



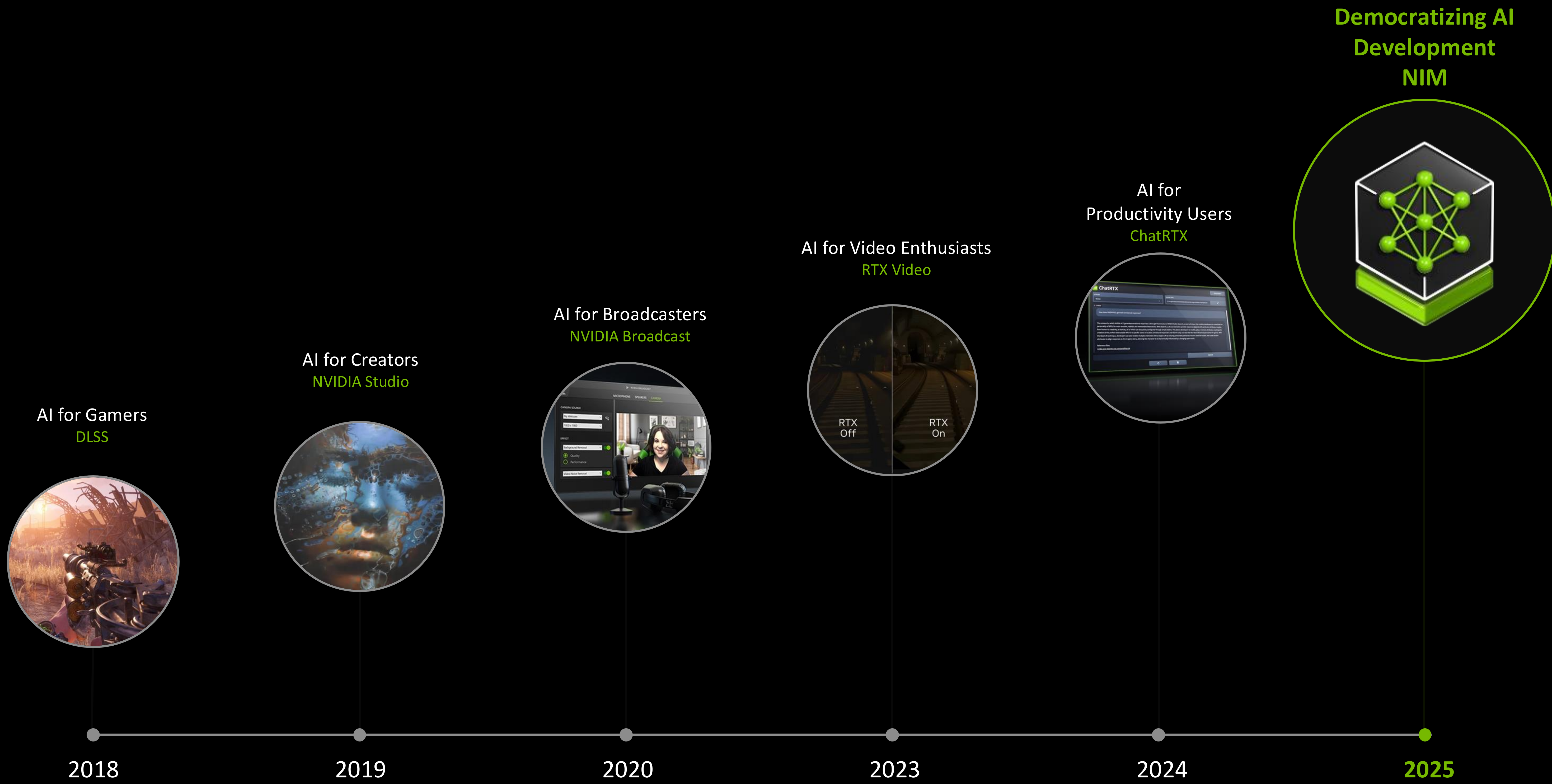
# RTX AI PC

CES Editor's Day 2025 Session 3

Jesse Clayton | Director of Product Management  
and Product Marketing, Windows AI at NVIDIA

# GeForce RTX: 7 Years of AI PC Innovation

100M RTX AI PCs | 600 RTX AI games and apps

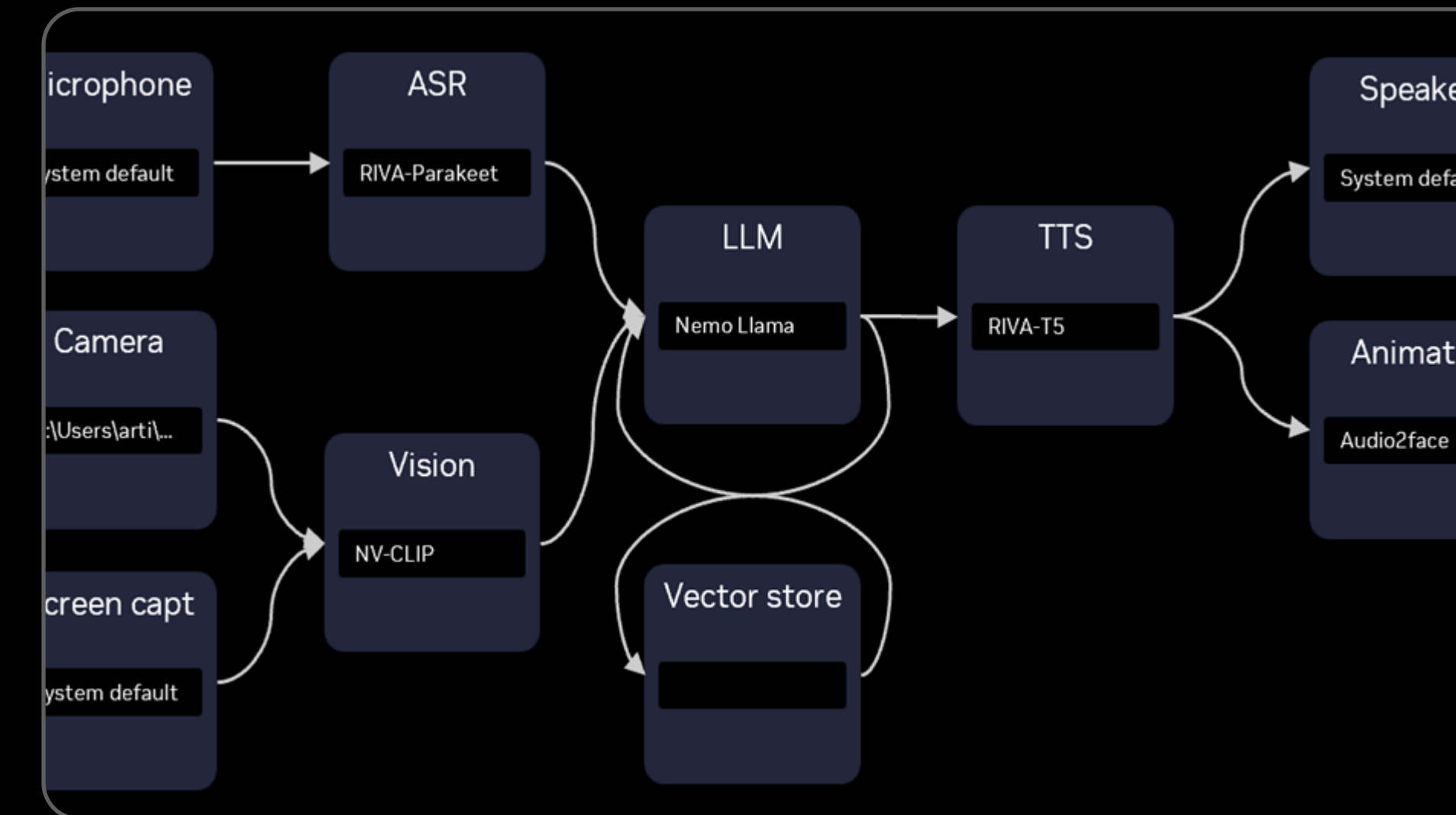


# The New Software Development Paradigm

With advances in generative AI and new tools, everyone is a developer

```
107. workOffset[0] = 0;
108. for(unsigned int i=0; i < ciDeviceCount; ++i)
109. {
110.     // Input buffer
111.     workSize[i] = (i != (ciDeviceCount - 1)) ? sizePerGPU : (uiWA - workOffset[i]);
112.     d_A[i] = clCreateBuffer(cxGPUContext, CL_MEM_READ_ONLY, workSize[i] * sizeof(float) * uiWA, NULL, NULL);
113.
114.     // Copy only assigned rows from host to device
115.     clEnqueueCopyBuffer(commandQueue[i], h_A, d_A[i], workOffset[i] * sizeof(float) * uiWA,
116.                        0, workSize[i] * sizeof(float) * uiWA, 0, NULL, NULL);
117.
118.     // create OpenCL buffer on device that will be initialize from the host memory on first use
119.     // on device
120.     d_B[i] = clCreateBuffer(cxGPUContext, CL_MEM_READ_ONLY | CL_MEM_COPY_HOST_PTR,
121.                            mem_size_B, h_B_data, NULL);
122.
123.     // Output buffer
124.     d_C[i] = clCreateBuffer(cxGPUContext, CL_MEM_WRITE_ONLY, workSize[i] * uiWC * sizeof(float), NULL, NULL);
125.
126.     // set the args values
127.     clSetKernelArg(multiplicationKernel[i], 0, sizeof(cl_mem), (void *) &d_C[i]);
128.     clSetKernelArg(multiplicationKernel[i], 1, sizeof(cl_mem), (void *) &d_A[i]);
129.     clSetKernelArg(multiplicationKernel[i], 2, sizeof(cl_mem), (void *) &d_B[i]);
130.     clSetKernelArg(multiplicationKernel[i], 3, sizeof(float) * BLOCK_SIZE * BLOCK_SIZE, 0);
131.     clSetKernelArg(multiplicationKernel[i], 4, sizeof(float) * BLOCK_SIZE * BLOCK_SIZE, 0);
132.     clSetKernelArg(multiplicationKernel[i], 5, sizeof(cl_int), (void *) &uiWA);
133.     clSetKernelArg(multiplicationKernel[i], 6, sizeof(cl_int), (void *) &uiWC);
134.     clSetKernelArg(multiplicationKernel[i], 7, sizeof(cl_int), (void *) &workSize[i]);
135.
136.     if(i+1 < ciDeviceCount)
137.         workOffset[i + 1] = workOffset[i] + workSize[i];
138. }
139.
140. // Execute Multiplication on all GPUs in parallel
141. size_t localWorkSize[] = {BLOCK_SIZE, BLOCK_SIZE};
142. size_t globalWorkSize[] = {shrRoundUp(BLOCK_SIZE, uiWC), shrRoundUp(BLOCK_SIZE, workSize[0])};
143.
144. // Launch kernels on devices
145. #ifdef GPU_PROFILING
146. int nIter = 30;
147.
148. for (int j = -1; j < nIter; j++)
149. {
150.     // Sync all queues to host and start timer first time through loop
151.     if(j == 0){
152.         for(unsigned int i = 0; i < ciDeviceCount; i++)
153.         {
154.             clFinish(commandQueue[i]);
155.         }
156.     }
157. }
```

Code



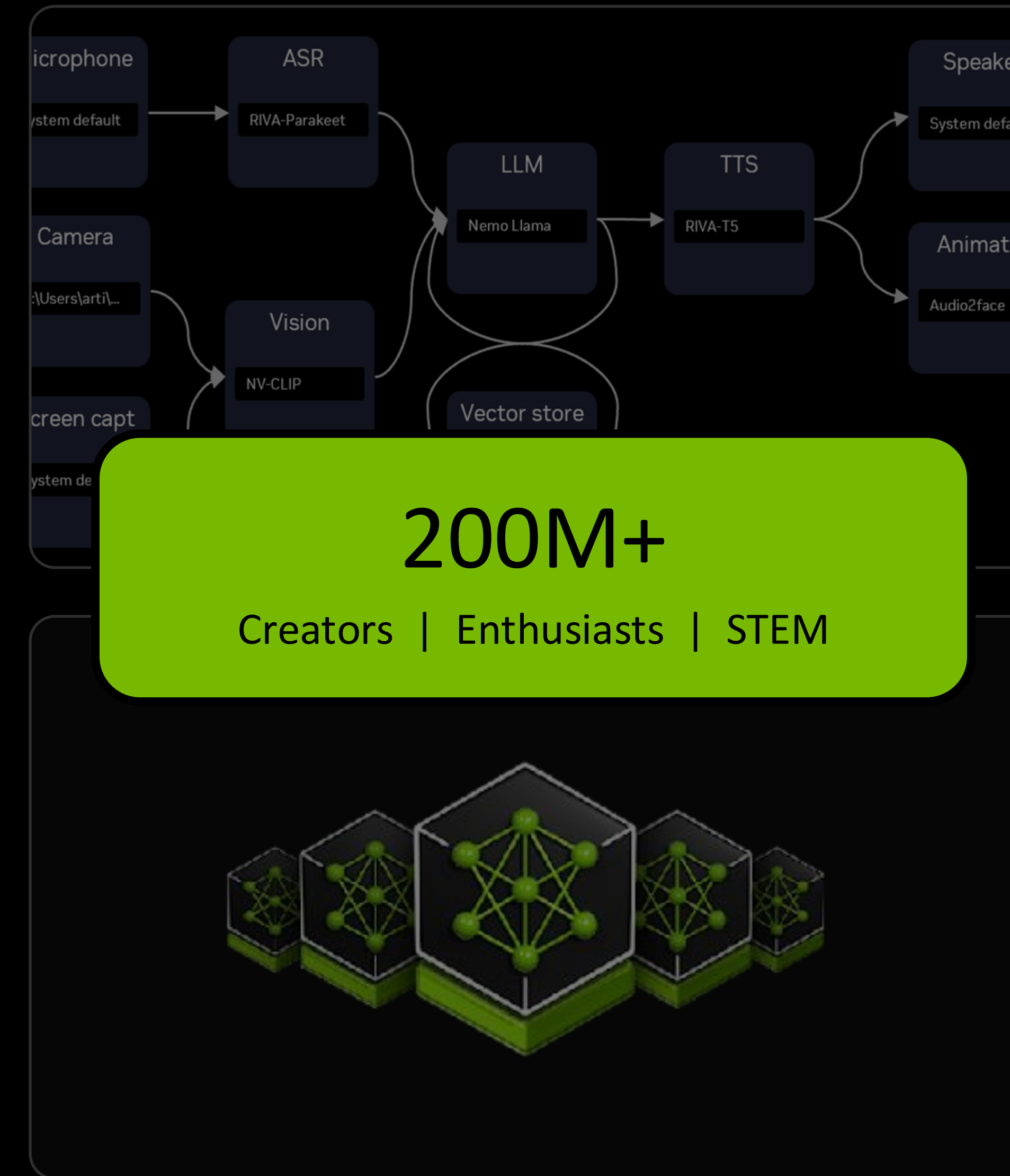
Graph + AI

# The New Software Development Paradigm

With advances in generative AI and new tools, everyone is a developer

```
107 workOffset[0] = 0;
108 for(unsigned int i=0; i < ciDeviceCount; ++i)
109 {
110     // Input buffer
111     workSize[i] = (i != (ciDeviceCount - 1)) ? sizePerGPU : (uIMA - workOffset[i]);
112     d_A[i] = clCreateBuffer(cxGPUContext, CL_MEM_READ_ONLY, workSize[i] * sizeof(float) * uIMA, NULL, NULL);
113
114     // Copy only assigned rows from host to device
115     clEnqueueCopyBuffer(commandQueue[i], h_A, d_A[i], workOffset[i] * sizeof(float) * uIMA,
116                        0, workSize[i] * sizeof(float) * uIMA, 0, NULL, NULL);
117
118     // create OpenCL buffer on device that will be initialize from the host memory on first use
119     // on device
120     d_B[i] = clCreateBuffer(cxGPUContext, CL_MEM_READ_ONLY | CL_MEM_COPY_HOST_PTR,
121                           mem_size_B, h_B_data, NULL);
122
123     // Output buffer
124     d_C[i] = clCreateBuffer(cxGPUContext, CL_MEM_READ_WRITE, workSize[i] * sizeof(float) * uIMA, NULL, NULL);
125
126     workOffset[i + 1] = workOffset[i] + workSize[i];
127 }
128
129 // Execute Multiplication on all GPUs in parallel
130 size_t localWorkSize[] = {BLOCK_SIZE, BLOCK_SIZE};
131 size_t globalWorkSize[] = {shrRoundup(BLOCK_SIZE, uIMC), shrRoundup(BLOCK_SIZE, workSize[0])};
132
133 // Launch kernels on devices
134 #ifdef GPU_PROFILING
135 int nIter = 30;
136 for (int j = -1; j < nIter; j++)
137 {
138     // Sync all queues to host and start timer first time through loop
139     if(j == 0){
140         for(unsigned int i = 0; i < ciDeviceCount; i++)
141         {
142             clFinish(commandQueue[i]);
143         }
144     }
145 }
```

30M  
Developers



Code

Graph + AI

# NVIDIA NIM for RTX

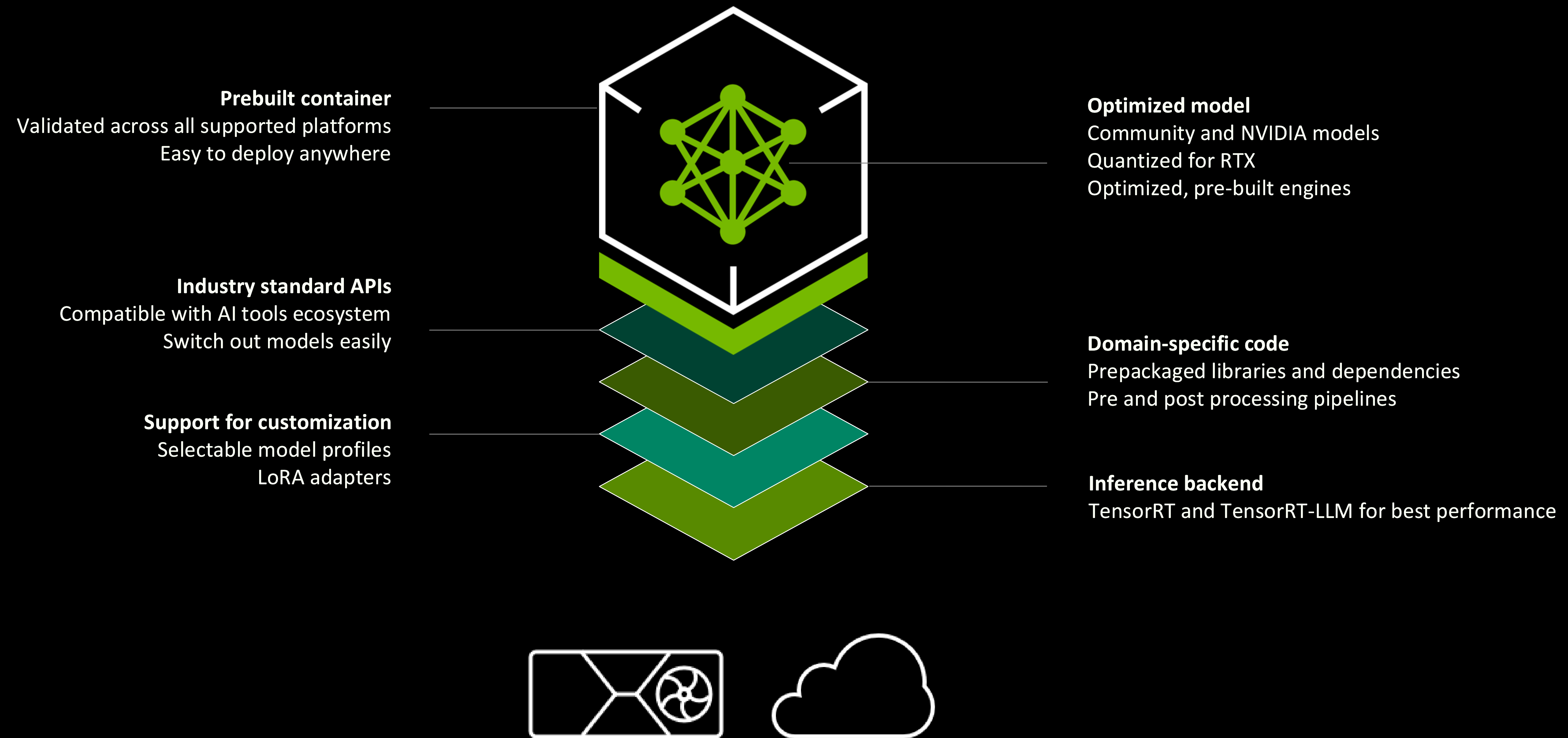
Optimized, prepackaged microservices for generative AI



- Easy to Use – download and connect with streamlined API
- Optimized for performance on RTX
- Top community and NVIDIA models
- Compatible with AI ecosystem tools
- Deploy anywhere – PC to Cloud

# NVIDIA NIM for RTX

Optimized, prepackaged microservices for generative AI



# Initial Wave of RTX NIMs – Coming in February

With many more to come...



## Language

Llama 3.1 8B  
instruct  
Llama 3.2 3B  
Mistral-nemo-12B-  
instruct  
Starcoder 2 15B  
Mixtral 8x7B



## Regional Language

RIVA Megatron  
1b-nmt



## Vision Language

NV-CLIP



## RAG

NV-EmbedQA-5-  
V5  
Llama-3.2-NV-  
RerankQA-  
1B-v1



## Speech

Riva  
Parakeet-  
ctc-1.1b-asr  
Riva TTS  
Maxine  
Studio Voice



## Animation

Audio2Face



## Computer Vision

PaddleOCR



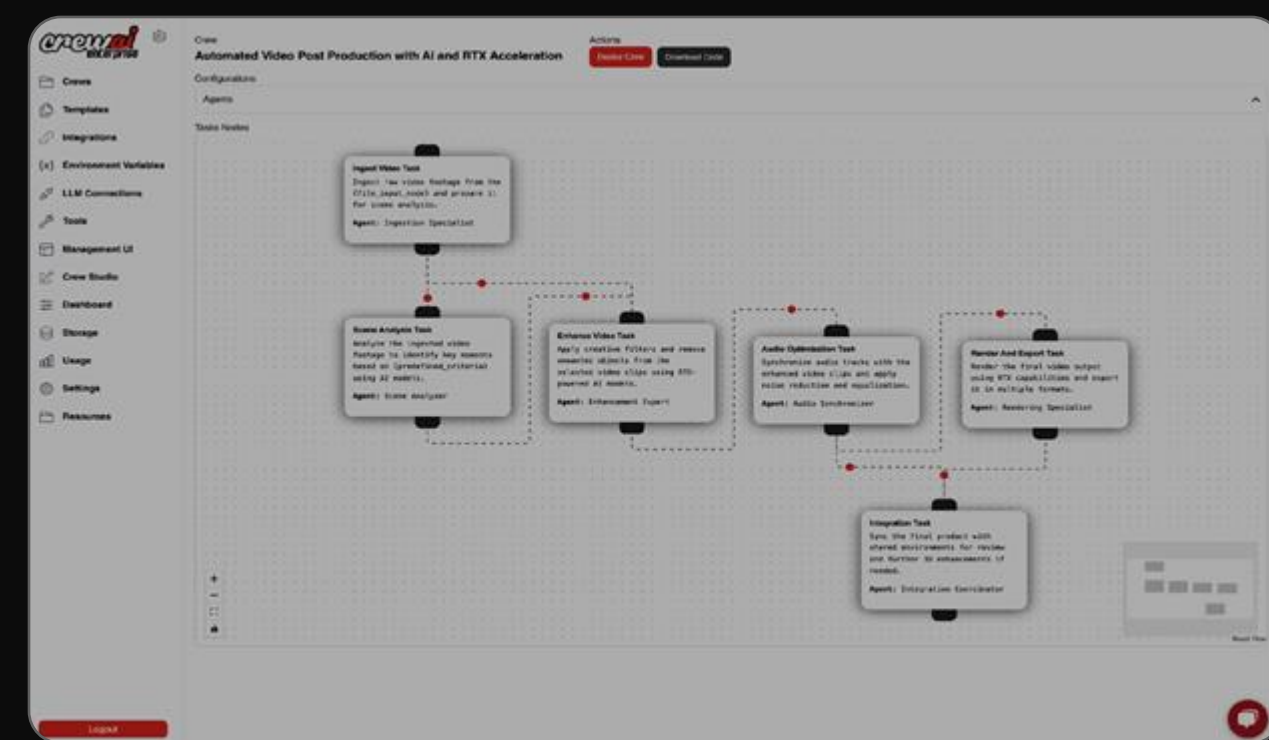
## Image

SDXL  
Flux Image

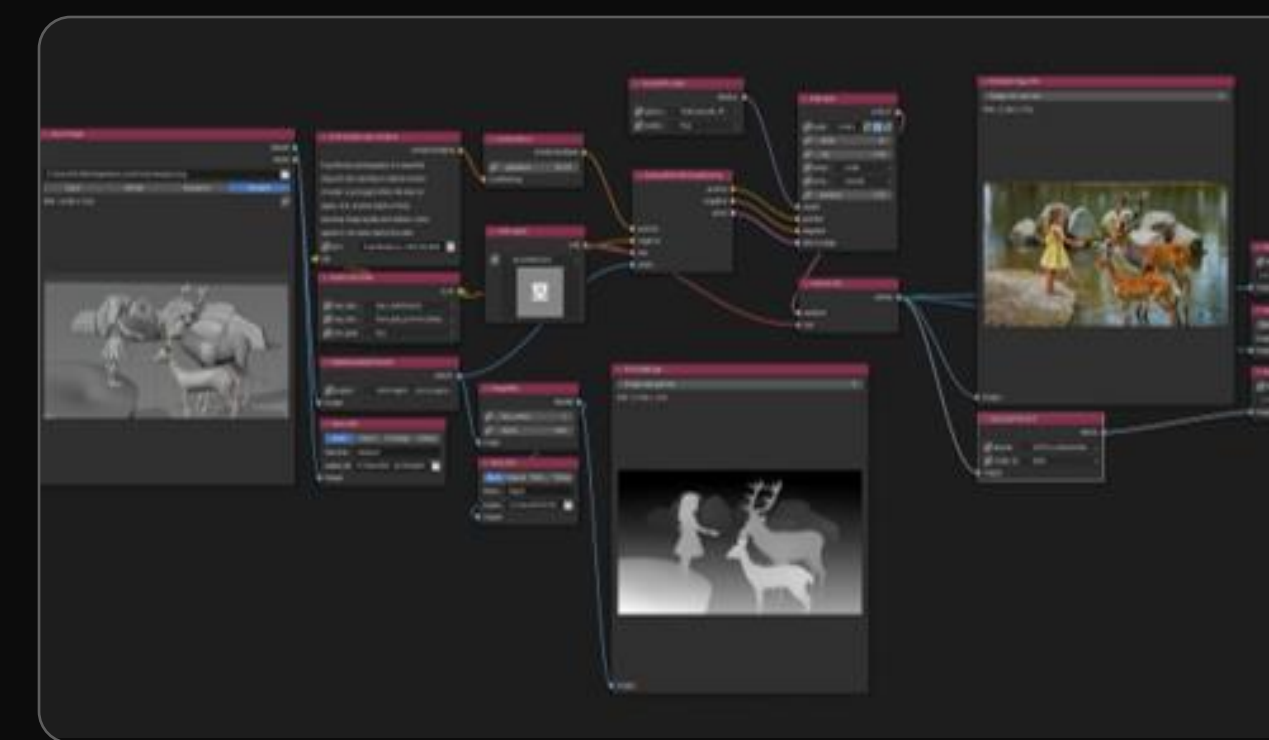
# Get Started with NIM in Top AI Tools

Build and customize chatbots, AI agents, and creative workflows

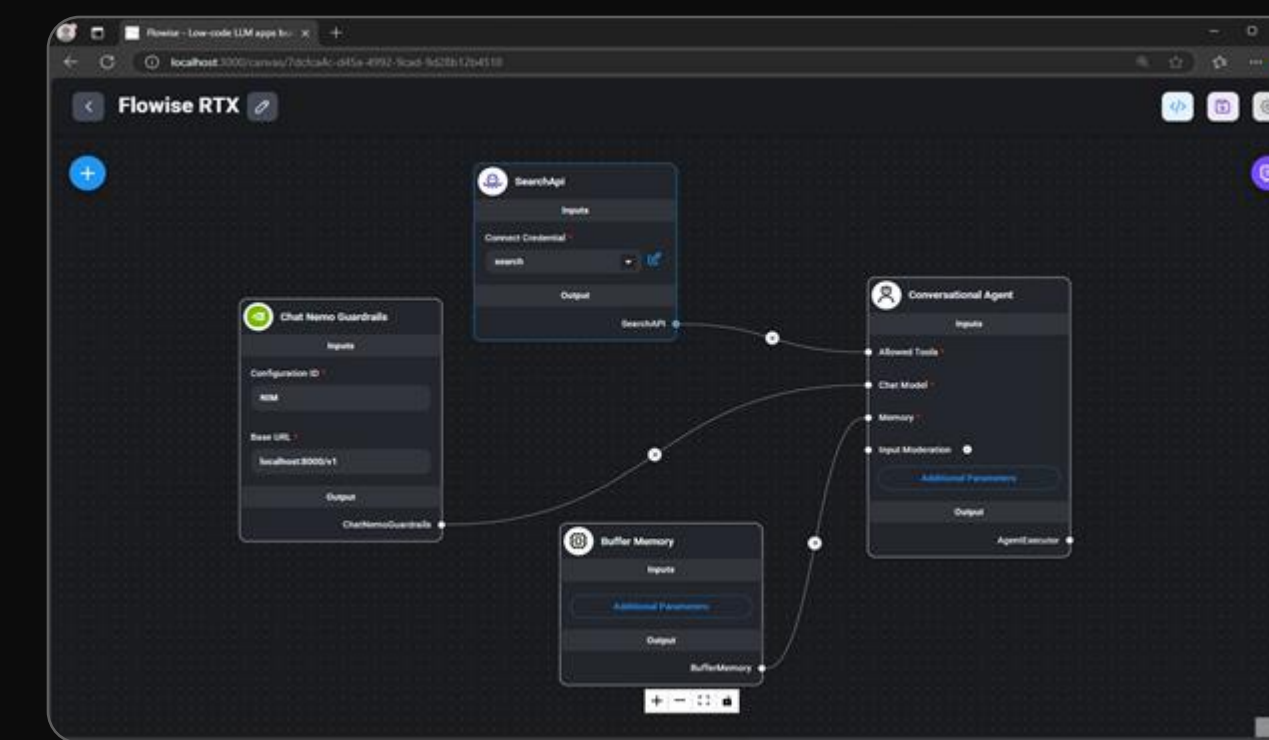
## Graph UI



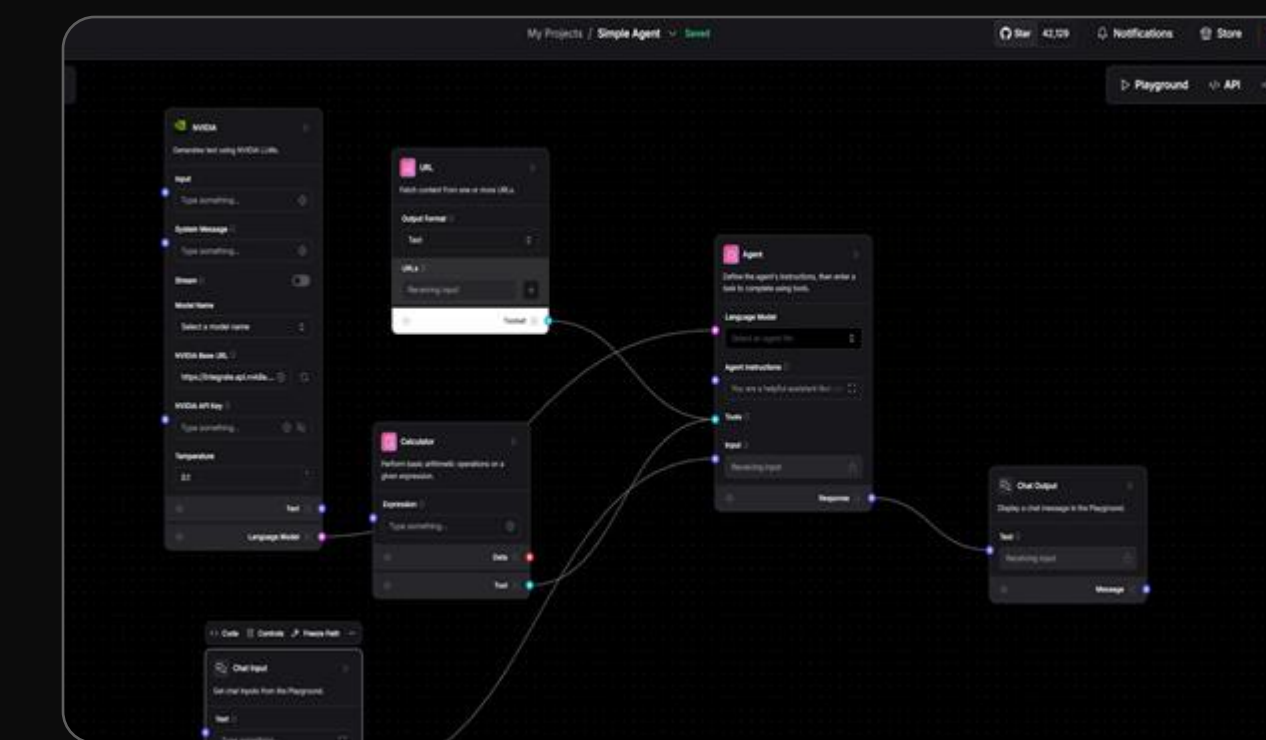
CrewAI



ComfyUI

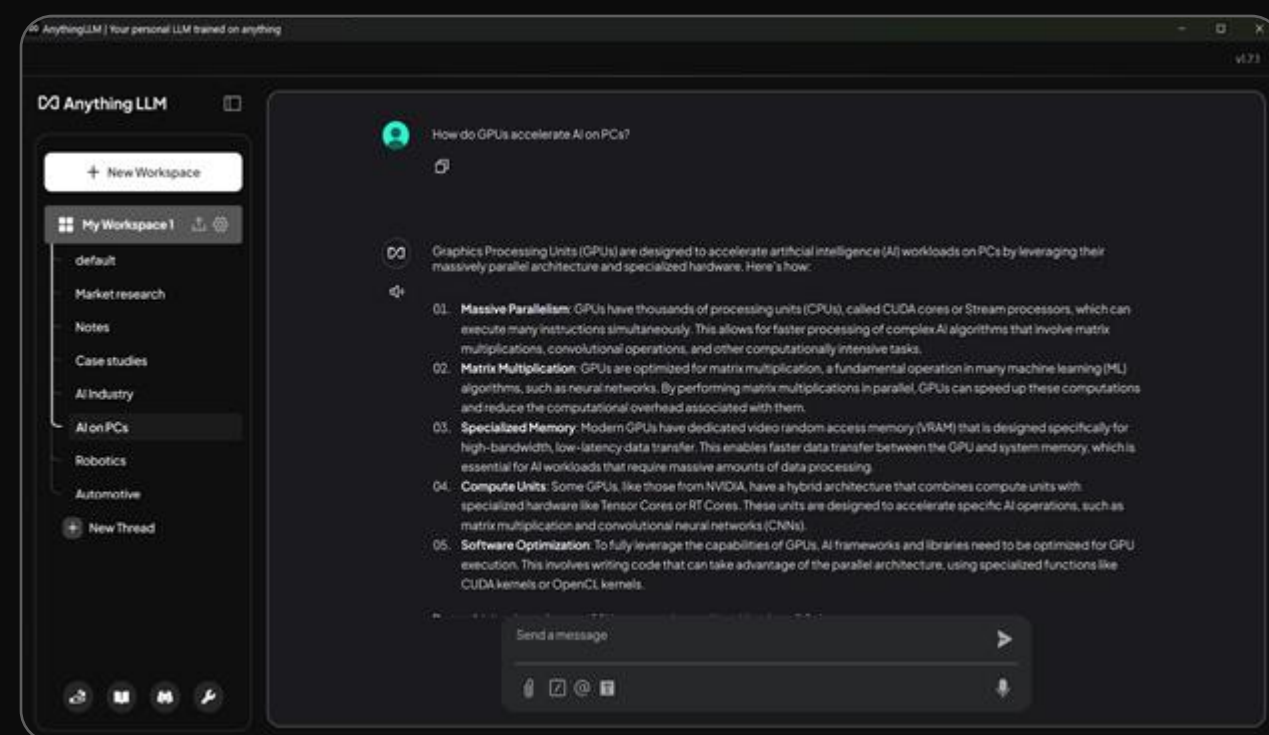


FlowiseAI

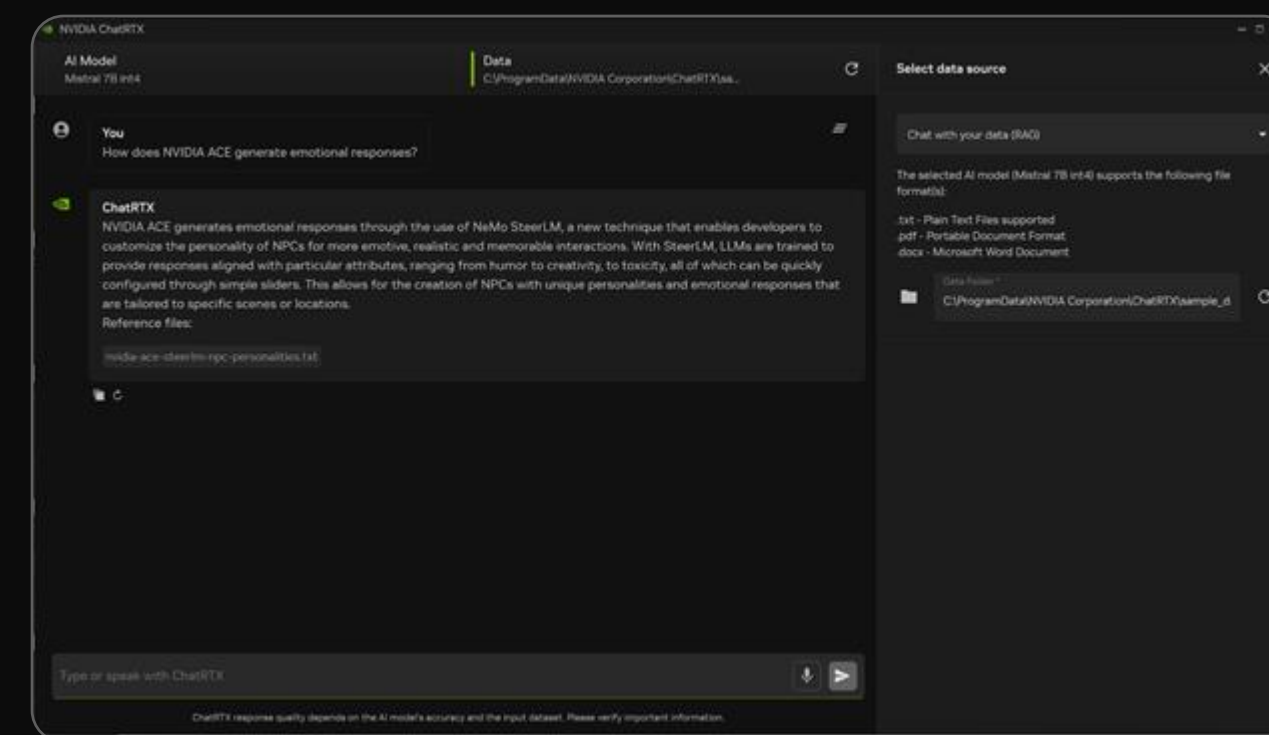


Langflow

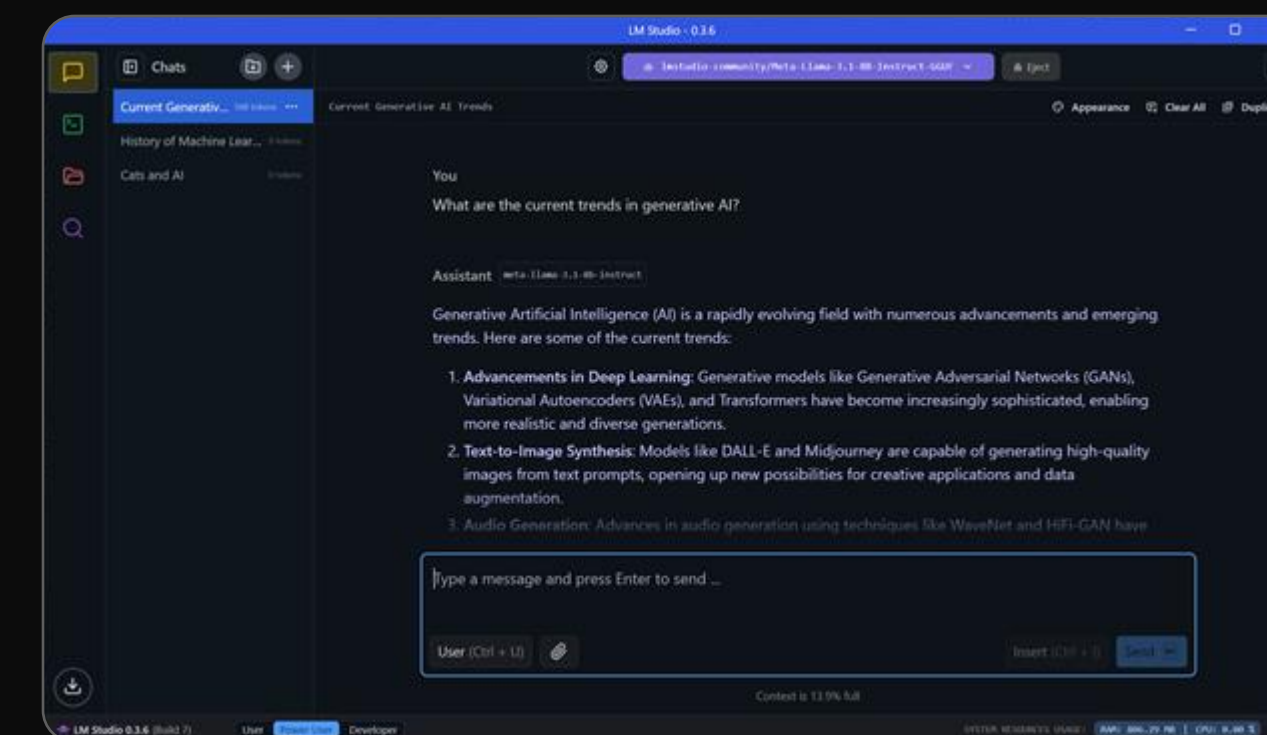
## Chat UI



AnythingLLM

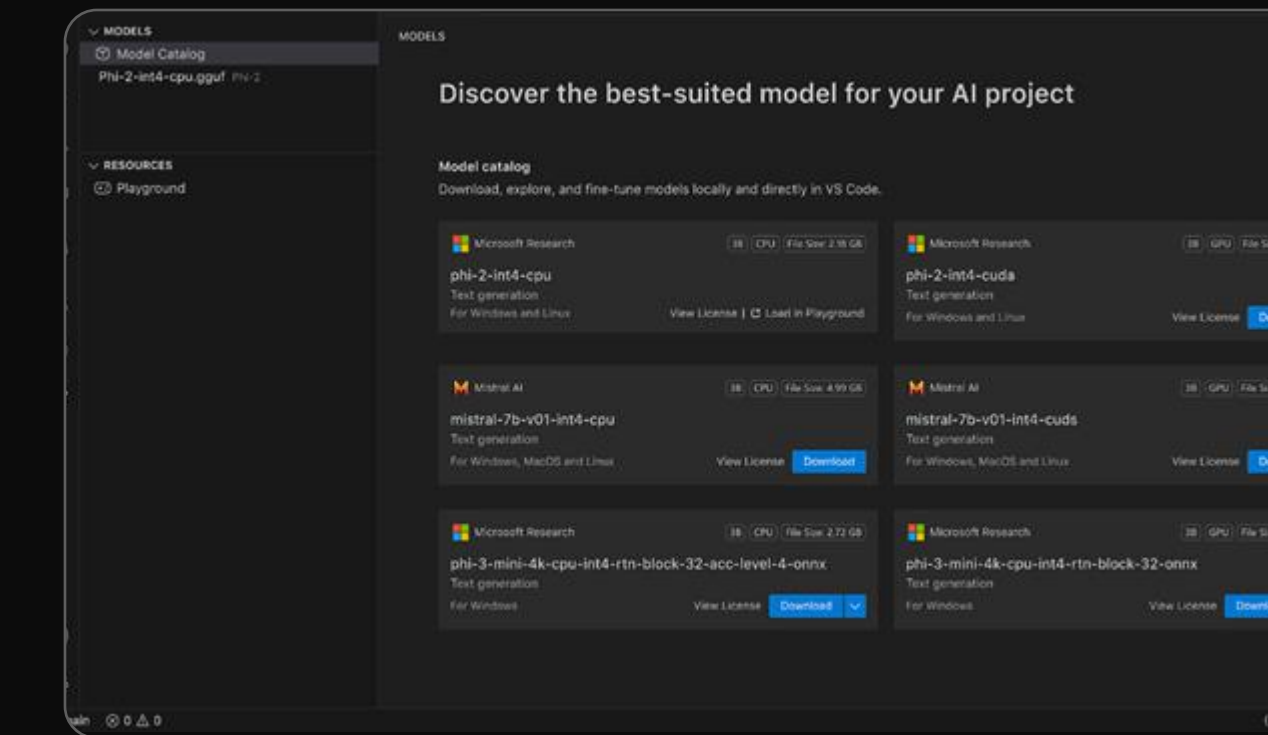


NVIDIA ChatRTX



LM Studio

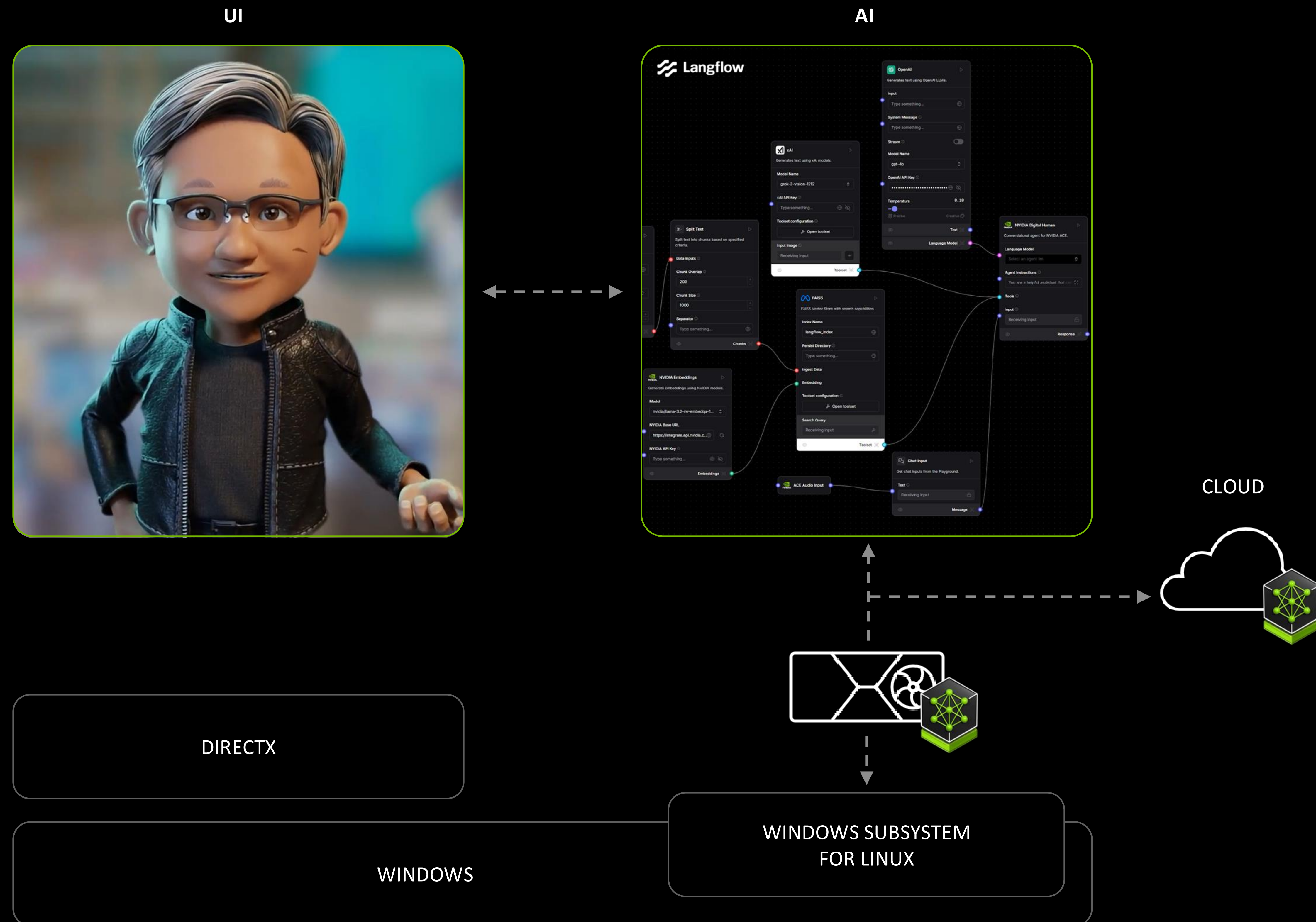
## Model Tuning



AI Toolkit for VS Code

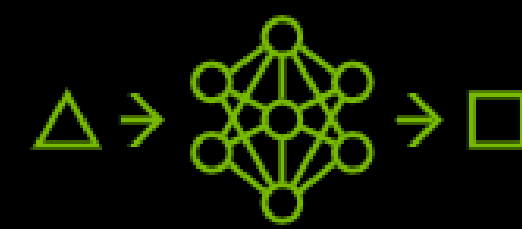
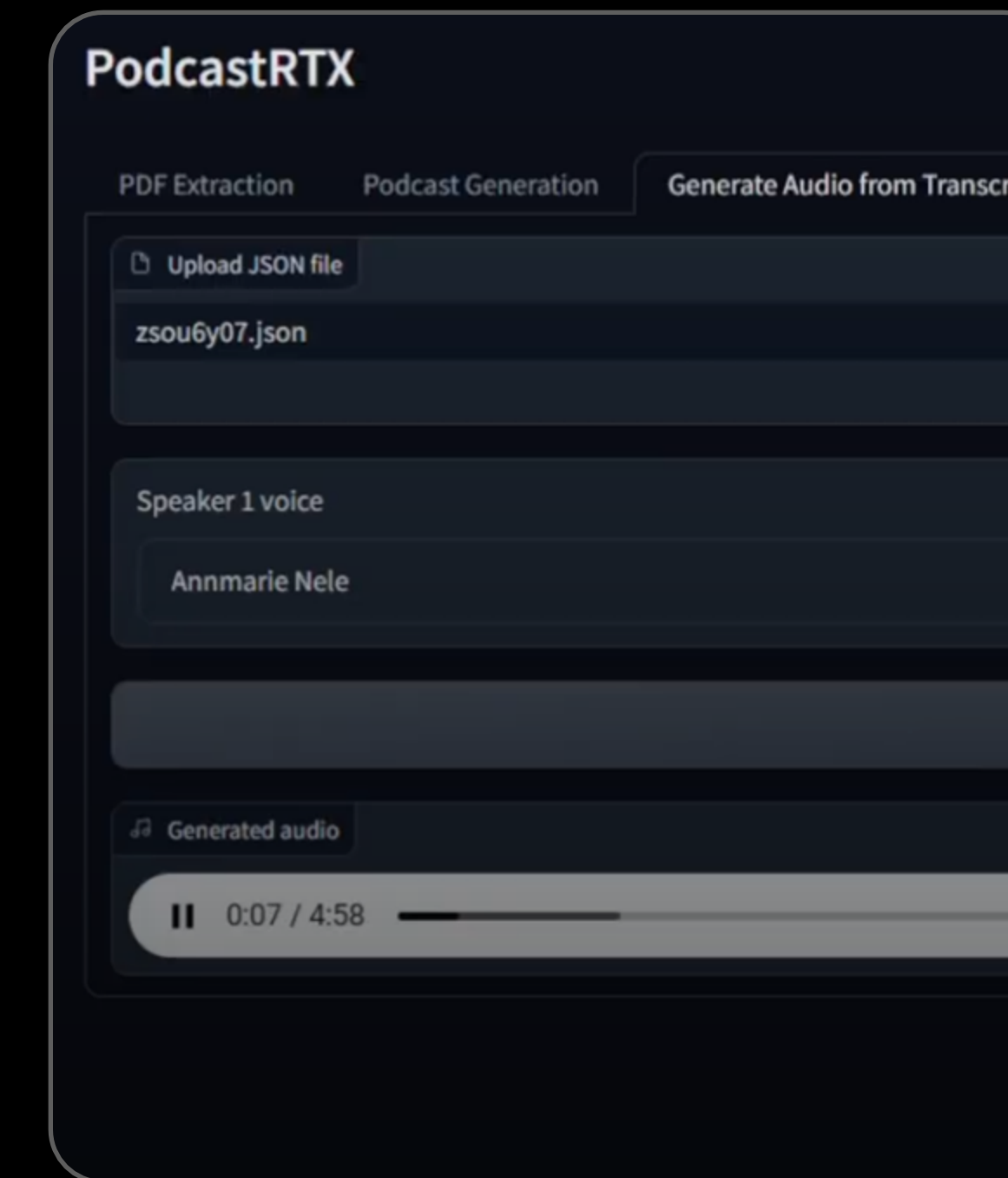
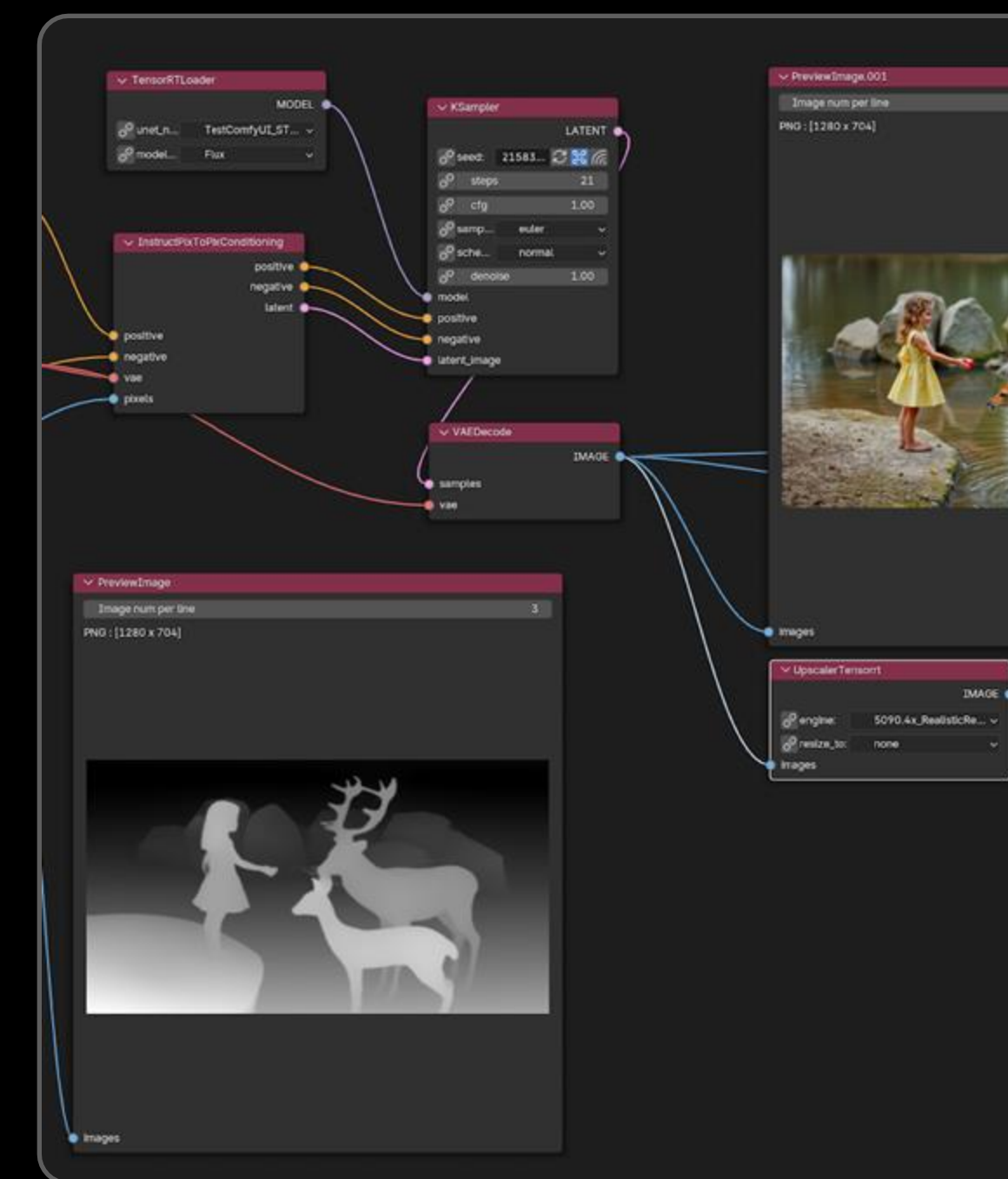


# The RTX AI PC



# NVIDIA AI Blueprints for RTX

Customizable, extensible reference implementations of NIM-powered AI workflows



Reference Application



Sample Data



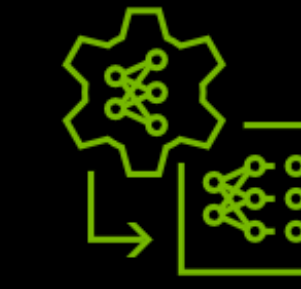
Reference Code



Architecture



Customization Tools



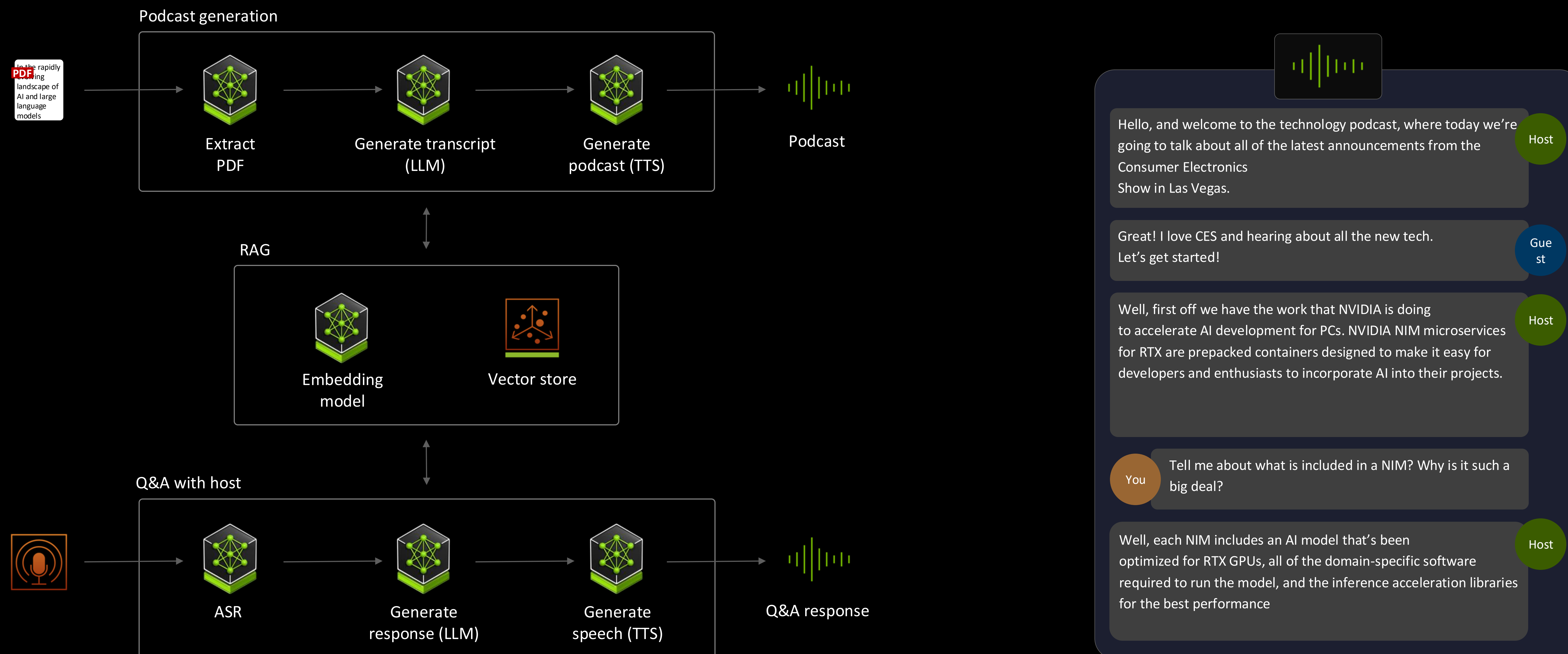
Orchestration Tools



NVIDIA NIM

# NVIDIA AI Blueprint for RTX: PDF to Podcast

Generate engaging podcasts from any PDF | Q&A with podcast host | End-to-end workflow, powered by NIM



# PDF-to-Podcast

PDF to Transcript   **Generate Audio from Transcript**   PDF RAG

Upload JSON file ✕

tech-final.json 9.2 KB ↓

Language

en ▼

Speaker 1 voice


sophia ▼

Speaker 2 voice

tom ▼

**Generate Audio**

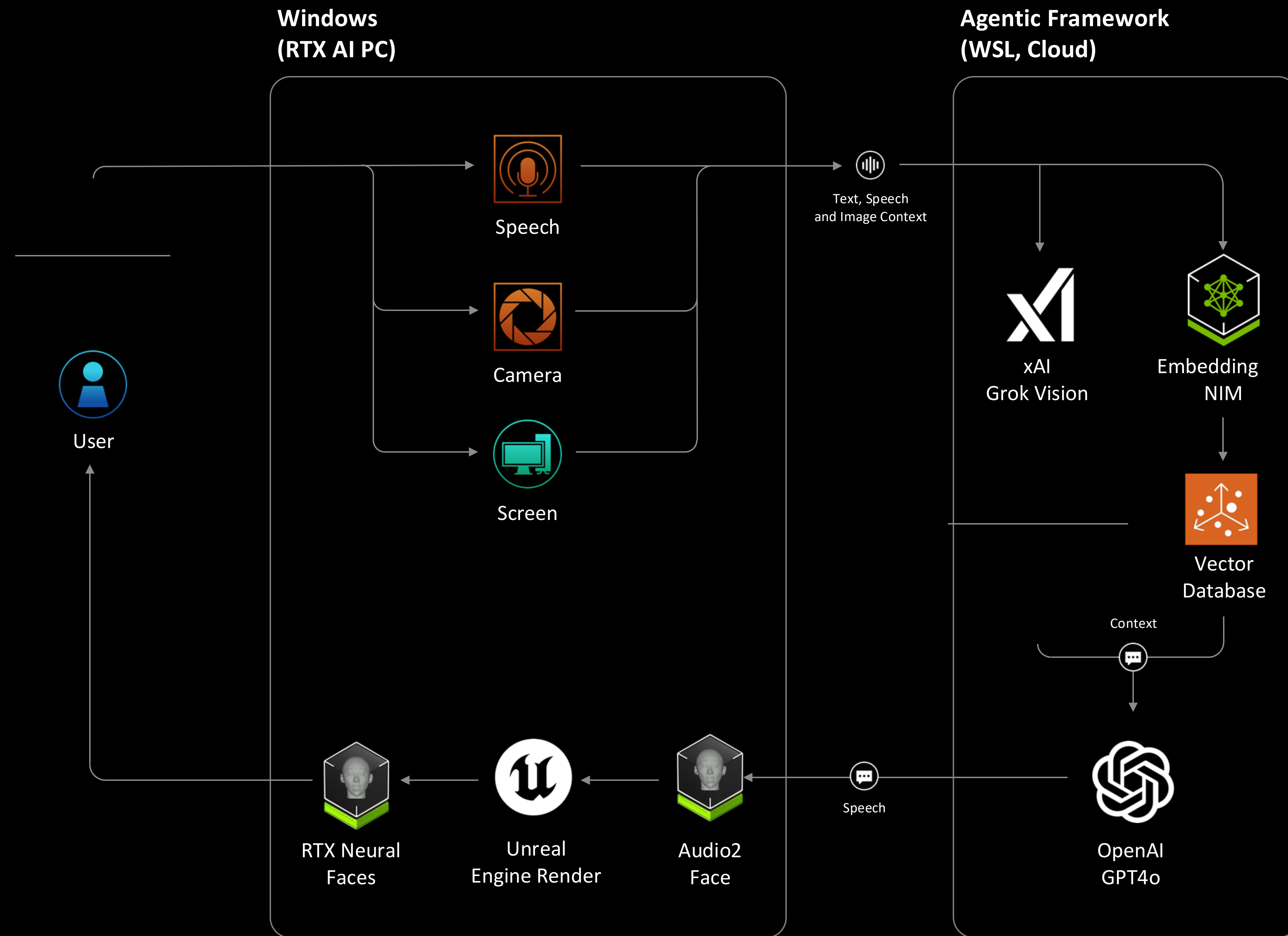
Generated audio 📄 ⬇️



0:06 7:51

🔊 1x ⏪ ⏩

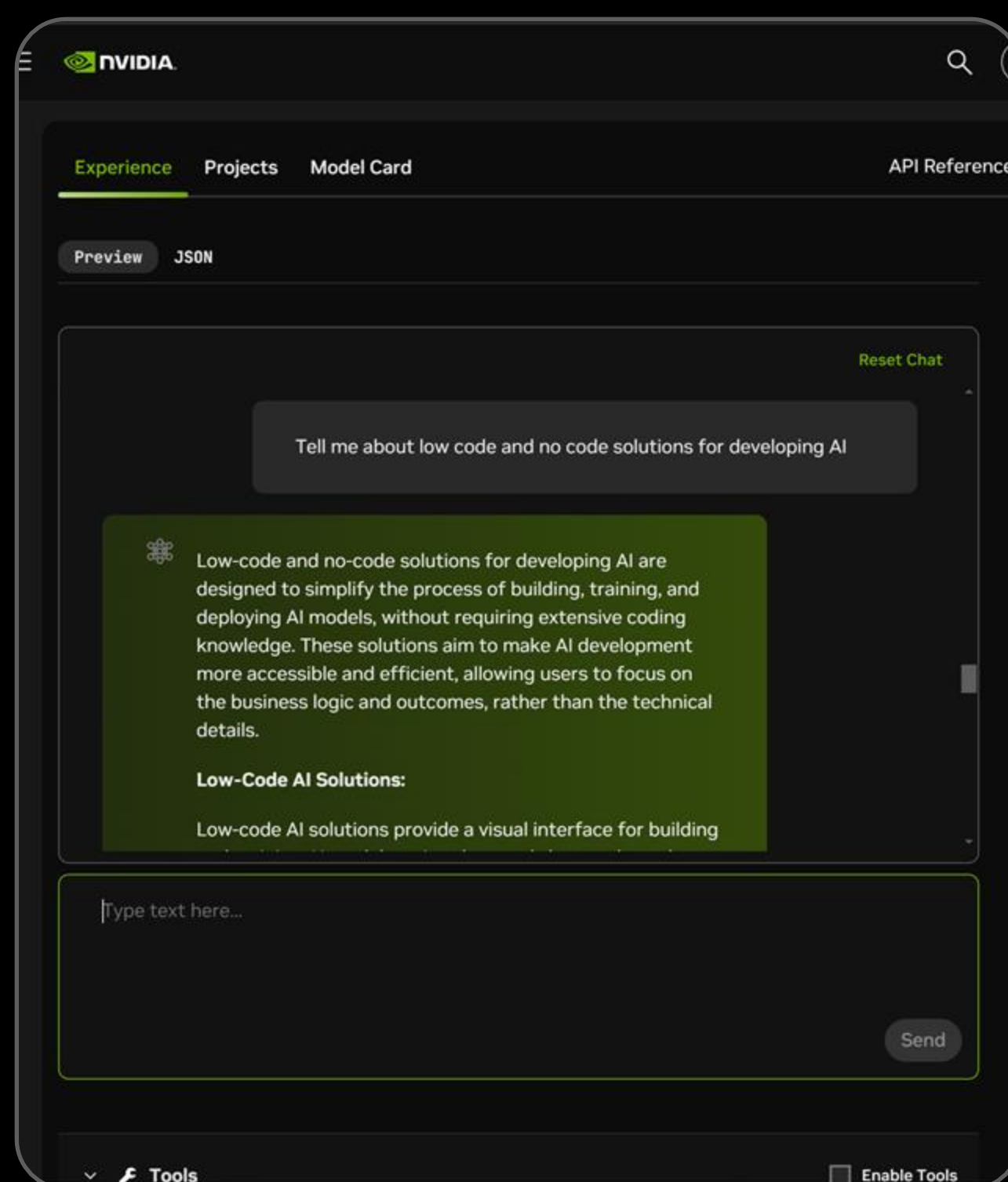
# NVIDIA AI Blueprint for RTX: Digital Human





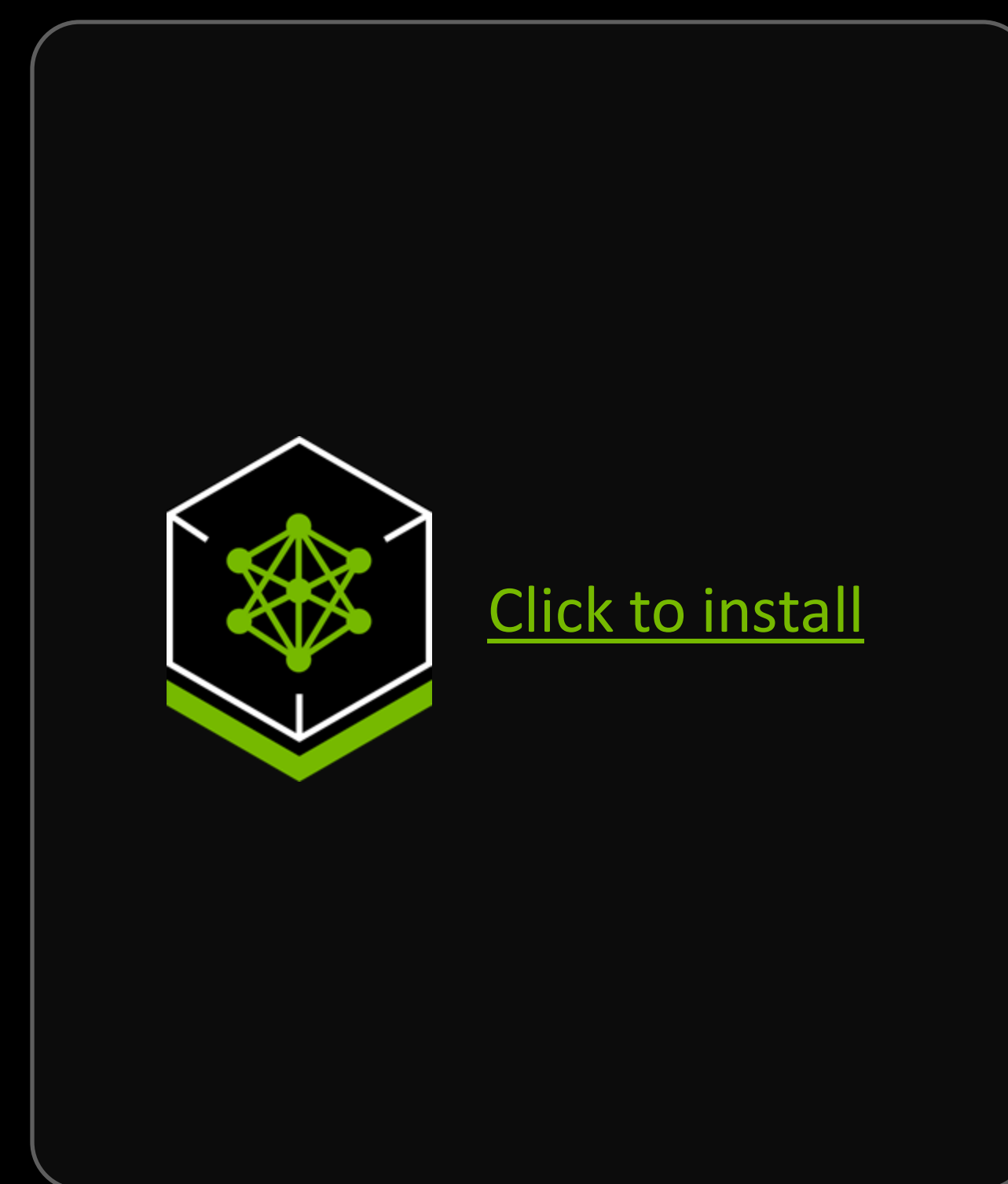
DEMO – R2X

# Develop with NVIDIA NIM and AI Blueprints on RTX



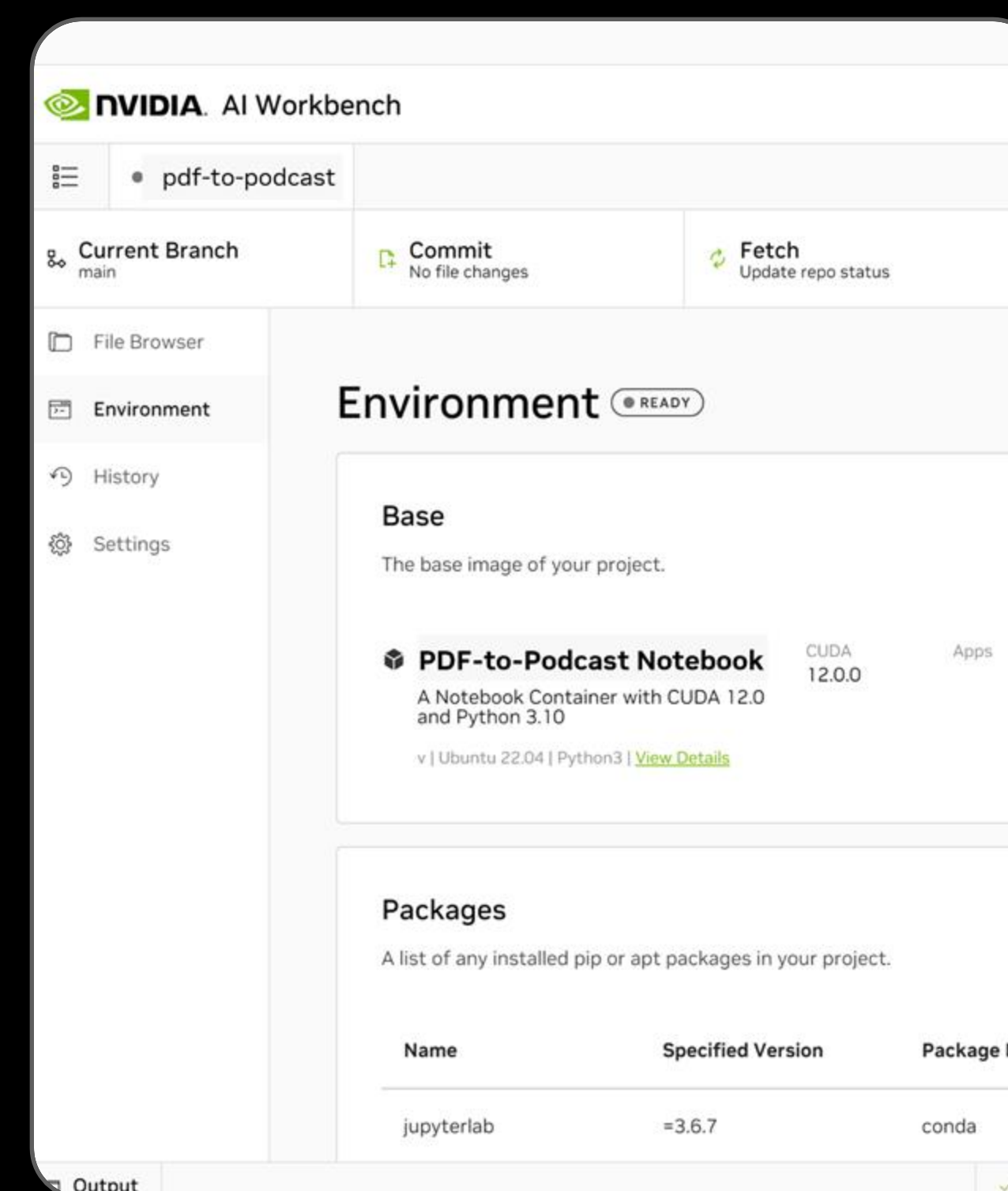
Experience NIM on Web

Try playground at [build.nvidia.com](https://build.nvidia.com)



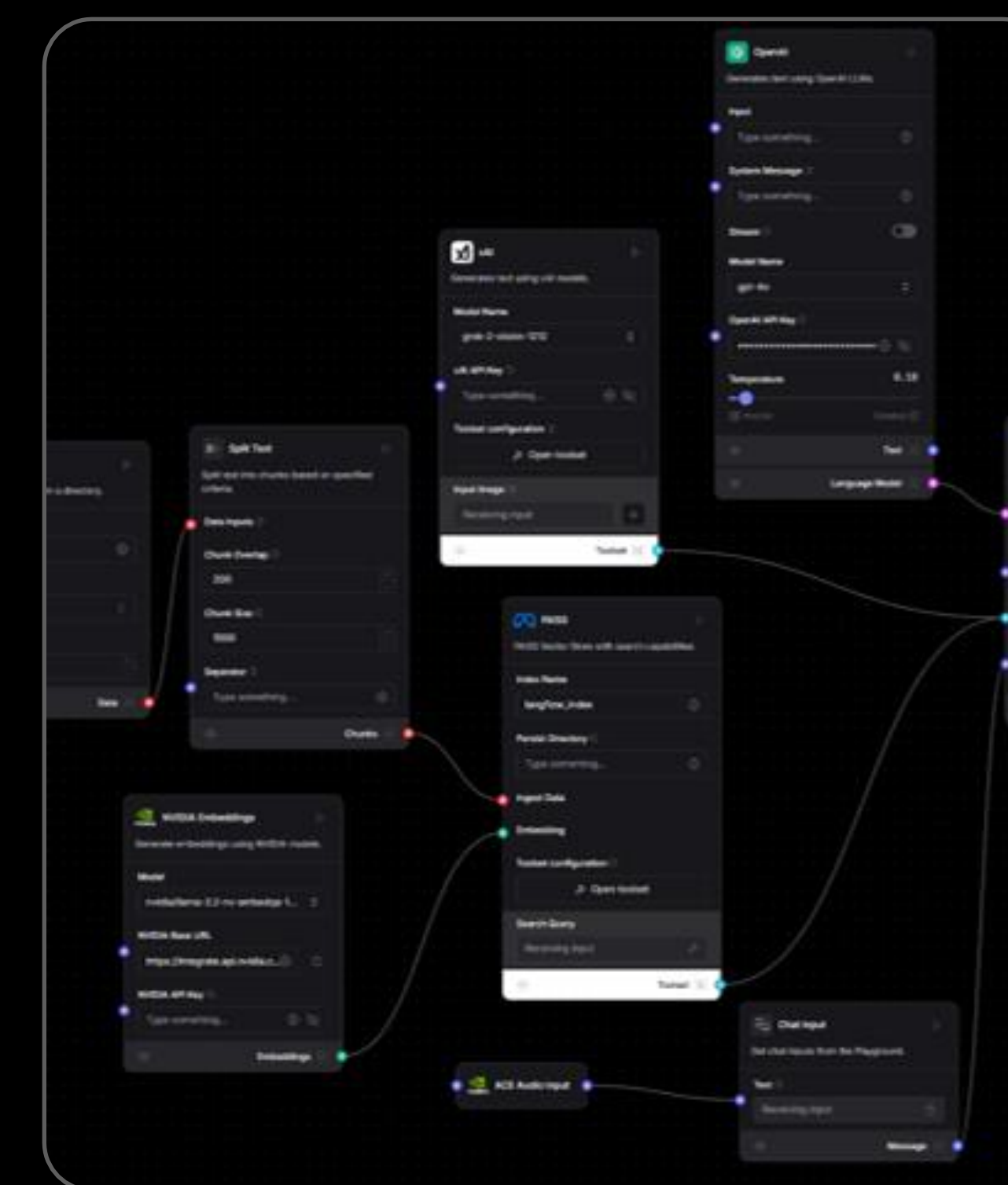
Run NIM on RTX

Download and install with one click



Leverage AI Blueprints

Deploy and run with AI Workbench



Integrate Agents

Connect with Graph UIs



Connect to UI

Build with NVIDIA digital human

# NIM on RTX

Coming soon to RTX AI PCs from all top OEMs



## MODELS



Meta

stability.ai



## EXPERIENCES

Anything LLM

ComfyUI



LM Studio

## DEVELOPER TOOLS

crewai

LangChain

FlowiseAI

Langflow

acer

ASUS

DELL

GIGABYTE™

HONOR



Lenovo

LG

机械革命  
MECHREVO



msi

RAZER

SAMSUNG



# NVIDIA NIM and AI Blueprints for RTX



- Easy-to-use prepackaged NIM microservices optimized for RTX
- Extensible reference blueprints for NIM-powered AI workflows
- Integration with top AI tools and frameworks
- Available for download in February



**Thank you**