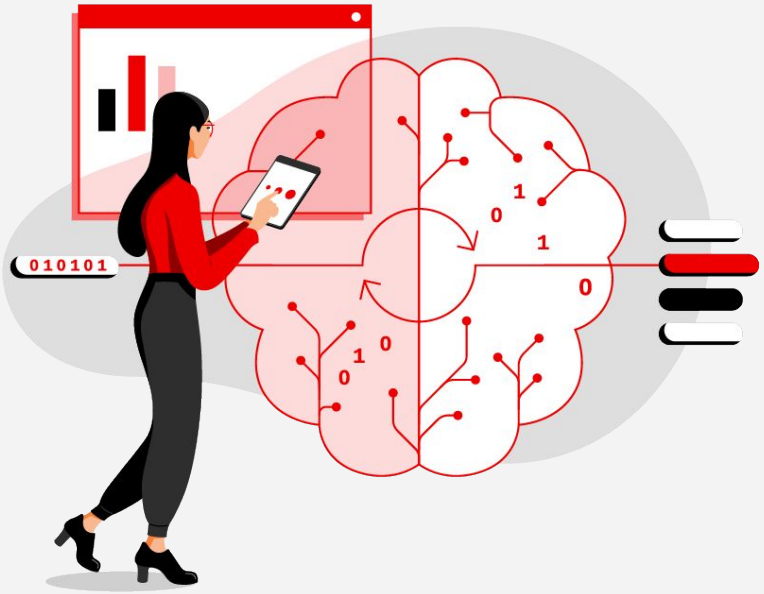
Development thrives through collective input, enabling diverse contributors to innovate and improve projects together.

No Vendor Lock-in

Users have the freedom to choose, modify, and migrate solutions without being tied to a single provider or proprietary system.



&



open source

AI

Data Scientists
"Developing Models"

Developers
"Writing Code"

these two personas get **most** of the attention...

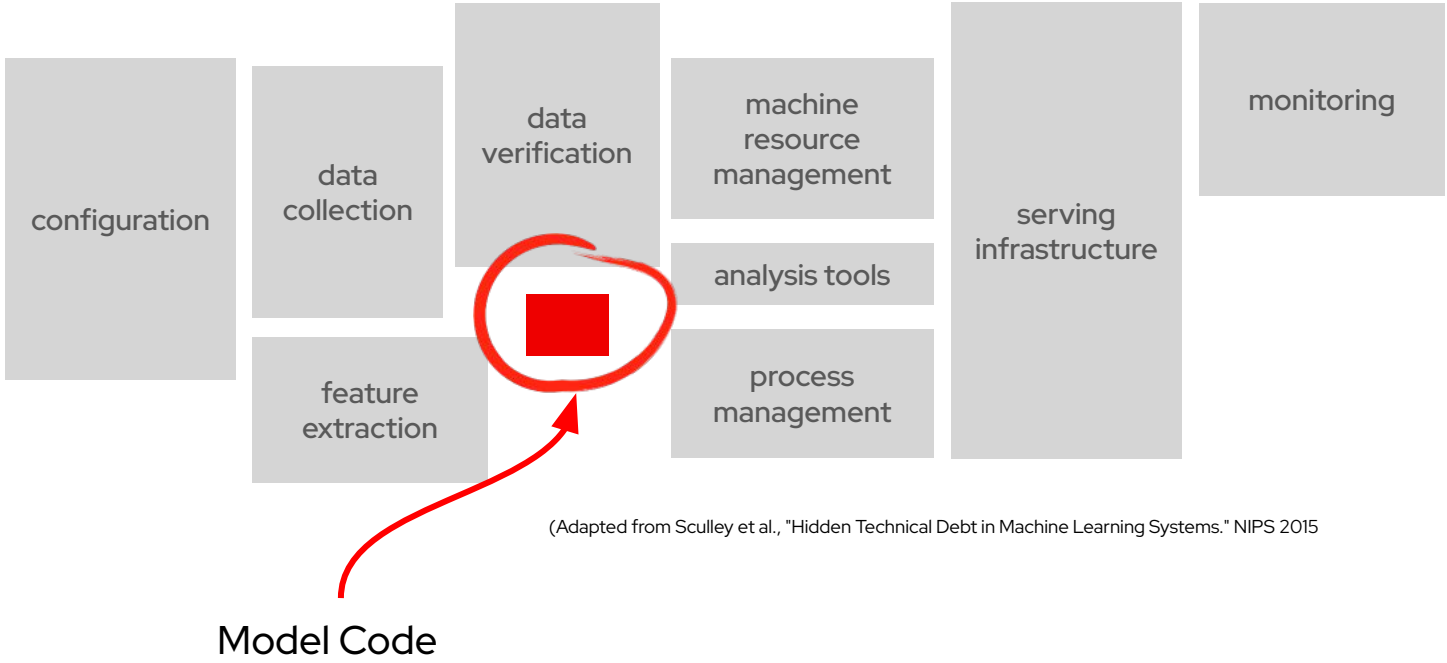
Operationalizing AI is one of the biggest challenges

Putting models into production can be quite a complicated process

Average timeline from idea to operationalizing the model

7-12 months to operationalize a model 50%

+1 year to operationalize a model 26%



(Adapted from Sculley et al., "Hidden Technical Debt in Machine Learning Systems." NIPS 2015

Source:
Gartner Peer Insights, Open Source AI for Enterprise survey, 2023

The reality of enterprise IT environments

Infrastructure

50%
of apps being run by enterprises are on hybrid cloud.[1]

41%
claim that IT Agility is one of main reasons to run apps on hybrid cloud[1]

Bare metal
Virtualization
Private cloud
Edge
Public cloud

Applications

AI/ML
Analytics
Serverless
Cloud-native services and microservices
Java™
.Net
ISV

Processes and people

Developer tools
Methods and practices
Culture and policy

Source: Red Hat detail. "2024 Global Tech Trends," Feb. 2024.

Red Hat

The reality of enterprise IT environments

Infrastructure

50%
of apps being run by enterprises are on hybrid cloud.[1]

41%
claim that IT Agility is one of main reasons to run apps on hybrid cloud[1]

Bare metal
Virtualization
Private cloud
Edge
Public cloud

Applications

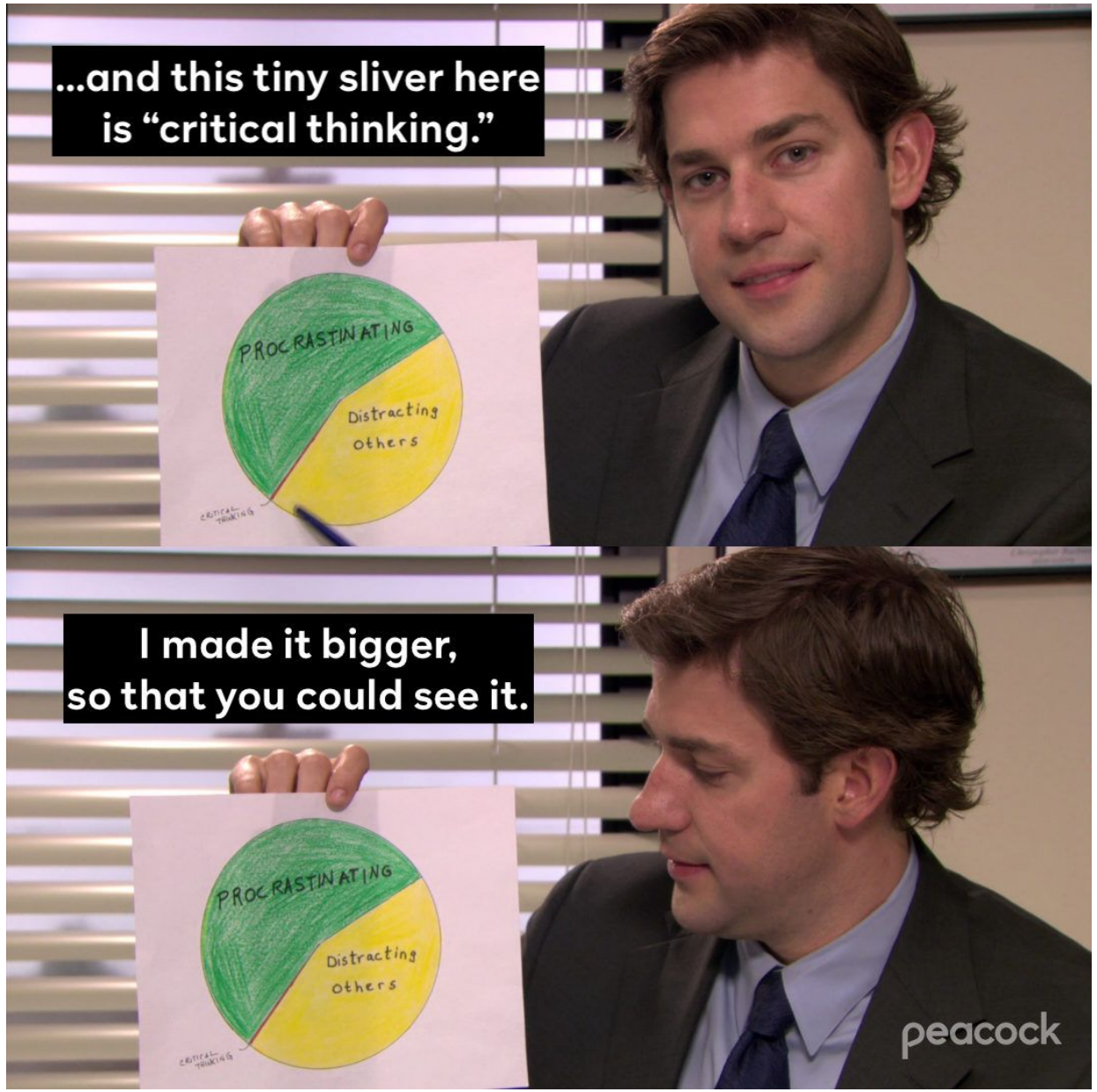
AI/ML
Analytics
Serverless
Cloud-native services and microservices
Java™
.Net
ISV

Processes and people

Developer tools
Methods and practices
Culture and policy

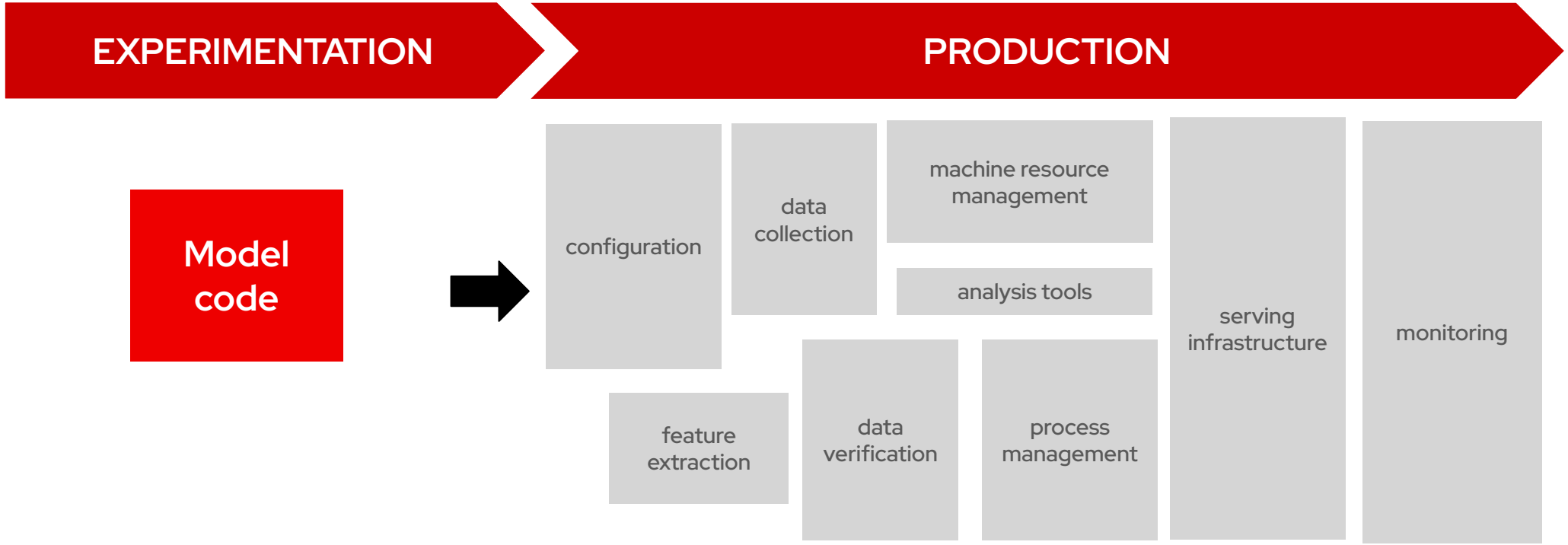
Source: Red Hat detail. "2024 Global Tech Trends," Feb. 2024.

Red Hat


...and this tiny sliver here is “critical thinking.”
PROCRASTINATING
Distracting others
I made it bigger, so that you could see it.
PROCRASTINATING
Distracting others
peacock

Poorly designed systems lead to failed ML projects

Lack of focus on end-to-end system builds technical debt



Technical debt is a barrier to production

Red Hat

There's a huge stack for AI/ML

AI models

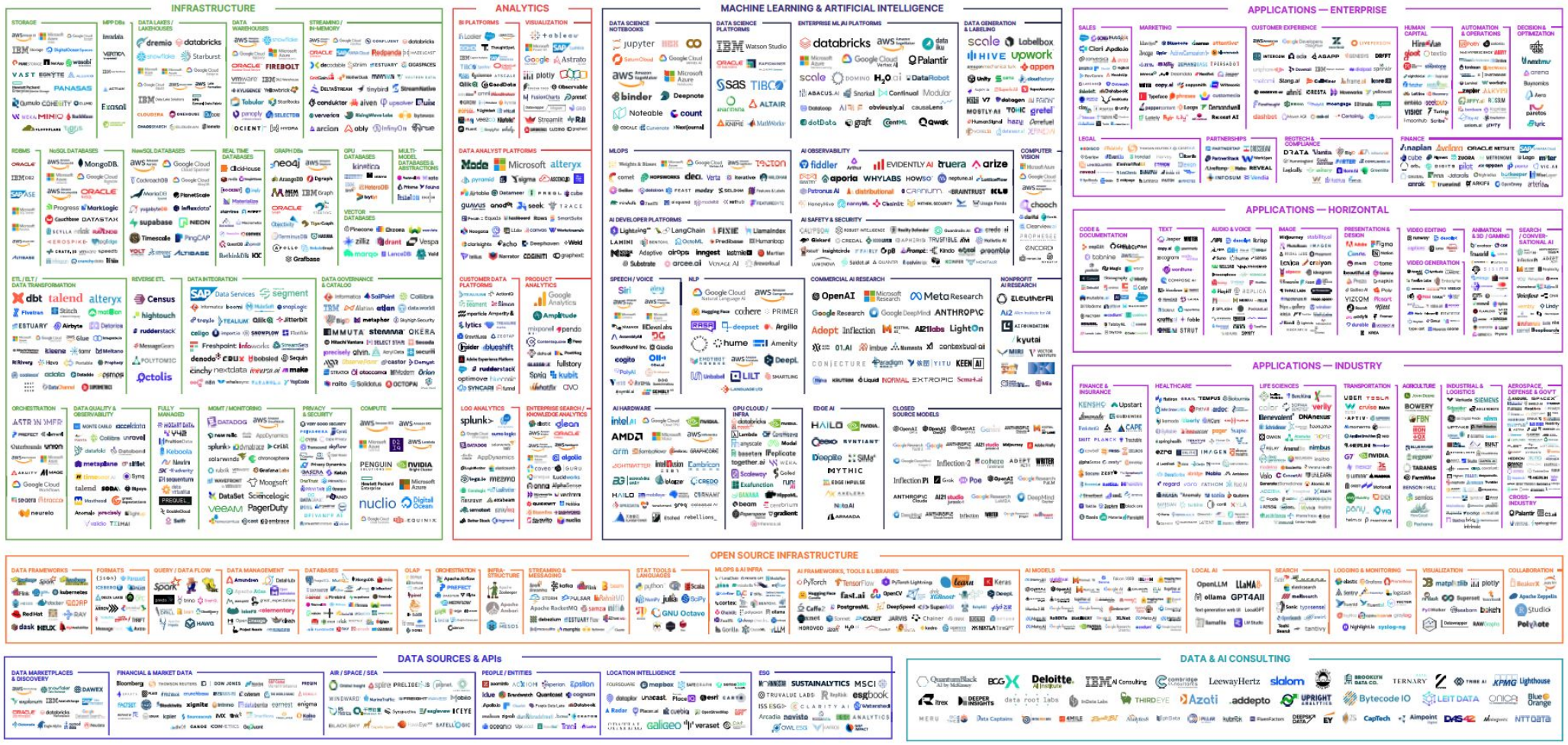
AI & MLOps Platform

- Machine learning libraries: TensorFlow, PyTorch, scikit learn, XGBoost, NumPy, pandas, OpenCV, LangChain
- Languages & development tools: python, R, jupyter, VS Code, Brain
- Data visualization, labeling, processing: Superset, Label Studio, trino, Pachyderm, Spark, Airflow, RAY, FEAST
- Experimentation & model lifecycle: Katib, Kubeflow, mlflow, intel OpenVINO, SELDON, Flask, KServe

Application Platform

- Software-defined storage: ceph, MINIO
- Integration: kafka, Istio, 3scale by Red Hat
- Operating containers at scale: etcd, K, HELM, Prometheus, Grafana
- Automated software delivery: TEKTON, argo
- Modernize & Accelerate app development: WildFly, KONVEYOR
- Containerization & container orchestration: docker, podman, CoreOS, kubernetes
- Process scheduling & hardware acceleration: LINUX, NVIDIA CUDA

There's a huge stack for AI/ML - It gets bigger!

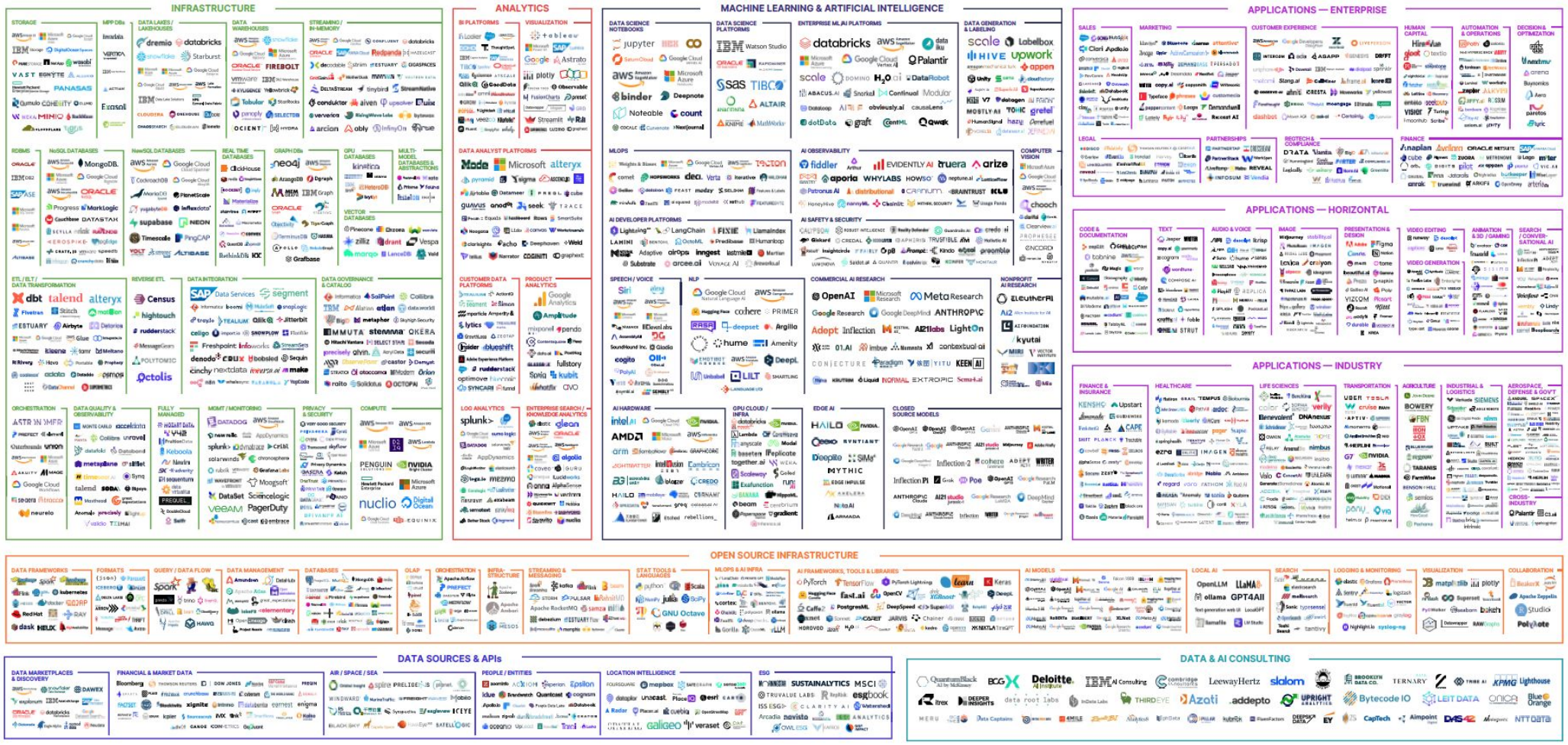


Version 1.0 - March 2024 © Matt Turck (@mattturck) , Aman Kabeer (@AmanKabeer11) & FirstMark (@firstmarkcap) Blog post: mattturck.com/MAD2024 Interactive version: MAD.firstmarkcap.com Comments? Email MAD2024@firstmarkcap.com

FIRSTMARK EARLY STAGE VENTURE CAPITAL

There's a huge stack for AI/ML - It gets bigger!

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



Version 1.0 - March 2024 © Matt Turck (@mattturck) , Aman Kabeer (@AmanKabeer11) & FirstMark (@firstmarkcap) Blog post: mattturck.com/MAD2024 Interactive version: MAD.firstmarkcap.com Comments? Email MAD2024@firstmarkcap.com

FIRSTMARK EARLY STAGE VENTURE CAPITAL

There's a huge stack for AI/ML - It gets bigger!

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

Scope For Today

❌ Training a Foundation Model from Scratch
No, that would be expensive

✅ Working off existing open source base models

✅ Integrating your data to the model

✅ Building applications with AI

✅ Platforms, serving, and operationalizing AI

Session Slides
red.ht/open-source-ai

Red Hat

How do we Develop Generative AI Applications?

Ideation & Prototyping

- Find LLMs
- Try prompts
- Experiment with your data
- Benchmarking

How do I **evaluate** models and pick the best one for my use case?

Building & Refining

- Connect to data source
- Exception handling
- Limited fine tuning
- Retrieval-Augmented Generation (RAG)
- Chaining
- Evaluate flows

As a developer, how do I **build** an application with LLMs?

Operationalizing

- Model serving
- Endpoints
- Monitoring
- Integrate with apps

How do I **deploy** my application with LLMs, scale, etc?

Red Hat

How do we Develop Generative AI Applications?

Ideation & Prototyping

- Find LLMs
- Try prompts
- Experiment with your data
- Benchmarking

How do I **evaluate** models and pick the best one for my use case?

Building & Refining

- Connect to data source
- Exception handling
- Limited fine tuning
- Retrieval-Augmented Generation (RAG)
- Chaining
- Evaluate flows

As a developer, how do I **build** an application with LLMs?

Operationalizing

- Model serving
- Endpoints
- Monitoring
- Integrate with apps

How do I **deploy** my application with LLMs, scale, etc?

Red Hat

How do we Develop Generative AI Applications?

Ideation & Prototyping

- Find LLMs
- Try prompts
- Experiment with your data
- Benchmarking

How do I **evaluate** models and pick the best one for my use case?

Building & Refining

- Connect to data source
- Exception handling
- Limited fine tuning
- Retrieval-Augmented Generation (RAG)
- Chaining
- Evaluate flows

As a developer, how do I **build** an application with LLMs?

Operationalizing

- Model serving
- Endpoints
- Monitoring
- Integrate with apps

How do I **deploy** my application with LLMs, scale, etc?

Red Hat

How do we Develop Generative AI Applications?

Ideation & Prototyping

- Find LLMs
- Try prompts
- Experiment with your data
- Benchmarking

How do I **evaluate** models and pick the best one for my use case?

Building & Refining

- Connect to data source
- Exception handling
- Limited fine tuning
- Retrieval-Augmented Generation (RAG)
- Chaining
- Evaluate flows

As a developer, how do I **build** an application with LLMs?

Operationalizing

- Model serving
- Endpoints
- Monitoring
- Integrate with apps

How do I **deploy** my application with LLMs, scale, etc?

Red Hat

Let's begin our journey thinking about the model!

What do you need to know about model selection & consequences?

Determining your use case

What problem are we solving?

Types of Models

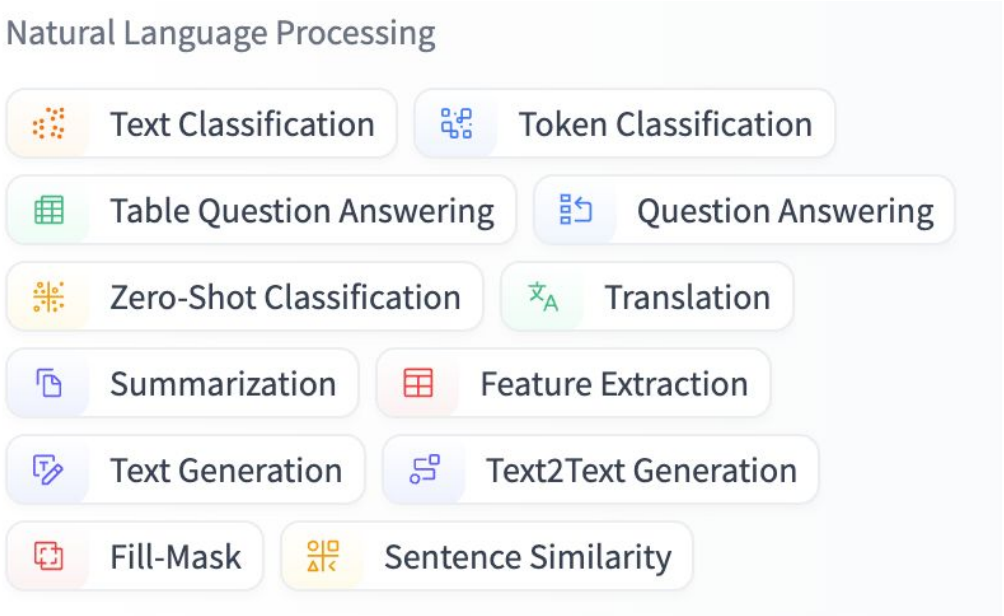
Instruct vs Base vs Embed...

Start

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.



Types of Models

Instruct Models

These are generative models (like ChatGPT) that have been fine-tuned to follow **natural language instructions**

Use Cases

- Summarize this Article
- Explain Quantum Physics in simple terms

Embedded Models

These models **convert text into vectors** (high-dimensional numerical representations) that capture semantic meaning.

Use Cases

- Text Classification by Semantic Similarity

Example output

- `[0.023, -0.448, ..., 1.205]` ⟵ (a vector representing that sentence's meaning)

You then **compare vectors** (using cosine similarity, etc.) to see how similar two pieces of text are.

Let's begin our journey thinking about the model!

What do you need to know about model selection & consequences?

Model Nomenclature

Understanding the naming conventions

Model Transparency

Is the training data biased & understanding openness

Benchmarks/Evaluations

Understanding "vibes" and model evaluations

Also! There's a naming convention

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

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

- Family name
- Model architecture and version
- Number of parameters
- Model fine-tuned to be a baseline

Mixtral-8x7B-Instruct-v0.1

- Family name
- Architecture type
- Number of parameters
- Model fine-tuned for instructive tasks
- Model version

Open Source provides transparency behind models

Understanding model architecture & training data is critical!

Training Data	Pre-processing	Training	Model	Deployment
Data sets	Code Filtering criteria	Code	Architecture Trained weights Evaluation results	Code Use guide License

LF AI & DATA

MODEL OPENNESS FRAMEWORK

Example: Llama 4 & DeepSeek

Check the Model Openness Framework for understanding model components & licensing

Integrating Your Data To The Model

How can we specialize a “generic” language model to our unique data?

RAG, Fine-Tuning, etc

Customizing responses

Processing Enterprise Documents

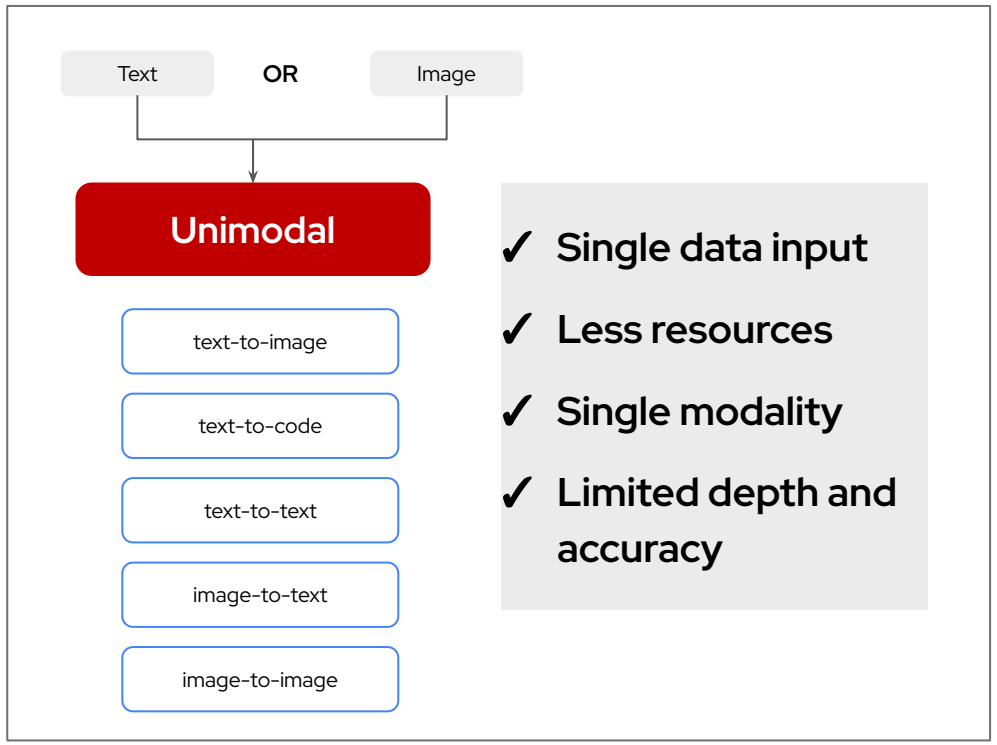
Not all data is AI-ready!

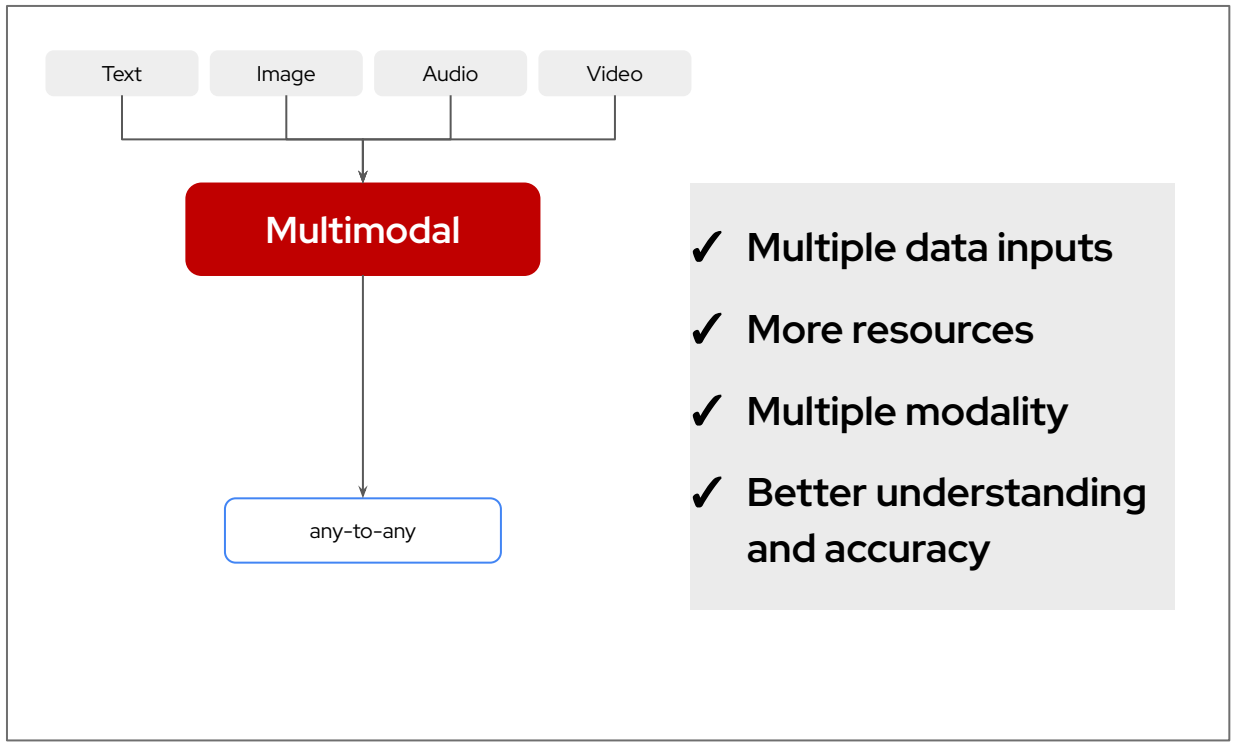
Add Your Data And Preferences

Make it your own

First, consider your data!

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





Customization of Models

What are the most common ways to add your data to AI?

Prompt Engineering

Prompt Engineering

The process of rafting precise inputs to guide AI systems toward generating the most relevant outputs.

RAG

Retrieval Augmented Generation

Enhance Gen AI model-generated text by retrieving relevant information from external sources, improving accuracy and depth of model's responses.

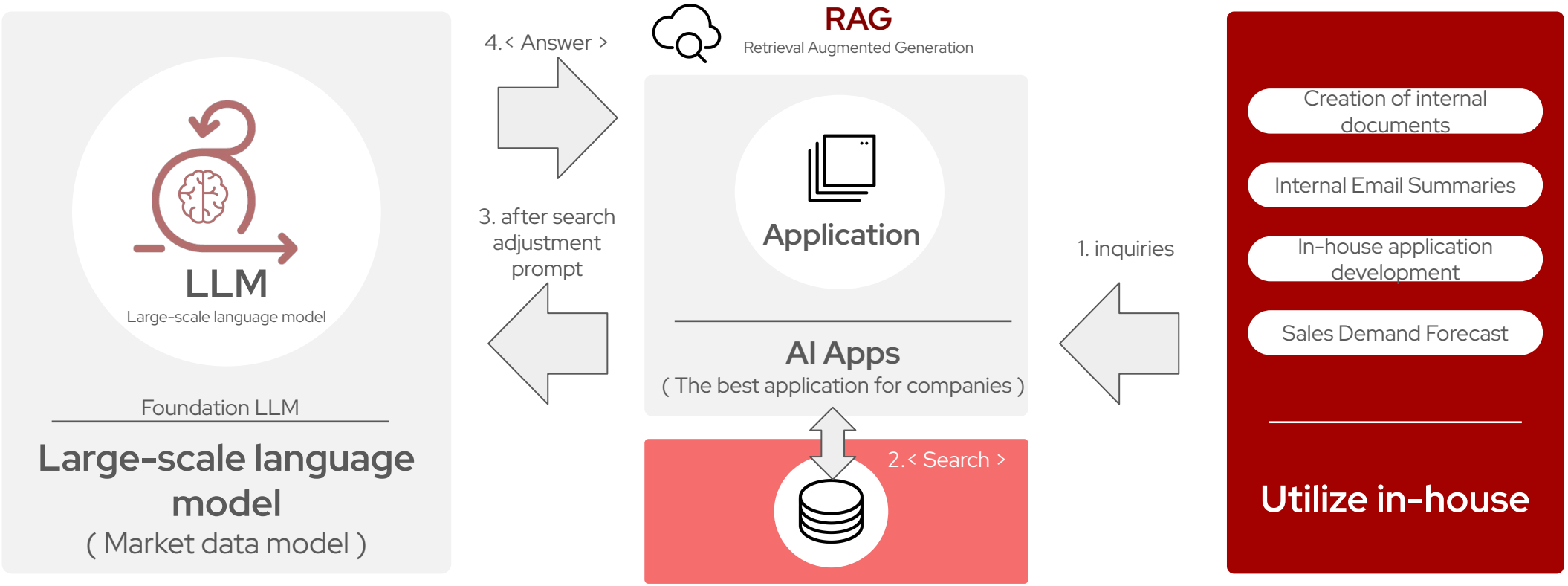
Fine tuning

Fine Tuning

Adjust a pre-trained model on specific tasks or data, improving its performance and accuracy for specialized applications without full retraining.

What is Retrieval Augmented Generation (RAG)?

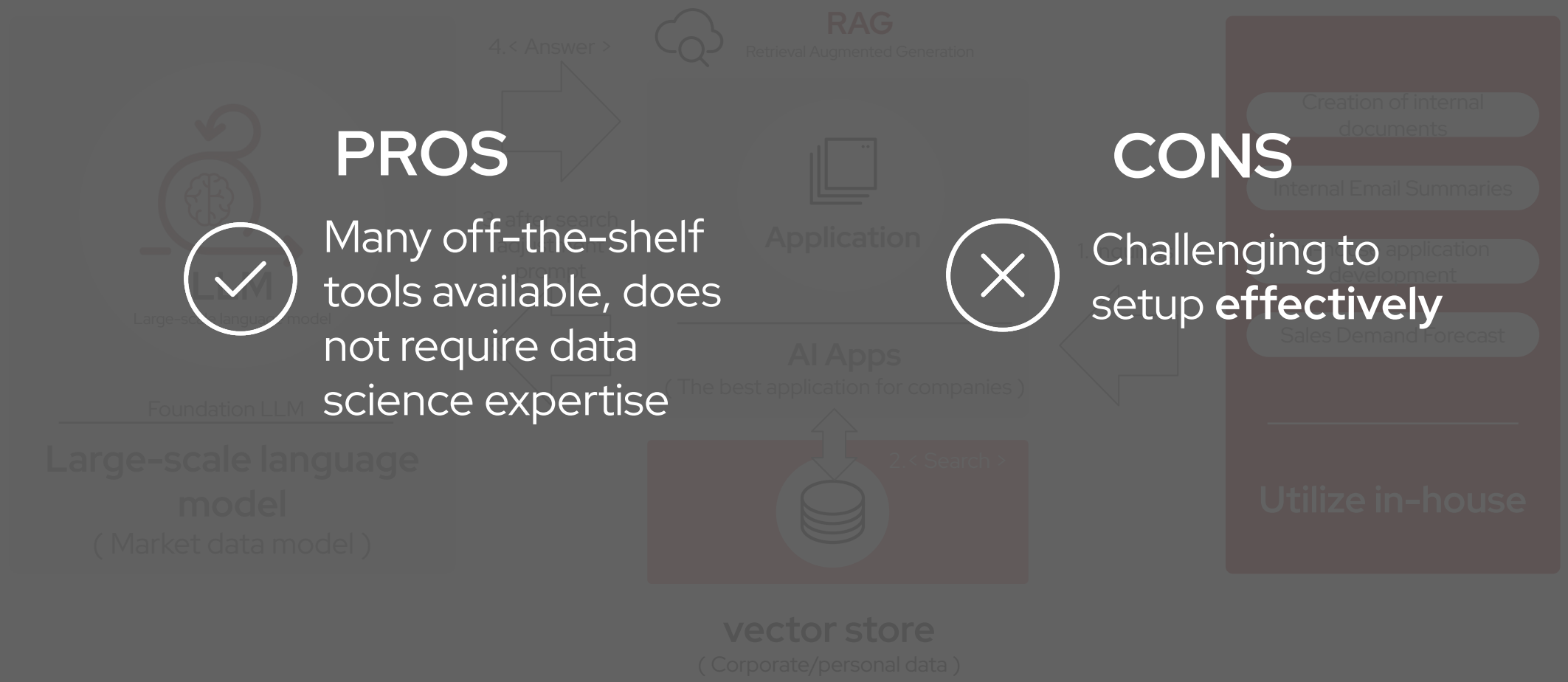
A method that retrieves facts from an external knowledge base and causes the LLM to generate answers **based on accurate information**. The **original LLM is not modified**.



vector store
(Corporate/personal data)

What is Retrieval Augmented Generation (RAG)?

A method that retrieves facts from an external knowledge base and causes the LLM to generate answers **based on accurate information**. The **original LLM is not modified**.



PROS

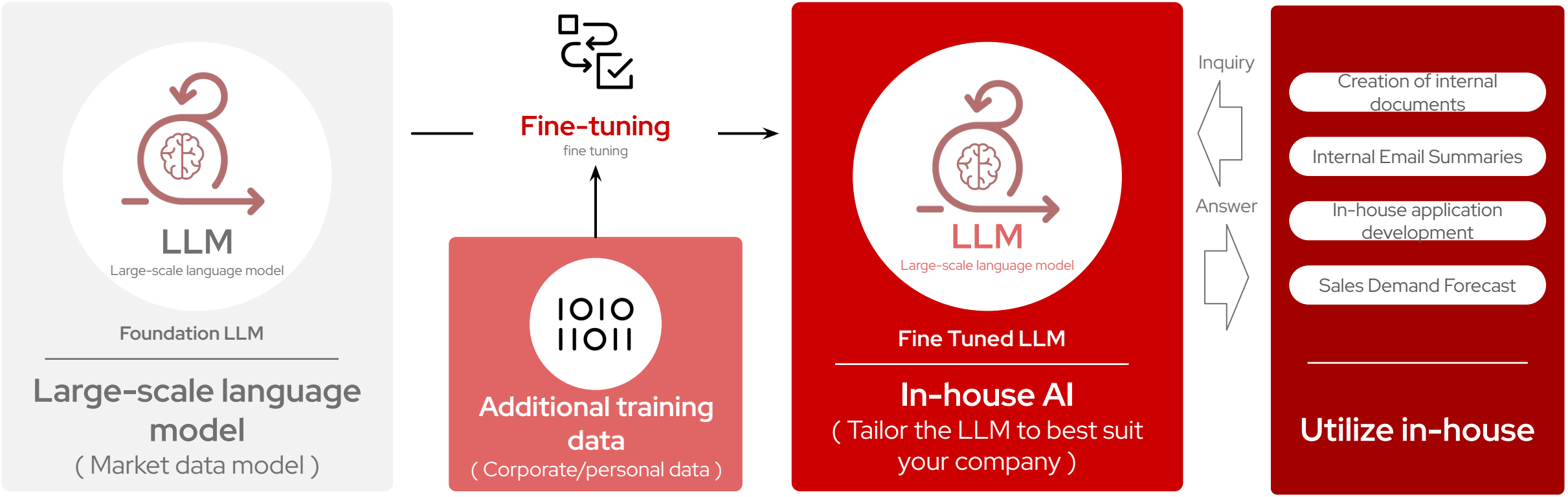
Many off-the-shelf tools available, does not require data science expertise

CONS

Challenging to setup **effectively**

What is Fine-tuning?

Fine-tune the original LLM with your own data by retraining part or the entire LLM **with a different data set**.



What is Fine-tuning?

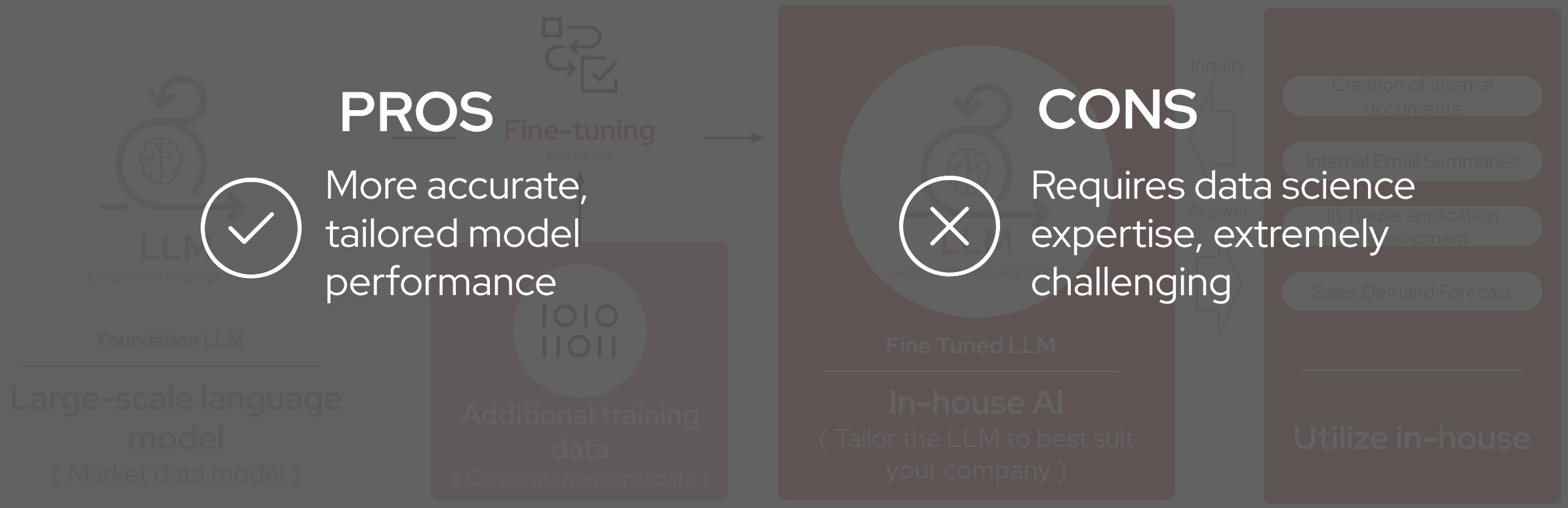
Fine-tune the original LLM with your own data by retraining part or the entire LLM **with a different data set**.

PROS

More accurate, tailored model performance

CONS

Requires data science expertise, extremely challenging



Are there other model customization options?

Yes! There are more efficient & effective ways to integrate your unique data

RAG

Retrieval Augmented Generation

Enhance Gen AI model-generated text by retrieving relevant information from external sources, improving accuracy and depth of model's responses.

NEW

InstructLab

Large-scale Alignment for chatBots

Leverage a taxonomy-guided synthetic data generation process and a multi-phase tuning framework to improve model performance.

Fine tuning

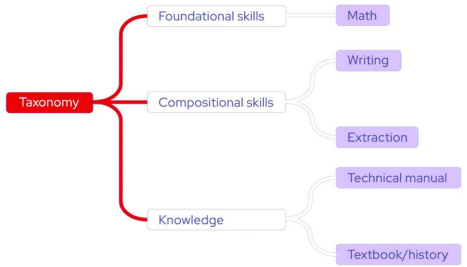
Fine Tuning

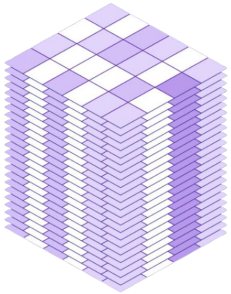
Adjust a pre-trained model on specific tasks or data, improving its performance and accuracy for specialized applications without full retraining.

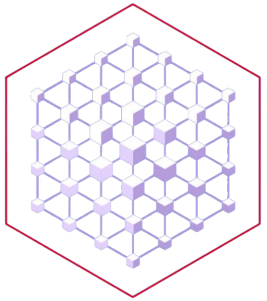
InstructLab provides **more accessible fine tuning** & **complements RAG**

InstructLab powers **simple & cost-effective** LLM refinements

It's a CLI and library to align language models with enterprise data







Taxonomy-Driven Data Curation

Folder structure with Q&A pairs for topics to teach a model

Large-Scale Synthetic Data Generation

Generate additional training data to expand dataset automatically

Multi-Phase Alignment Tuning

Full parameter, phased alignment tuning for custom knowledge and skills

Let’s Customize AI!

❌ $$$ for data scientists and expensive GPU’s
No, that’s not accessible for most!

✅ Easy model customization with plain text

✅ Unstructured data parsing from PDF’s, etc

✅ Simple CLI or python library for the team

✅ Powered by open source models & software!

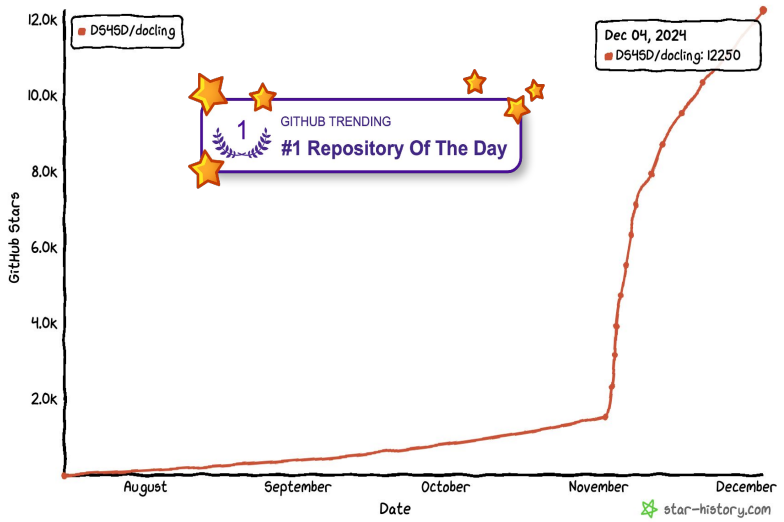
Workshop Link
red.ht/world-summit-lab

Red Hat

Turning Unstructured Data to Structured

With the Docling project, which uses OCR + advanced Vision Models for PDF's, etc!

- **Document Parsing:** Extracts context, tables, graphs, and valuable data from PDF, Docx, etc
- **AI-Ready:** Direct integrations into LlamaIndex, LangChain, etc. for RAG and Dataset Gen.
- **Performative:** Fastest among all open-source parsers on CPU (+ 200k pages/day on GPU)
- **For Developers:** Released as a CLI for simple usage or a library for integrating into applications.



Red Hat

Building applications that use AI models

How can developers use, build, and test AI capabilities?

Serving Models Locally

How to not depend on 3rd party LLM API's

Building Apps with AI Capabilities

Think RAG, Agentic, etc

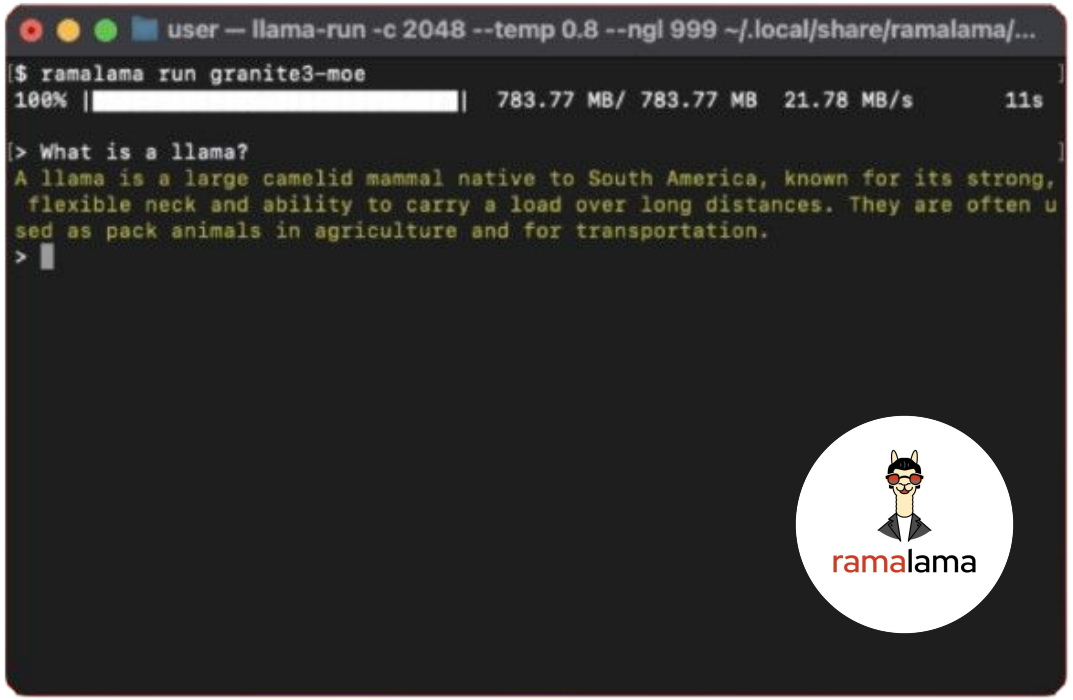
Testing Model Output

How do we test non-deterministic output?

Introducing: 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.
-

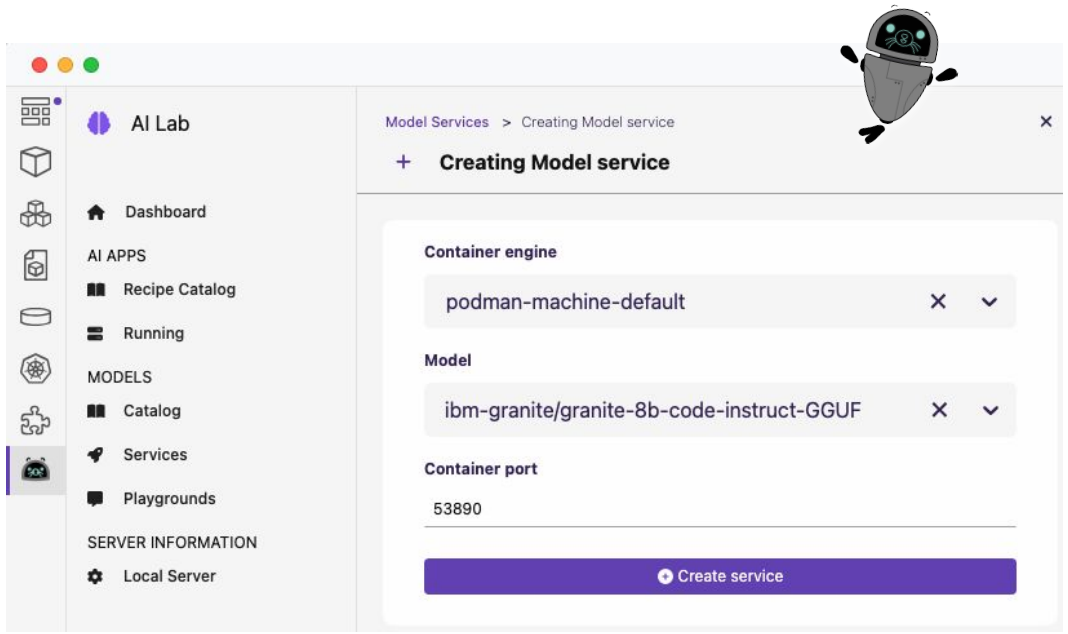


Red Hat

Introducing: 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.



Red Hat

Run Model in Production

Serve the Model Efficiently
via vLLM

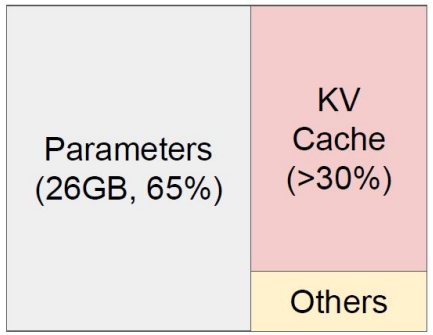
Monitor, Update, and Improve
via KServe, Ray, &
Prometheus/Grafana

Allow users or apps to
call it via API

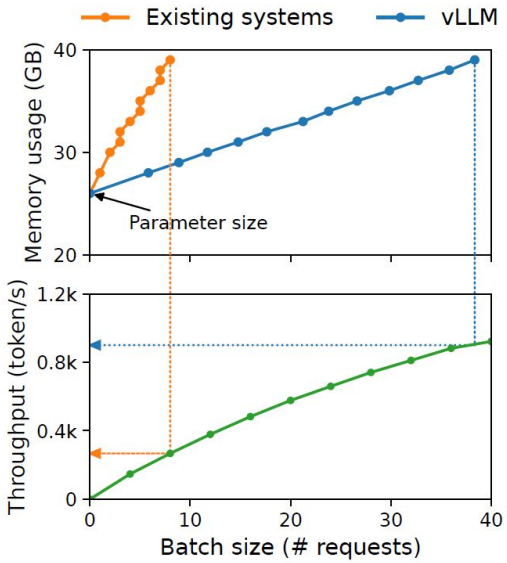
vLLM

For LLM inference serving 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



NVIDIA A100 40GB

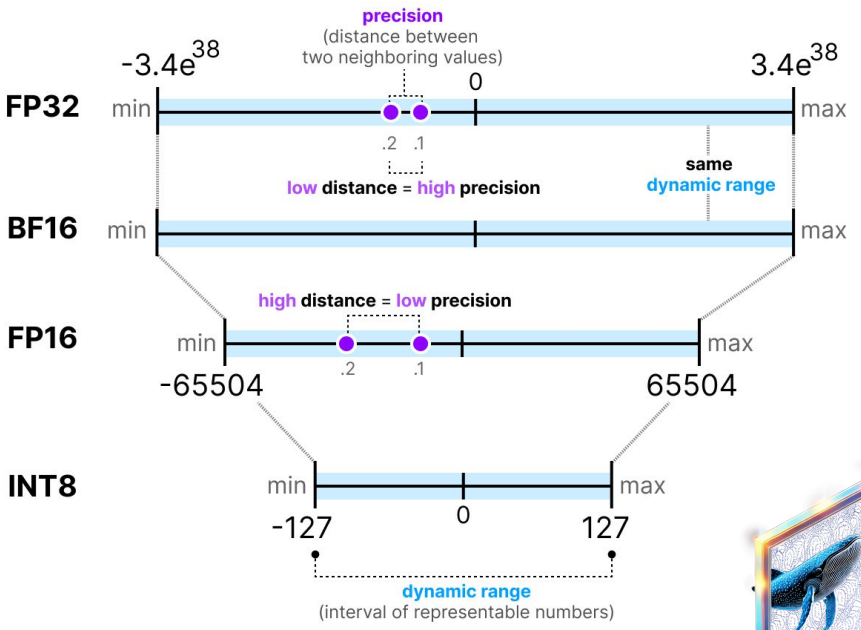


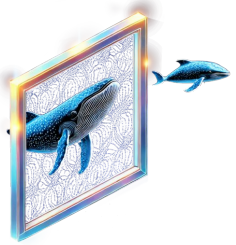
Red Hat

How can you save on GPU resources?

Quantization! It's a way to compress models, think like a .raw to .jpg

- **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.



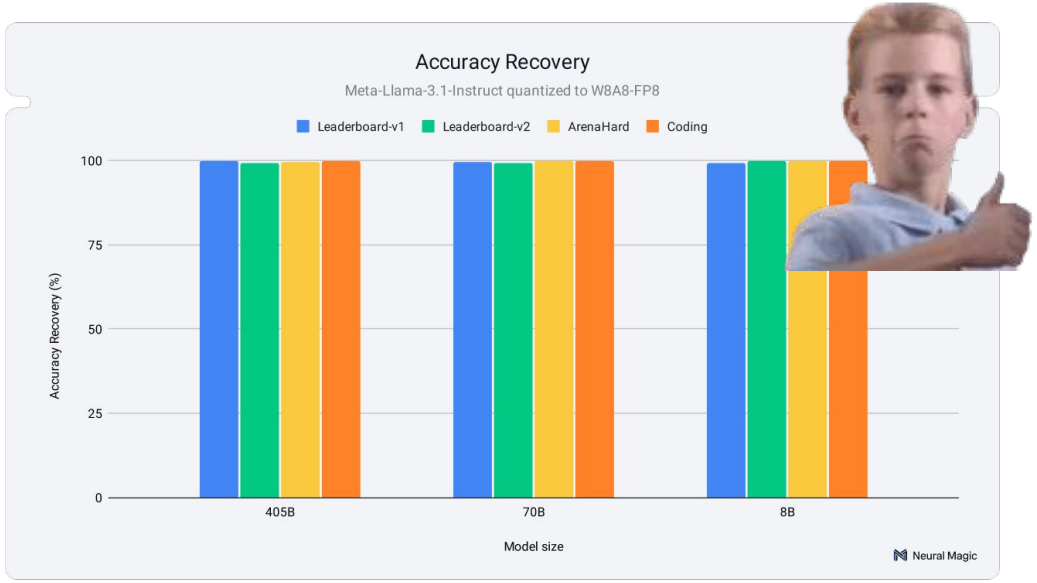


Red Hat

How can you save on GPU resources?

Quantization! It's a way to compress models, think like a .raw to .jpg

- **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!**



Source:
https://neuralmagic.com/blog/we-ran-over-half-a-million-evaluations-on-quantized-llms-heres-what-we-found

& 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

NVIDIA GPUs
Google TPUs
AMD Instinct
intel CPUs

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

Red Hat

But what does an AI platform need to be successful?

Hint: It stems from open source technology!

	Model Development & Training		Model Storage	Model Serving			
Development Challenges	**Gather & Prepare Data** Data Extraction, exploration, transformation, Feature engg, labeling, Taxonomy, Synthetic Data Generation	**Model Development** Foundational Models, Workbenches, ISV Images, Model Training Tools, Collaboration, Hardware Support	**Model Storage** Highly Available,, Eventing, Compliant, Tiering Capabilities, Retention	**Model Deployment** Model Packaging,Deployment,, Serving, MLOps Automation, Infrastructure provisioning, Connect app with Model, Distributed Workloads,	**Retrain** Model Versioning, Data validation, Data manipulation, Drift detection. Retraining Pipelines. Compliance and Governance, Serving Engine, Runtimes	**Model Monitoring** Logging and Data Collection, Real-time monitoring , Alerts Model Drift. Performance audit Resource usage, Model Explainers, Quality Metrics	
Technology Area	Data Engineering Tools	Models & Model Development Tools	Model Registry & Storage	Model Deployment & Pipelines	Model Serving	Model Monitoring & Governance	
Core Open Source Community Projects	Pachyderm, trino, Rapid Studio, jupyter, PyTorch, NVIDIA, OpenCV, OpenVINO, CodeFlare, ANACONDA, scikit learn, pandas, NumPy		MINIO, RCLONE	CodeFlare, Kubeflow, RAY, Katib, KServe, PyTorch, NVIDIA TRITON INFERENCE SERVER, OpenVINO, ModelOp, Kubeflow, Caikit, vLLM, CORE		TensorBoard, Katib, Grafana loki, CORE, Pachyderm	

Red Hat

We contribute to and productize open source for AI

From frameworks like PyTorch to pipelines and more, all for Kubernetes!

Upstream projects

RAY
CodeFlare
ModelMesh
Kubeflow
PyTorch
TensorFlow
KServe
TrustyAI
jupyter

→

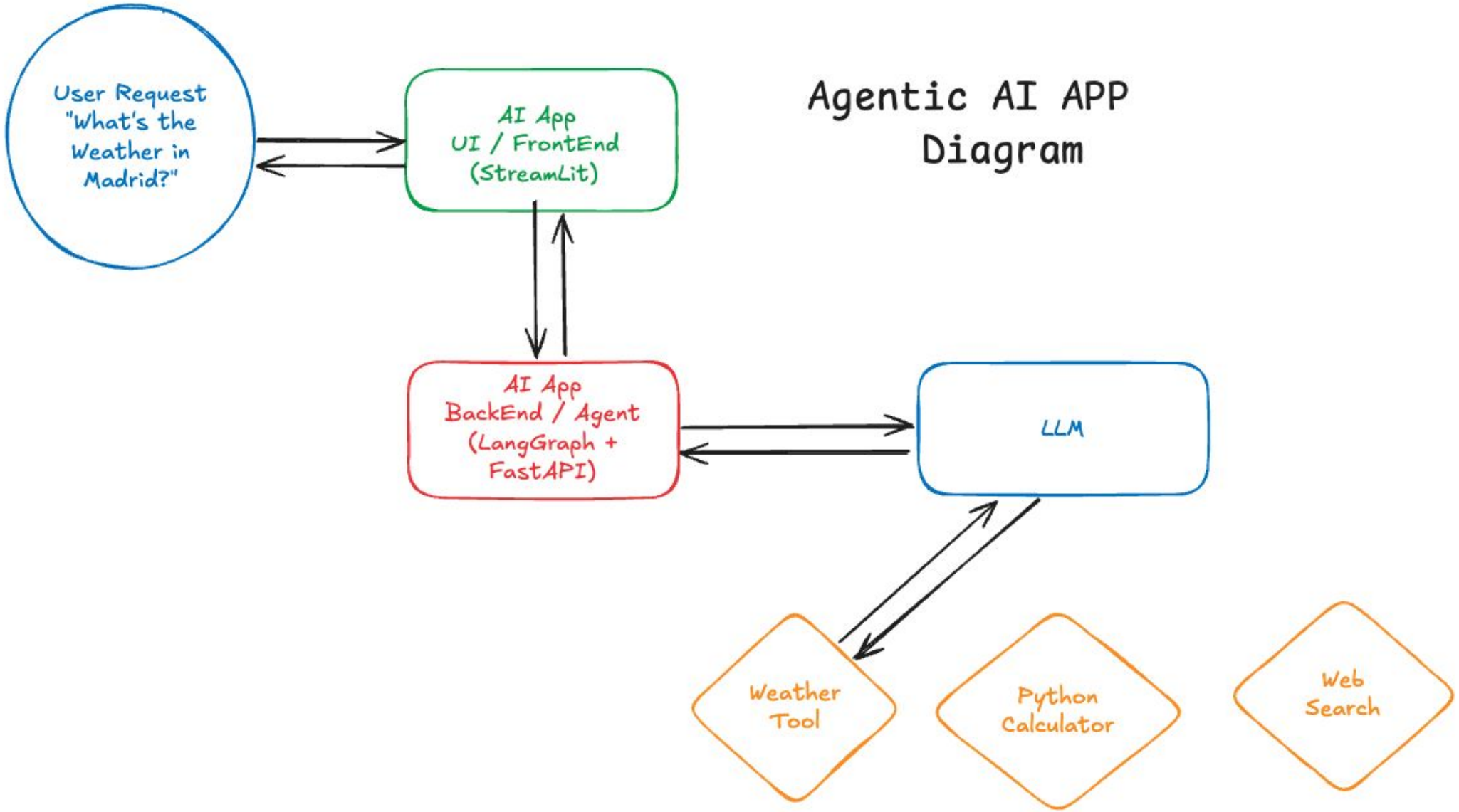
Community projects

OPEN DATA HUB
AI Platform powered by Open Source

→

Product

Red Hat OpenShift AI


Agentic AI APP Diagram
User Request "What's the Weather in Madrid?"
AI App UI / FrontEnd (StreamLit)
AI App BackEnd / Agent (LangGraph + FastAPI)
LLM
Weather Tool
Python Calculator
Web Search

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.

linkedin.com/company/red-hat

youtube.com/user/RedHatVideos

facebook.com/redhatinc

x.com/RedHat

Red Hat