NVIDIA Inference Stack Demystified

Amit Kumar | Manager | Industries | Solutions Architecture & Engineering | Generative AI | NVIDIA

https://www.linkedin.com/in/amit-kumar-b4b43ab/

Enterprise Generative AI Journey

From insights to outcomes

Perceptive AI

Detecting Patterns

Generative AI

Generating Answers

Agentic AI

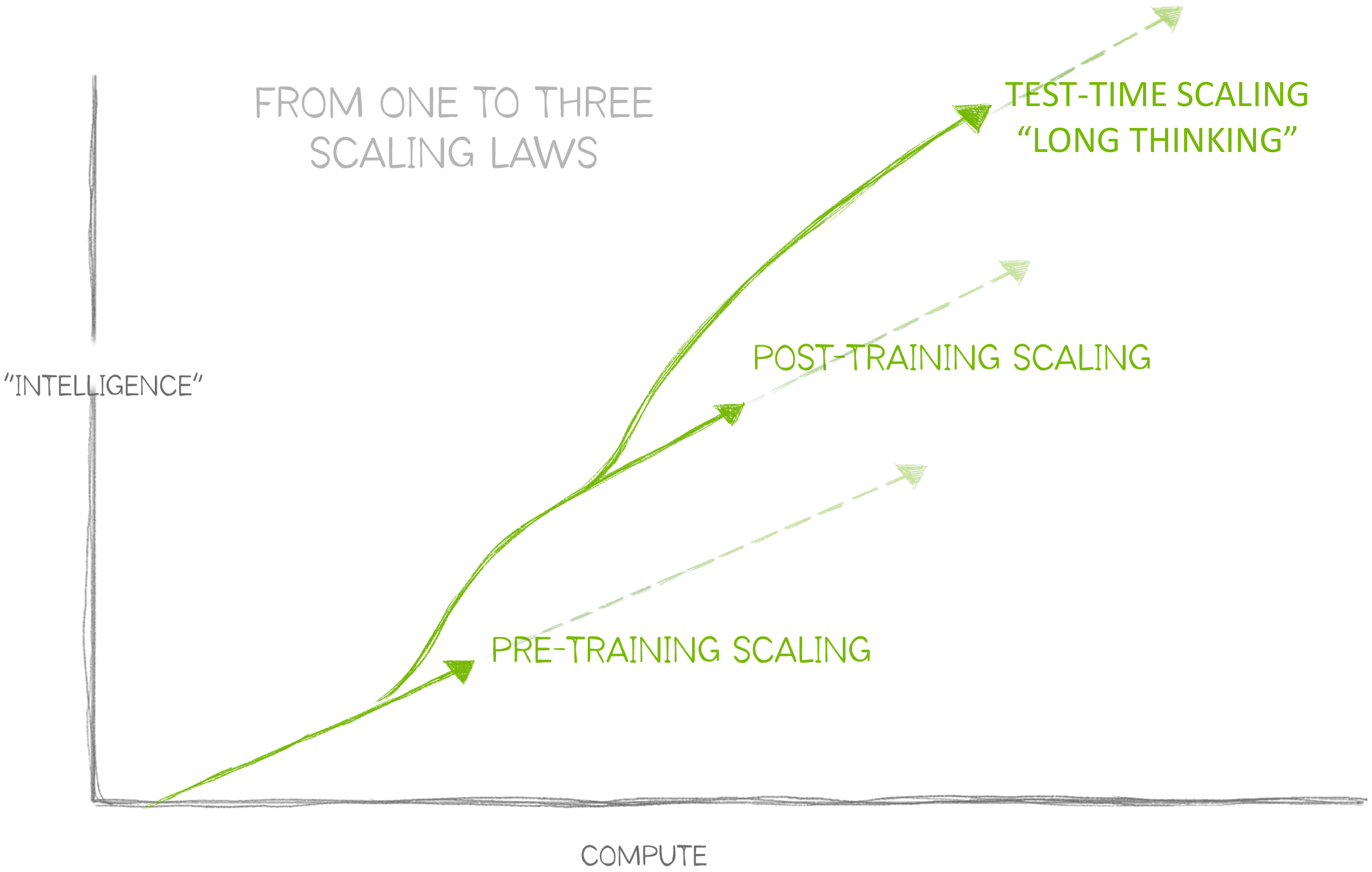
Generating Outcomes

Single application use cases

AI Factories
creating agents at massive scale

AI Scaling Laws Drive Exponential Demand for Compute

New "long thinking" supercharges inference scaling



Inference Compute Requirements Scaling Exponentially

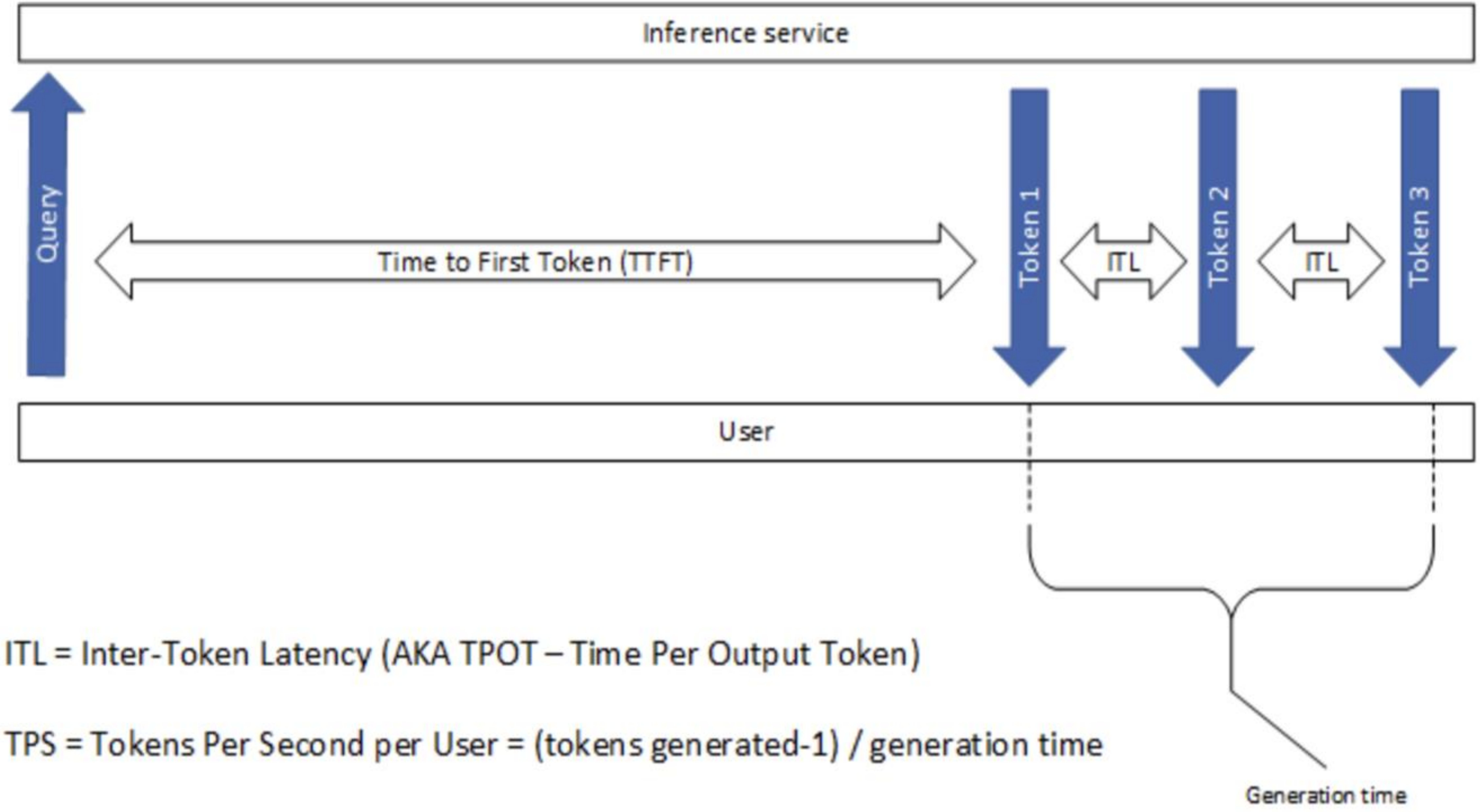
Fueled by reasoning models and AI agents

Hundreds of billions of parameters

100x more thinking tokens

Millions of input tokens

Metrics Definition

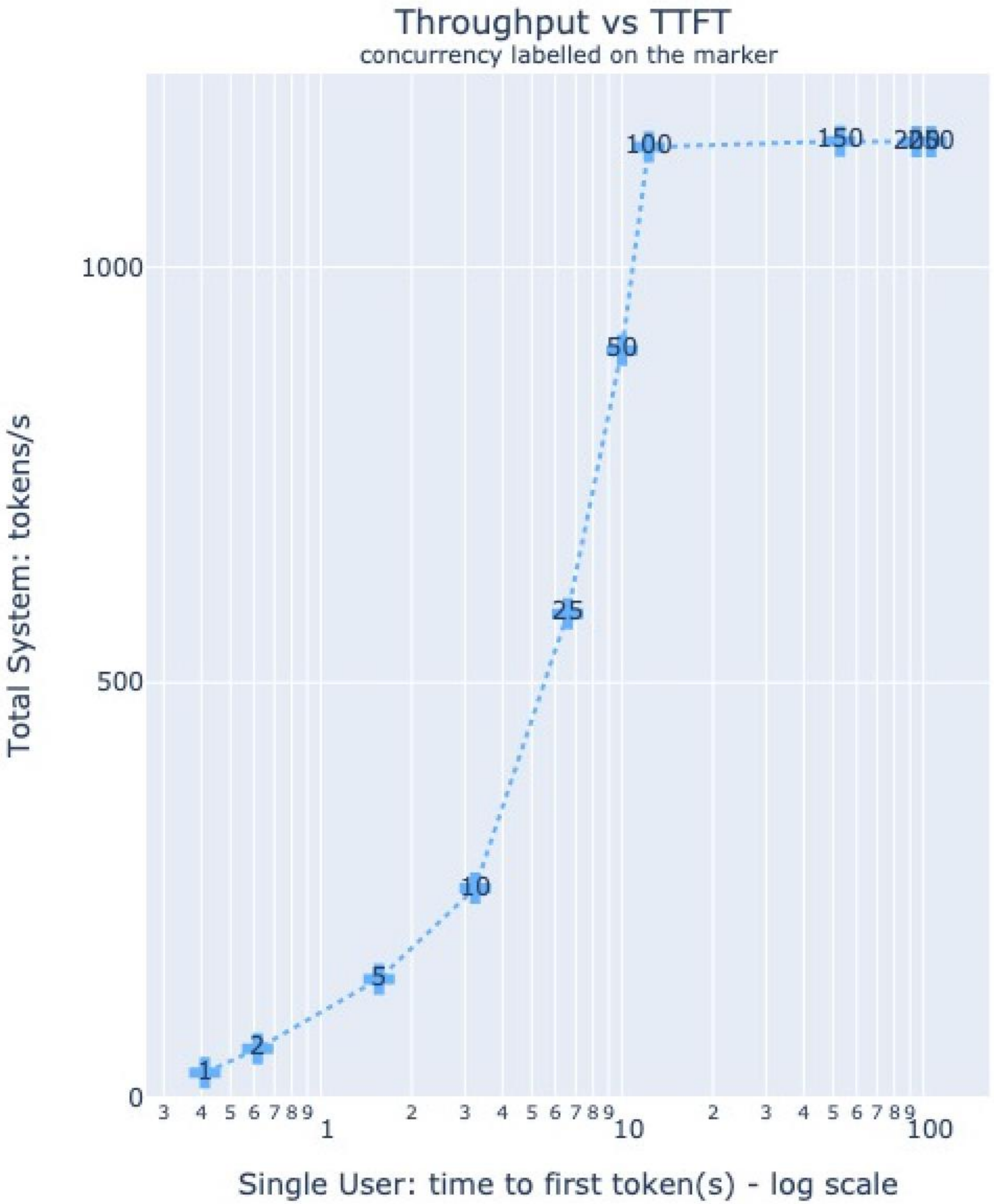


ITL = Inter-Token Latency (AKA TPOT – Time Per Output Token)

TPS = Tokens Per Second per User = (tokens generated-1) / generation time

Request latency (e2e) = TTFT + ITL*(tokens generated-1)

A word about Latency and Throughput Trade-off across concurrecies



X-axis: Latency (Time to first token)

y-axis: Throughput (total system tokens/s)

Values on the marker: concurrency values

Uses for the chart

- Find Maximum Throughput Under a Latency Budget
- Find the Throughput and Latency under a known concurrency value
- Find the throughput saturation threshold beyond which only TTFT increases when concurrencies increase.
- Similar chart for Throughput vs ITL

500/2000

Meta-Llama3.1-70B-Instruct Inference 500/2000 on H100 TP4

Throughput vs TTFT
concurrency labelled on the marker

Throughput vs E2E latency
concurrency labelled on the marker

Total System: tokens/s
0
2000
4000
6000
Single User: time to first token(s) - log scale
5
0.1
2
5
1
2
5
10
2
5
1
5
25
50
100
150
200
250
NIM ON FP8
NIM ON BF16
NIM OFF FP8
NIM OFF FP16
Single User: E2E latency(s) - log scale
30
40
50
60
70
80
90
100
110
120
130
140
150
Meta-Lama3.1-70B-Instruct Inference H100 500_2000 tp4 Precision

NVIDIA TensorRT Model Optimizer

Product Overview

NeMo

HF

Megatron-L
M

Optimize model *with*
in-framework or
post-training
methods

Inputs FWs:
PyTorch
ONNX
…

Model Optimizer

APIs

Recipes & Algorithms

Training/Fine-tuning

Distillation

QAT

Sparsity

Prune

Speculation

Post-Training

PTQ

Model Optimization Infrastructures

Exporters

Deploy optimized model
with FW of
choose

TensorRT-LLM
/TRT

SGLang

vLLM

Pre-Quantized
HF Checkpoints

🟩 Nvidia AI SW Stack

Links:
[TensorRT-Model-Optimizer]
[HF Checkpoints]

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

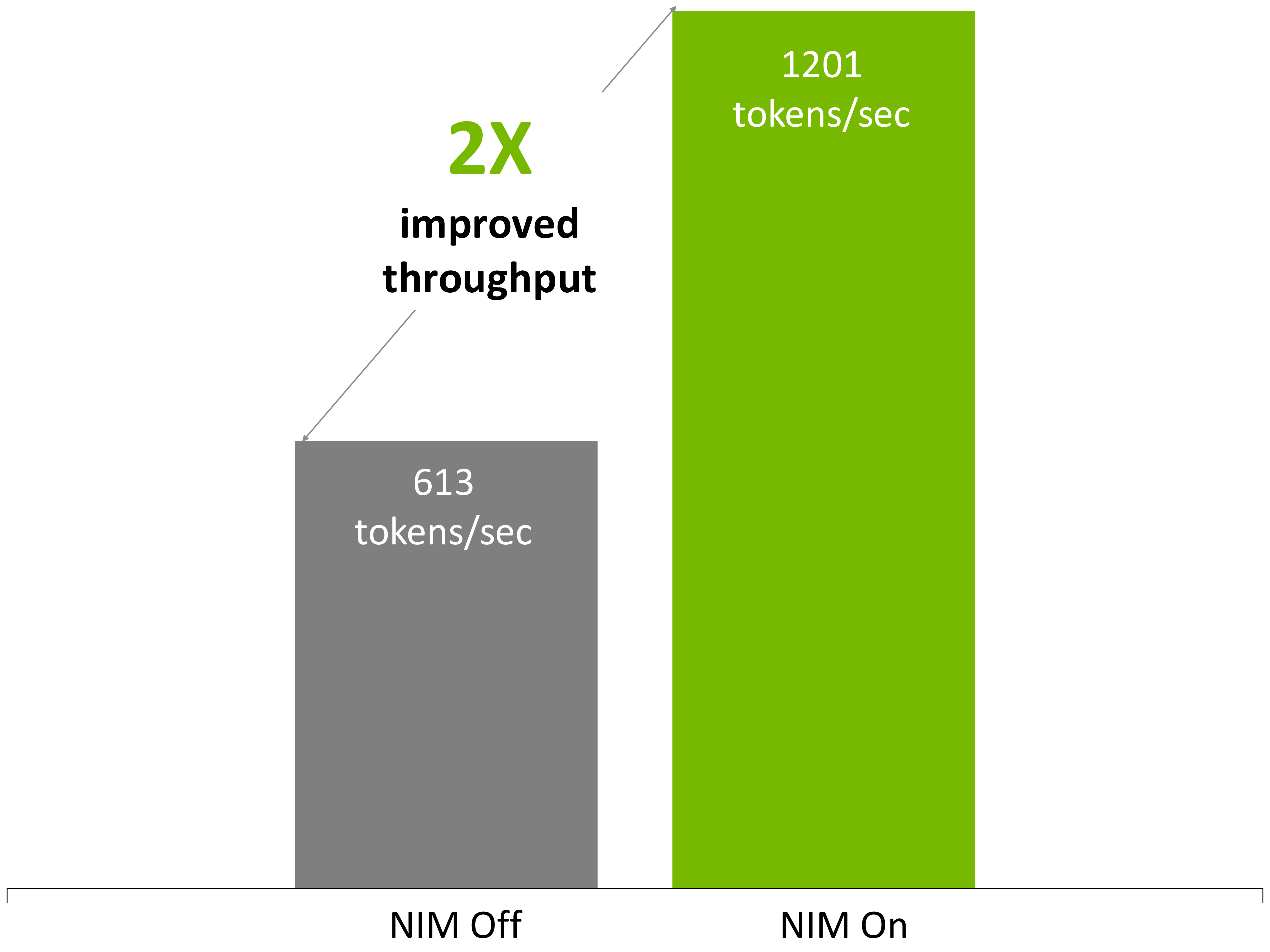
Enterprise Supported Gain confidence with stable APIs, quality assurance, continuous updates, security patching, and support

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

Optimized Efficiency Out of the Box

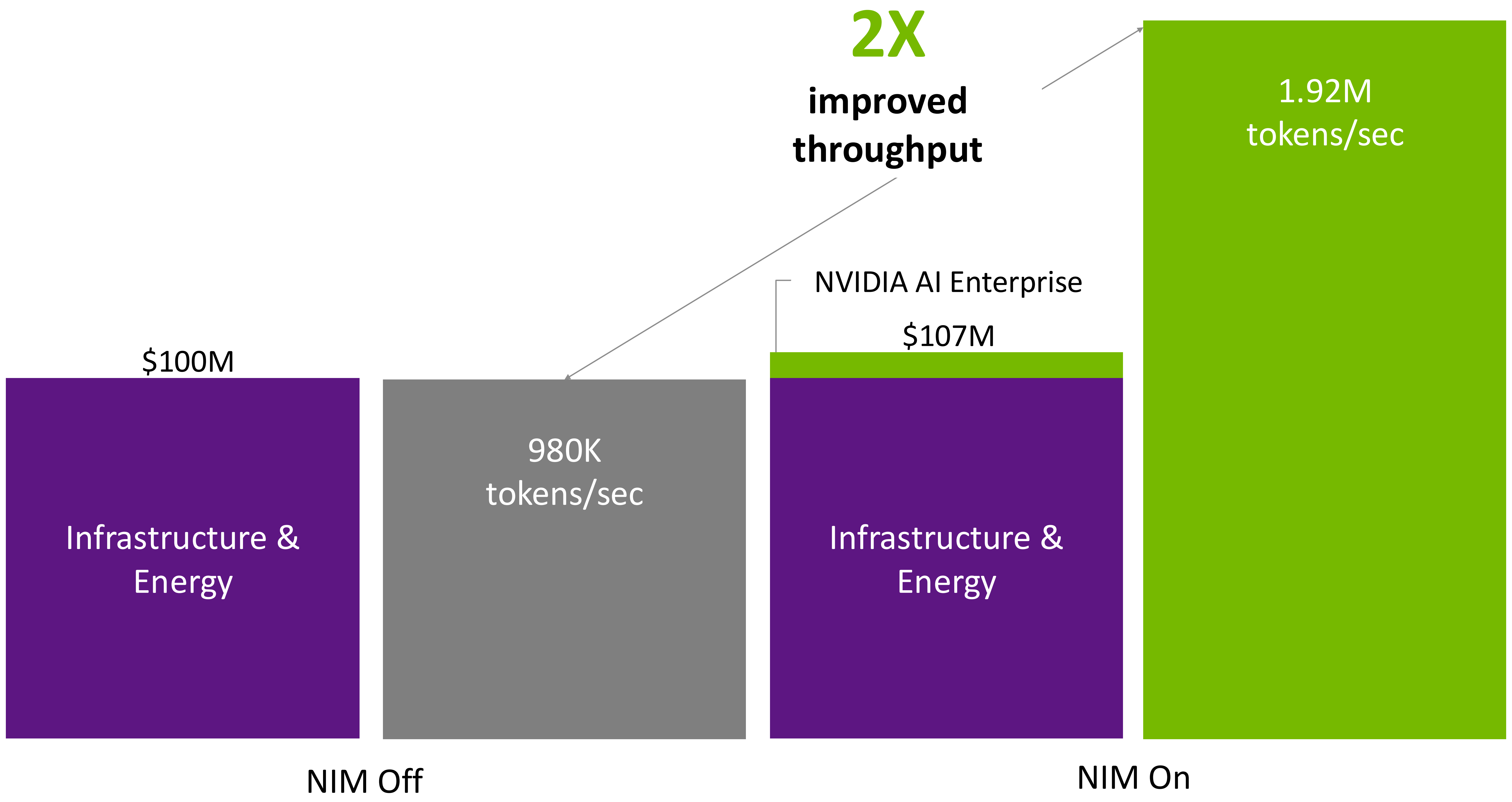
Improved performance on the same infrastructure with every release



Llama 3.1 8B, 1 x H100SXM, concurrent requests: 200. NIM ON: NIM v1.3, FP8. throughput 1,201 tokens/s, ITL 32ms. NIM OFF: FP8. throughput 613 tokens/s, ITL 37ms.

NIM Accelerates the AI Factory

Maximize token generation, or minimize total cost of ownership



NVIDIA AI Enterprise adds only 7% cost to the AI factory, while drastically increasing throughput

Also includes:
Support SLA
Security patching
Lifecycle management

Infrastructure includes hardware systems with GPU, CPU, Networking, Storage, as well as infrastructure operations. Energy includes 4-year operating costs.
Tokens/second based on Llama 3.1-8B-Instruct running on 1x H100 SXM compared to leading open-source inference solutions

NIM Deployment

Easily manage cost-efficient autoscaling across NVIDIA accelerated cloud infrastructure

Microsoft Azure

NIM on baremetal, VMI, AKS, Azure ML and Azure AI Studio

NIM Deploy - Azure

aws

NIM on baremetal, AMI, EKS and Amazon SageMaker

NIM Deploy - AWS

Google Cloud

NIM on baremetal, VMI, GKE and Google Cloud Vertex AI

NIM Deploy – Google Cloud

ORACLE

NIM on baremetal, CMI, OKE, and OCI Data Science Service

How-to Blog

Hewlett Packard Enterprise

Red Hat OpenShift

KServe

Canonical

NUTANIX

NIM on KServe Inference Platform on Kubernetes

NIM Deploy - KServe

HELM

Simplified NIM Deployment on Kubernetes

NIM Deploy - Helm

NVIDIA

NIM Operator

NIM Deployment Lifecycle Management on Kubernetes

NIM Operator

NVIDIA Agent Intelligence (AI-Q) Toolkit

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

Agent Interconnect

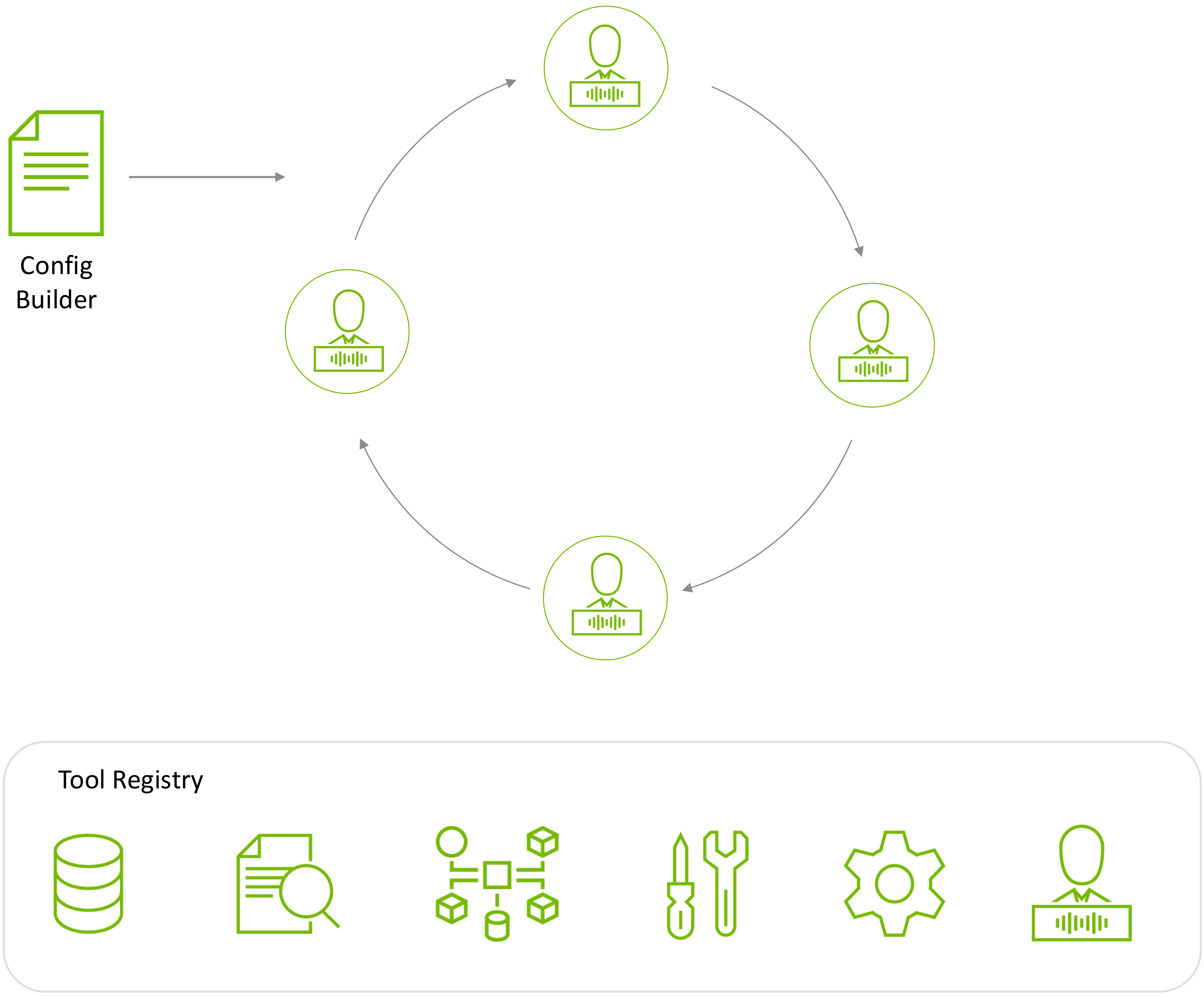
- Universal descriptors for agents, tools, and workflows across frameworks
- Reusable Agent/Tool registry
- Workflow Configuration/Builder

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



Enabling Software Development Lifecycle for AI Agents

NVIDIA AI-Q Toolkit

Plan
Analyze
Design
Test
Integrate
Deploy
Maintain

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.

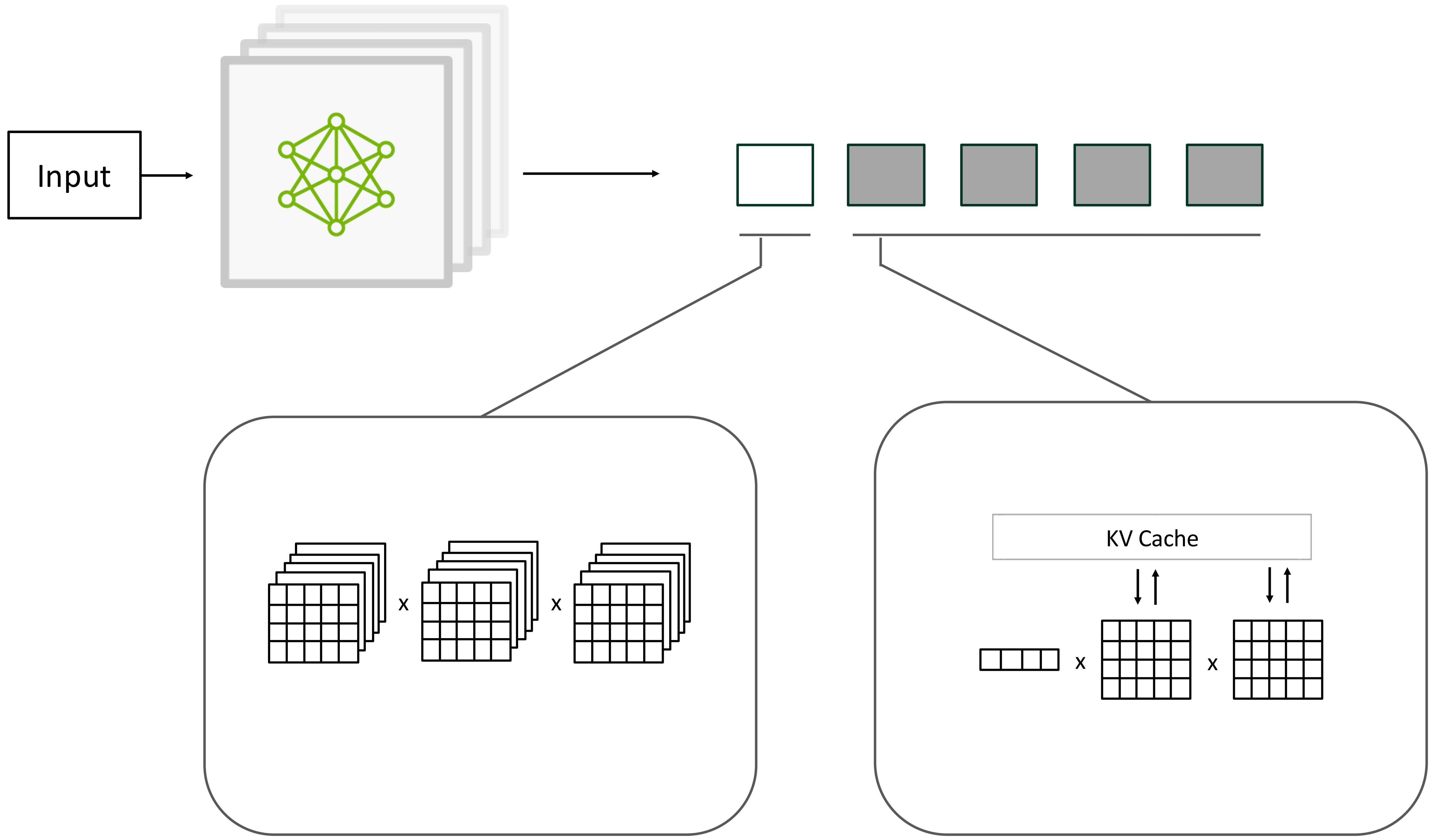
IMPROVE BUSINESS OUTCOMES

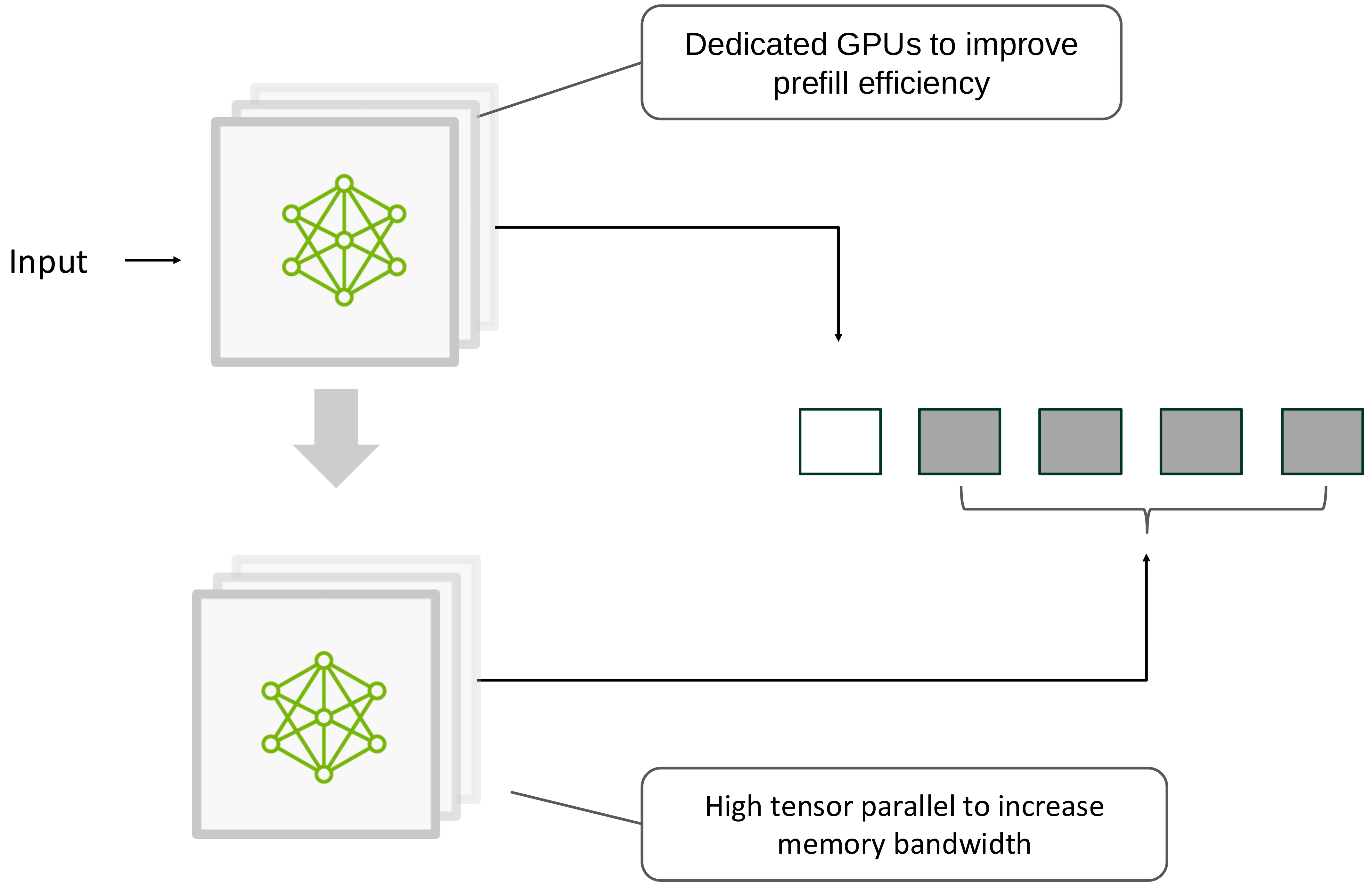
Increase agentic system accuracy

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

New Inference Optimization Techniques to Boost Inference

Disaggregated serving separates prefill and decode allowing each to be optimized independently





Dedicated GPUs to improve prefill efficiency

High tensor parallel to increase memory bandwidth

More flexibility to optimize cost and user experience

Announcing NVIDIA Dynamo

AI Inference Software for Reasoning Inference at Scale

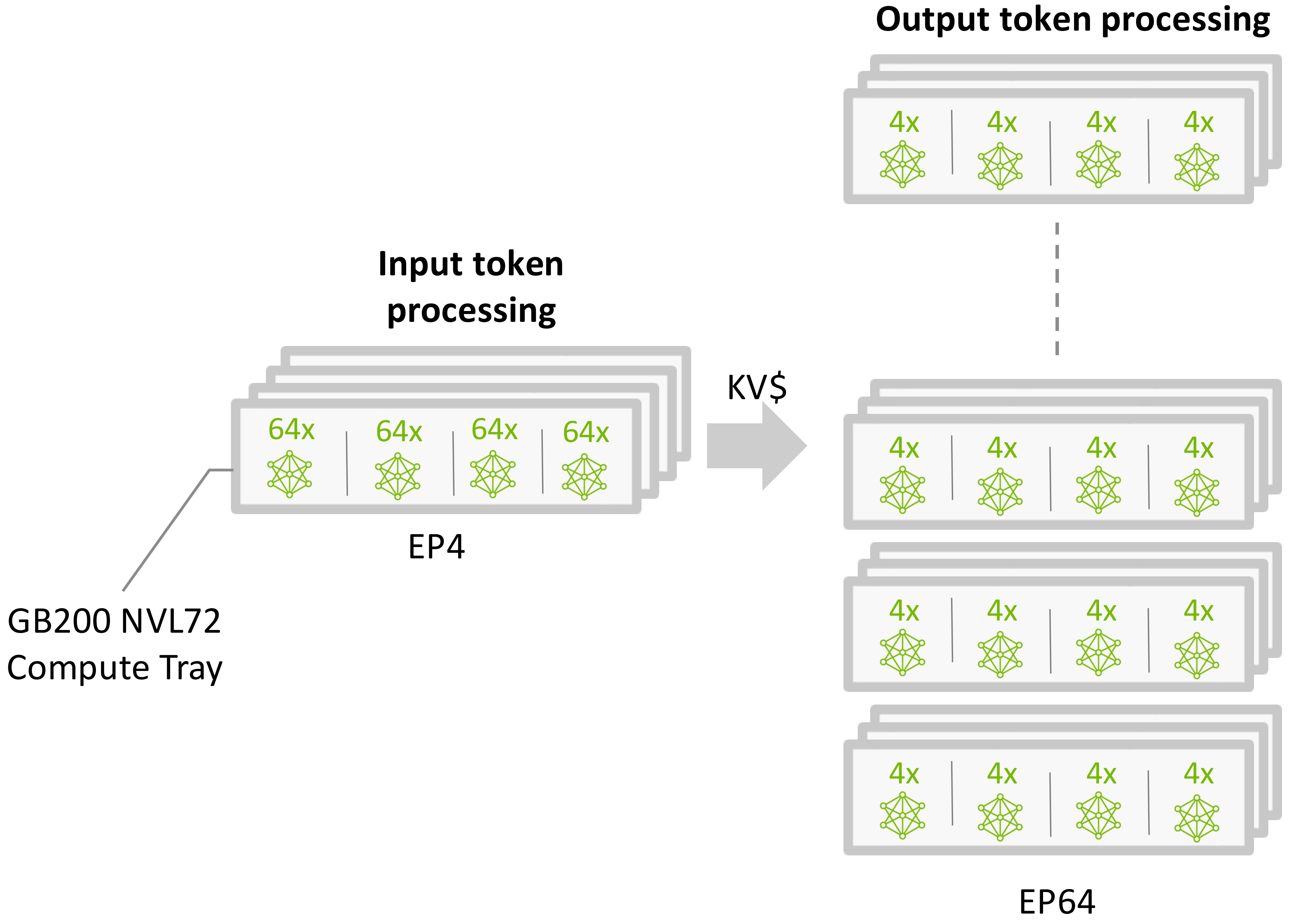
30X
AI Factory Throughput
& Revenue
DeepSeek models
on Blackwell

1000+
GPU Scale for
a single query

2X
Throughput & Revenue
Llama Models
On Hopper

Distributed and Disaggregated Serving

DeepSeek R1



Output token processing

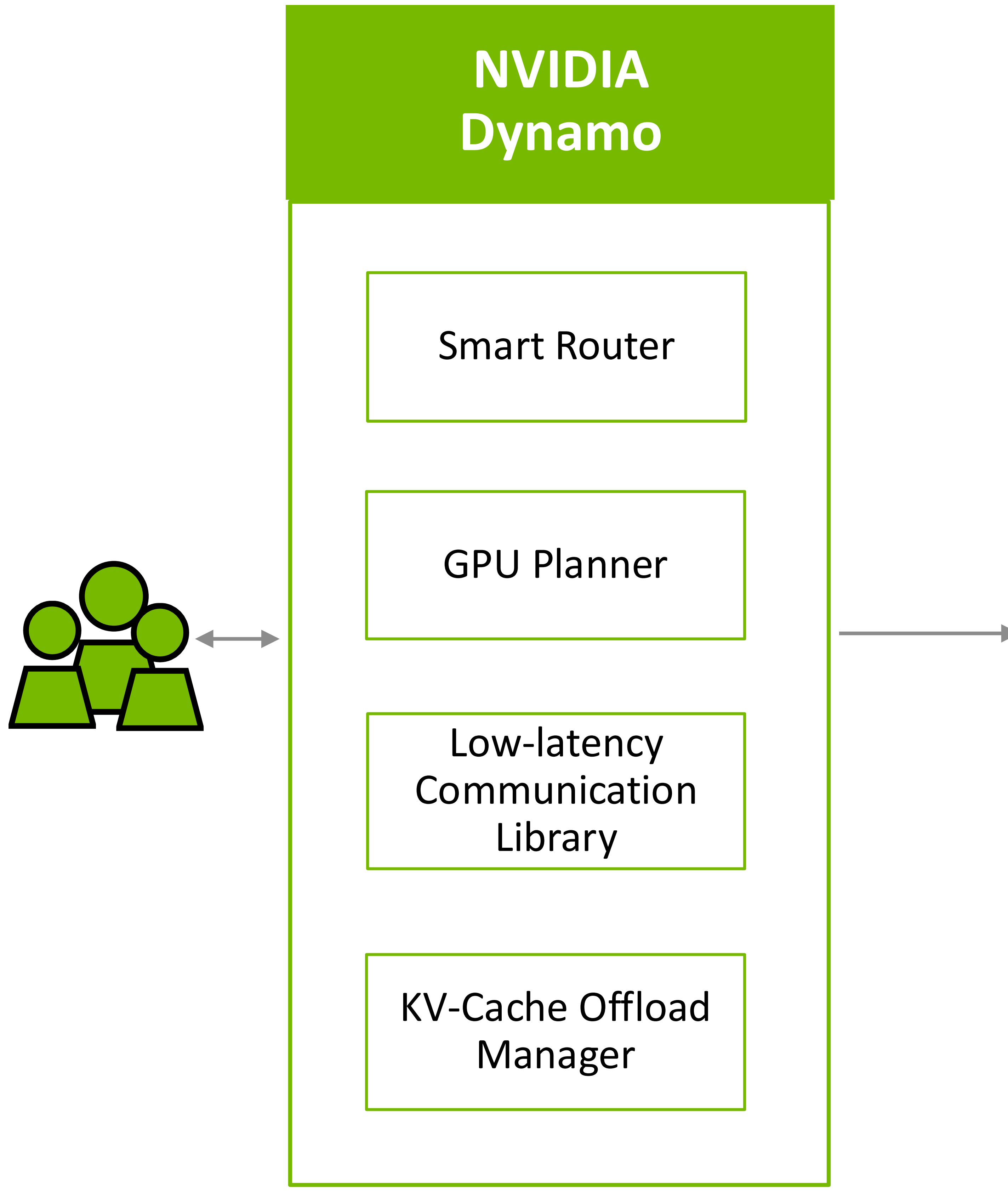
Input token processing

KV$

EP4

EP64

GB200 NVL72
Compute Tray



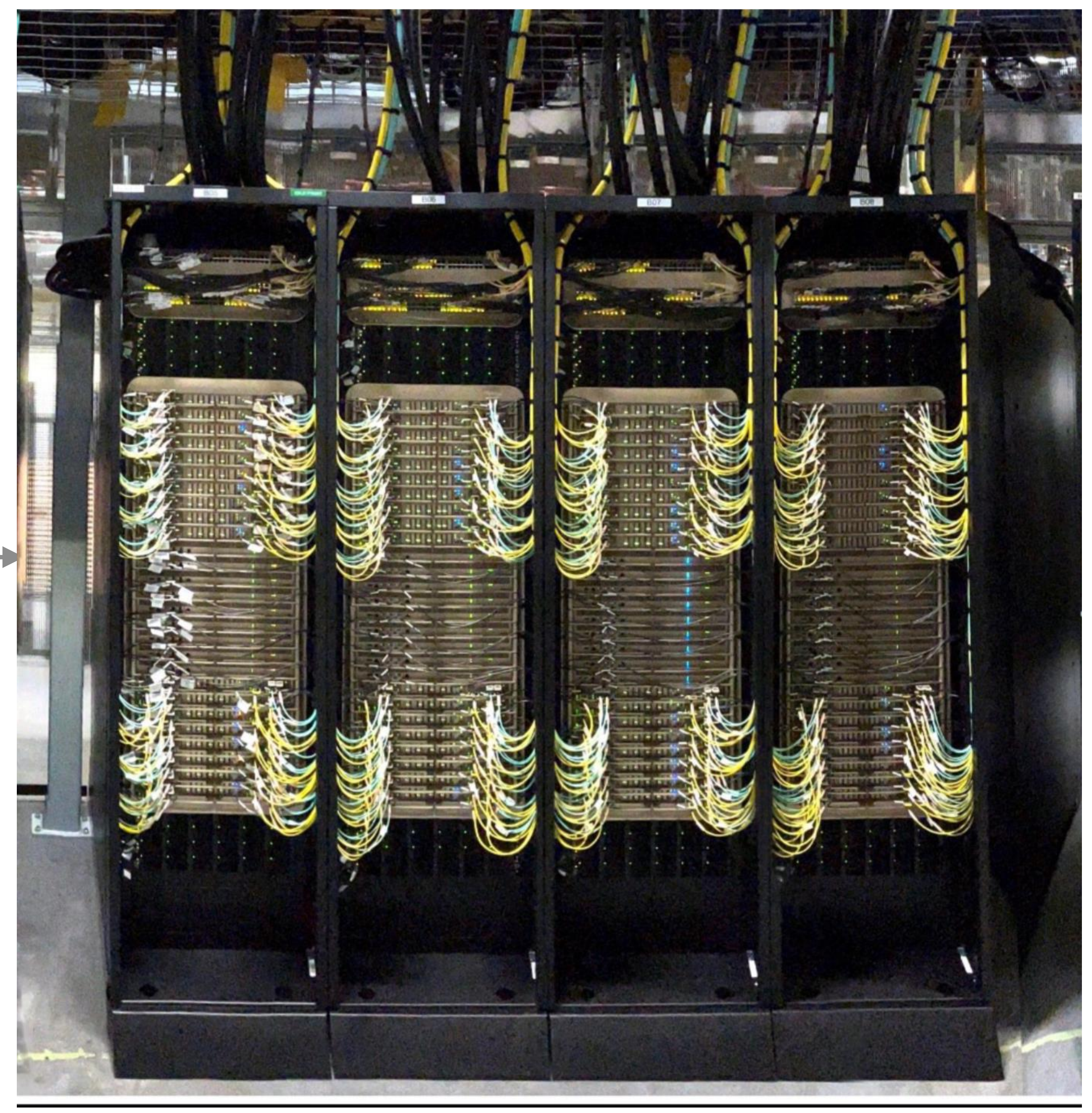
NVIDIA Dynamo

Smart Router

GPU Planner

Low-latency Communication Library

KV-Cache Offload Manager



NVIDIA Dynamo Use Cases

Unlock the full protentional of reasoning models and AI agents

Serving Reasoning Models

Hi,

I'm an AI-powered virtual agent and can answer questions, fulfill request, and generally point you in the right direction. I'm still learning, so I might make mistakes sometimes. Check my answers for accuracy.

How can I help you today?

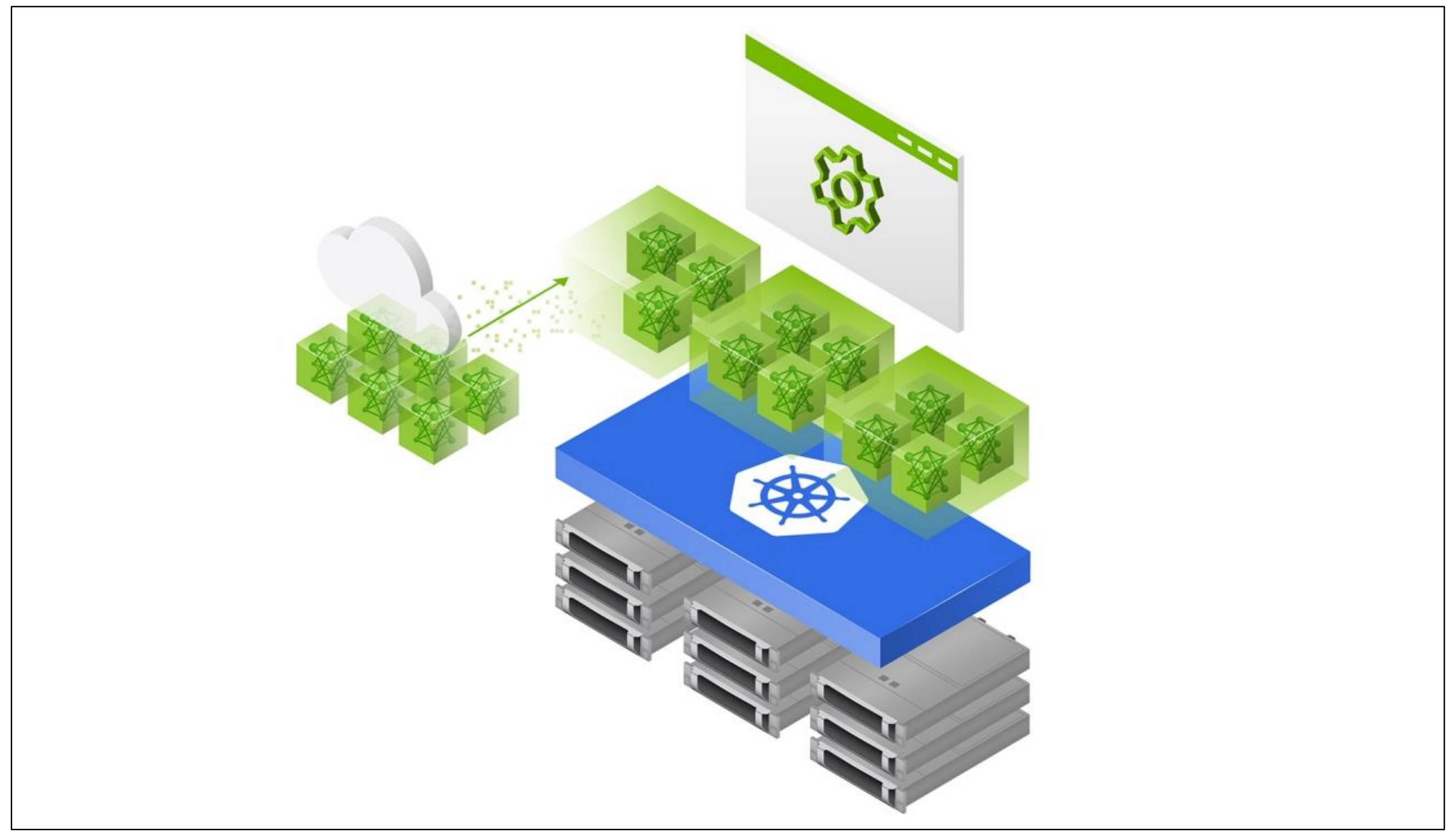
just now

What kinds of phone are included in our plan?

```
from simulator2D import Simulator2D
from visualizer2D import Visualizer2
from analytics2D import Analytics2D
from scene2D import *
from utils import state_from_files
from time import sleep
import datetime
#Arge parse
#Get xml and yaml file

xml_file = "warehouse_demo.drawio"
yaml_file = "warehouse_demo.yaml"
timesteps = int(0.5 * 60 * 60) #seco

starting_state = state_from_files(xm
```



Code Generation

NVIDIA Dynamo Breakthrough Features

A modular generative AI inference server designed for distributed and disaggregated serving

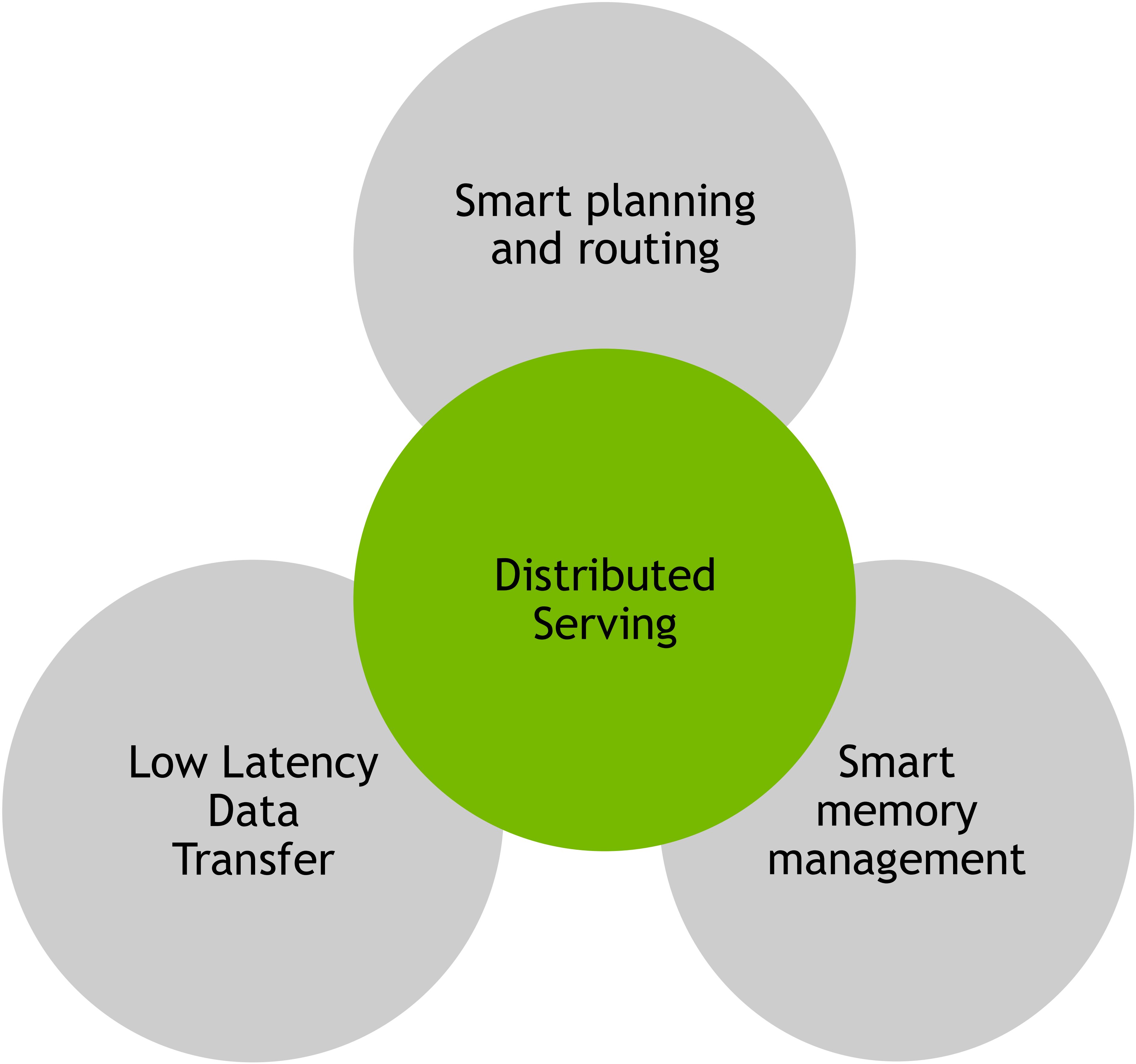
NVIDIA Dynamo

Distributed Inference Serving	Seamlessly scale LLMs from a single GPU to thousands of GPUs
GPU Planning & Scheduling	Meet changing demand patterns w/o over or under provisioning of resources
Smart Request Router	Free up GPU resources by reducing re-computations for similar requests
Low-latency Inference Data Transfer Library	Accelerate GPU-to-GPU communication to enhance user experience
KV Cache Manager	Preserve GPU memory by offloading context (KV$) to cheaper storage

NIM with NVIDIA Dynamo

Easy button for high performance inference at data center scale

- **Turbocharge agentic AI outcomes** with 30X AI reasoning throughput on NVIDIA accelerated datacenter GPUs
- **Unlock game-changing use cases** with distributed scale-out of complex requests across 1000+ GPUs
- **Maximize token revenue generation** for AI factories with distributed serving without bottlenecks
- **Push button deploy** in 3 simple steps in 5 minutes or less without coding



Accelerating AI Ecosystem

Fully open source and supports all major AI frameworks

aws
Azure
cohere
DELL
Google Cloud
Meta
NetApp™
OpenAI
perplexity
PyTorch
SGL
together.ai
VAST
vLLM

Learn with NVIDIA (DLI)

Register free for NVIDIA GTC