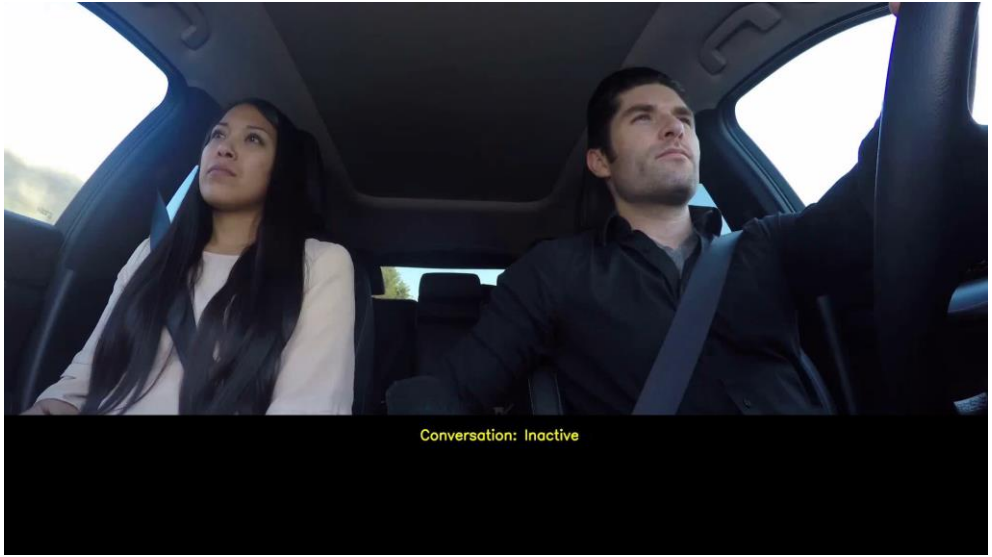
Jarvis and NeMo

GTC China


Jarvis

JARVIS

Platform to develop and deploy conversational AI applications
Designed for sensor fusion



Gaze & Speech
https://www.youtube.com/watch?v=r264lBi1nMU

USE CASES

ACROSS ALL VERTICALS

Online Store

Provide conversational interface for shopping

Industrial

Collaborative robots - Robots and humans collaborate in close proximity

Engineer troubleshooting with the help of AI assistant

Finance

Call center: Sentiment of customers calling

Insurance chatbot: “Add a wedding ring to an insurance policy via an image and receive policy price quote”

Energy / Oil & Gas

Use camera and ask, “what are the safety guidelines for this chemical”?

Loud environment - virtual assistant using lip reading

Consumer Internet

Video diarization - Meeting/conversation transcription per person with timestamps

Content tagging with Image, text, Audio - Recommendation, Ads

In car experience

Autonomous Driving: Enhanced In-car experience combining visual inputs with speech

CHALLENGES OF CONVERSATIONAL AI

Custom models	Deployment	Multiple sensors	High accuracy	Real Time
Cloud services not customizable High costs Data Sovereignty	Existing software not designed for modern production environments	Difficult to use multiple sensors efficiently	Need state-of-the-art algorithms and models	Requires low latency for natural interaction

JARVIS BENEFITS

Custom models

Start from base model, train with *your* data on *your* infrastructure

Deployment

Micro-service approach
Designed for K8s
Simple APIs, easy to integrate

Multiple sensors

Framework for training and deploying models across modalities
Tools to simplify fusion

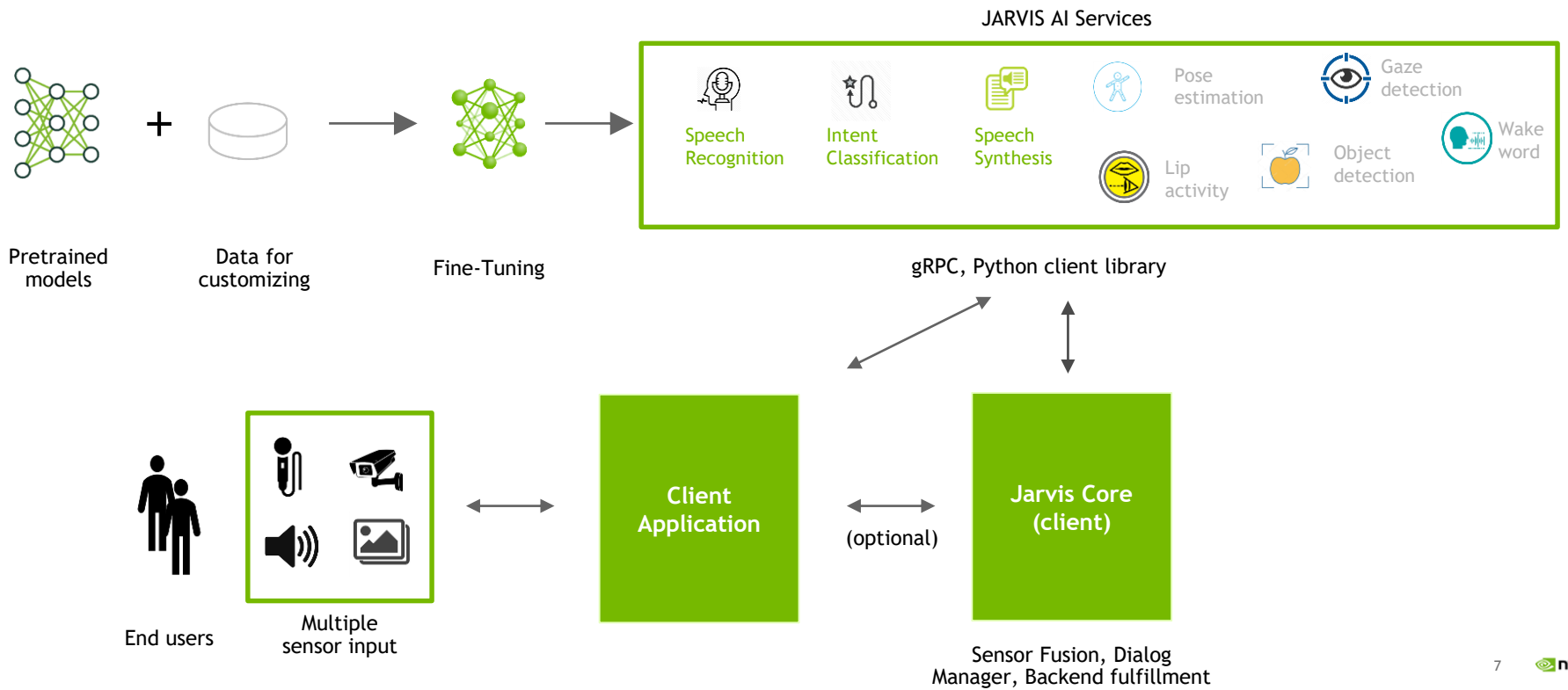
High accuracy

Best-in-breed algorithms
Direct access to cutting-edge research

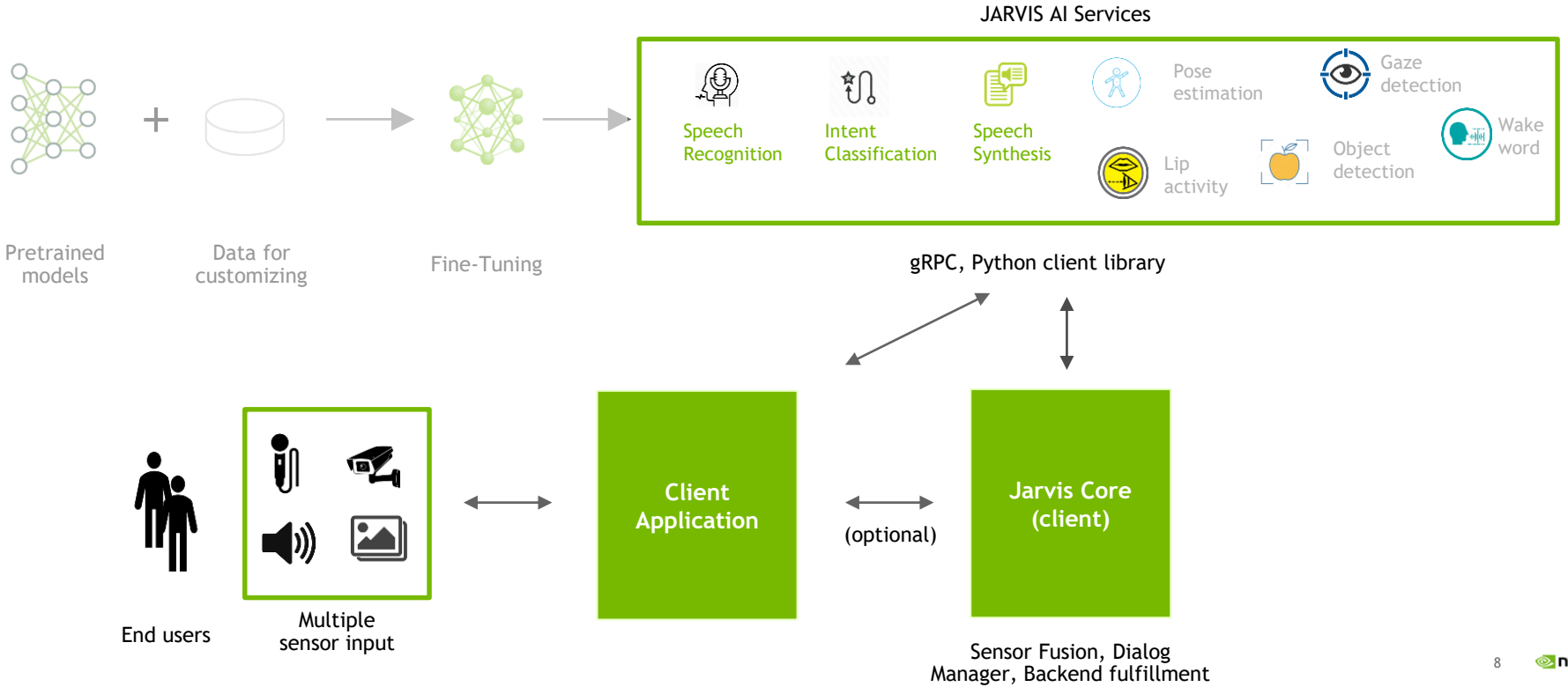
Real Time

End-to-end inference on GPUs optimized to reduce latency

JARVIS WORKFLOW OVERVIEW



JARVIS WORKFLOW OVERVIEW



Visual Diarization

Multiple speaker transcription based on video and audio streams

Interaction: Jupyter notebook with live video stream overlaying gaze detection and lip activity detection and producing a text transcript per person from the audio stream

Technology of sensor fusion:

- Video stream
  - Gaze detection to engage the system
  - Lip activity to determine who is speaking
- Audio stream:
  - Transcribe the audio
  - Label transcriptions per individual speaker

Implementation:

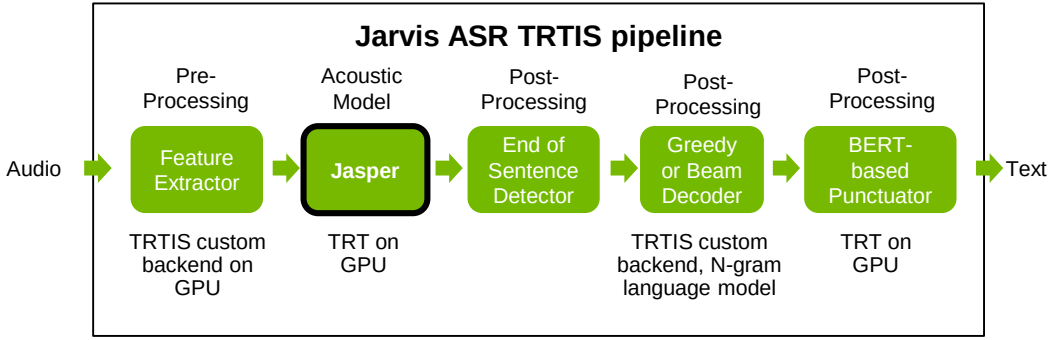
- Fusion graph via JSON to combine the multiple inference models
- gRPC end points for direct interaction with the inference models
- Jupyter notebook demonstrates Python APIs for interaction

Model Developer: Improve the conversational model accuracy via fine-tuning with NeMo

Developer Operations: Deploy via docker containers from NGC into Kubernetes (EGX)

Transcription
Driver: Where is a good sushi restaurant?
Passenger: What's the weather in Chicago

Jarvis ASR Service



Jarvis ASR API

Method Name	Description
Recognize	Given audio file as input, return transcript
StreamingRecognize	Process audio from a file or a microphone as it's being captured, returning partial transcripts

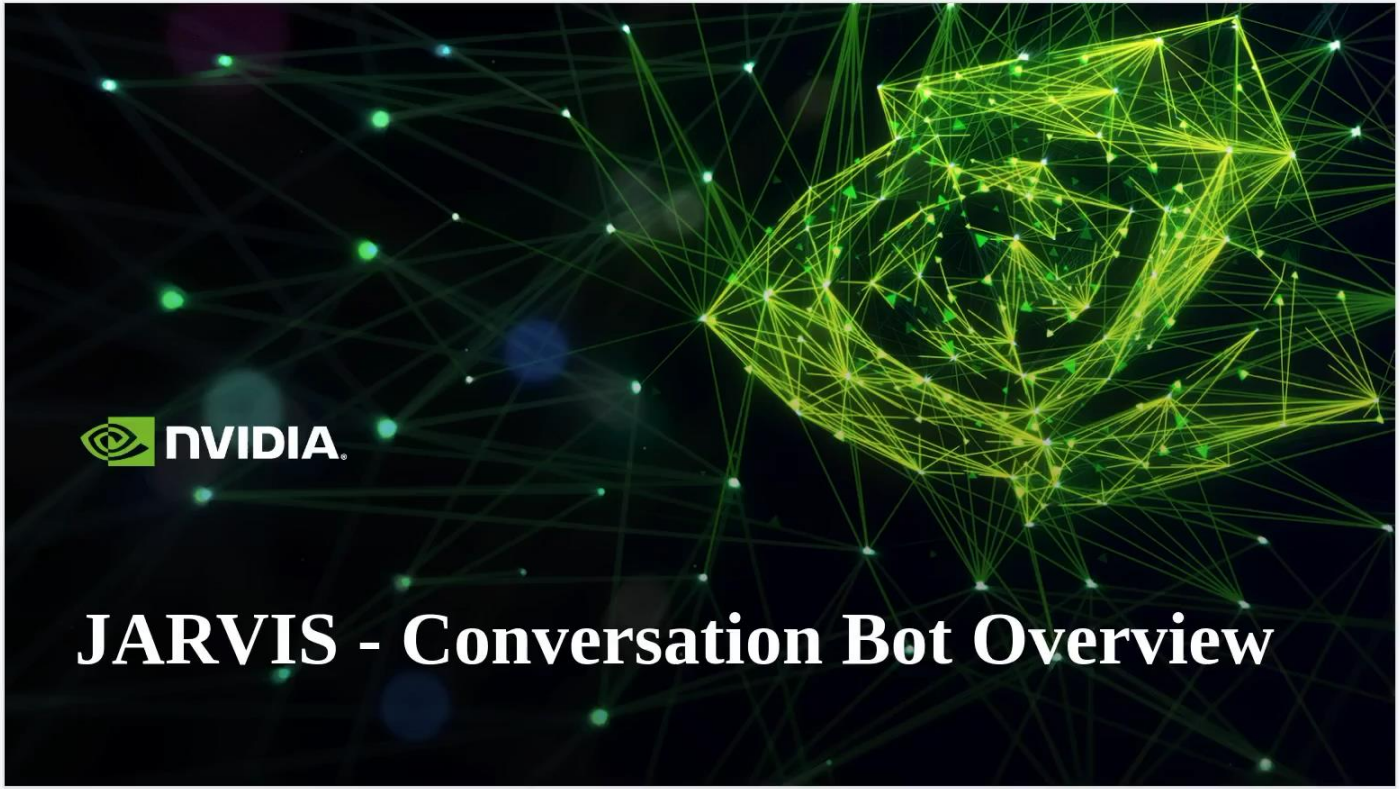
Jarvis Conversational Bot - Purnendu - Google Slides - Google Chrome

Jarvis Conversational Bot - Purnendu

File Edit View Insert Format Slide Arrange Tools Add-ons Help All changes saved in Drive

Present Share

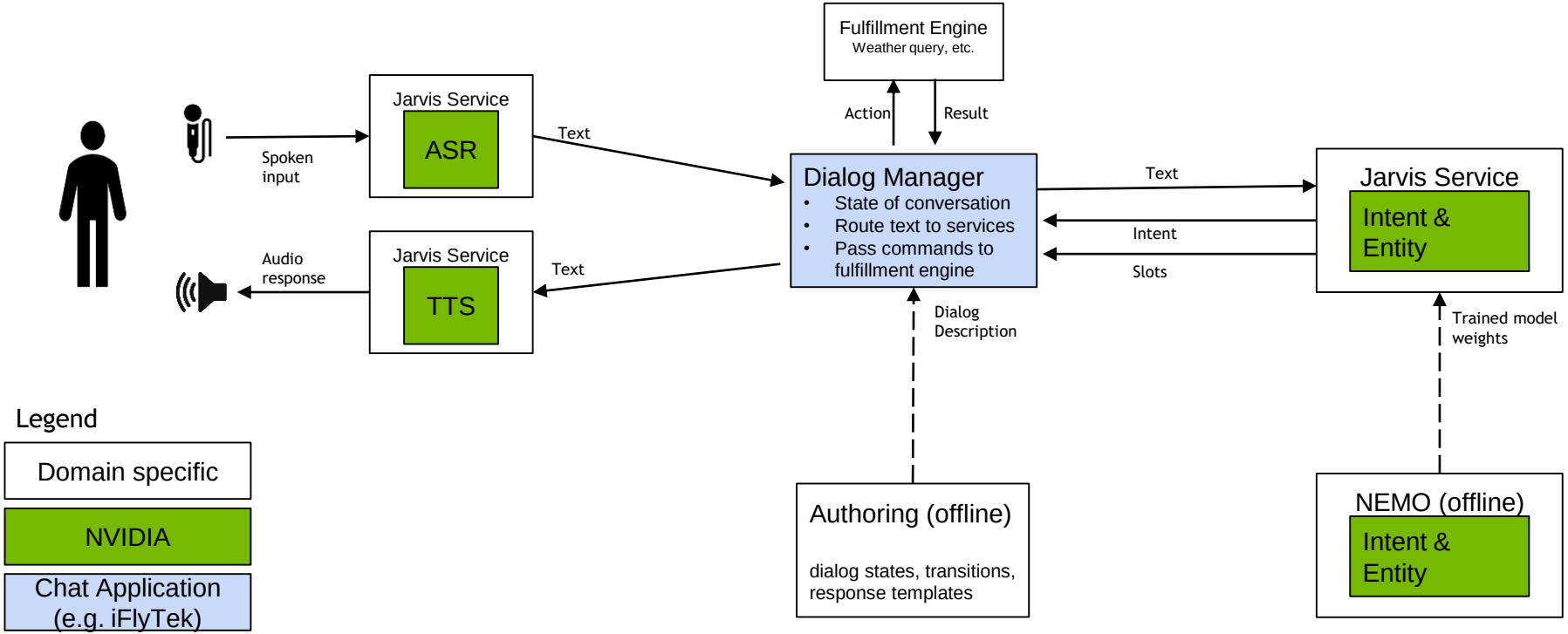
Background Layout Theme Transition



Explore

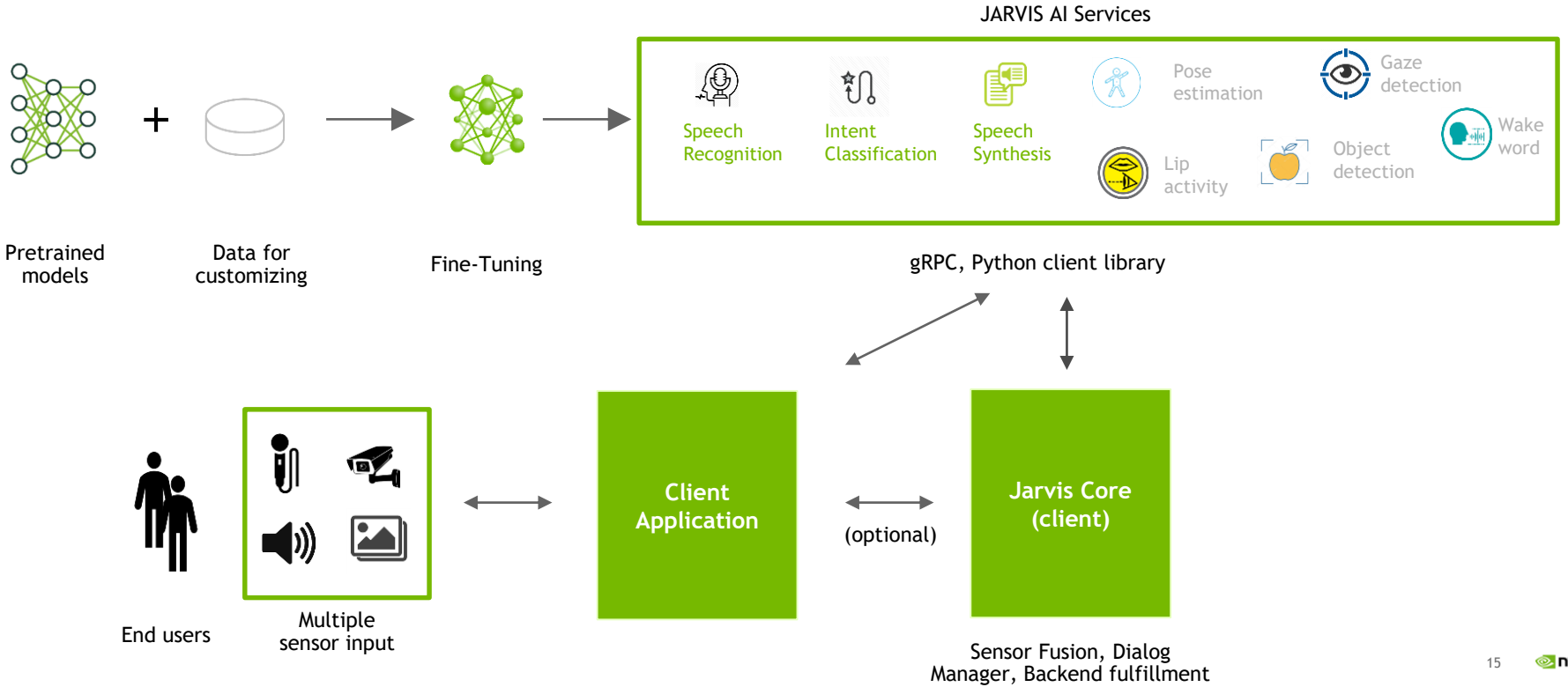
Jarvis - Weather Bot Architecture

Deployment of Jarvis components with simple dialog manager

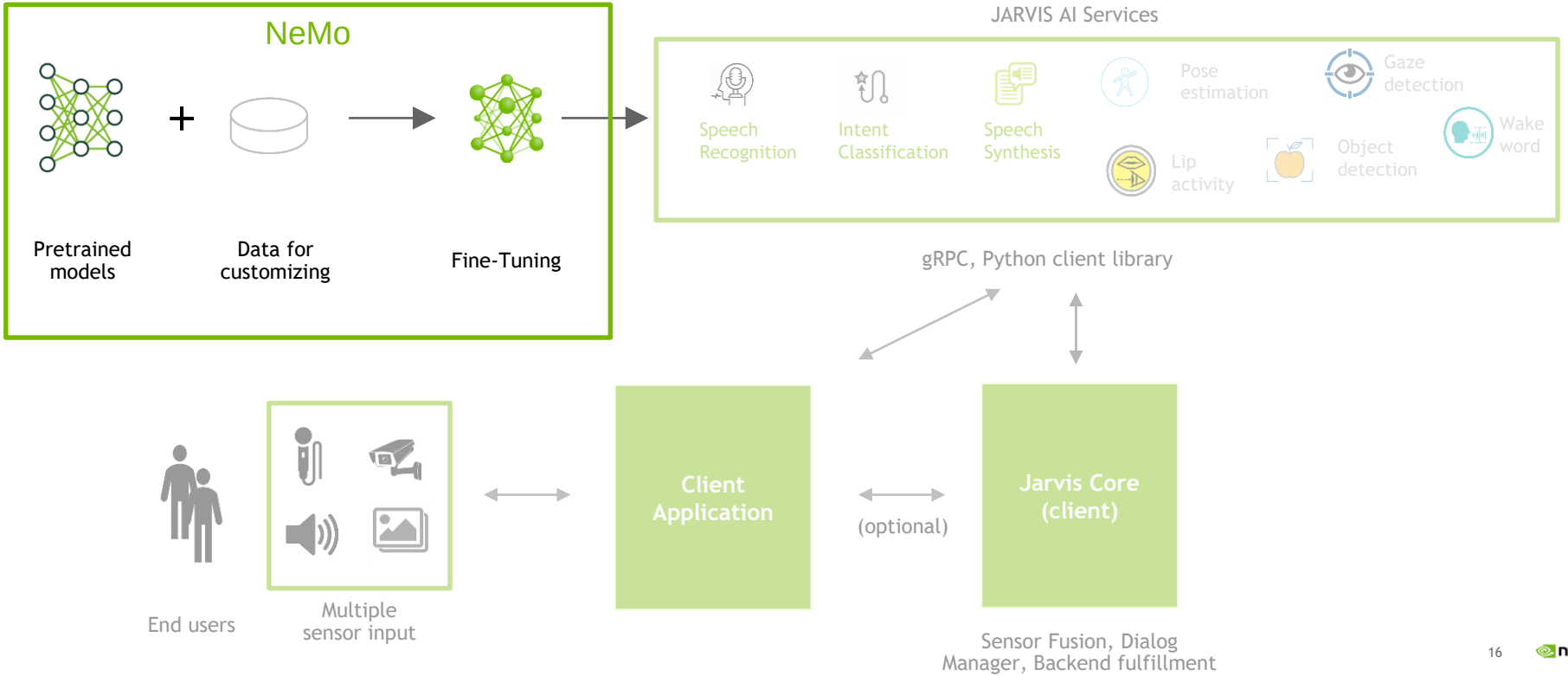


Neural Modules (NeMo)

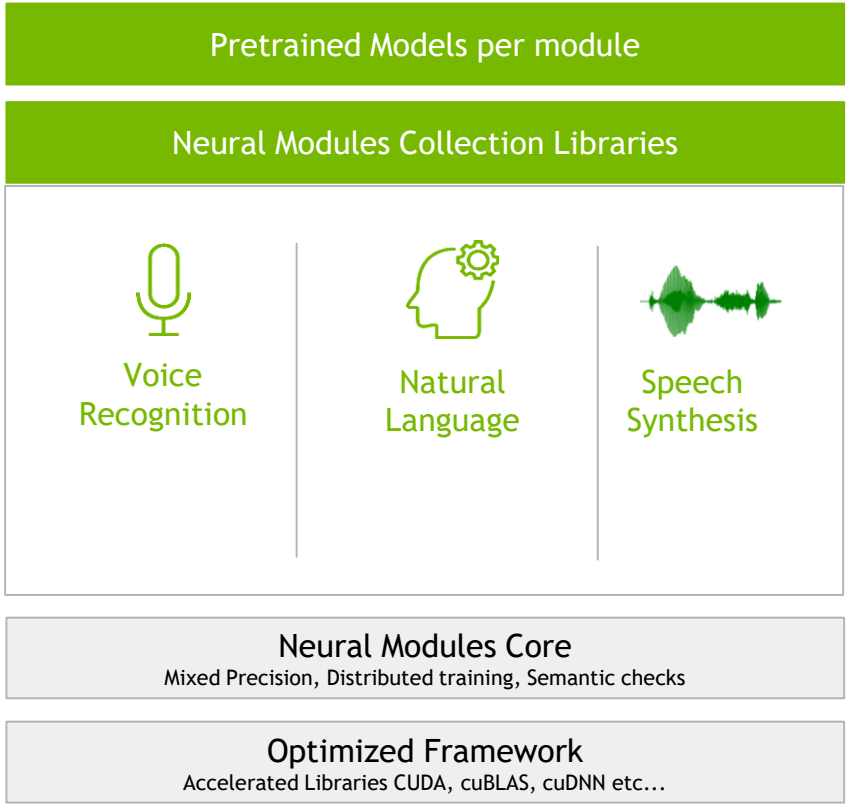
CONVERSATIONAL AI WORKFLOW



CONVERSATIONAL AI WORKFLOW



NEMO: TRAINING CONVERSATIONAL AI MODELS



- Open source deep learning Python toolkit for training speech and language models
- High performance training on NVIDIA GPUs
  - Uses TensorCores
  - Multi-GPU
  - Multi-Node
- Based on concept of Neural Module – reusable high level building block for defining deep learning models
- PyTorch backend (TensorFlow on Roadmap)

NEMO COLLECTIONS

pip install nemo_asr

nemo_asr
(Speech Recognition)

- Jasper acoustic model
- QuartzNet acoustic model
- RNN with attention
- Transformer-based
- English and Mandarin tokenizers and dataset importers

pip install nemo_nlp

nemo_nlp
(Natural Lang Processing)

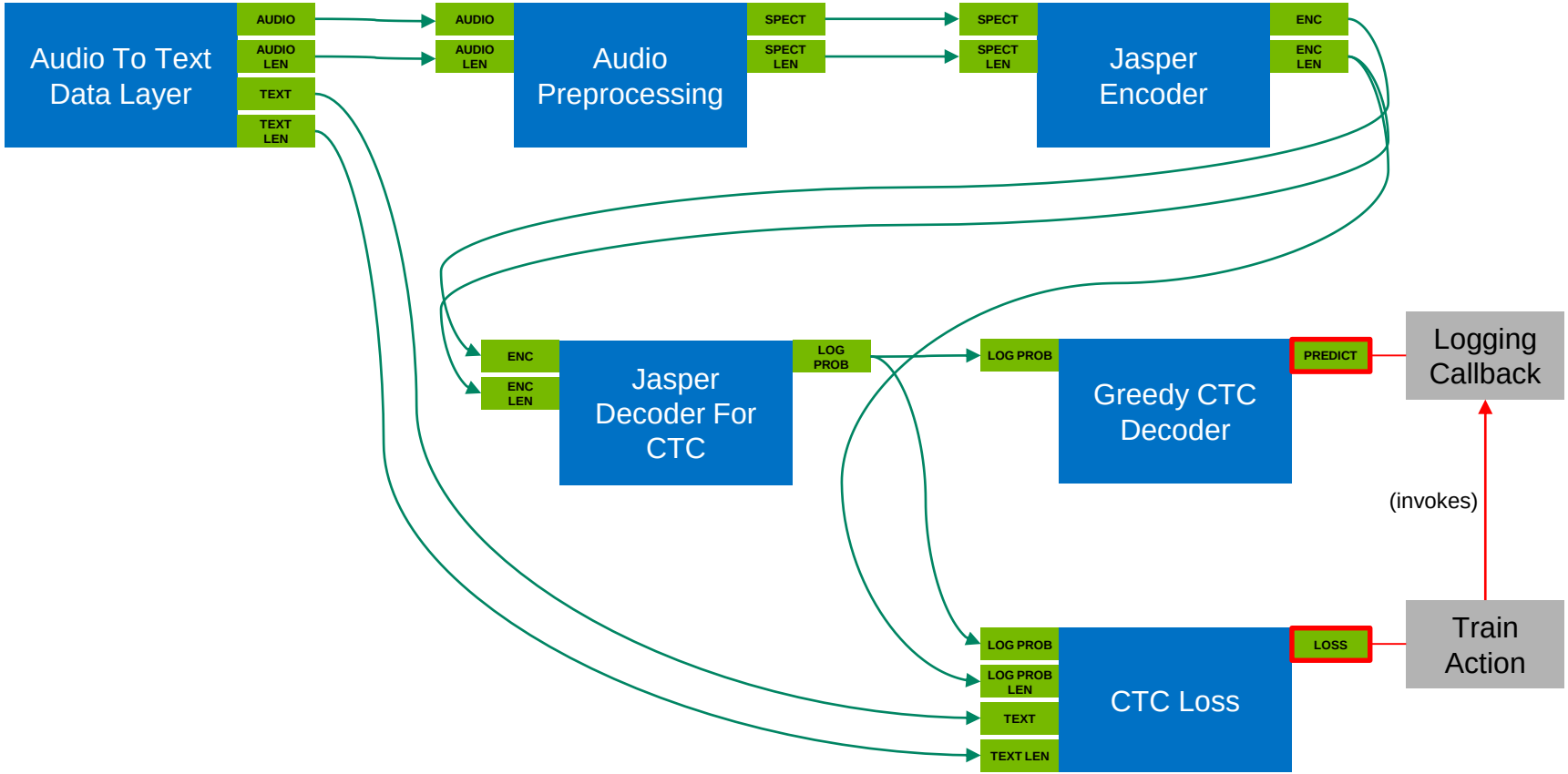
- BERT pre-training & finetuning
- GLUE tasks
- Language modeling
- Neural Machine Translation
- Intent classification & slot filling
- ASR spell correction
- Punctuation
- English and Mandarin dataset importers

pip install nemo_tts

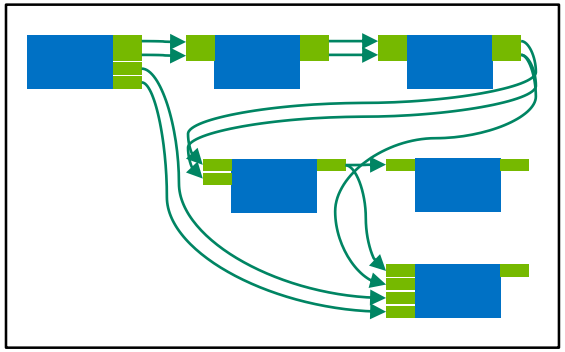
nemo_tts
(Speech Synthesis)

- Tacotron 2
- WaveGlow
- English and Mandarin output and datasets importers

NEMO EXAMPLE: JASPER ASR



NEMO EXAMPLE: JASPER ASR



```python
import nemo
import nemo_asr
```

```python
data_layer = nemo_asr.AudioToTextDataLayer(...)
data_preprocessor = nemo_asr.AudioPreprocessing()
jasper_encoder = nemo_asr.JasperEncoder(...)
```

```python
jasper_decoder = nemo_asr.JasperDecoderForCTC(...)
ctc_loss = nemo_asr.CTCLossNM(...)
greedy_decoder = nemo_asr.GreedyCTCDecoder()
```

Create modules

```python
audio, audio_len, text, text_len = data_layer()
hist, hist_len = data_preprocessor(audio, audio_len)
encoded, encoded_len = jasper_encoder(hist, hist_len)
```

```python
log_probs = jasper_decoder(encoder_output=encoded)
predictions = greedy_decoder(log_probs=log_probs)
loss = ctc_loss(log_probs, targets=text, encoded_len, text_len)
```

Connect them

```python
from nemo_asr.helpers import monitor_asr_train_progress
train_callback = nemo.core.SimpleLossLoggerCallback(
    tensors=[loss, predictions, transcript, transcript_len],
    print_func=lambda x: monitor_asr_train_progress(x, labels=labels))

neural_factory = nemo.core.NeuralModuleFactory()
neural_factory.train(tensors_to_optimize=[loss],
                     callbacks=[train_callback],
                     optimizer="novograd",
                     optimization_params={ "lr": 1e-2})
```

Define training and evaluation actions

"Jasper: An End-to-End Convolutional Neural Acoustic Model" by Li et al. INTERSPEECH 2019 https://arxiv.org/pdf/1904.03288.pdf

ASR COMPARISONS

English LibriSpeech dataset %WER

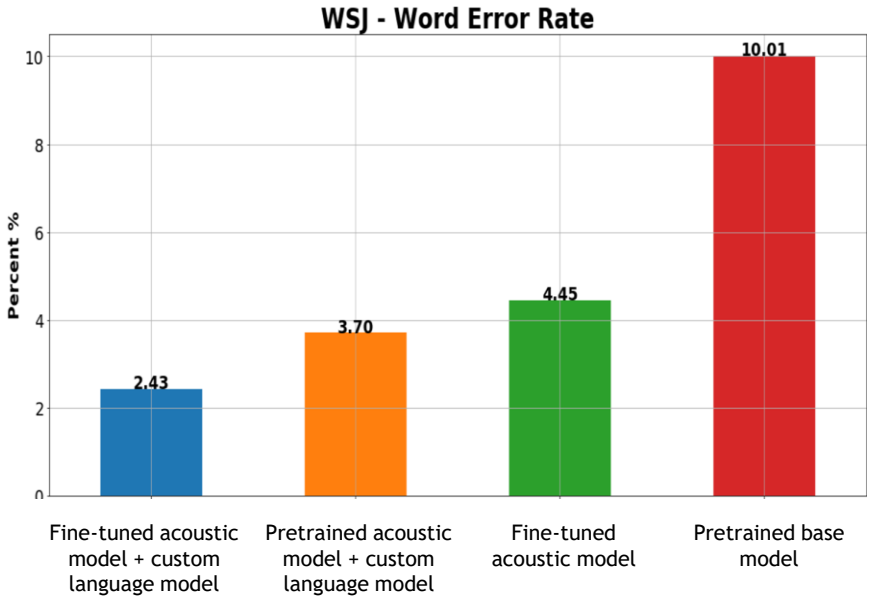
Model	Language Model	Test-Clean	Test-Other	Params, M
DeepSpeech 2	5-gram	5.33	13.25	>70
wav2letter++	ConvLM	3.26	10.47	208
Listen-Attend-Spell (with SpecAugment)	RNN	2.5	5.8	360
Jasper 10x5	-	3.77	11.08	333
	6-gram	3.19	9.03	
	Transformer-XL	2.86	8.17	
QuartzNet 15x5	-	3.90	11.28	19
	6-gram	2.96	8.07	
	Transformer-XL	2.69	7.25	

DOMAIN SPECIFIC ASR

Jupyter notebook transfer learning tutorial

- Start with pretrained base QuartzNet model
- Fine tune with WSJ data (newspaper read aloud)
- Add custom language model to base model
- Add custom language model to fine-tuned model for best performance

- **Achieves Word Error Rate of < 2.5 !**



Tutorial here: https://ngc.nvidia.com/catalog/containers/nvidia:nemo_asr_app_img

TRANSFER LEARNING CUSTOMER STORY

KENSHO

An S&P Global Company

NVIDIA Corporation (NasdaqGS:NVDA)
Earnings Call Transcript
Thursday, August 15, 2019 5:30 PM

S&P Global

Executives
Colette M. Kress - Executive VP & CFO
Jen-Hsun Huang - Co-Founder, CEO, President & Director
Simona Jankowski - Vice President of Investor Relations

Analysts
Aaron Christopher Rakers - Wells Fargo Securities, LLC, Research Division
Christopher James Muse - Evercore ISI Institutional Equities, Research Division
Harlan Sur - JP Morgan Chase & Co, Research Division
Joseph Lawrence Moore - Morgan Stanley, Research Division
Matthew D. Ramsay - Cowen and Company, LLC, Research Division
Stacy Aaron Rasgon - Sanford C. Bernstein & Co., LLC., Research Division
Timothy Michael Arcuri - UBS Investment Bank, Research Division
Toshiya Hari - Goldman Sachs Group Inc., Research Division
Vivek Arya - BofA Merrill Lynch, Research Division

Presentation

Operator
Good afternoon. My name is Christina, and I will be your conference operator today. Welcome to NVIDIA's financial results conference call. [Operator Instructions]

I'll now turn the call over to Simona Jankowski from Investor Relations to begin your conference.

Simona Jankowski
Thank you. Good afternoon, everyone, and welcome to NVIDIA's Conference Call for the Second Quarter of Fiscal 2020. With me on the call today from NVIDIA are Jen-Hsun Huang, President and Chief Executive Officer; and Colette Kress, Executive Vice President and Ch

I'd like to remind you that our call is being webcast live on NVIDIA's Investor Relations website. financial results for the third quarter of fiscal 2020. The content of today's call is NVIDIA's proper

l the conference call to discuss our thout our prior written consent.

During this call, we may make forward-looking statements based on current expectations. These are subject to a number of significant risks and uncertainties, and our actual results may differ materially. For a discussion of factors that could affect our future financial results and business, please refer to the disclosure in today's earnings release, our

Global Scope

Speakers

Meta Tags

Financial Jargon

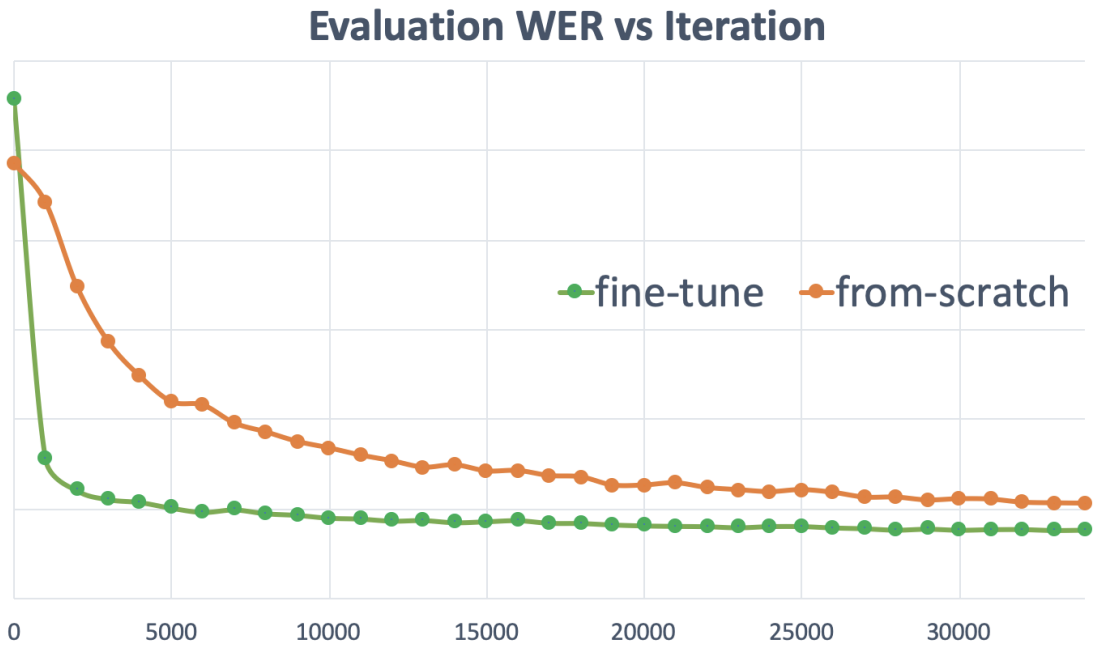
Domain Experts

External Knowledge

- S&P Global produces transcriptions of earnings calls – 10,000 hours of high quality data
- Scribe application works with ASR models
- Recognizes domain specific financial jargon
- Additional language models provide meta tags, punctuation

GTC Talk: https://events.rainfocus.com/widget/nvidia/gtcdc19/catalog-short?search=nemo

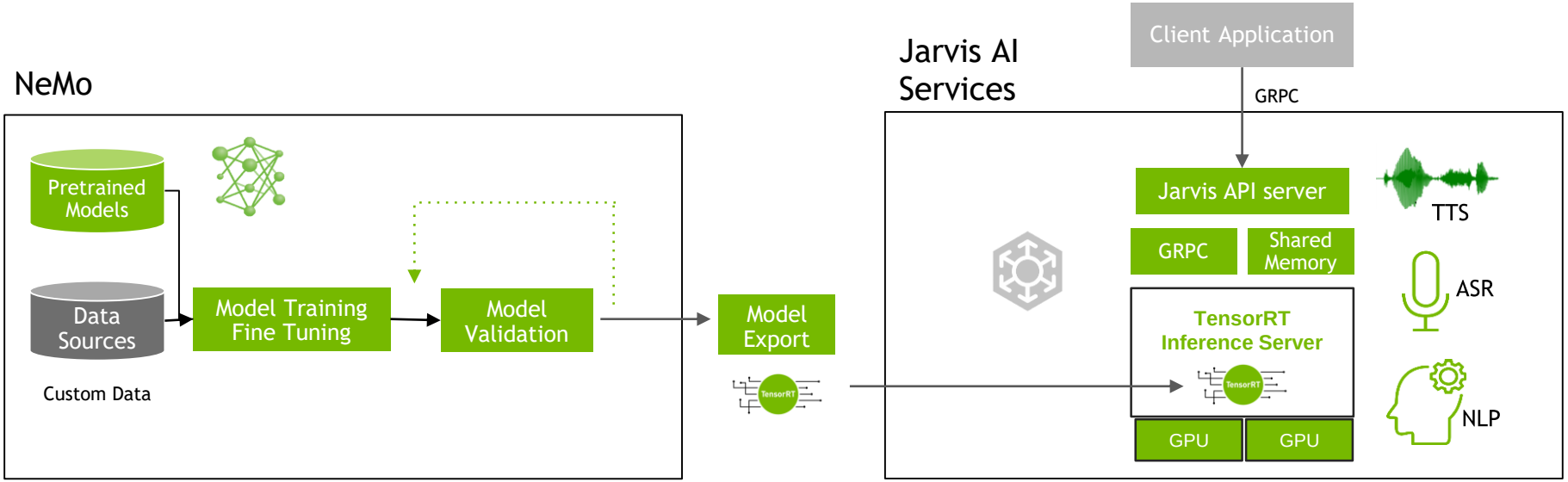
KENSHO ASR RESULTS



- QuartzNet trained on domain specific financial data outperformed all leading ASR models
- Fine tuning was faster and had higher accuracy than training from scratch

JARVIS AND NEMO TOGETHER

EXPORT TO JARVIS



DEPLOYING JARVIS AI SERVICES TO K8S

One Helm Chart to Deploy

Stage 1: Create TensorRT engines

A. Get PyTorch checkpoints
   - Download from NGC or use local checkpoint

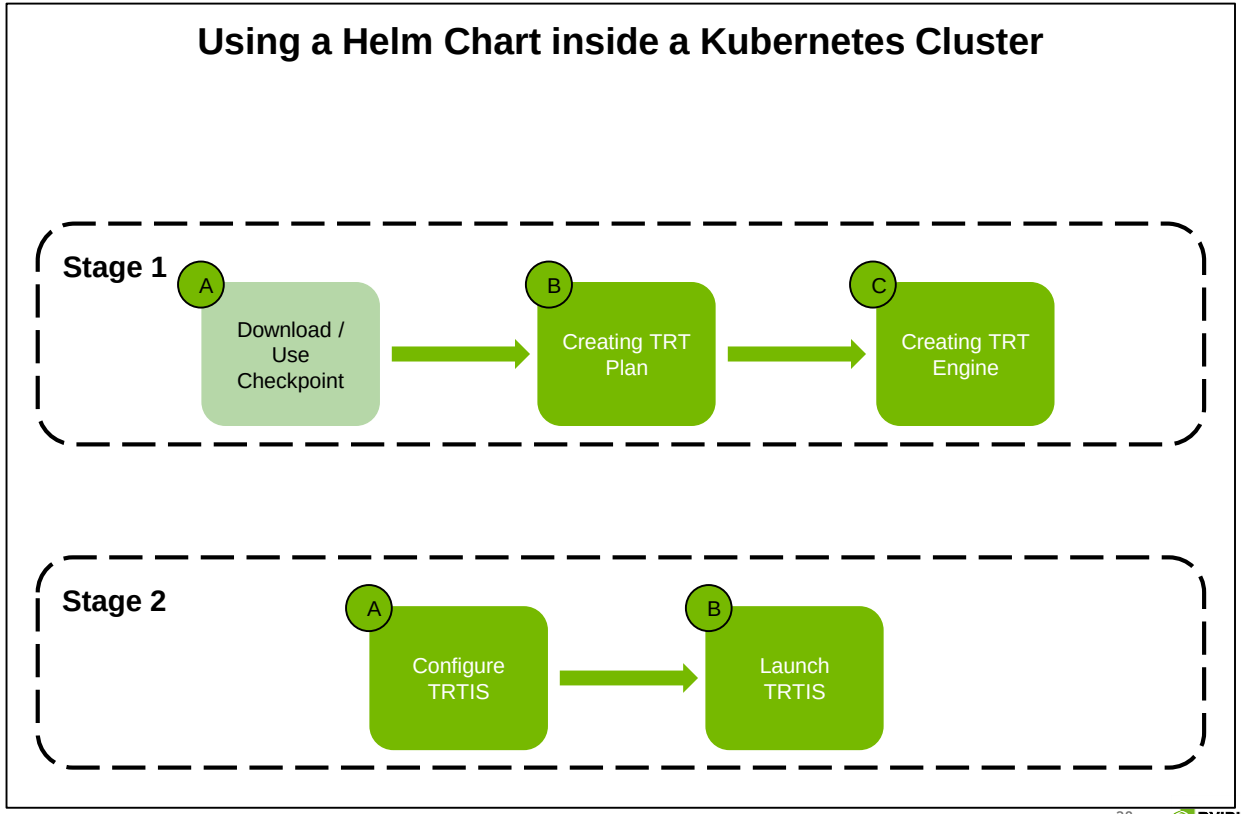
B. Create TRT plans
   - Convert PyTorch checkpoint to TRT

C. Create TRT engines
   - Generate optimized engines

Stage 2: Launch TRTIS

A. Configuring TRTIS
   a. Setup model directory structure and server configuration

B. Launching TRTIS



JARVIS PARTNERSHIP

BENEFITS & EXPECTATIONS

Partnership Benefits

Cutting-edge Technology

- Opportunity to collaborate with NV Eng to accelerate development

Performance

- SW optimizes on GPUs by design
- Prioritization of feature requests

Capability Growth

- Proof-of-concept to solve more challenges

Collaboration

Try and Implement

- API on preferred model/data
- Validation of performance metrics

Help us improve

- Provide Feedback
- File bugs and issues
- Feature requests

Success together

- Your business growth
- Testimonial & Success case study

Try NeMo today!

https://nvidia.github.io/NeMo/zh/index.html

```
> pip install nemo_toolkit
nemo_asr nemo_nlp nemo_tts
```

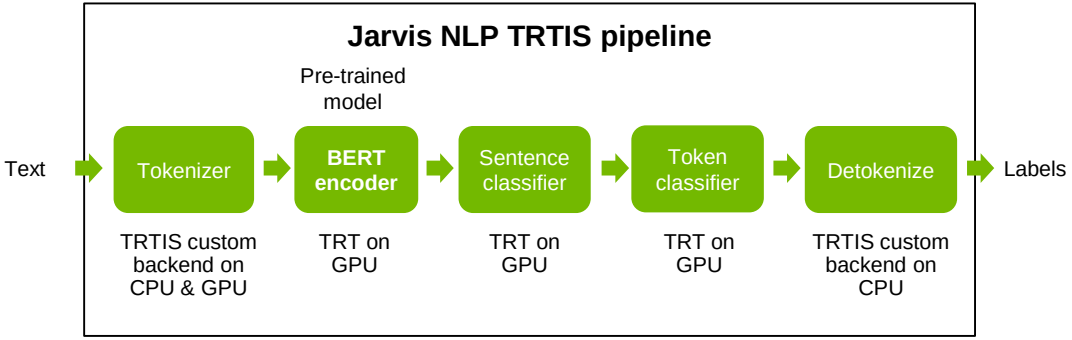
Register for Jarvis Early Access (January)

https://developer.nvidia.com/nvidia-jarvis


NVIDIA®

Backup

Jarvis NLP Service



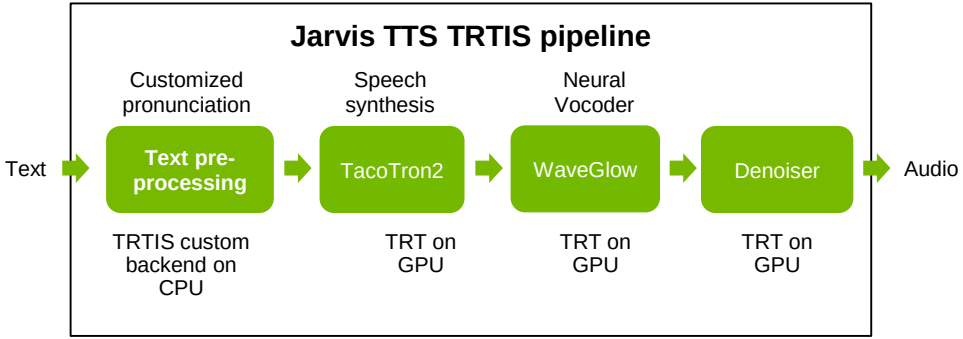
Jarvis NLP API

Method Name	Description
ClassifyText	Given text input, return a class label and score
ClassifyTokens	Given text input or array of tokens, return a class label and score per token
TransformText	Given input text, return output text

Jarvis NLP Provided Models

Method Name	Description
AnalyzeSentiment	Run sentiment detection on input and return label/score
AnalyzeEntities	Given text input, return named entities (NER)
Punctuate	Take text without punctuation (e.g. ASR output) and add periods, commas, question marks

Jarvis TTS Service



Jarvis TTS API

Method Name	Description
Synthesize	Given text input, return audio of spoken version as a single audio clip
SynthesizeOnline	Given text input, return audio of spoken version as an audio stream