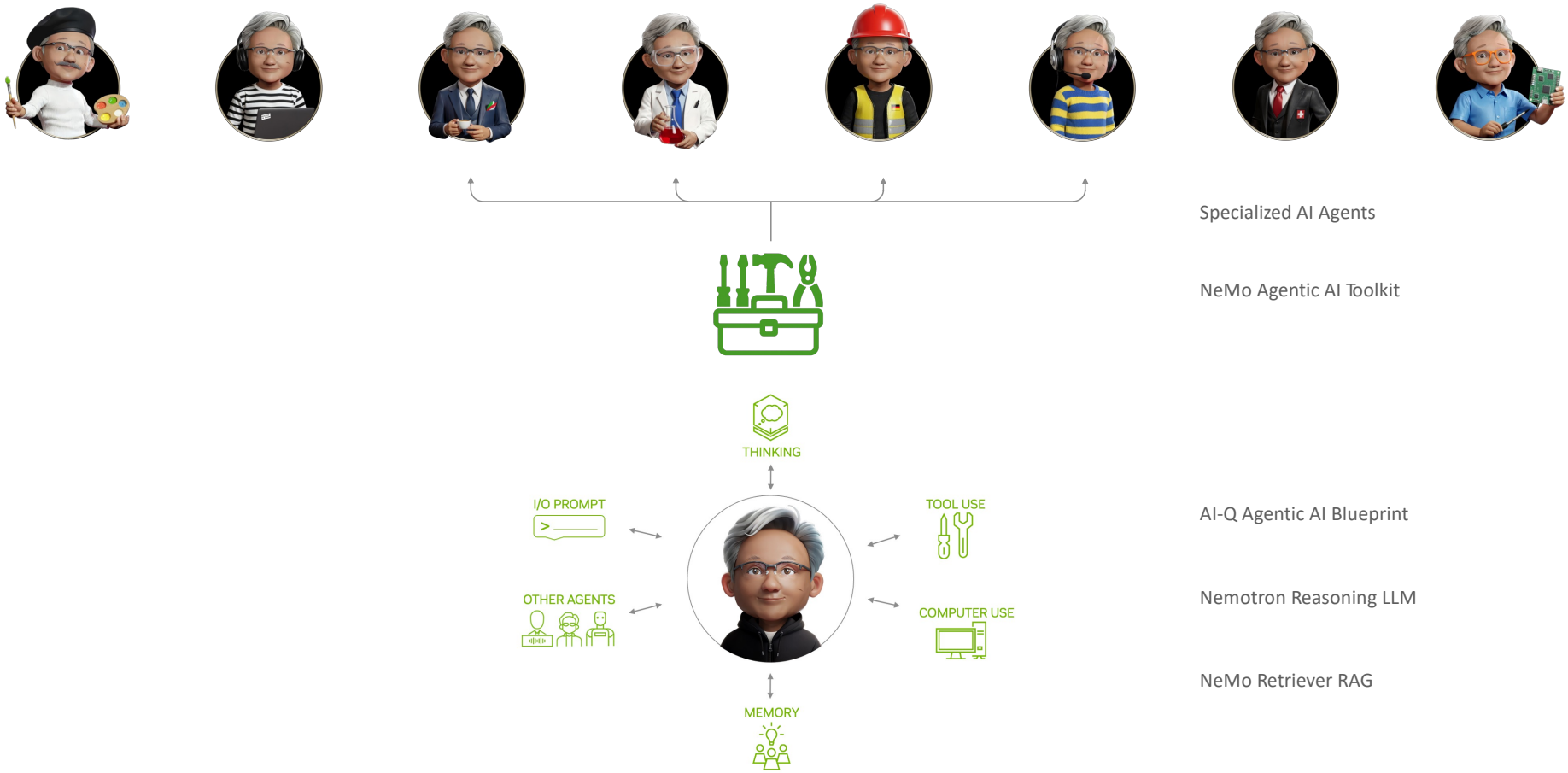
Practical Agentic AI Systems

Tran Minh Quan, PhD,
Senior Developer Technologist, NVIDIA

NVIDIA Enterprise AI Agent Platform



Specialized AI Agents

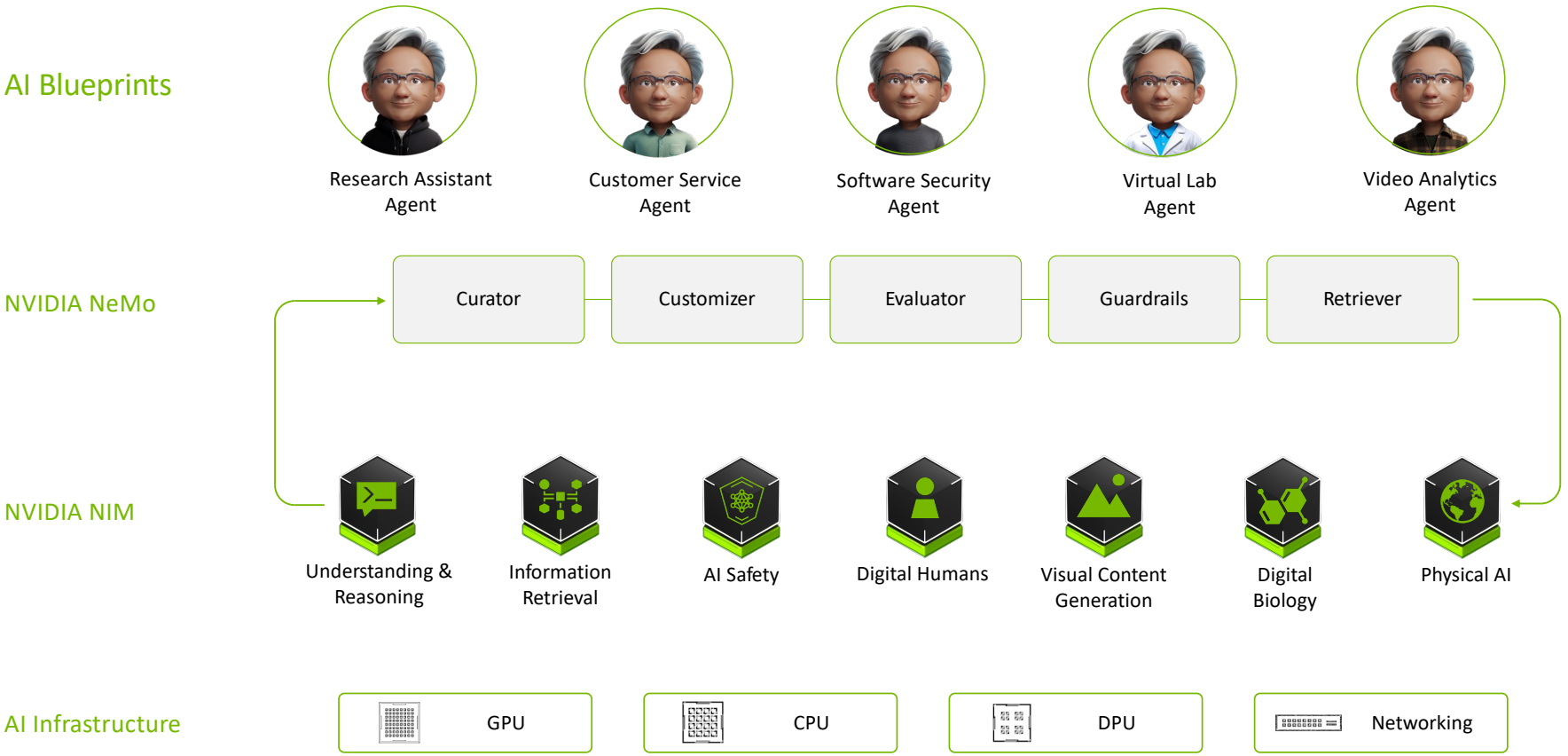
NeMo Agentic AI Toolkit

AI-Q Agentic AI Blueprint

Nemotron Reasoning LLM

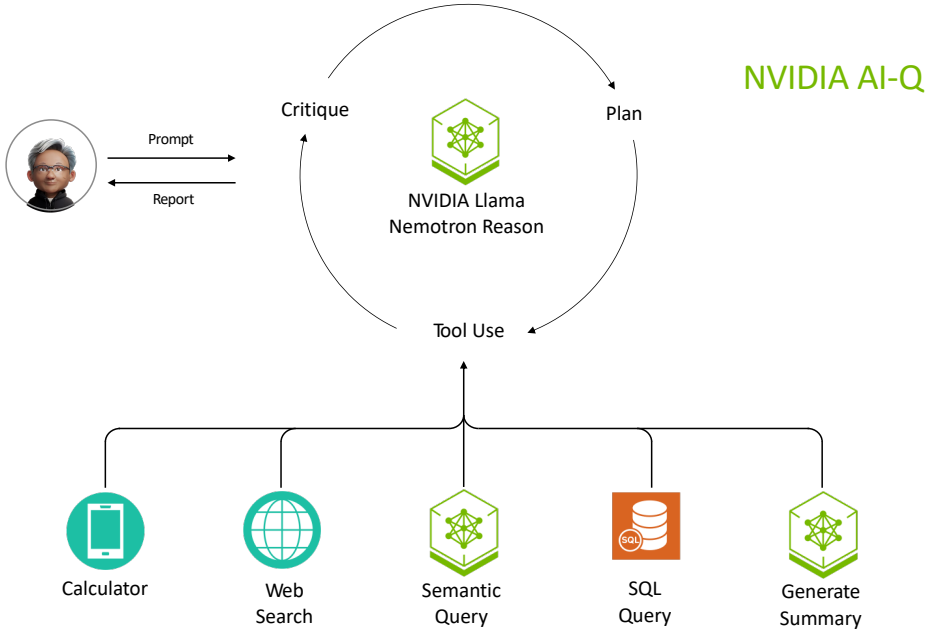
NeMo Retriever RAG

NVIDIA Provides the Building Blocks for Agentic AI



AI-Q: AI Agent Interface to Enterprise Data Stores

AI Data Platform Reference Design



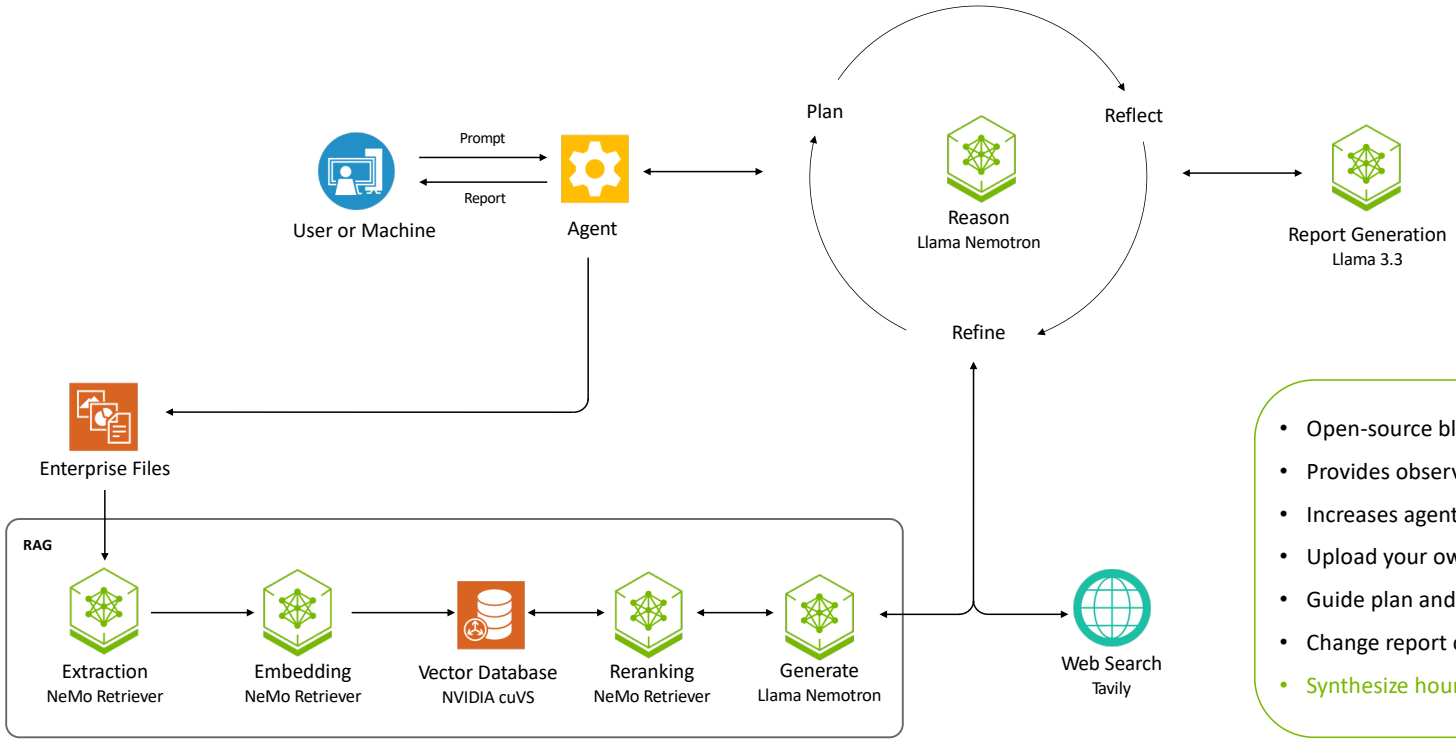
ddn DELL Technologies Hewlett Packard Enterprise Hitachi Vantara IBM

NetApp NUTANIX PURESTORAGE VAST WEKA

NVIDIA

AI-Q (Pronounced ‘IQ’) NVIDIA Blueprint

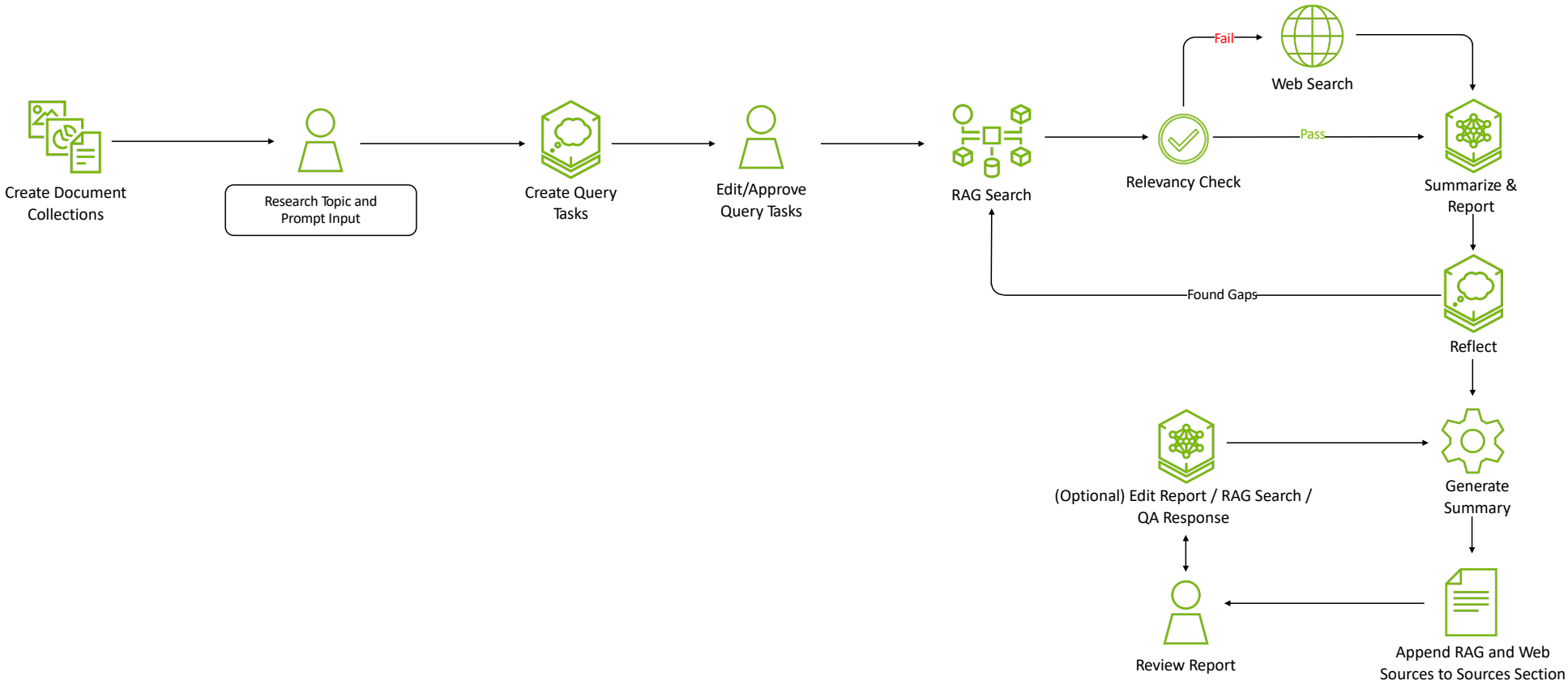
Use Reasoning to Connect AI Agents, Data, and Tools



- Open-source blueprint for research & reporting
- Provides observability and transparency
- Increases agent accuracy with reasoning
- Upload your own data sources
- Guide plan and report in real-time
- Change report output dynamically
- Synthesize hours of material in minutes

AI-Q Research Assistant Blueprint Process Flow

Deep Research with On-Premise Data: Chat with Your Enterprise Data



NVIDIA NeMo Agent Toolkit

An open-source library for building enterprise-ready agentic systems

Profiling & Optimizations

- Fine-grained AI workflow telemetry collected can be used to implement agentic system accelerations.

Evaluation & Observability

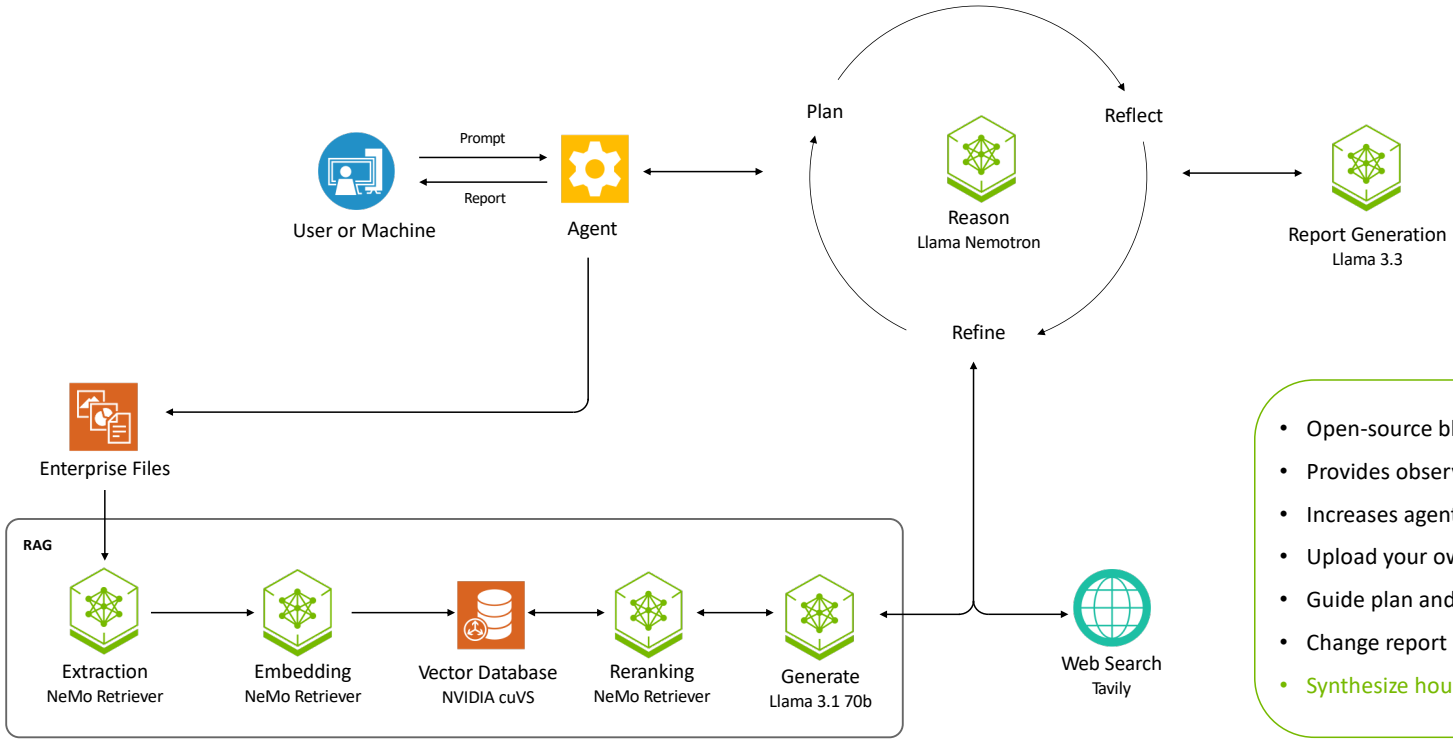
- Evaluate system level accuracy
- Understand and debug inputs and outputs for each component in the AI workflow

Agent Interconnect

- Universal descriptors for agents, tools, and workflows across frameworks
- Reusable Agent/Tool registry
- Workflow Configuration/Builder

AI-Q (Pronounced ‘IQ’) NVIDIA Blueprint

Use Reasoning to Connect AI Agents, Data, and Tools

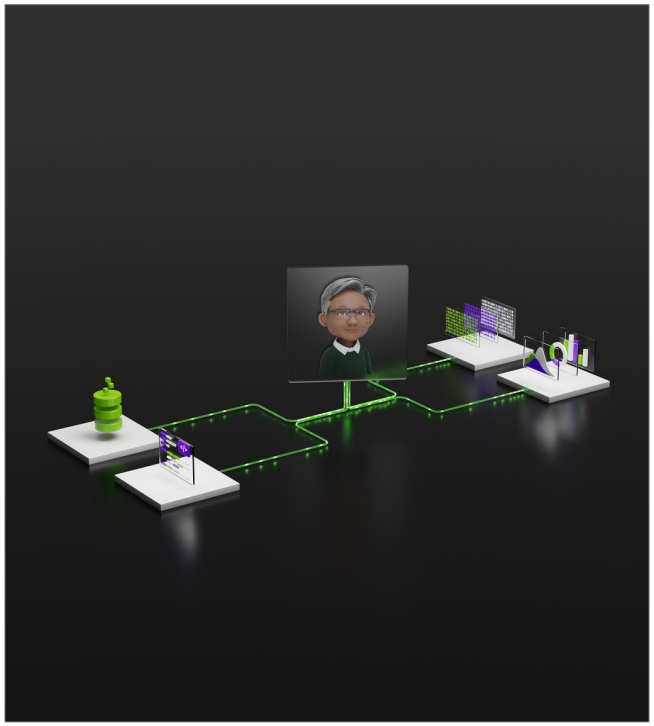


- Open-source blueprint for research & reporting
- Provides observability and transparency
- Increases agent accuracy with reasoning
- Upload your own data sources
- Guide plan and report in real-time
- Change report output dynamically
- Synthesize hours of material in minutes

Building Blocks of the AI-Q NVIDIA Blueprint

Use Reasoning to Connect AI Agents, Data, and Tools

NeMo Retriever
Connects AI Agents to Data



Llama Nemotron
Equips AI Agents with Reasoning Skills

Agent Intelligence Toolkit
Connects, Evaluates, Optimizes AI Agents

Llama Nemotron Reasoning Model Family

Leading Open Reasoning NIM Microservices for Agentic AI

Nano

Super

Ultra

NVIDIA NeMo Framework

Post-Training

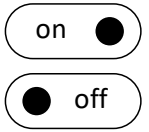
Open NVIDIA Datasets

Model-Building Recipe

60B Tokens, 360K H100 Hours, 45K Annotation Hours

Leading Accuracy

Highest Efficiency



Reasoning ON/OFF

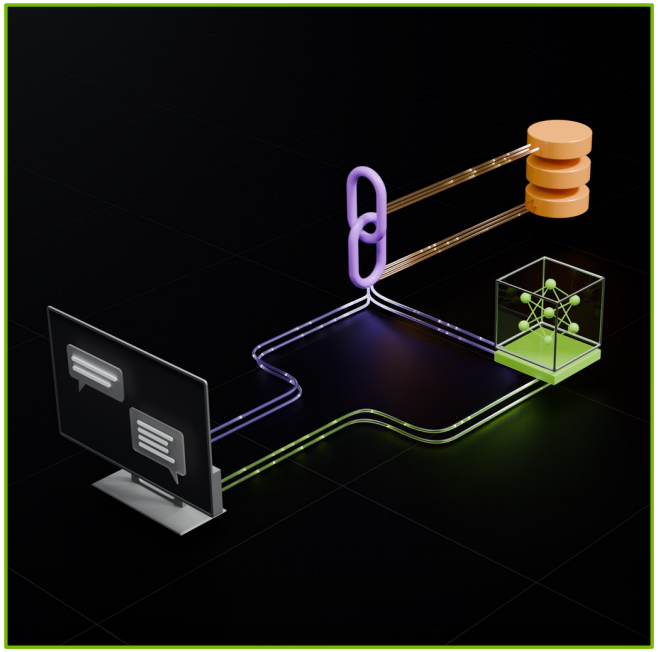
Enterprise Ready

Open

NVIDIA NeMo Retriever: Accurate, Real Time Insights from Enterprise Data

Now Supports Even More Data Types – Including Text, Tables, Charts, and Infographics from Millions of PDFs

NVIDIA AI Blueprint for RAG



NeMo Retriever

Extraction

Embedding

Reranking

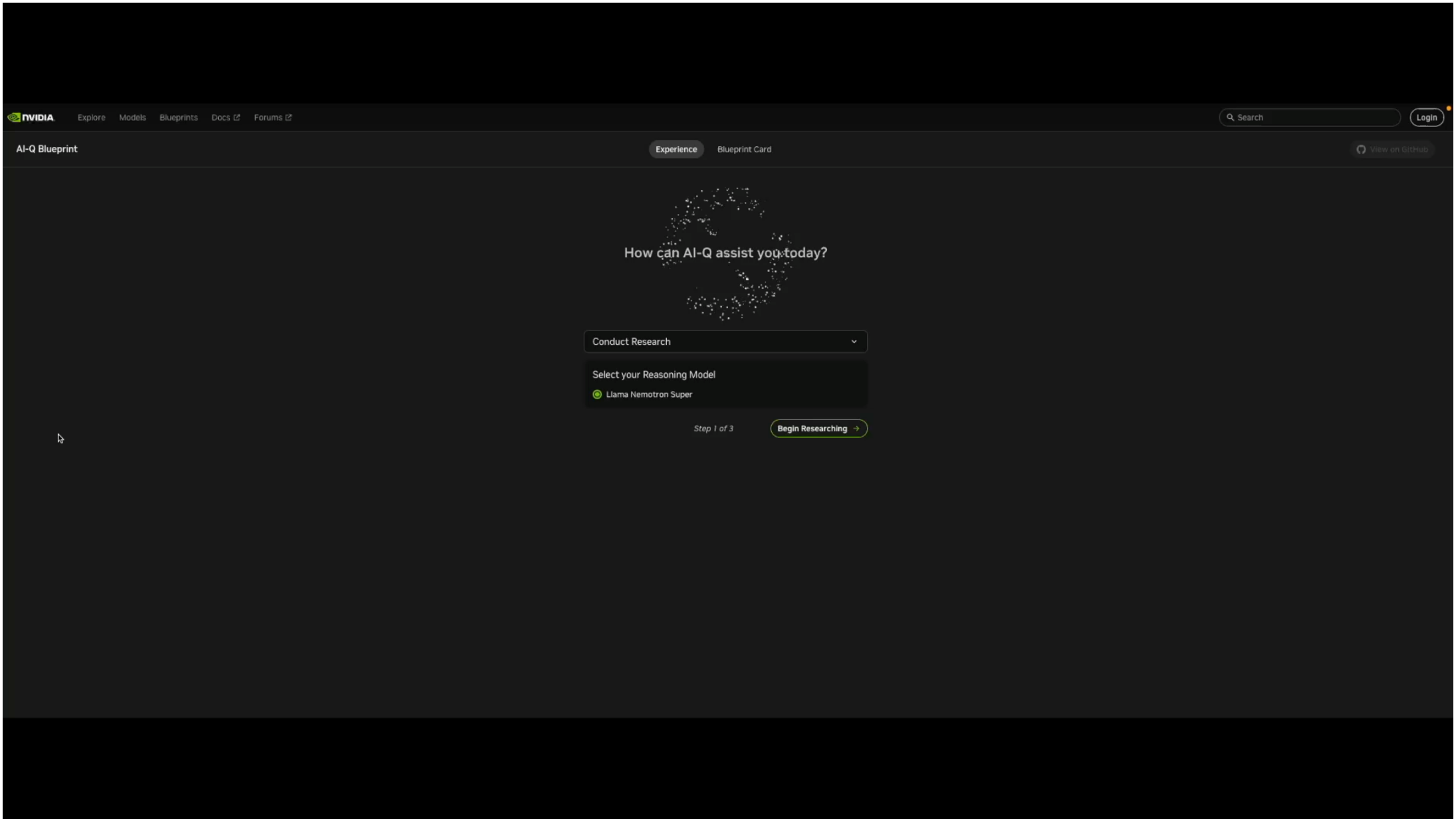
50%
fewer incorrect answers

15x
improved multimodal extraction throughput

35x
improved data storage efficiency

20%
reduced RAG costs

NVIDIA


NVIDIA
Explore
Models
Blueprints
Docs
Forums
Search
Login
AI-Q Blueprint
Experience
Blueprint Card
View on GitHub
How can AI-Q assist you today?
Conduct Research
Select your Reasoning Model
Llama Nemotron Super
Step 1 of 3
Begin Researching

NVIDIA Agent Intelligence (AI-Q) Toolkit

An open-source library for building enterprise-ready agentic systems

Profiling & Optimizations

- Fine-grained AI workflow telemetry collected can be used to implement agentic system accelerations.

Evaluation & Observability

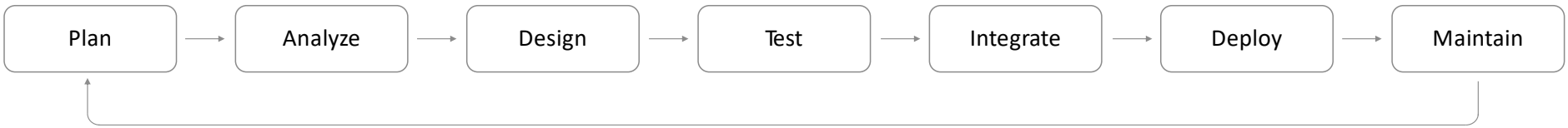
- Evaluate system level accuracy
- Understand and debug inputs and outputs for each component in the AI workflow

Agent Interoperability

- Universal descriptors for agents, tools, and workflows across frameworks
- Reusable Agent/Tool registry
- Workflow Configuration/Builder

Enabling Software Development Lifecycle for AI Agents

NVIDIA AI-Q Toolkit



NVIDIA AI-Q Toolkit

Agent Interconnect

- ✓ Convert tools into Agents
- ✓ Workflow configuration/builder
- ✓ Create agent registry

Implementation & Testing

- ✓ Evaluate agent and system accuracy
- ✓ Debug prompts and workflow
- ✓ Profile performance

Deployment & Monitoring

- ✓ Collect telemetry
- ✓ Monitor workflow behavior and performance
- ✓ Optimize agent scaling and deployment

NVIDIA AI-Q Toolkit

Accelerate AI Agents and Streamline Agentic Workflow Optimization

SAVE TIME

Simplify the development of agentic systems

- Flexibly choose, and connect, agent frameworks best suited for each task
- Easily reuse existing and new RAG pipelines, different Agentic workflows, and tools across your Enterprise
- Quickly elevate existing Gen AI workflows to Agentic AI workflows

REDUCE COSTS

Accelerate agent responses—do more with what you have

- System level optimizations provide accelerated Agentic AI performance
- NVIDIA AgentIQ collects telemetry that provides opportunities for optimization, driving efficiency for an agentic workflow.

IMPROVE BUSINESS OUTCOMES

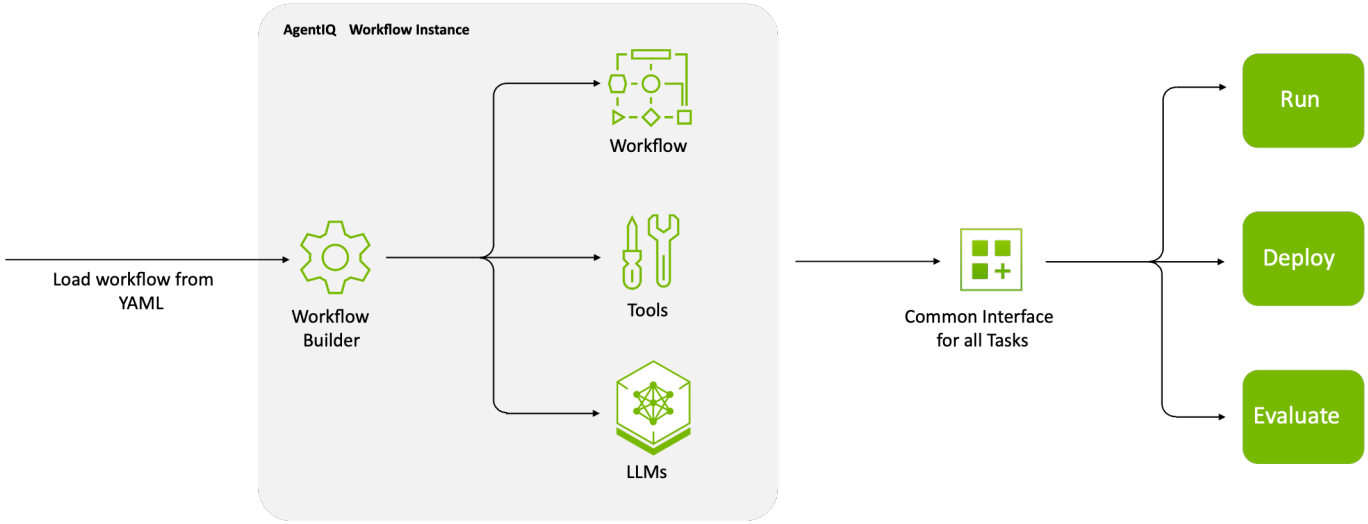
Increase agentic system accuracy

- Evaluate agentic system response accuracy
- Understand and debug inputs and outputs for each component in the system
- Traceability and auditing of agent communications

Configuring a Heterogeneous AI Query Engine

```yaml
general:
  use_uvloop: true

functions:
  llama_index_rag_tool:
    _type: llama_index_rag
    llm_name: nim_llm
    embedding_name: nim_embedder
  langchain_rag_tool:
    _type: langchain_rag
    llm_name: nim_llm
    embedding_name: nim_embedder
  semantic_kernel_rag_tool:
    _type: semantic_kernel_rag
    llm_name: nim_llm
    embedding_name: nim_embedder
  react_agent:
    _type: tool_calling_agent
    tool_names:
      - llama_index_rag_tool
      - langchain_rag_tool
      - semantic_kernel_rag_tool
    llm_name: nim_llm

llms:
  nim_llm:
    _type: nim
    model_name: meta/llama-3.3-70b-instruct
    temperature: 0.0
    max_tokens: 1024
  nim_reasoning:
    _type: nim
    model_name: nvidia/llama-3.3-nemotron-49b-instructllama_nemotron
    temperature: 0.5
    max_tokens: 1024
    top_p: 1
    max_tokens: 1024

workflow:
  _type: reasoning_agent
  llm_name: r1_model
  augmented_fn: react_agent
  verbose: true
```

- **Reasoning Agent**
  - Plans the execution strategy using a Nemotron reasoning NIM
- **Tool Calling Agent**
  - Executes instructions generated by the Reasoning Agent using a Llama instruct model
- **Heterogenous RAG**
  - Each RAG engine exposed as a tool providing a heterogeneous AI Query Engine.



Transforming Banking with AI

RBC Aiden

NVIDIA AI-Q

AI Data

Structured

- Trade & Reference
- Market

Unstructured

- Internal research analyst reports
- News
- Regulatory Filings (SEC 10K, 10Q, financial)

Data Prep

AI Data Platform

KX
Data Set Curation
Domain data sets
Model Store

NVIDIA RAPIDS

NeMo Curator

Model Telemetry

Train Model

Quant Fin HPC Models

ETL/ML models (ex. NVIDIA RAPIDS)

Algos

Hybrid RAG, Prompt & Fine Tuning

NVIDIA AI Enterprise

Validation

Bias

Deployment

Inference Serving

NVIDIA NIM

Models
LLMs
AI/ML

AI Agents

Monitoring

NeMo Guardrails

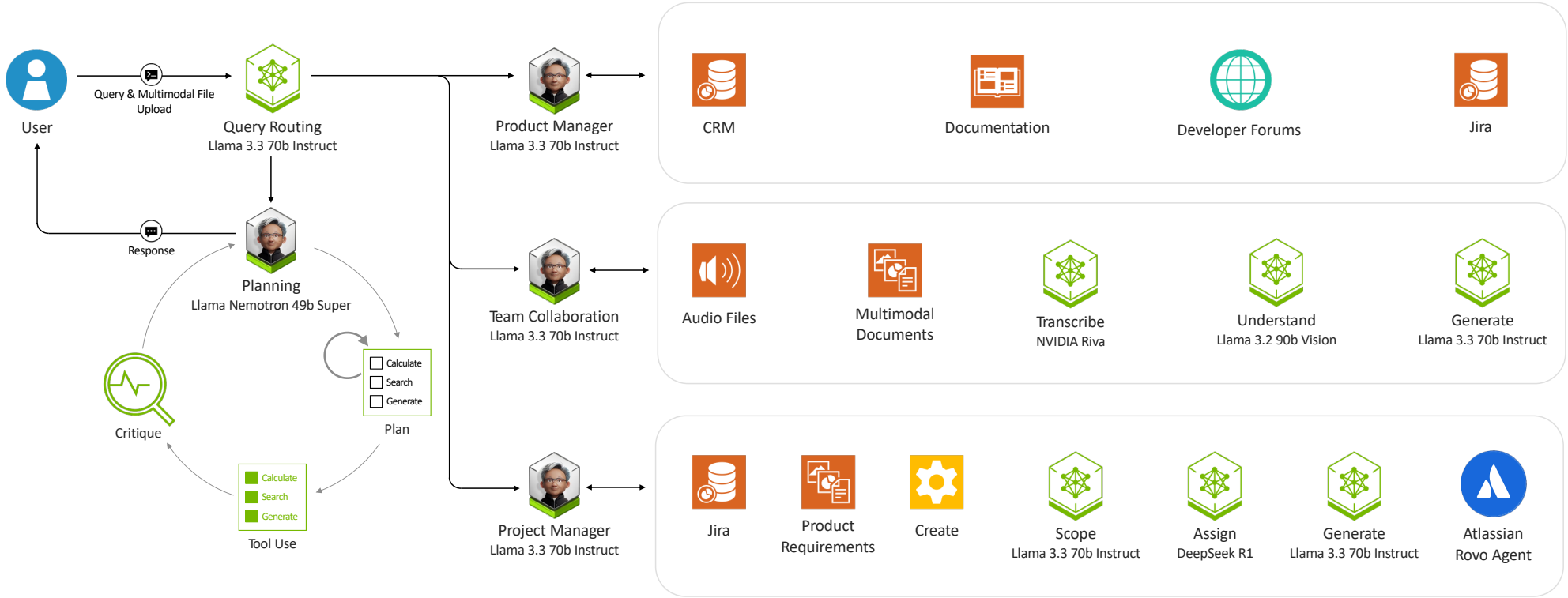
AI Governance

Use Cases

- **Hyper-personalized client experience**
- **Research efficiency:** earning call analytics, trend discovery, enhancing insights
- **Knowledge retrieval**
- **Trading intelligence**
- **Operational automation**

Using AI-Q for Product Lifecycle Management

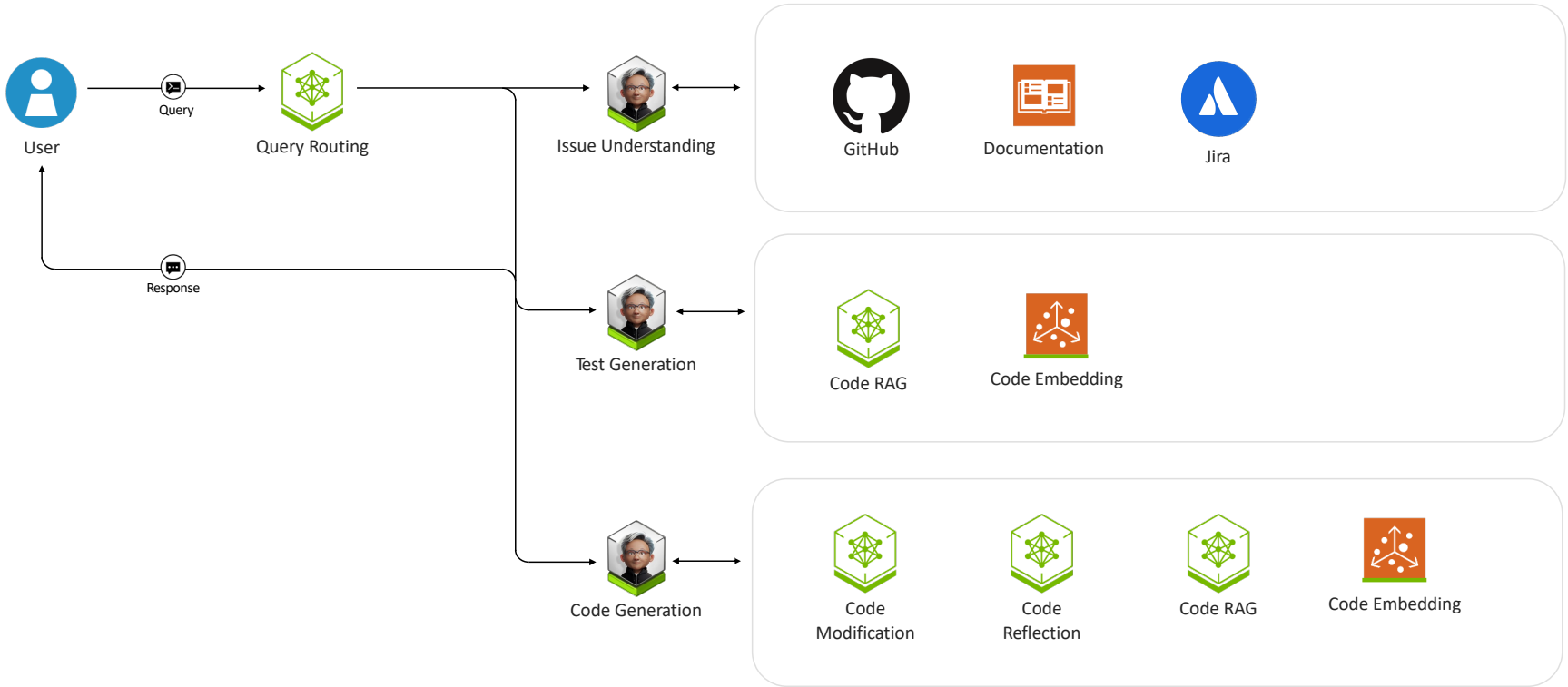
Respond to customer requests in hours, not weeks



Future of Work Demo Video

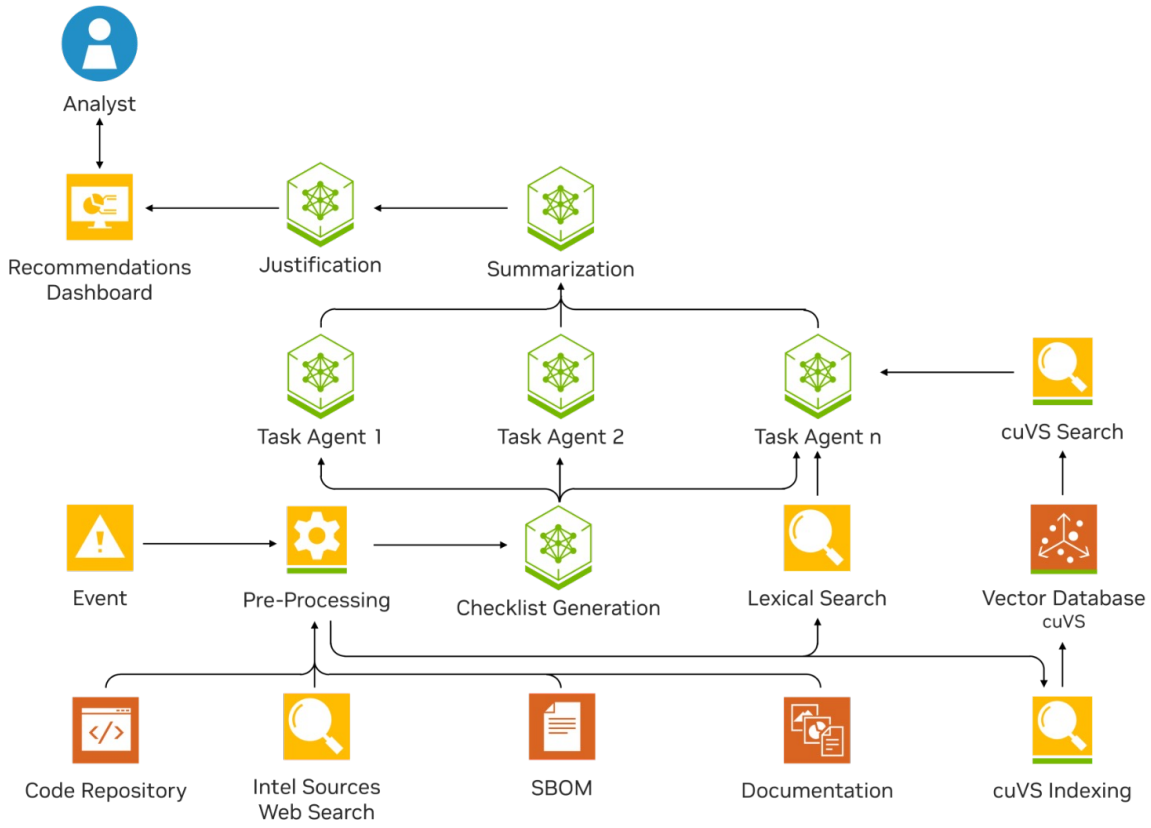
Product and Program Management AI Agents

Using AI-Q for Test Driven Development



Using AI-Q for CVE Analysis (Agent Morpheus)

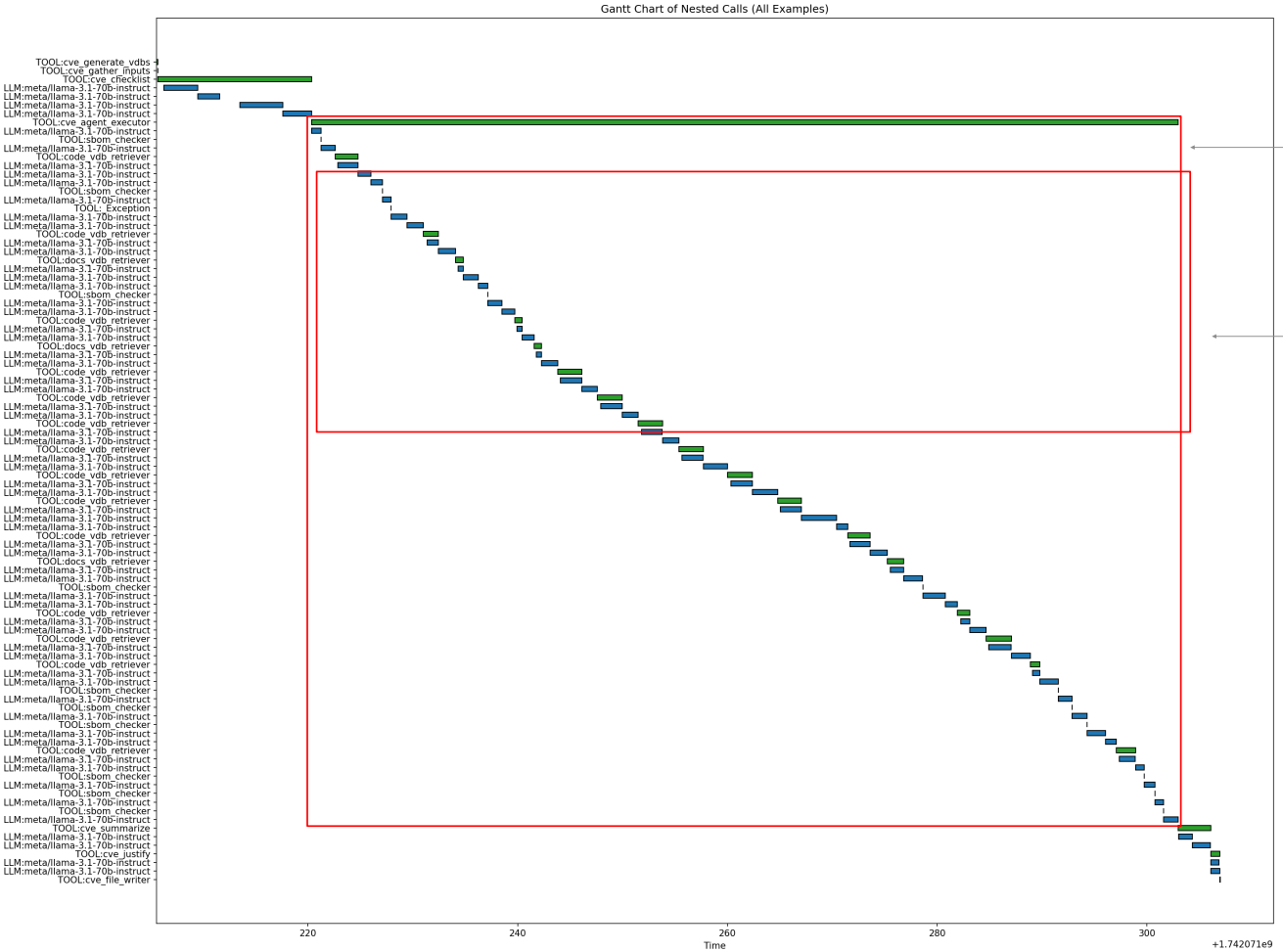
Built using AI-Q toolkit, triage vulnerabilities automatically



- Triage software for vulnerabilities and exposures in seconds versus days or hours
- LLM agent can autonomously perceive, reason, act
- Presents summary of findings to human analyst
- Empowers human analyst to make decisions faster

Identifying Workflow Bottlenecks with the Profiler

Pre-optimization: Understand where agents spend most of their time



It appears that the Agentic Triage step is taking the longest time, at **82.62 seconds***

Most of the latency appears to be coming from the sequential execution of tools and agents* for each checklist item. **Could we do better?**

Key Ideas for Optimization:

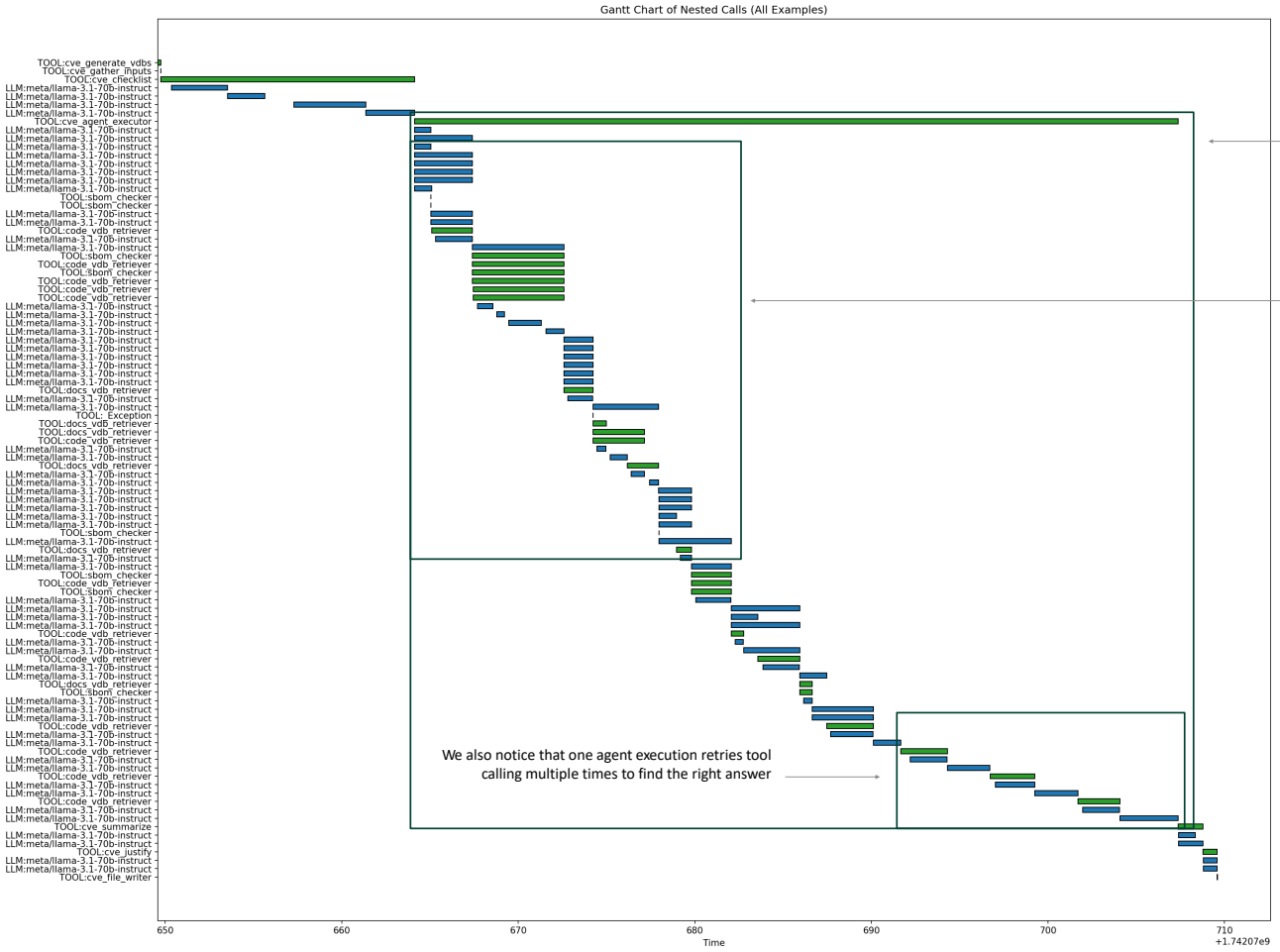
- CVE's can be processed independently and instead of having to wait for one CVE to process before another does!
- Checklist items for each CVE can also be processed in parallel to reduce macro-latency.

* All metrics are extracted from the AgentIQ profiling report, not pictured here for brevity.

NVIDIA

Identifying Workflow Bottlenecks with the Profiler

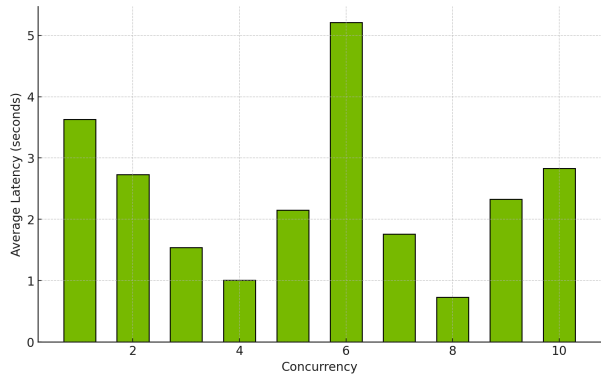
Targeted Optimization of Identified Bottlenecks



We notice a **2x reduction in latency by remediating identified bottlenecks** using optimization strategies. The total runtime is now 42.26 seconds.

Agents are executed in parallel for each CVE and checklist item. **Reducing end-to-end latency while saturating the GPU**. This will ultimately reduce TCO.

The AgentIQ Profiler also provides insight into LLM latency at various observed concurrencies, allowing us to optimize our LLM deployment strategy.

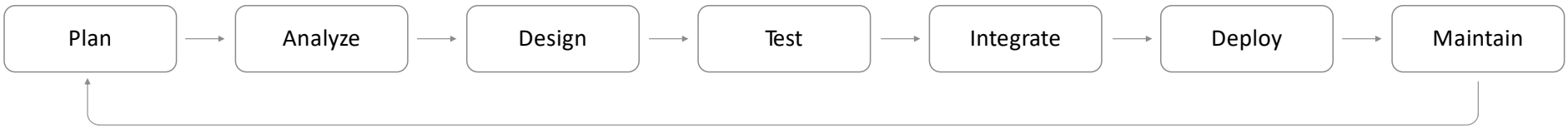


* All metrics are extracted from the AgentIQ profiling report, not pictured here for brevity.

NVIDIA.

Enabling Software Development Lifecycle for AI Agents

NVIDIA NeMo Agent Toolkit for Developing AI Agent Teammates



NVIDIA NeMo Agent Toolkit

Implementation & Testing

- ✓ Evaluate agent and system accuracy
- ✓ Debug prompts and workflow
- ✓ Profile performance

Deployment & Monitoring

- ✓ Collect telemetry
- ✓ Monitor workflow behavior and performance
- ✓ Optimize agent scaling and deployment

Agent Interconnect

- ✓ Convert tools into Agents
- ✓ Workflow configuration/builder
- ✓ Create agent registry

Unified Agentic AI: Collaboration, Communication, and Tool Access

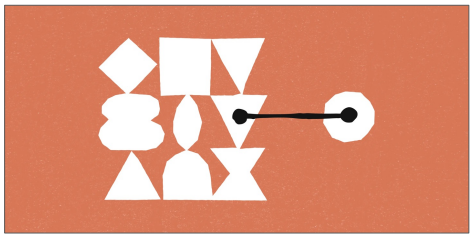
NVIDIA NeMo Agent Toolkit, Anthropic MCP, and Google A2A

NVIDIA Agent Toolkit

Optimization, profiling, and framework agnostic connections

- Comprehensive toolkit for building, profiling, and optimizing teams of AI agents across any framework
- Provides universal descriptors for agents and tools, enabling cross-framework compatibility
- Focuses on performance optimization, evaluation, and observability of entire agentic systems

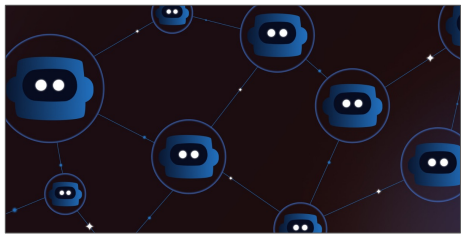
MCP



Standardized agent-to-tool integration

- Focuses on vertical integration—connecting individual AI models with external tools and data
- Optimized for direct request-response flows between a model and its tools
- Designed for controlled tool access rather than peer-to-peer agent collaboration

A2A



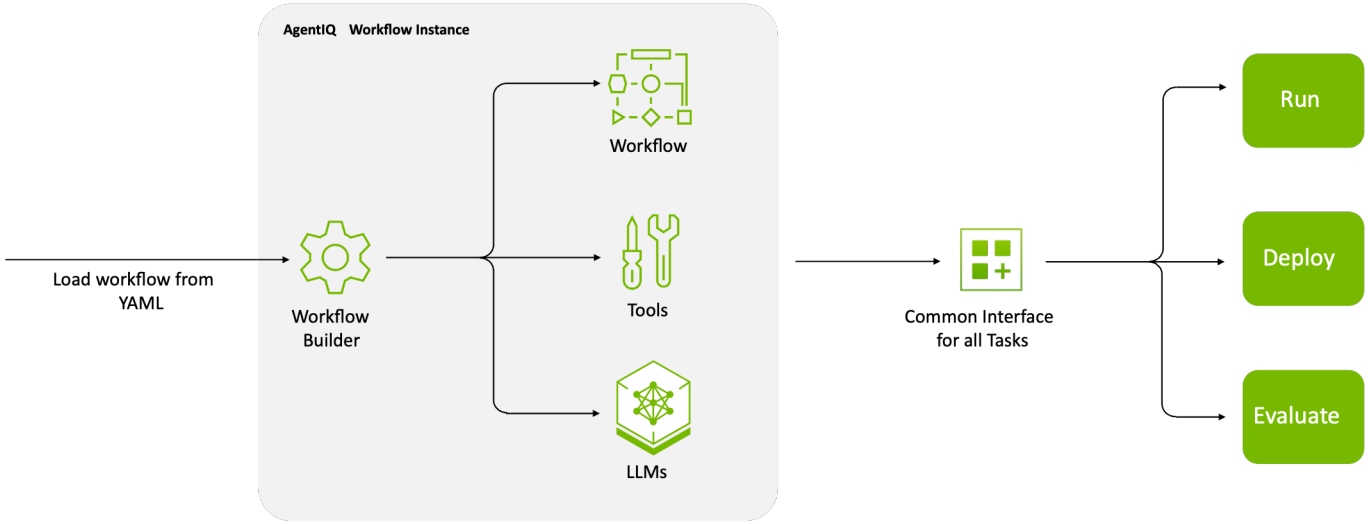
Standardized agent-to-agent communication

- Focuses on horizontal integration—enabling communication between different AI agents
- Creates standardized protocols for agents to discover each other and collaborate across platforms
- Designed primarily for agent-to-agent communication rather than tool usage or optimization

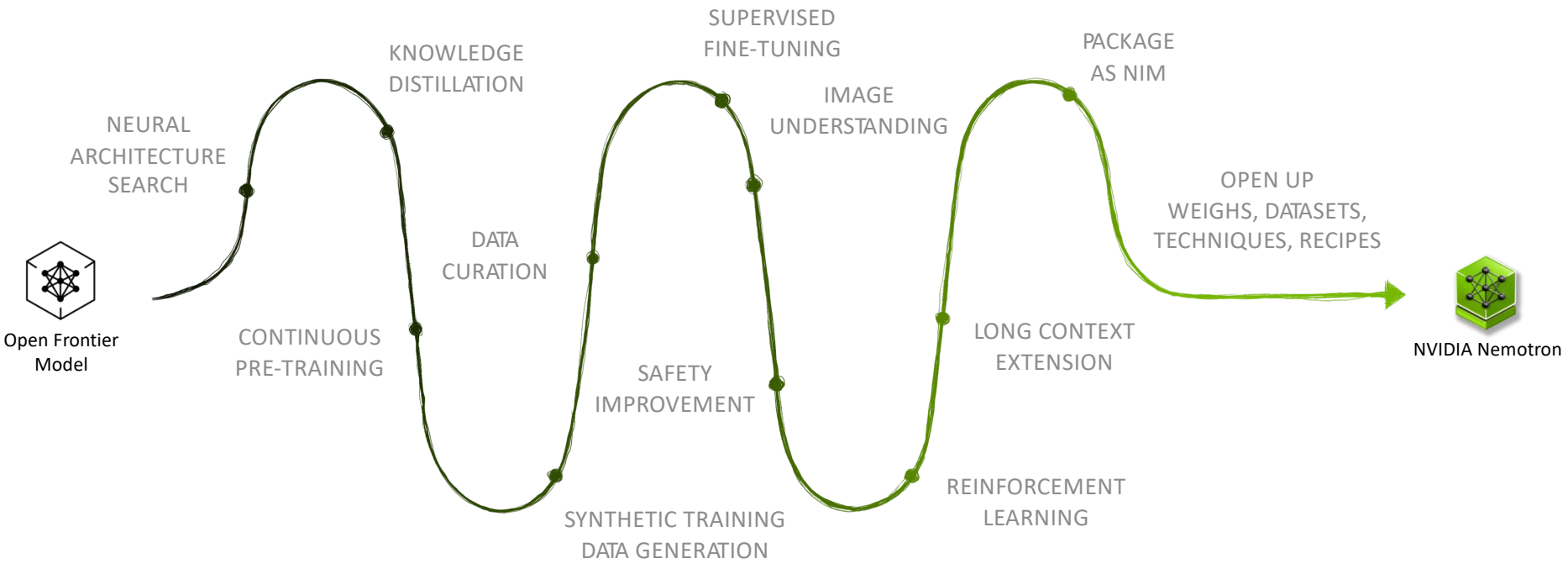
Configuring a Heterogeneous AI Query Engine

```yaml
general:
  use_uvloop: true

functions:
  llama_index_rag_tool:
    _type: llama_index_rag
    llm_name: nim_llm
    embedding_name: nim_embedder
  langchain_rag_tool:
    _type: langchain_rag
    llm_name: nim_llm
    embedding_name: nim_embedder
  semantic_kernel_rag_tool:
    _type: semantic_kernel_rag
    llm_name: nim_llm
    embedding_name: nim_embedder
  react_agent:
    _type: tool_calling_agent
    tool_names:
      - llama_index_rag_tool
      - langchain_rag_tool
      - semantic_kernel_rag_tool
    llm_name: nim_llm

llms:
  nim_llm:
    _type: nim
    model_name: meta/llama-3.3-70b-instruct
    temperature: 0.0
    max_tokens: 1024
  nim_reasoning:
    _type: nim
    model_name: nvidia/llama-3.3-nemotron-49b-instructllama_nemotron
    temperature: 0.5
    max_tokens: 1024
    top_p: 1
    max_tokens: 1024

workflow:
  _type: reasoning_agent
  llm_name: r1_model
  augmented_fn: react_agent
  verbose: true
```

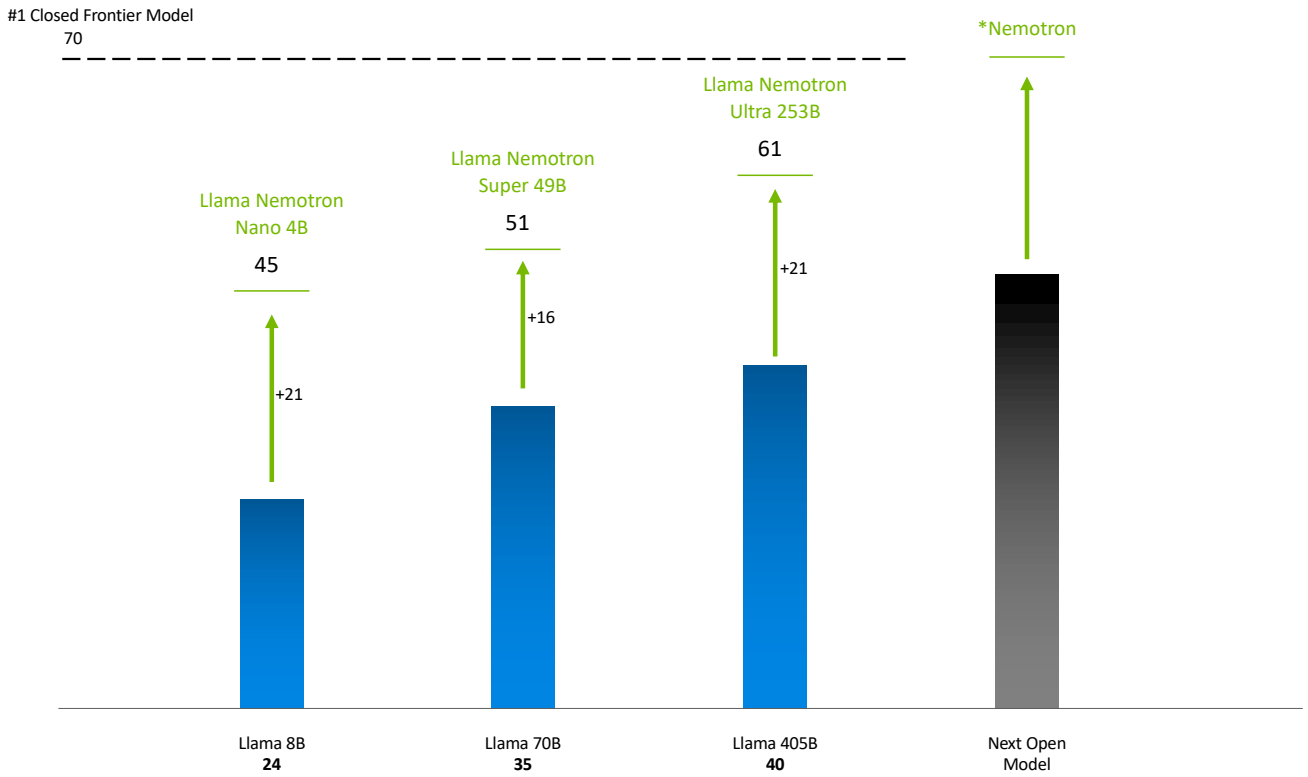
- **Reasoning Agent**
  - Plans the execution strategy using a Nemotron reasoning NIM
- **Tool Calling Agent**
  - Executes instructions generated by the Reasoning Agent using a Llama instruct model
- **Heterogenous RAG**
  - Each RAG engine exposed as a tool providing a heterogeneous AI Query Engine.



NVIDIA Nemotron Further Advances Leading Open Models

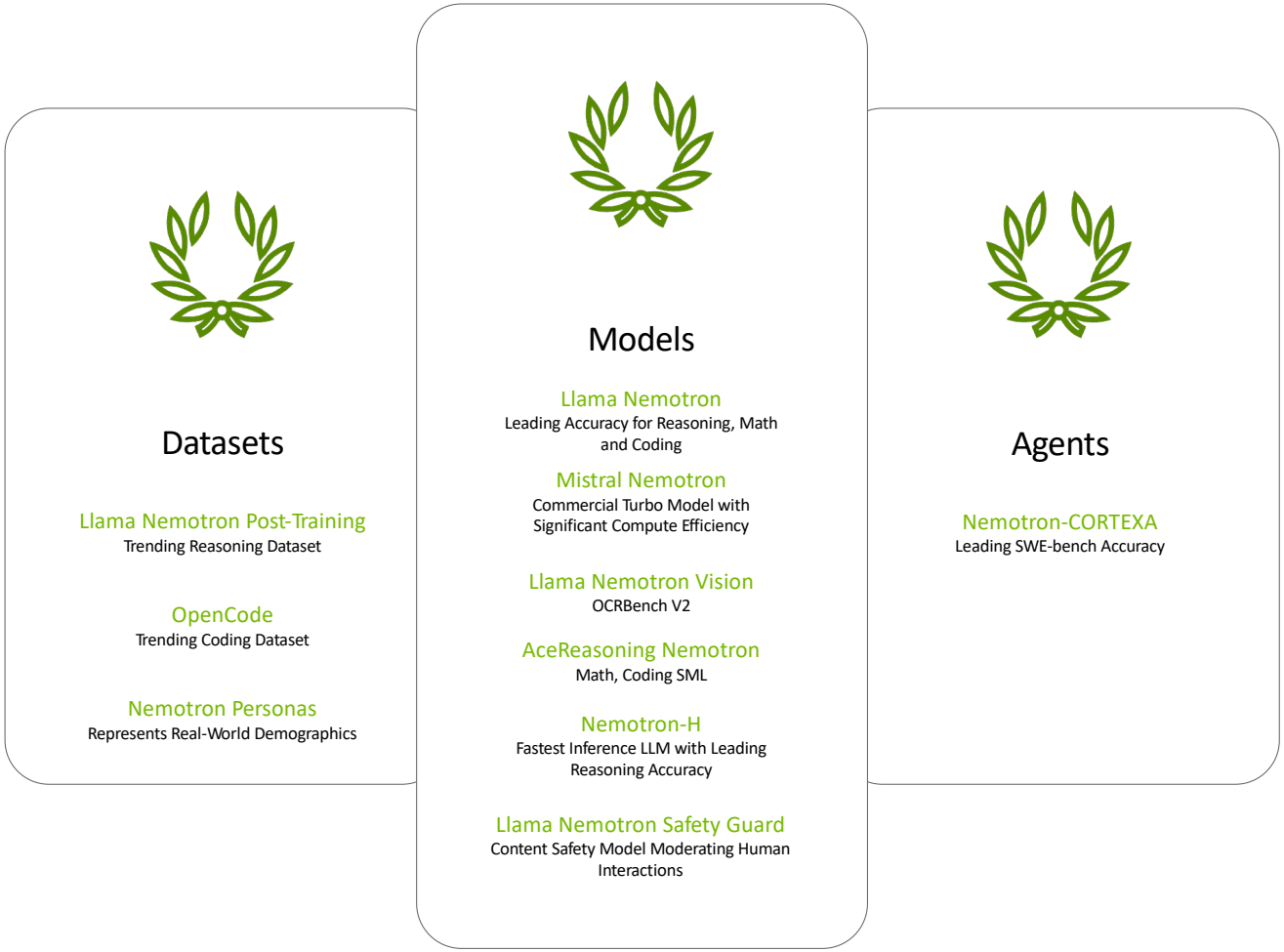


NVIDIA Nemotron Increases Open Model Efficiency and Accuracy



Artificial Analysis Intelligence Index

NVIDIA Nemotron Achieves World-Class Accuracy

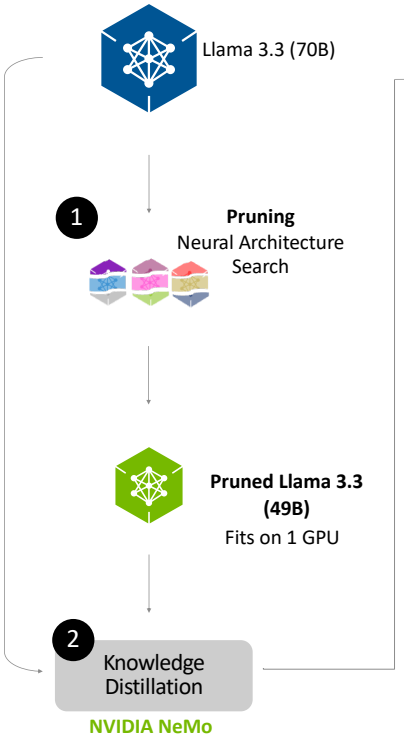


Detailed

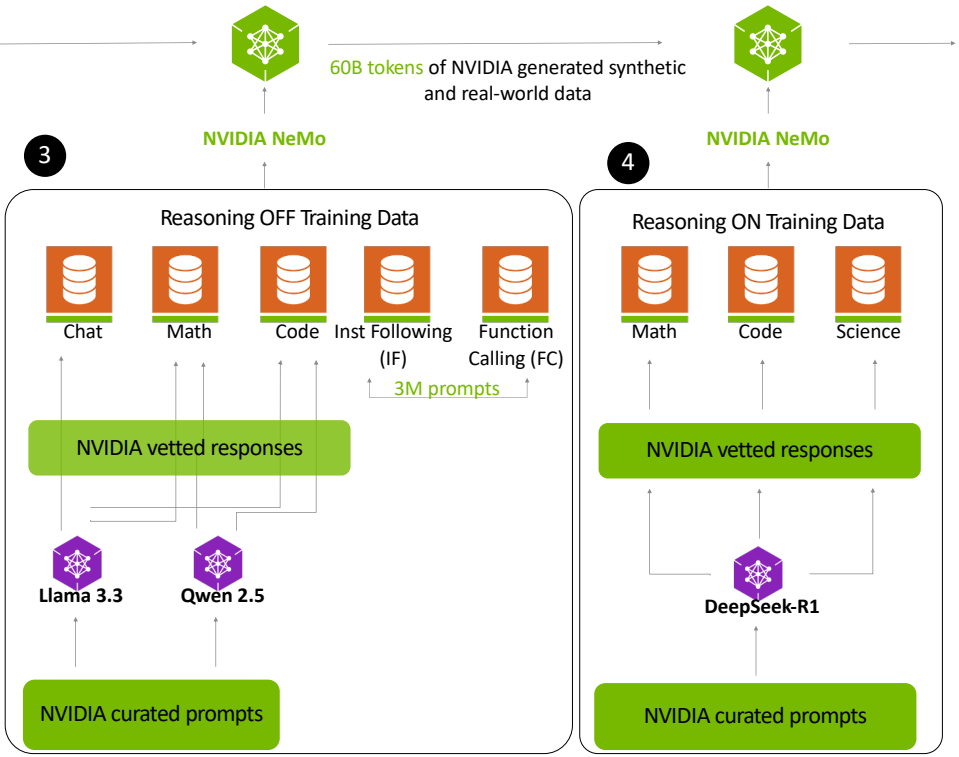
The Making of Llama Nemotron with Reasoning

Built on Internet-Scale Knowledge, Trained with NVIDIA Curated Reasoning Skills

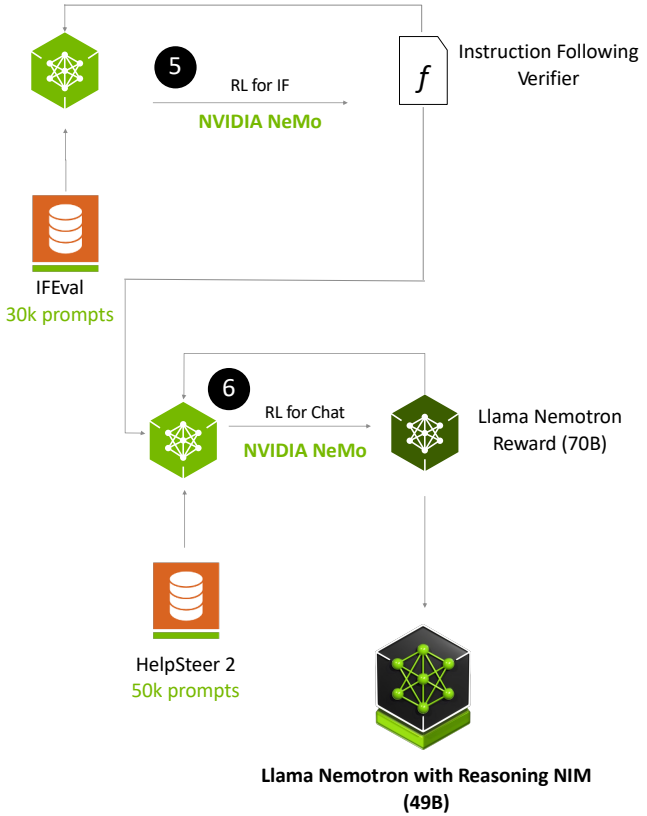
Distillation
Improve model efficiency



Supervised Fine-Tuning
Improving Agentic Skills with Reasoning



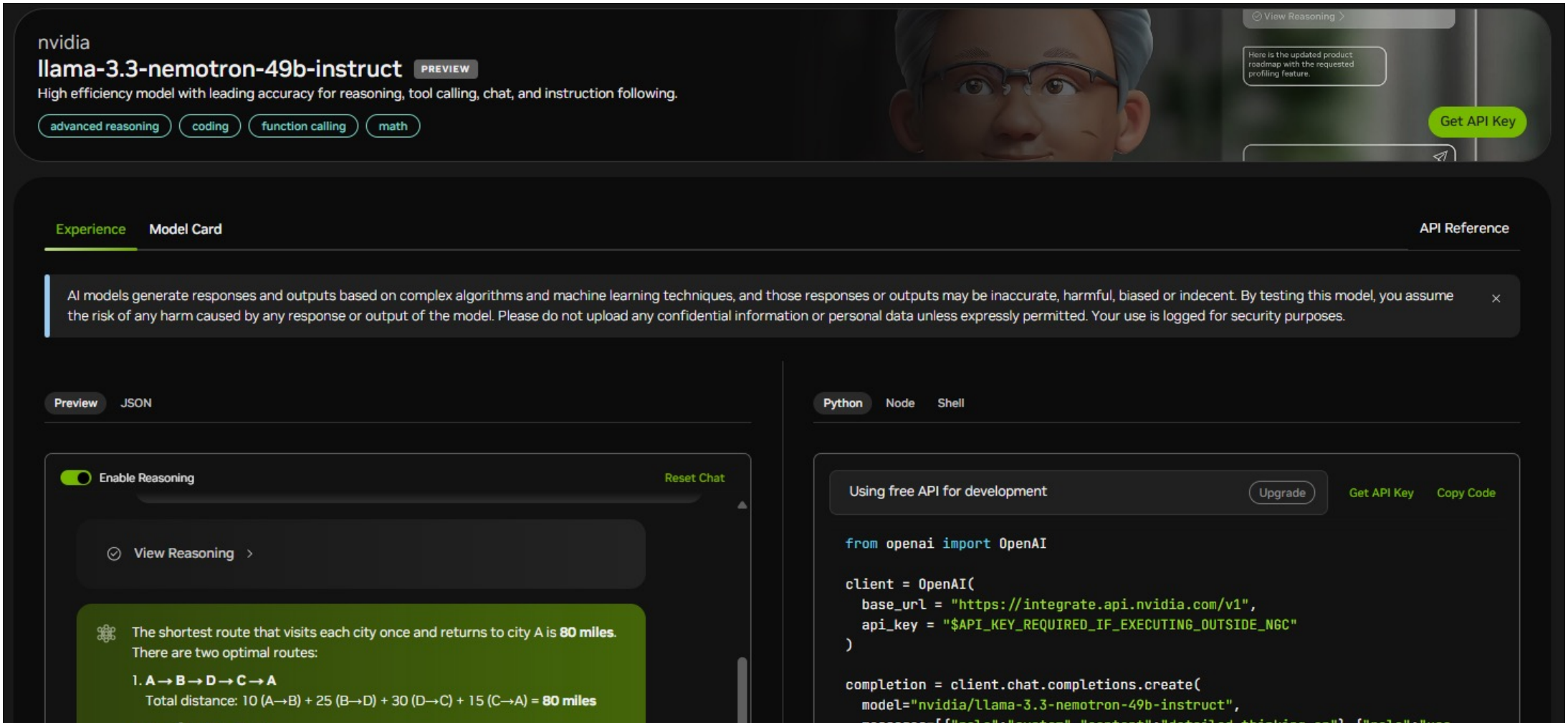
Reinforcement Learning (RL)
Aligning for Human Preferences



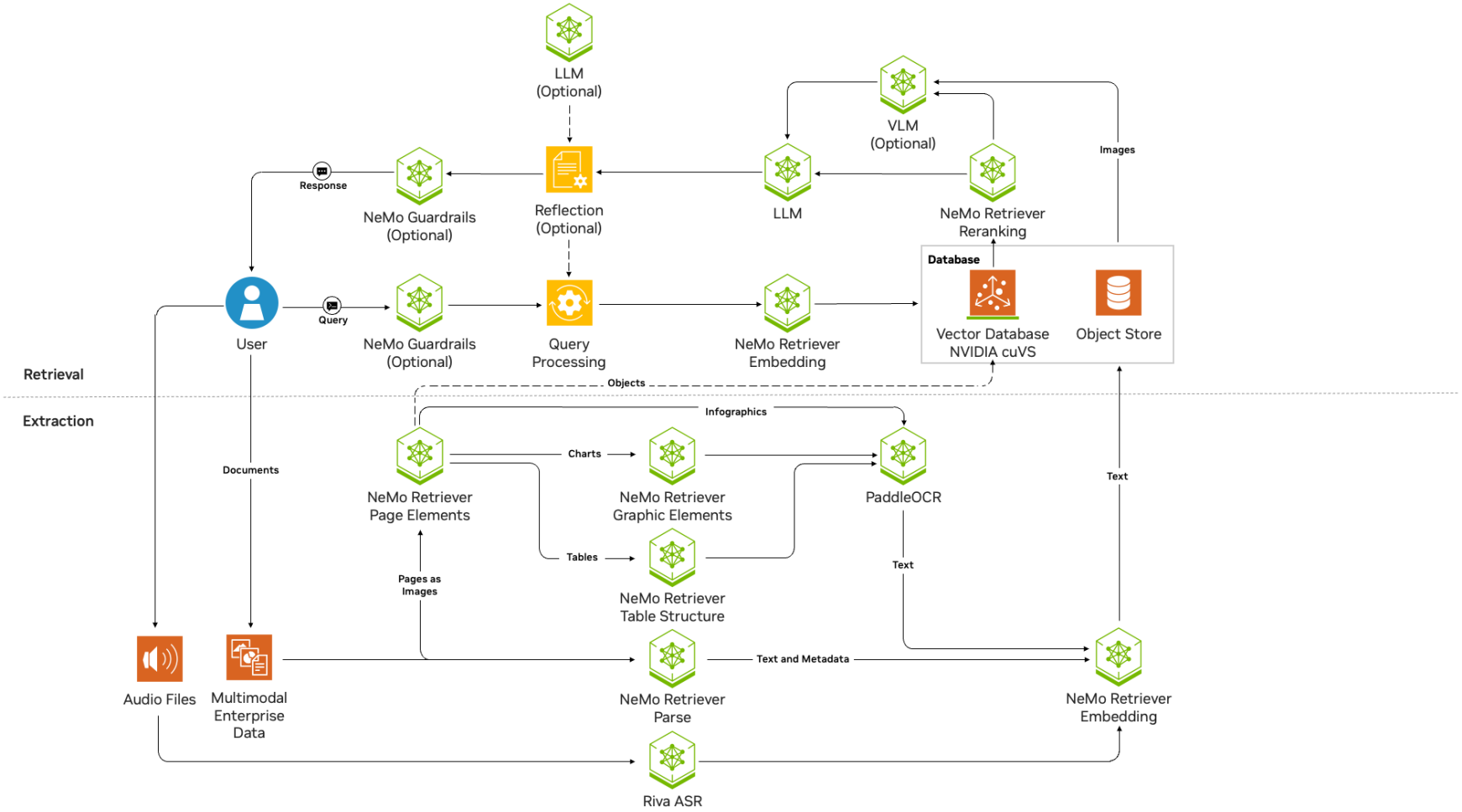
NVIDIA

Demo

Try Nemotron NIM Now **build.nvidia.com**

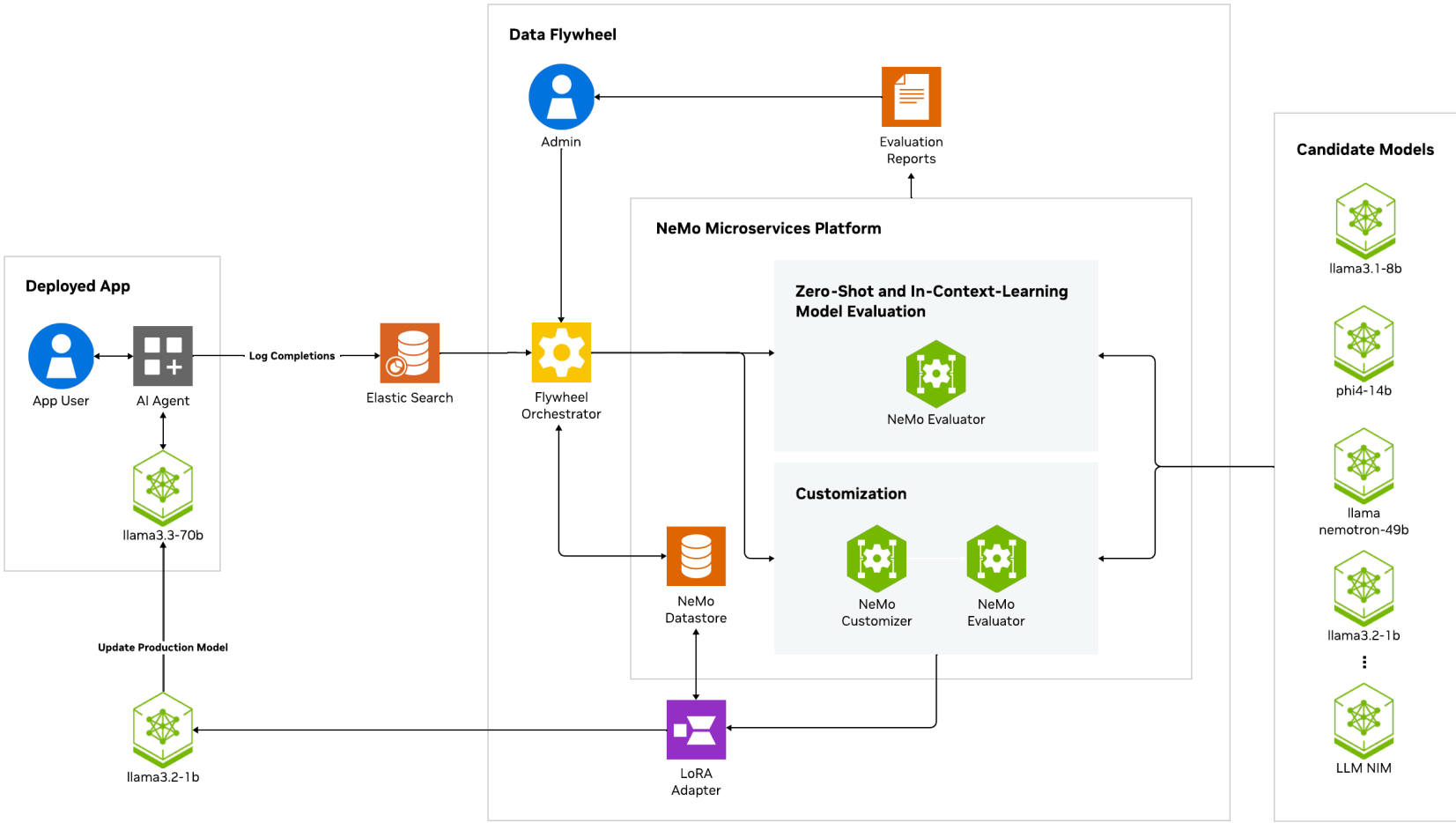


NVIDIA RAG Blueprint with NeMo Retriever and Riva



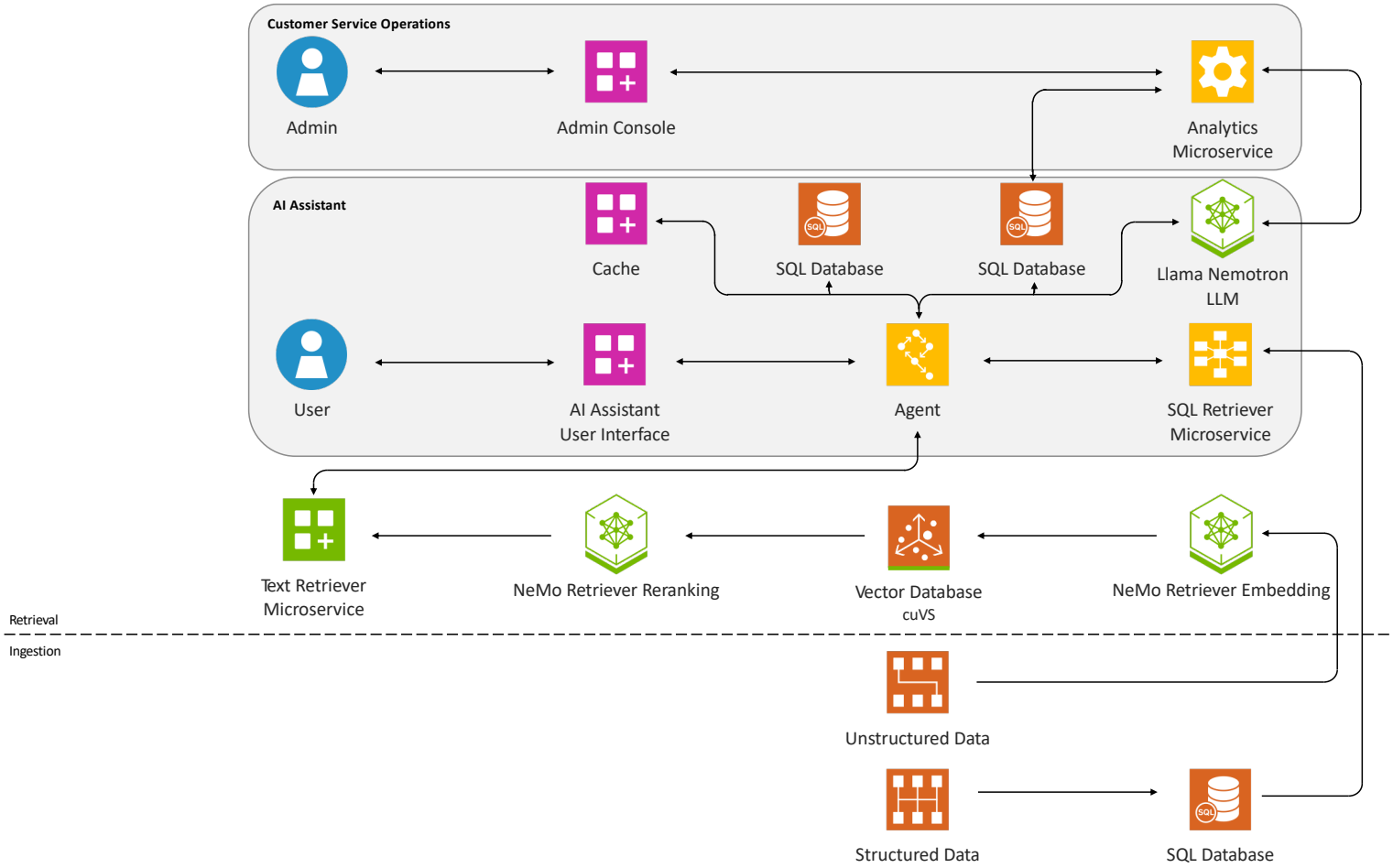
NVIDIA AI Blueprint for Building Data Flywheels

Set-up Self-Optimizing Agentic Workflow With Ease Powered by NeMo Microservices



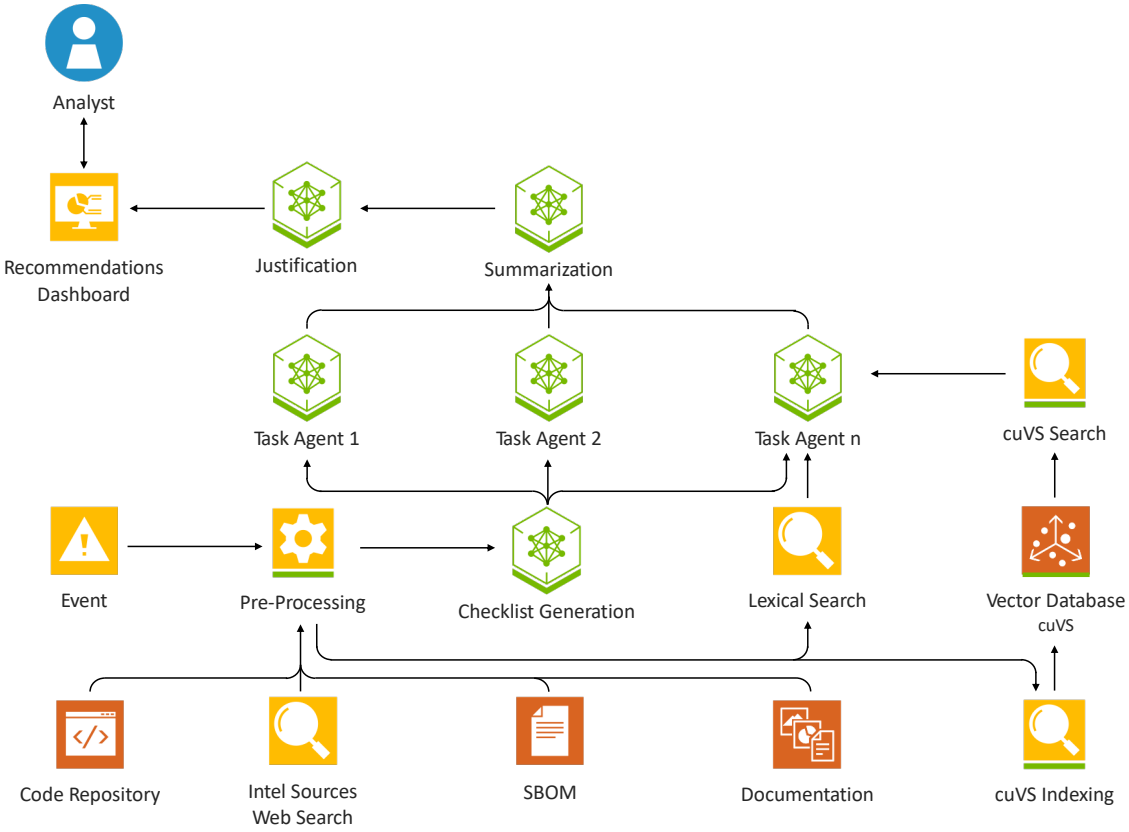
AI Virtual Assistant for Customer Service

Help address the nearly 1 billion daily contact center calls with personalized service



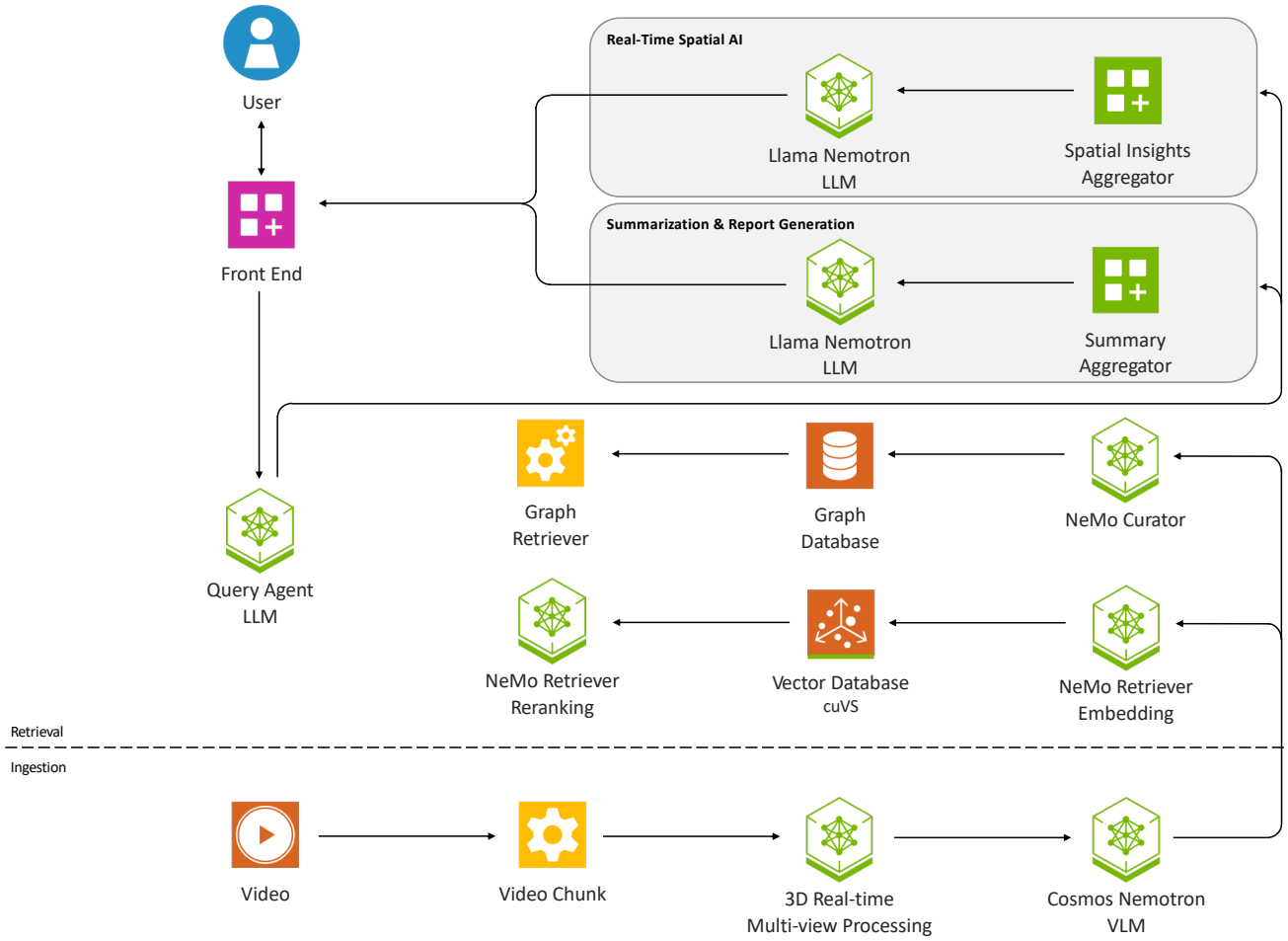
Software Security AI Agent

Analyze software vulnerabilities, reducing mitigation time from days to seconds



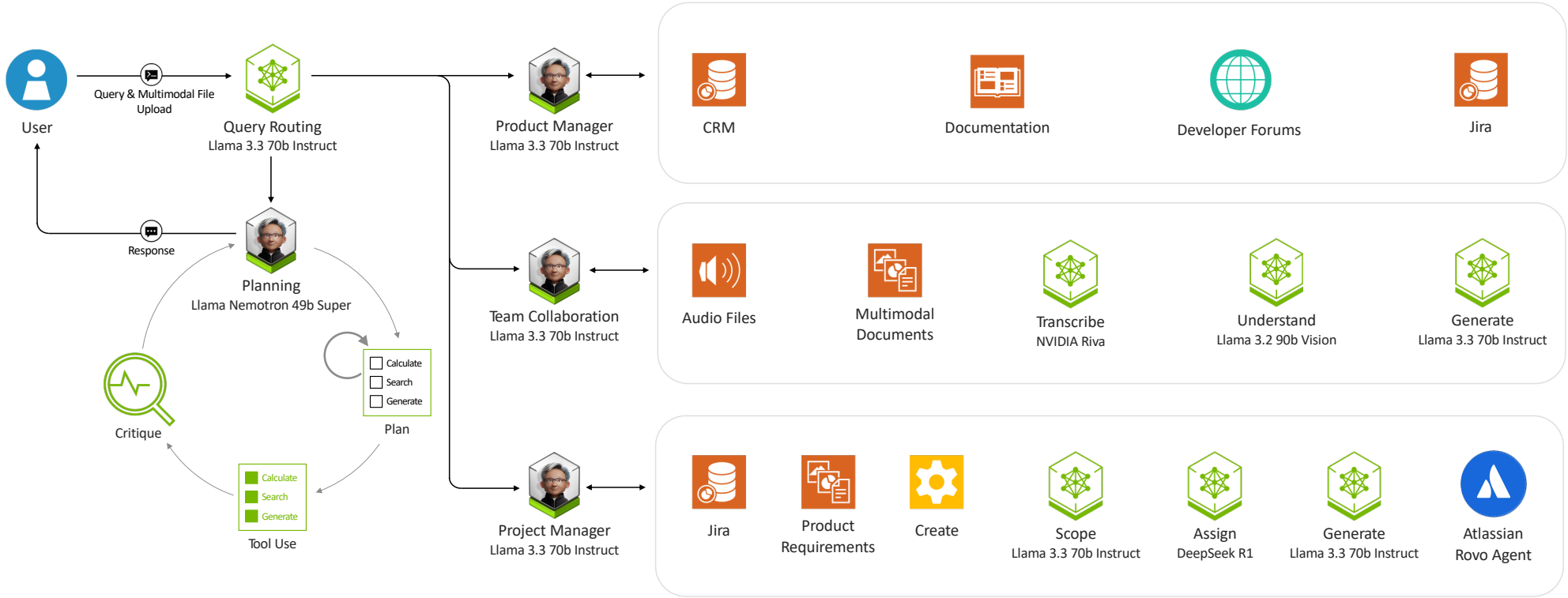
Video Analytics AI Agent

Analyze content from the billions of cameras generating 100 thousand petabytes of video per day



Using the Toolkit for Product Lifecycle Management

Respond to customer requests in hours, not weeks



How to Get Started

Build, Connect, Evaluate, Optimize AI Agents

Try

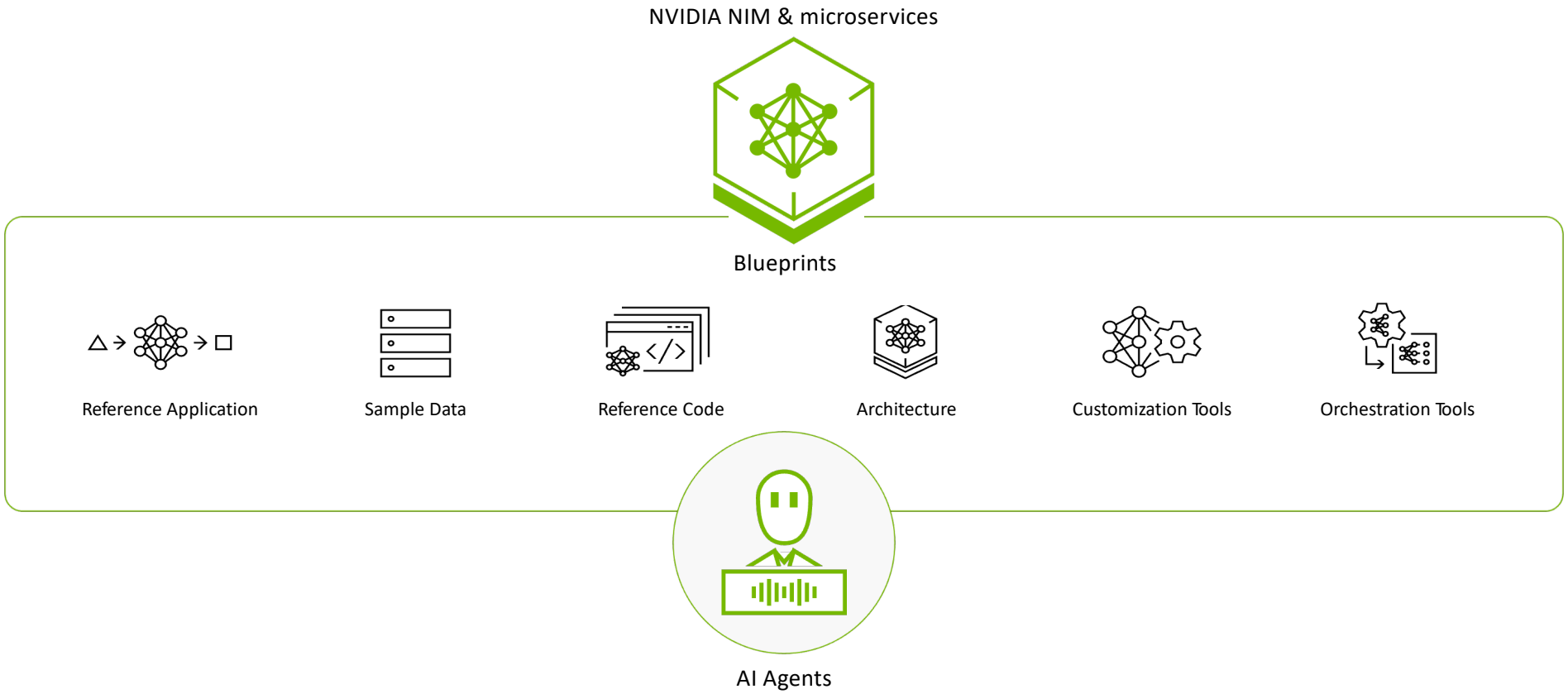
Use the AI-Q NVIDIA Blueprint in GitHub or use the Launchable starting at build.nvidia.com/nvidia/aiq.

Develop

Access NVIDIA Agentic AI Toolkit code in GitHub, read the documentation, watch ‘how-to’ videos.
developer.nvidia.com/AIQ or
github.com/NVIDIA/AIQ

NVIDIA Blueprints

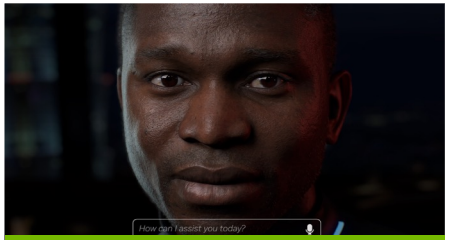
Available on build.nvidia.com



NVIDIA Blueprints

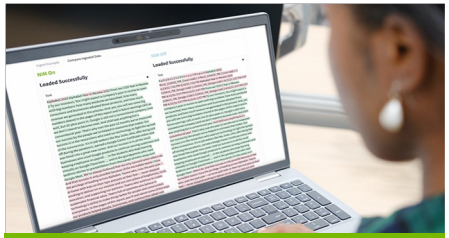
Available on build.nvidia.com

Digital Humans
for Customer Service



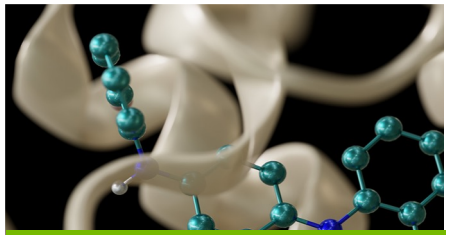
NVIDIA AI Blueprint

Multimodal PDF Data Extraction
for Enterprise RAG



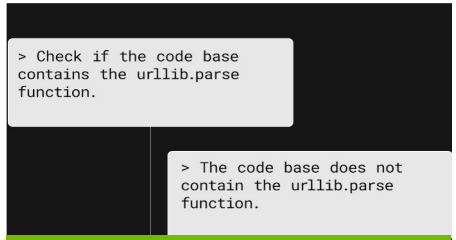
NVIDIA AI Blueprint

Generative Virtual Screening
for Drug Discovery



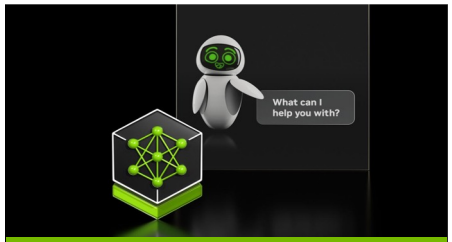
NVIDIA BioNeMo Blueprint

Vulnerability Analysis
for Container Security



NVIDIA AI Blueprint

AI Virtual Assistants
for Customer Service



NVIDIA AI Blueprint

Visual AI Agent
for Video Search and Summarization

NVIDIA AI Blueprint

3D Conditioning for
Precise Visual Generative AI

NVIDIA Omniverse Blueprint

Build a Digital Twin for
Interactive Fluid Simulation

NVIDIA Omniverse Blueprint

NVIDIA NIM Powers Generative AI Factories

Building blocks of multi-modal agentic AI

Language & Reasoning

Regional and Domain Specialized

Visual

RAG

Speech

Digital Human

Healthcare

Computer Vision & Simulation

Safety

build.nvidia.com

NVIDIA NIM Optimized Inference Microservices

Rapidly deploy reliable building blocks for accelerated generative AI anywhere

Portable Run cloud-native microservices anywhere, maintaining security and control of data and apps

Easy to Use Move fast with the latest agentic AI building blocks for reasoning, retrieval, images and more, deployed in minutes with standard APIs

Enterprise Supported Gain confidence with stable APIs, quality assurance, continuous updates, security patching, and support

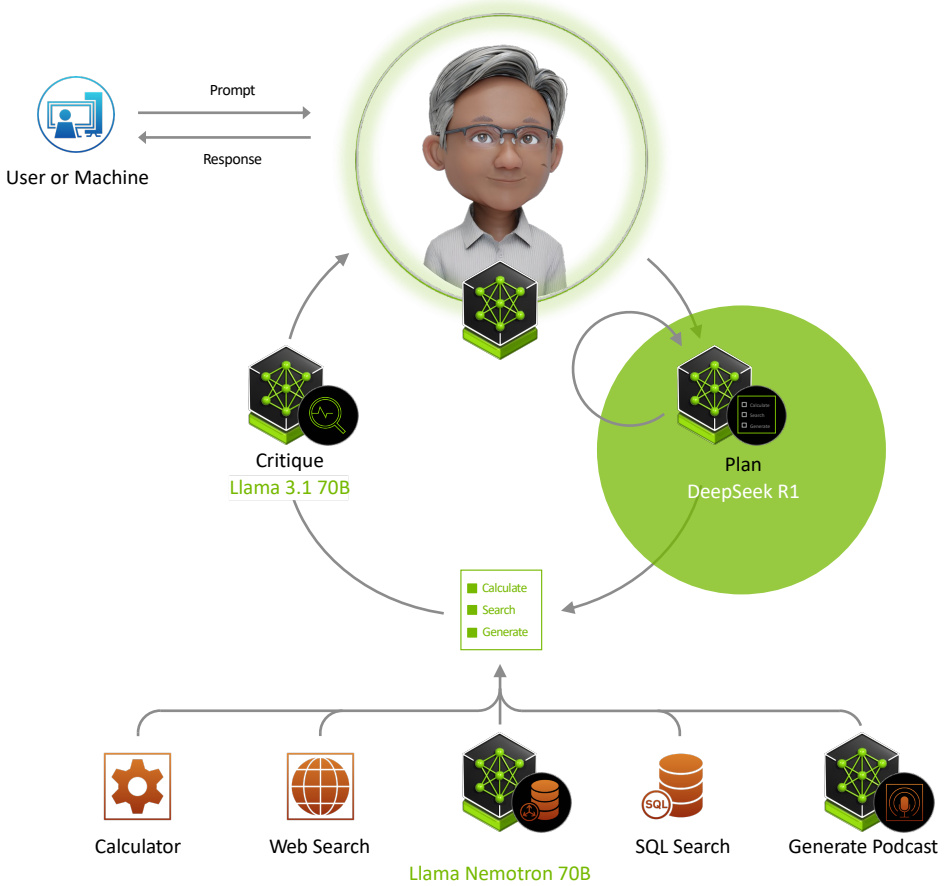
Performance Optimize accuracy, latency and throughput to meet requirements with lowest TCO

Microsoft Azure · aws · Google Cloud · ORACLE · DGX & DGX Cloud · cisco · DELL Technologies · Hewlett Packard Enterprise · Lenovo · SUPERMICRO

NVIDIA

AI Agents Require Multiple Models and Tools

Developers want to rapidly integrate the latest AI building blocks

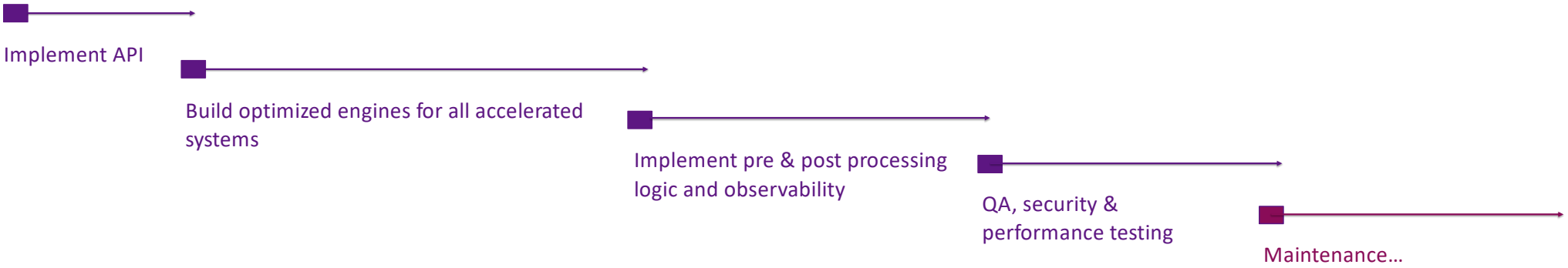



```
import openai
  client = openai.OpenAI (
    base_url = “MY_LOCAL_ENDPOINT_URL”
  )

critique_completion = client.chat.completions.create (
  model = “llama-3.1-70b”
  )
plan_completion = client.chat.completions.create (
  model = “deepseek-r1” // model = “llama-3.1-405b”
  )
```

Rapid Integration Requires Rapid Endpoint Deployment

NIM handles the heavy lifting

SELF-HOST DIY: 5+ Days



5 MINUTES TO NIM

DEPLOY

```
docker run /path/new-model-name
```

RUN

```
curl –X ‘POST’ \
‘local_host/v1/completions \

‘{

    model = “new-model-name”
}’
```

BUILD

```
completion =
client.chat.completions.create (

    model = “new-model-name”

    )
```

NVIDIA NIM for LLMs

LLM-specific and multi-LLM compatible container options

LLM-specific NIM

```
docker run [...]\
nvcr.io/nim/meta/llama-3-8b-instruct
```

Model config layer from NVIDIA NIM Factory (llama3-8b-instruct)

Base container image (NIM for LLMs)

multi-LLM compatible NIM

```
docker run [...]\
-e NIM_MODEL_NAME="hf://mistralai/Codestral-22B-v0.1"
```

Base container image (NIM for LLMs)

NVIDIA NIM for LLMs

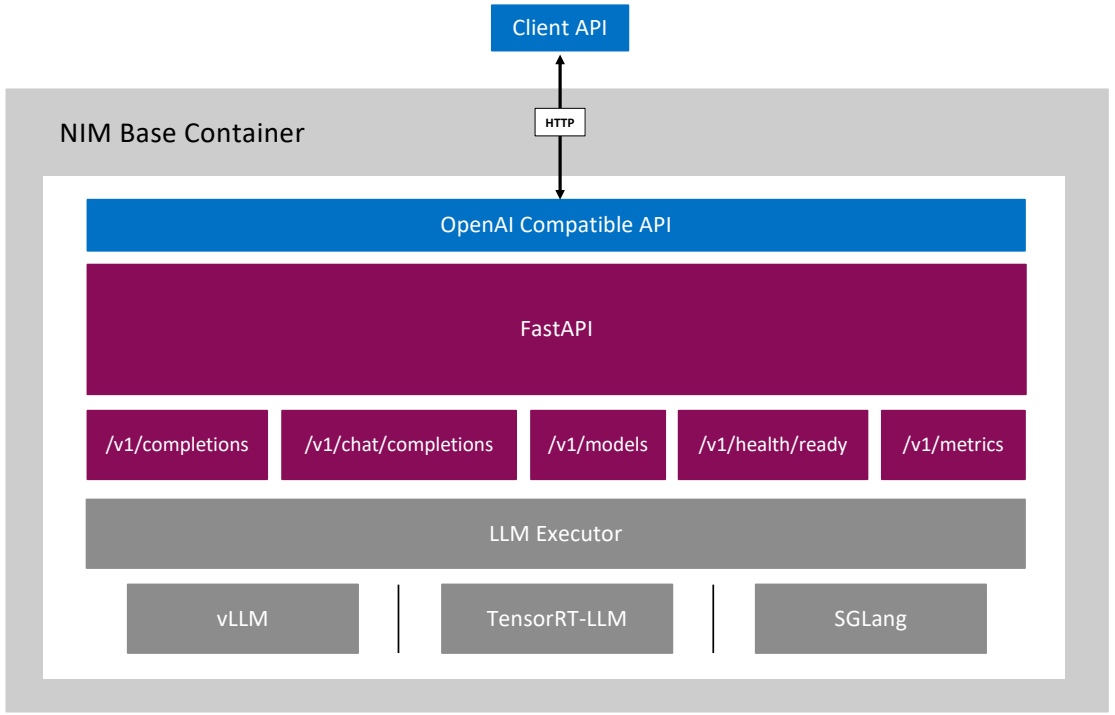
NIM features based on container option

LLM-specific NIM		multi-LLM compatible NIM
Max performance with pre-built, optimized engines for specific model-GPU combinations	Performance	Good baseline performance plus optimized engine local build for supported models
Limited to single model per container	Flexibility	Broad range of models, formats, and quantization types
Models and containers curated and security-scanned by NVIDIA	Security	User responsible for verifying model safety and integrity

NVIDIA NIM for LLM Architecture

HTTP REST API conforms to OpenAI specification for easy developer integration
- Liveness,
- health check
- and metrics endpoints for monitoring and enterprise management

NVIDIA NIM includes multiple LLM runtimes:
- TensorRT-LLM,
- vLLM
- and SGLang



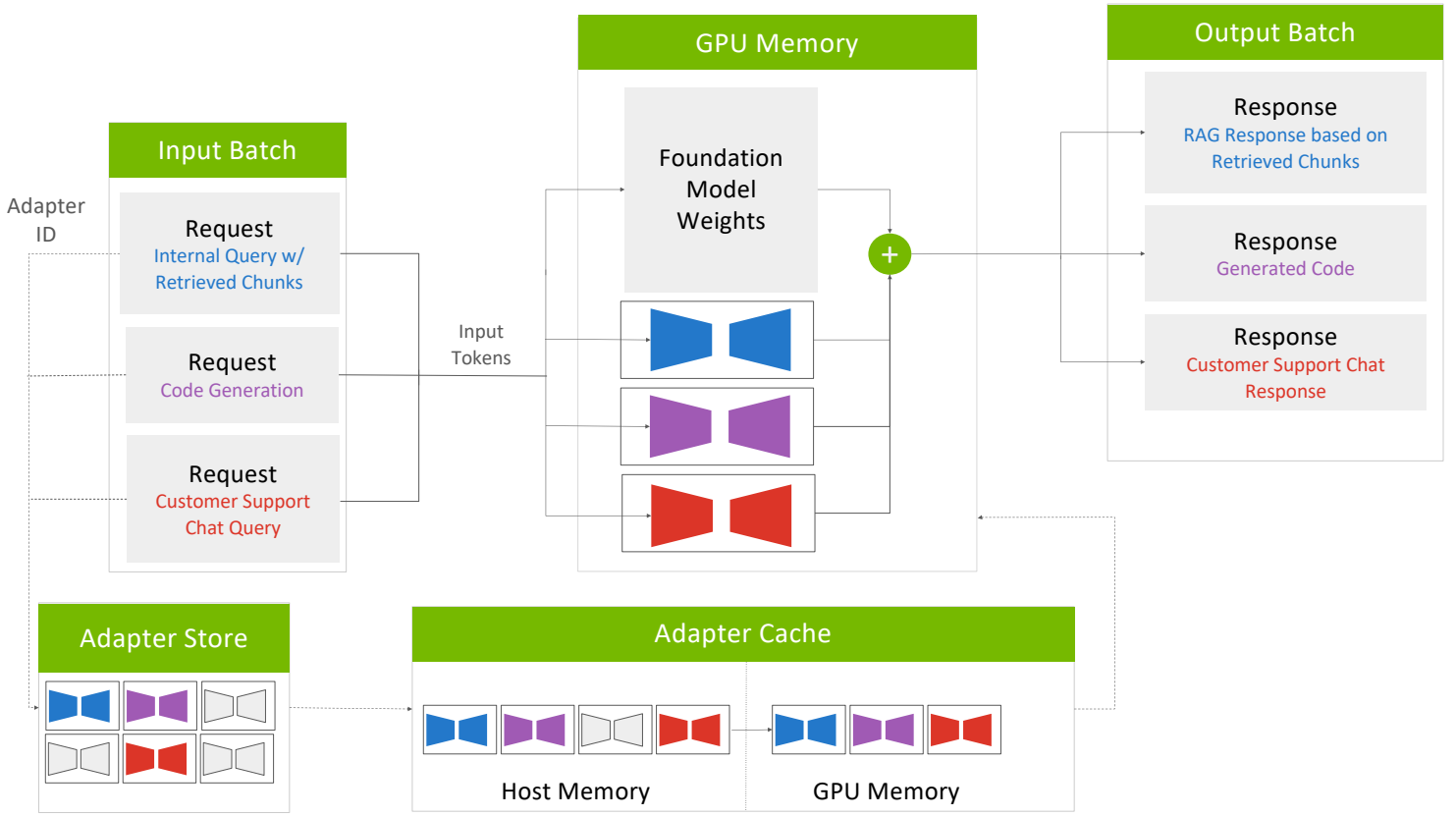
Best Accuracy For Enterprise

Seamlessly Deploy One Foundation Model PEFT Fine-Tuned for Multiple Tasks

Best accuracy with NIMs customized for domain-specific tasks using NeMo Framework or Hugging Face PEFT

Optimization with custom CUDA kernels for simultaneous multi-LoRA and base model inference, plus automatic multi-tier cache management

Seamless deployment with a single command for both base and LoRA models



Locally Built TensorRT-LLM Inference Engines

Seamlessly deploy SFT models and ensure TensorRT-LLM optimized performance on any NVIDIA GPU

Deploy NIM with custom model weights with a single command

NIM automatically builds and loads an optimized TensorRT-LLM inference engine and deploys the fine-tuned model for inference.

```
export CUSTOM_WEIGHTS=/path/to/customized/llama
docker run -it --rm --name=llama3-8b-instruct \
    -e NIM_FT_MODEL=$CUSTOM_WEIGHTS \
    -e NIM_SERVED_MODEL_NAME="llama3.1-8b-my-domain" \
    -e NIM_CUSTOM_MODEL_NAME=custom_1 \ # set this to cac
    -v $CUSTOM_WEIGHTS:$CUSTOM_WEIGHTS \
    -u $(id -u) \
    $NIM_IMAGE
```

Optionally, use `list-model-profiles` and `-e NIM_MODEL_PROFILE=<profile_name>` to list and specify alternative prebuilt and locally buildable, optimized engine profiles for single command deployment

Specify tensor parallelism for N GPUs, or latency versus throughput optimization preference (supported GPUs only).

Locally built optimized engines ensure TensorRT-LLM optimized performance on *any* NVIDIA GPU

If a prebuilt, TensorRT-LLM optimized inference engine is not available at NIM launch, NIM automatically builds one in the local environment, loads it and deploys the model through the same single-command deployment process.

```
docker run -it --rm --name=meta-llama3-8b-instruct \
    --runtime=nvidia \
    --gpus all \
    --shm-size=16GB \
    -e NGC_API_KEY \
    -v ~/.cache:/opt/nim/.cache \
    -u $(id -u) \
    -p 8000:8000 \
    nvcr.io/nvidian/nim-llm-dev/meta-llama3-8b-instruct
```

Same single-command deploy

Tool Use

Tool calling with Llama 3.1

Tool Definition

- OpenAI-compatible tool-calling API
- User passes one or more tools to the model
- Tools are defined using description, and their parameters
- Tool can be defined to always be called by the LLM

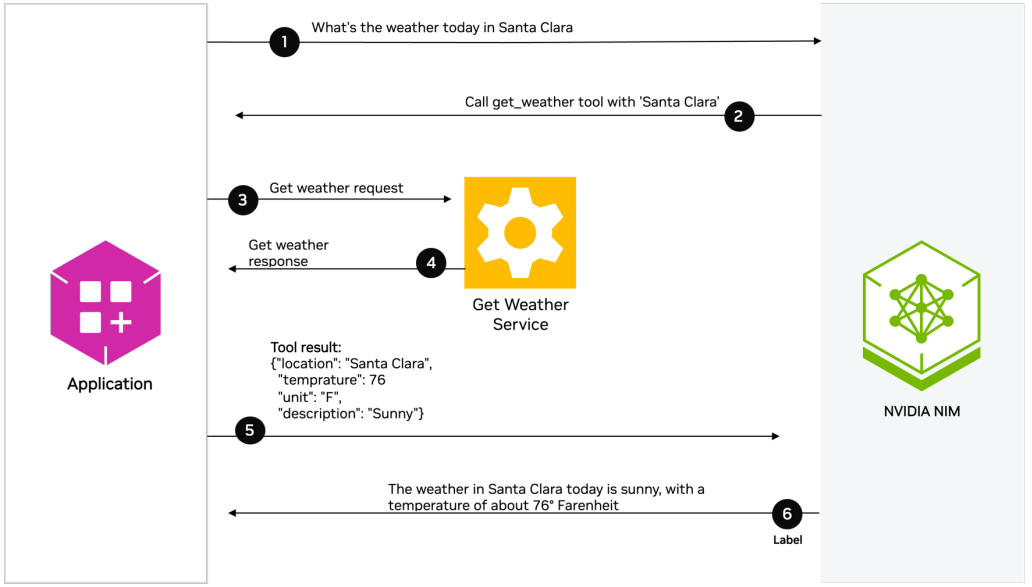
```
from openai import OpenAI

client = OpenAI(
  base_url = "https://integrate.api.nvidia.com/v1",
  api_key = "$API_KEY_REQUIRED_IF_EXECUTING_OUTSIDE_NGC"
)

completion = client.chat.completions.create(
  model="meta/llama-3.1-405b-instruct",
  messages=[{"role":"user","content":"Write a limerick about the wonders of GPU computing."}],
  temperature=0.2,
  top_p=0.7,
  max_tokens=1024,
  stream=True,
  tools=[{"type":"function","function":{"name":"get_current_weather","description":"Returns the
current weather at a location, if one is specified, and defaults to the user's location.","para
meters":{"type":"object","properties":{"location":{"type":"string","description":"The location
to find the weather of, or if not provided, it's the default location."},"format":{"type":"stri
ng","enum":["celsius","fahrenheit"],"description":"Whether to use SI or USCS units (celsius or
fahrenheit)."}},"required":[]}}}],
  tool_choice="auto"
)

for chunk in completion:
  if chunk.choices[0].delta.content is not None:
    print(chunk.choices[0].delta.content, end="")
```

NIM Tool Calling

Tool calling with Llama 3.1

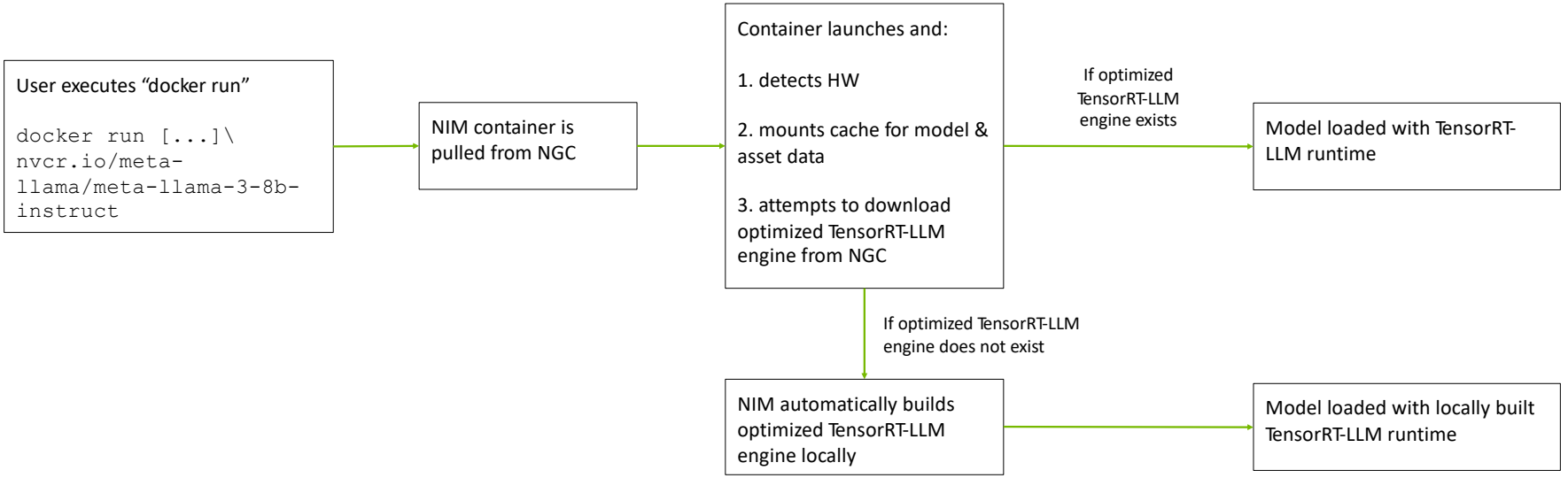
Tool Calling Execution Sequence

1. User prompts LLM
2. LLM decides to call `get_weather` tool and returns a structured output with the appropriate arguments
3. App call Get Weather API
4. Weather API returns structured response
5. App sends response to LLM
6. LLM returns answer in natural language



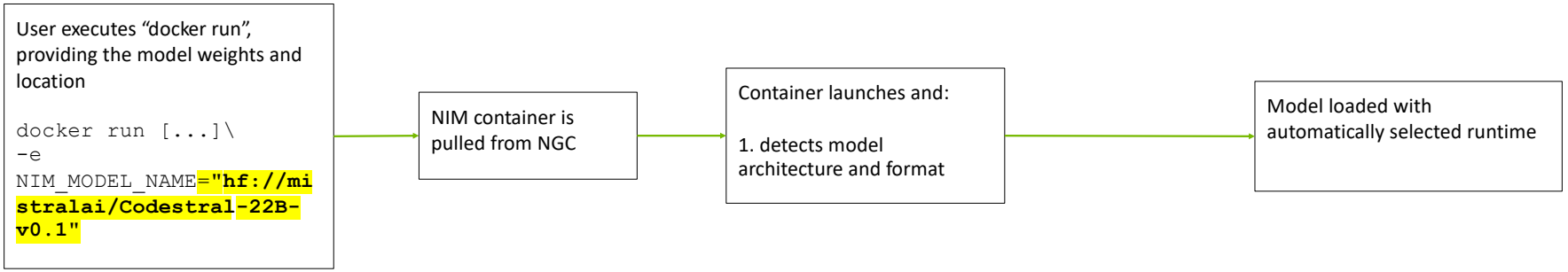
NVIDIA LLM NIM Pull Sequence

LLM-specific NIM



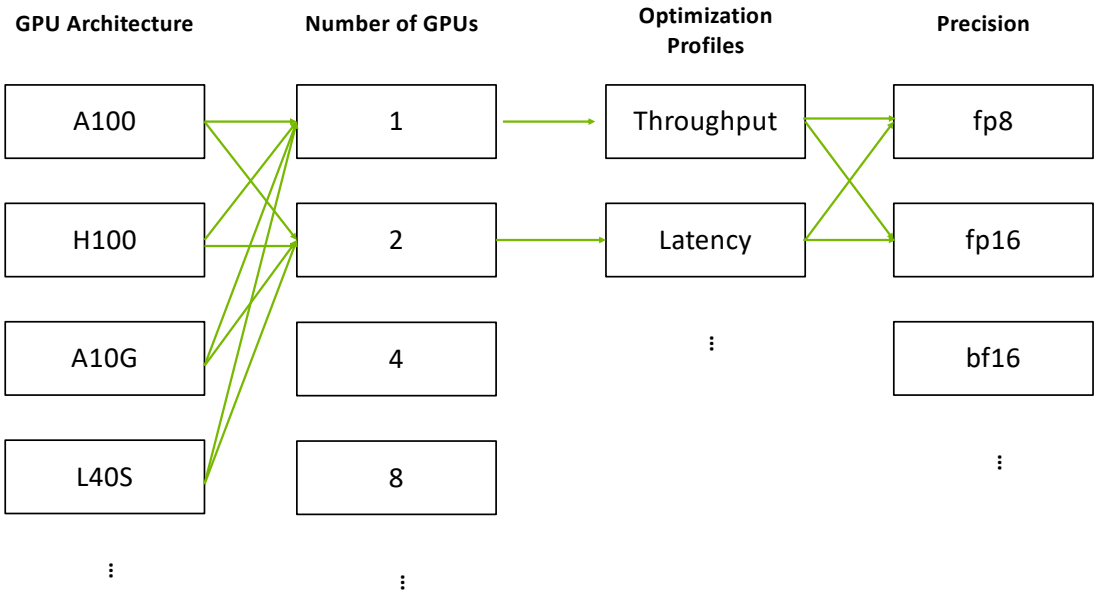
NVIDIA LLM NIM Pull Sequence

Multi-LLM compatible NIM



How NIMs Leverage Optimized Engines

Llama-3-8B-Instruct Optimization for NVIDIA GPUs with LoRA Support Option



Optimized Models

llama3-8b-instruct:0.10.0+cbc614f5-**h100**x1-fp8-**throughput**
llama3-8b-instruct:0.10.0+cbc614f5-**h100**x2-fp8-**latency**
llama3-8b-instruct:0.10.0+cbc614f5-**h100**x1-fp16-**throughput**
llama3-8b-instruct:0.10.0+cbc614f5-**h100**x1-fp16-**lora**

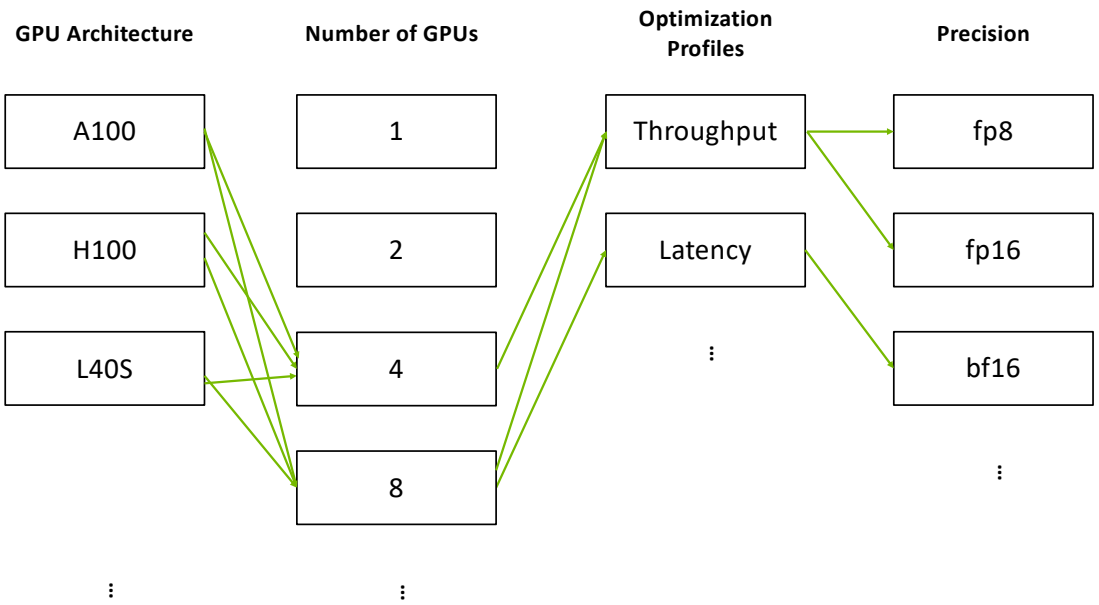
llama3-8b-instruct:0.10.0+cbc614f5-**a100**x1-fp16-**throughput**
llama3-8b-instruct:0.10.0+cbc614f5-**a100**x1-fp16-**lora**
llama3-8b-instruct:0.10.0+cbc614f5-**a100**x2-fp16-**latency**
llama3-8b-instruct:0.10.0+cbc614f5-**h100**x2-fp16-**latency**

llama3-8b-instruct:0.10.0+cbc614f5-**a10g**x2-fp16-**latency**
llama3-8b-instruct:0.10.0+cbc614f5-**a10g**x1-fp16-**throughput**
llama3-8b-instruct:0.10.0+cbc614f5-**a10g**x1-fp16-**lora**
llama3-8b-instruct:0.10.0+cbc614f5-**l40s**x2-fp8-**latency**
llama3-8b-instruct:0.10.0+cbc614f5-**l40s**x2-fp16-**latency**
llama3-8b-instruct:0.10.0+cbc614f5-**l40s**x1-fp8-**throughput**
llama3-8b-instruct:0.10.0+cbc614f5-**l40s**x1-fp16-**throughput**
llama3-8b-instruct:0.10.0+cbc614f5-**l40s**x1-fp16-**lora**

```
Detected 3 compatible profiles.
Valid profile: 17190a87573ad181e4a55b96d1a84b52f1c82c6542e2404473b25011ebfb403e (tensorrt_llm-A100-bf16-tp1-balanced)
Valid profile: 4ca08b7e7f1d59d5da5f761969320738d6922f25f26f334ab344c21f2e57348f (tensorrt_llm-A100-fp16-tp1-throughput)
Valid profile: 15fc17f889b55aedaccb1869bfeadd6cb540ab26a36c4a7056d0f7a983bb114f (vllm)
Selected profile: 17190a87573ad181e4a55b96d1a84b52f1c82c6542e2404473b25011ebfb403e (tensorrt_llm-A100-bf16-tp1-balanced)
Profile metadata: llm_engine: tensorrt-llm
```

How NIMs Leverage Optimized Engines

Llama-3-70B-Instruct Optimization for NVIDIA GPUs with LoRA Support Option



Optimized Models

lllama3-70b-instruct:0.10.0+cbc614f5-**a100**x8-bf16-**latency**
llama3-70b-instruct:0.10.0+cbc614f5-**a100**x4-fp16-**throughput**
llama3-70b-instruct:0.10.0+cbc614f5-**a100**x4-fp16-**lora**

llama3-70b-instruct:0.10.0+cbc614f5-**h100**x8-fp8-**latency**
llama3-70b-instruct:0.10.0+cbc614f5-**h100**x8-fp16-**latency**
llama3-70b-instruct:0.10.0+cbc614f5-**h100**x4-fp8-**throughput**
llama3-70b-instruct:0.10.0+cbc614f5-**h100**x4-fp16-**throughput**
llama3-70b-instruct:0.10.0+cbc614f5-**h100**x4-fp16-**lora**

llama3-70b-instruct:0.10.0+cbc614f5-**l40s**x8-fp16-**throughput**
llama3-70b-instruct:0.10.0+cbc614f5-**l40s**x8-fp8-**latency**
llama3-70b-instruct:0.10.0+cbc614f5-**l40s**x4-fp8-**throughput**
llama3-70b-instruct:0.10.0+cbc614f5-**l40s**x8-fp16-**lora**