Infrastructure Benchmarking Study for Quick Service Restaurants to implement AI-powered Customer experiences.

Abstract

DaveAI conducted a benchmarking study of small form factor hardware devices that manage AI workloads deployed at the edge. The hardware under test included a variety of devices manufactured by Intel, Advanced Micro Devices (AMD) and Nvidia.

The generated workloads involved the use of a DaveAI software implementation deployed as a quick service restaurant (QSR) self-service kiosk. The recent impact of COVID-19 increased the demand for and adoption of QSR kiosks, making the implementation a relevant choice for sample workloads. The software's components include automated speech recognition (ASR), natural language processing (NLP), computer vision (CV), an AI Affinity Engine, and DaveAI's Virtual Avatar, developed to decrease latency and provide a rich graphical experience.

This hardware and vendor-agnostic report contains test plans, methodologies, frameworks, and results. It also describes environment setup and compares hardware details, such as processor specification and thermal capacity.

The tests were conducted in two parts.

- In Phase 1(results published on 2nd August 2022), the benchmarking study was conducted between Intel and AMD devices. The AI models used were standard DaveAI models (not optimized for OpenVINO) based on Libri Speech and Aspire Chain. In Phase 2, the benchmarking study was further extended to Nvidia devices.
- In Phase 2(published on 07th Nov 2022), the AI models utilized for ASR were based on QuartzNet optimized using Intel OpenVINO toolkit for Intel devices and NVIDIA NEMO for Nvidia devices. Results are also published for Intel devices comparing between regular and OpenVINO Optimized models.
- In both Phase1 and Phase 2, the NLP models were used are DaveAI Proprietary models.

Key Findings

Benchmarking tests involved three setup profiles of increasing software complexity, as shown in Table 1. For terminology and software components, see Table 3 and Table 4.

Table 1: Test Profiles

Profile	Software Components

Basic Profile	Avatar, Speech (WAKE and ASR), NLP, Ordering App
Intermediate Profile	Avatar, Speech (WAKE and ASR), NLP, Ordering App, Face Detection
Comprehensive Profile	Avatar, Speech (WAKE and ASR), NLP, Ordering App, Face detection, Vehicle recognition

At a high level, the results per profile suggest:

- **Basic:** A single Intel device is sufficient to handle the entire workload of the audio pipeline of a QSR kiosk (ASR + NLP + Avatar), making it a very cost-effective option. With an end-to-end audio response time of 2.2 seconds and high graphic performance(30FPS), the Intel i7-1165G7 still had more than 70% CPU headroom for other tasks.
  Basic profiles with only ASR and NLP workloads (Voice enabled kiosks), the Intel i5-1135G7 running OpenVINO optimized AI models performed the best, with a response time of 1.04 seconds.
- **Intermediate:** A pure edge setup with multiple Intel devices fared better than a hybrid or cloud setup. The edge setup demonstrated a faster (than Hybrid or cloud) audio pipeline response and inference speed for image processing.
- **Comprehensive:** The hybrid setup demonstrated best performance with a superior end-to-end response time of 4.1 seconds in Intel i5-11 device.

Hardware

The QSR use case's space requirements favored small devices that run on mobile processors with minimum (thermal design power) TDP.

The most important criteria for choosing devices for the benchmarking tests were:

- **Specification Similarities:** The devices chosen converged on similar specification details, such as clock speed and number of cores.
- **Small Form Factors:** A mini-PC form factor or laptop fit the QSR use case.
- **Unit Price and Availability in India:** The benchmarking effort required easily procured, inexpensive devices.

DaveAI chose two Intel® NUC Mini PCs with 11th Generation Intel® Core™ Processors and two comparable AMD devices in Phase 1. For optimized models, two comparable Nvidia devices were considered. For an in-depth comparison of device specifications, see Table 2.

Table 2. Hardware Device Specification Comparison

Spec list	**Hardware devices**					
Processor	**Intel i5-1135G7**	**Intel i7-1165G7**	**AMD Ryzen 3 4300U**	**AMD Ryzen 7 4700U**	**Jetson AGX***	**Jetson NX***
Generation	11th	11th	4th	4th	8th	8th
Clock Speed	2.40 GHz	2.80GHz	Min: 1400MHz Max: 2700MHz	Min: 1400MHz Max: 2000MHz	Max: 2265MHz	Max: 1100MHz
CPU(s)	8	8	4	8	8	6
Thread(s) per core	2	2	1	1	1	1
Core(s) per socket	4	4	4	8	2	2
Max TDP	15W	15W	15W	15W	>30W	20W
Memory	16GB	16GB	8GB	8GB	16GB	16GB
Hard-disk Type	SSD	SSD	SSD	SSD	SSD	SSD
L1d cache	48K	48K	32K	32K	512K	384K
L1i cache	32K	32K	32K	32K	1M	768K
L2 cache	1280K	1280K	512K	512K	8M	8M
L3 cache	8192K	12288K	4096K	4096K	4M	4M
Operating System	Ubuntu 18.04	Ubuntu 18.04	Ubuntu 18.04	Ubuntu 18.04	Ubuntu 18.04	Ubuntu 18.04

Software

DaveAI QSR Self-service Kiosk Software

DaveAI chose to conduct the benchmarking with a QSR self-service kiosk composed of several modular software components. See terminology related to component descriptions in Table 3 and component descriptions of the QSR kiosk software in Table 4.

Table 3. Terminology

Term	Description
Automatic Speech Recognition (ASR)	Component that recognizes spoken natural language, processes it, and converts it to text
Natural Language Processing (NLP)	Component that classifies speech based on intent and entity and suggests a response
Wake-up Words or Wake Words	List of words that trigger the kiosk to expect conversational speech
Intent	Actions a user wants to accomplish
Entity	Modifier words which change the user's intent
Utterance	Anything a user says (i.e., natural language)

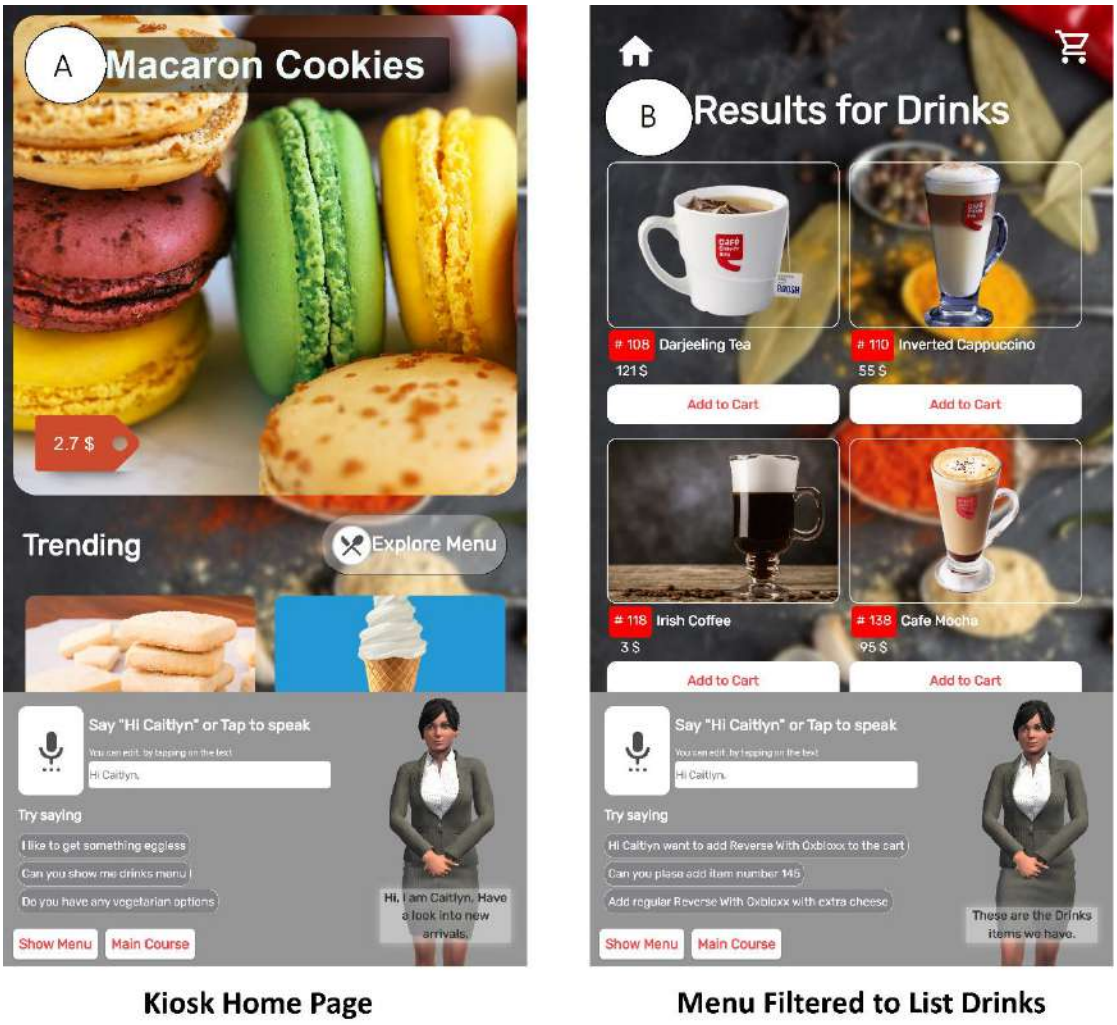
Table 4. Software Component Descriptions

Component	Description	Note
Avatar + Digital Signage	Web-based application using the WebGL interface	Runs on Google Chrome (WebGL-3D)
ASR	Component that recognizes speech as audio and converts it to text	Employs a model customized for restaurant use
Wake-up (WAKE)	Mini version of the ASR and a mic that listens to audio for a wake word	Triggers larger ASR component to listen for conversational speech
NLP + Ordering App	Component which predicts user input from keywords and suggests a response	Uses intent and entity classification
Image Processing (IMAGE)	Group of image processing components for detection and recognition	Uses face detection and vehicle and vehicle number recognition

Place an Order – A Quick Look

The numbered steps below describe the QSR use case implemented for this benchmarking test. The steps represent a simplified, idealized description of the customer and kiosk interactions. To see a more detailed list of use case variations, see Place an Order – Details and Variations.

1. Figure 1 shows the kiosk home page (A) in the idle state, waiting for the arrival of a customer.



Kiosk Home Page **Menu Filtered to List Drinks**

Figure 1. User Interface: Home Page and Drink Page

2. The customer approaches the kiosk in a vehicle.
3. Vehicle plate number recognition and facial detection activate the kiosk to start a greeting exchange with the customer.
4. The customer indicates menu choices verbally (e.g., “I’d like an order of the Macaron Cookies and two mocha coffees.”) or uses the touch screen, navigating various filter options, such as Trending (A) and Results for Drinks (B).
5. Items are added to the shopping cart with the Add to Cart (B) button.

6. Finally, as shown in Figure 2, the customer views the shopping cart (C), and on the Review Page, makes modifications to finalize the order.

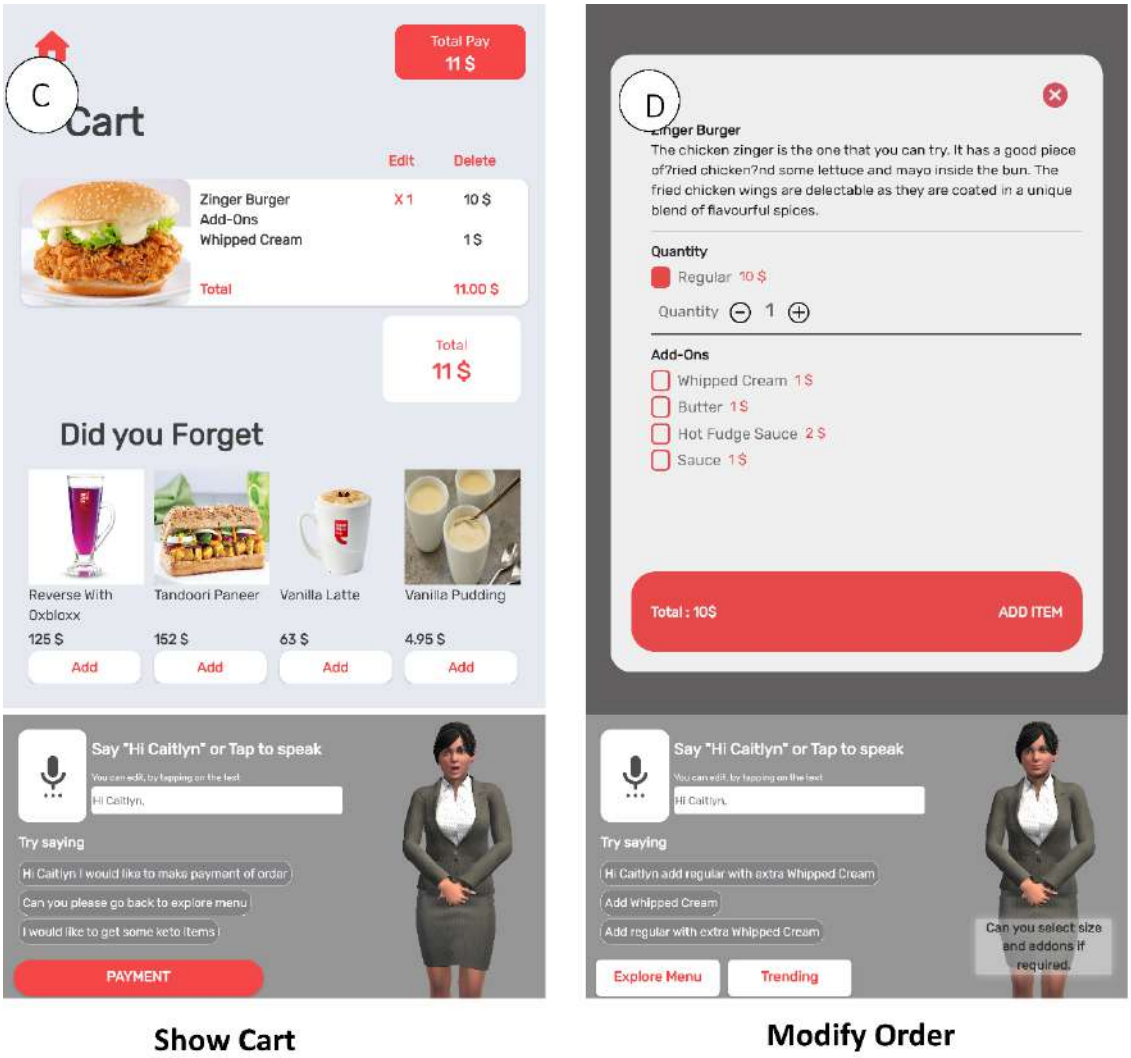
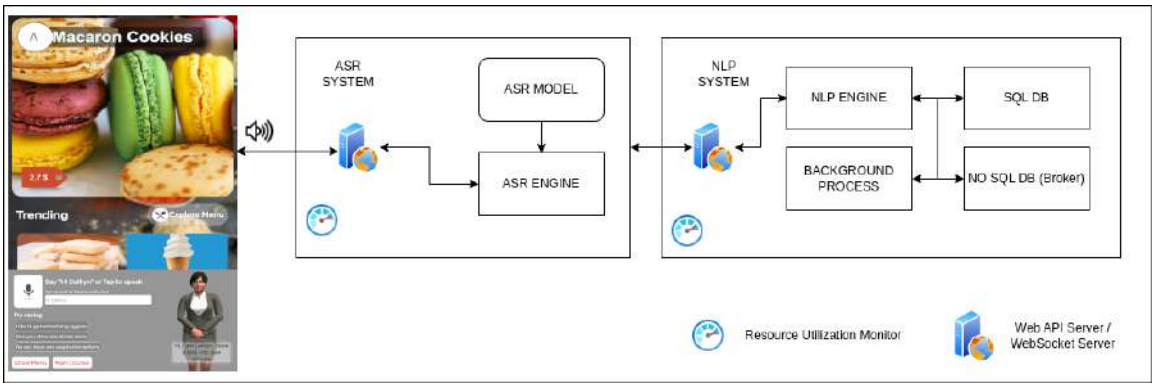


Figure 2. User Interface: Shopping Cart and Modify Order Pages

7. Customers select method of payment and complete the transaction.

Test Framework



Test Framework Architecture Diagram.

DaveAI developed a testing framework in Python to benchmark the QSR implementation's ASR, NLP, and Image Processing components. Download the framework from the repository [DaveAI to provide link]. The repository's README file contains instructions and example usage.

The test framework uses the python tools and packages listed in Table 5.

Table 5. Python Dependencies

Python dependency	**Description**
Locust	Load testing websites with concurrent users
Python-Socketio	Bidirectional communication between clients using Socket.IO protocol
Psutil	System usage metrics including CPU and memory
Jiwer	Word error rate (WER) measurement
Pandas	Data analysis

The test framework features:

1. System resource and component metrics
2. Support for single and concurrent users at different spawn rates
3. Automatic download of test cases per domain selection
4. Report generation, including a summary and detailed analysis

Methodology

Test Plan

Testing measured the performance of individual software components and combinations of software components on each hardware device.

Table 6 and Table 7 list tests conducted on single components and component combinations along with the data types and streams involved. All software components are implemented as containers.

Table 6. Single Component Tests

Test	Software component Tested	Data type	Test description	Component behavior
1	WAKE	Audio File	Test occurs on audio files containing: • wake-up words • non-wake-up words	WAKE only responds if the audio contains wake-up words.
2	ASR	Audio File	Test occurs on audio files streamed as: • whole files • file broken into audio chunks, with default chunk size of two seconds	Every two seconds an audio chunk will be streamed to the ASR for recognition.
3	NLP	Text Data through an API	Test occurs on the speech to text data conversions from the ASR. Test occurs on the text data that are collected for the benchmarking.	NLP predicts intent and entities.
4	IMAGE	Video Frames	Two-channel tests occur simultaneously: • face detection • vehicle number recognition	By default, the system will send a single frame/second.

Table 7 lists the types of testing conducted on combinations of components. These tests build in complexity, testing multiple components and data streams in tests 2 through 4 simultaneously.

Table 7. Multiple Component Tests

Test	Component(s) tested	Data type	Test description
1	WAKE + ASR	Audio File	Test first starts with WAKE. Audio data that contain wake up word will be sent to ASR for recognition.
2	ASR + NLP	Audio File	Test start with ASR for recognition. The recognized text will be sent to NLP for prediction.
3	WAKE + ASR + NLP	Audio File	Test start with WAKE. Audio data that contain wake up word will be sent to ASR for recognition. Recognized text will be sent to NLP for prediction.

4	WAKE + ASR + NLP + IMAGE	Audio File and Video Frames	Test will run simultaneously for both audio-pipeline (WAKE + ASR + NLP) and image pipeline.

Number of Samples

Data samples used for testing and training included:

- Audio
- Text
- Video
- Image

Table 8 summarizes the test data samples used. Samples were chosen from large sample file sets. For example, 2000 conversational text data samples were collected and 200 were selected as the final test set. All data are stored in the DaveAI cloud and are downloaded during testing.

Table 8:Test Data Type

Data type	Description	Number of samples	Sample notes
Audio	Audio files of different lengths	1500	LENGTH Longest: 6.9 seconds Shortest: .98 seconds NOISE PROFILE :12dB
Text	Conversational data samples	200	Files were manually annotated with expected intent, entities, and responses.
Image	Images of faces and vehicle plates	Face: 200 Vehicle Plate: 200	Files were manually annotated.

NOTE: The data selected for benchmarking tests are not used in the training of in-house models.

Metrics

Common Speech Recognition Metrics

Word Error Rate (WER)

Word error rate (WER) is a common performance metric of speech recognition systems. The WER *w* for a given sample of natural language speech, an utterance, can be calculated with this formula:

$$w = (n + i + d)/s,$$

where

- *n* equals the total number of word substitutions

- *i* equals the total number of word insertions
- d equals the total number of word deletions
- s equals the total number or words in the utterance

Table 9. Variables in WER Formula with Examples

Equation variable	**Definition**	**Natural language utterance**	**Recognized speech**
Substitutions (*n*)	Total number of words in the utterance that have been replaced	Do you have red sauce and pasta?	Do you have a rare lhasa apsa?
Insertions (*i*)	Total number of words in the utterance that have been added	Can I get a soda with lime?	And can I get a soda with lime?
Deletions (*d*)	Total number of words in the utterance that have been removed	Can you list the salad dressings?	Can you list the dressings?

The benchmarking calculated both the overall WER value**,** as defined by the formula above, and a weighted average WER, which normalizes the data to account for the different number of words in utterances.

Table 10 presents example text with WER calculations. An overall WER of 0 indicates the model performed well, and a value of greater than 0.5 indicates the model performed poorly.

Table 10. Text Examples with WER Calculations

Natural language utterance	**Recognized speech**	**Overall WER**	**Weight**
What's special today?	What's special today	0	3

Is there anything on offer today?	Is that anything I'm all for today	0.67	6
Show me the variety of offers I can get today.	Show me debate the awful flows I can get today	0.4	10
Can I pay cash on delivery?	And they paid cash on delivery	0.5	6

Accuracy

The NLP container predicts the intent and the entity of an utterance. For a single conversation, the prediction accuracy is assigned a value of

- 1: Indicates intent and all entities are predicted correctly
- 0: Indicates intent prediction is incorrect

If the intent prediction is correct but some entities were predicted incorrectly, the accuracy of the NLP c is calculated with a simple ratio of:

$$c = a/b,$$

where a equal the number of correctly predicted entities and b equals the total number of entities.

Average Response Time

The framework measured various response times, such as the duration between API calls, and calculated the averages. An average response time is calculated as the average of all the measured durations that occurred during benchmarking.

1. **Average response time:** The response time in seconds is measured as the duration between one event type (e.g., audio sent) and another event type (e.g., recognition audio has been received).

Component	**|---Response measurements between two events ---| Measured in seconds**	
	Event one	**Event two**
ASR – single audio file	Audio sent	Recognition received
ASR – streamed as chunks	First chunk sent	Final recognition
NLP	API Call	API Call
ASR + NLP	Data sent to ASR	Response received from NLP

NOTE: For both NLP and ASR+NLP, text-to- speech (TTS) conversion occurs for each response, and this is included in the average response time calculation.

2. **Average response time per second audio:** In ASR (non-streaming), the duration of audio files is not the same for each audio data. To understand the response time in seconds level, the response time per second is calculated for each audio data and averaged for a single benchmarking session.
3. **Average first recognition time:** In ASR (streaming), the ASR device will respond to every audio chunk it receives. Single audio contains multiple responses. First recognition is calculated as the duration between the first response received and the first chunk sent to the ASR.
4. **Average final recognition time:** Like the *average first recognition time*, the final recognition time is calculated as the duration between the last response received and the last chunk sent to the ASR.

Device Metrics

The test framework collects device metrics, CPU and memory data for each second of testing to calculate performance averages and determine peak usage (maximum):

1. **CPU Utilization**:
   a. Average CPU utilization (Percentage)
   b. Maximum CPU utilization (Percentage)
2. **Memory Utilization**: The metrics for this data are calculated in both percentage and actual memory usage.
   a. Average memory utilization (Percentage)
   b. Average memory usage (MB or GB)
   c. Maximum memory utilization (Percentage)
   d. Maximum memory usage (MB or GB)

Usability Score

Usability ratings are calculated from three measurements:

1. Response time for audio pipeline
2. Inference speed for image processing
3. Frames per second (FPS) for avatar display

Ratings were low, medium, and high. See the Table 9 for details.

Table 11: Usability Rating

Usability	Description	Audio pipeline response time	Image processing inference speed	Avatar display FPS	Note

LOW: Poor User Experience	A user will struggle to complete an order.	Greater than 7 seconds	Greater than 2 seconds	Less than 10	A user will face the lag in the avatar and the overall response will be high.
MEDIUM: Average User Experience.	A user can still complete his/her food order with a slight lag.	5-7 seconds	1-2 seconds	Range of 20-30 FPS	When the load is high, a user might face a slight lag in avatar or a slight delay in response intermittently.
HIGH: Good User Experience	A user can order food without any lag.	Less than 5 seconds	Less than 1 second	Greater than 30 FPS	The avatar will keep pace with conversation.

Component Metrics

To test WAKE, DaveAI fine-tuned the ASR model for QSR kiosks. The list below describes component-level testing. See Table 13 for results.

1. **WAKE**

   Data Samples: 200 audio samples

   Method: Accuracy of WAKE's recognition of the wake-up word.

   NOTE: After tuning the model, tests were conducted using audio samples, with and without wake-up words.

2. **ASR**

   Data Samples: 450 audio samples

   Method: WER of ASR.

   NOTE: Data were sampled at 16KHz. The signal-to-noise ratio (SNR) of samples varied from clean to noisy (See Table 12). The noisy data set consisted of samples that had one or more of these issues:

   a. low volume
   b. non-standard pronunciation of the words
   c. truncation at the beginning or end of words
   d. poor recording quality

3. **NLP + Ordering App**

   Data Sample: 200 text samples

   Method: Accuracy of NLP's predictions of intent and entities.

NOTE: One conversation request means a client sent a request and a response was received from the NLP server.

4. **ASR + NLP + Ordering App**

   Data Sample: 450 audio samples + 200 text samples

   Method: For this compound test, both accuracy and WER were calculated.

5. **Image**

   Data Sample: 200 images

   Method: Accuracy in facial recognition.

   NOTE: Video frames were sent to both Face Detection and Vehicle Recognition classes simultaneously.

Table 12: Clean and Noisy Data Samples

Original utterance	**Recognized text** **Clean: SNR more than 20 db**	**Recognized text** **Noisy: SNR -6 db**
please get me old fashioned doughnut	please get me old fashioned doughnut	please get me old fashioned don't
please get me big crunch chicken cheeseburger	please get me a big crunch chicken cheeseburger	please get paid the chicken cheeseburger
please add hazelnut karat celebration cake to my order	please add hazelnut carrot celebration cake to my order	lease and hazelnut celebration cake to my order

NOTE: The current gold standard WER is between 0.1 to 0.2. The WER calculation usually depends on the type of dataset used for the calculation.

Table 13: Component Results

Component	**Data samples**	**Metric**	**Result**
WAKE	200 Audio	Accuracy	91%
ASR	450 Audio	WER	CLEAN: 0.117 NOISY: 0.35
NLP + Ordering App	200 Text	Accuracy in prediction of entities and intents	95%
ASR + NLP + Ordering App	450 Audio 200 Text	Accuracy and WER	CLEAN AUDIO Accuracy: 83% WER: 0.11

			NOISY AUDIO Accuracy: 75% WER: 0.39
Image Recognition (Face)	400 Images (Face and No Face)	Accuracy	95%

Google Cloud-based ASR vs. DaveAI Local Deployment

The cloud-based ASR from Google recognizes an extremely large vocabulary as input and responds roughly twice as fast as Edge deployments. Using available Google APIs, DaveAI tuned the Google ASR engine for better QSR use case accuracy.

Tests were conducted on Google (EN-IN) ASR and the in-house ASR model. For these tests, a test set of 490 audio files were used with a total of 1361.71 seconds of audio.

The list below summarized in Table 14: ASR Performance Comparison.

Table 14: ASR Performance Comparison, Intel Devices running Models that were not optimized for OpenVINO.

	Clean Audio		**Noisy Audio**	
	Local deployment (Intel i5-1135G7)	**Google**	**Local deployment (Intel i5-1135G7)**	**Google**
WER	0.11	0.08	0.39	0.26
Response time	0.8614	1.9863	1.0247	2.1847

Clean audio results were within a range of the gold standard WER with a faster response time. Google delivered good WER but with a longer response time.

Achieving a better WER is always challenging for noisy data. In adverse conditions, DaveAI's in-house ASR achieved a WER of 0.39 with a shorter response time.

Basic Profile Benchmarking Results

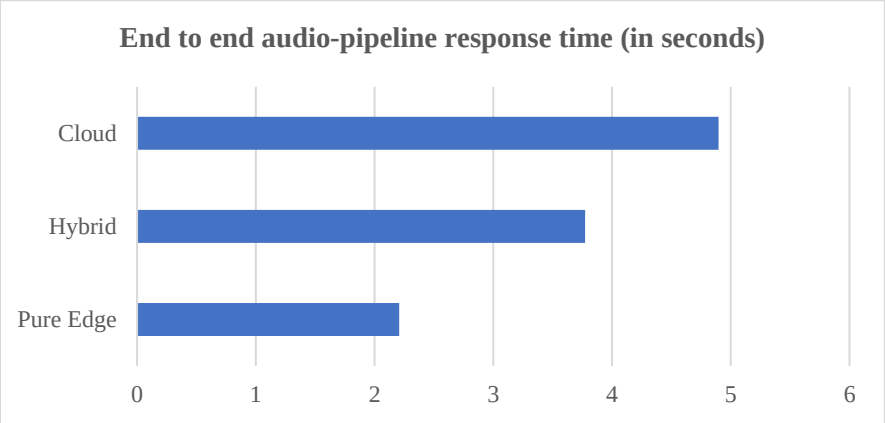
Components: Avatar, Speech (WAKE + ASR), NLP, Ordering App

The basic profile contains the components listed above. The audio pipeline uses a 3D interactive human avatar as the front-end display. In this profile, tests ran on both Intel and AMD devices. Speech is a combination of the Wake-UP component and ASR component.

Summary and Recommendations

A single Intel device is sufficient to handle the entire workload of the audio pipeline in a pure edge deployment. The pure edge setup outperformed hybrid and cloud setups.

Table 15. Basic Setup: Overall Response Times



DaveAI recommends a pure-edge deployment with a single Intel i7-1165G7, a cost-effective option delivering good performance in use cases with Speech, NLP and Avatar workloads. Find a detailed analysis in the Annexure.

DaveAI recommends using OpenVINO optimized AI models in all deployments. This is from the evidence captured while benchmarking ASR and NLP workloads on Intel i5 devices, which outperformed other configurations.

For more recommendations, see Total Cost of Ownership (TCO) Analysis.

NOTE: The response time for the audio pipeline is less than 3 seconds, an acceptable industry standard for a satisfactory user interaction. The QSR self-service kiosk software uses conversational filler in the interactive responses. According to the sources listed below, the use of a conversational filler response times of up to 3 seconds shows no degradation in user perception with reasonable user engagement up to around 6 seconds.

1. How Quickly Should a Communication Robot Respond: Delaying Strategies and Habituation Effects
2. Towards reaction and response time metrics for real-world human-robot interaction

Details

Pure Edge Setup

DaveAI benchmarked with:

- devices connected locally through LAN or WLAN
- a single user and two concurrent users
- component-level tests and entire audio pipeline tests

A monitor was connected to the devices to display the avatar.

The results are presented in tabular form below along with observations about the performance.

ASR and NLP Loaded Separately, Single User

Table 16.1: Basic, Pure Edge Setup - ASR Loaded Separately, Single User, Models not optimized for OpenVINO, these numbers in the below table are the outcome of the benchmarking tests conducted during phase 1.

300 Seconds and Single Concurrent Session	**Intel i5-1135G7**	**AMD Ryzen 3 4300U**	**Intel i7-1165G7**	**AMD Ryzen 7 4700U**
Average CPU Utilization %	8.91	16.28	7.93	5.59
Average Memory Utilization	1.9GB	1.5GB	1.9GB	2.4GB
Average response time (in seconds)	1.5325	1.2261	1.3268	4.6044

Table 16.2: Basic Pure Edge Setup – ASR Loaded Separately, Single User (AI Models were updated & optimized to conduct the below tests for a fair comparison), these numbers in the below table are the outcome of the benchmarking tests conducted during phase 2

300 Seconds and Single Concurrent Session	**Intel i5-1135G7 (Standard - Non OpenVINO)**	**Intel i5-1135G7 (OpenVINO)**	**Jetson AGX***	**Jetson NX ***
Average CPU Utilization %	8.36	10.35	14.19	17.80
Average Memory Utilization	2.10GB	3.30GB	3.60GB	4.00GB
Average response time (in seconds)	0.24524	0.20654	0.25723	0.34521

Table 17: Basic, Pure Edge Setup - NLP Loaded Separately, Single User, Models not optimized for OpenVINO, these numbers in the below table are the outcome of the benchmarking tests conducted during phase 1.

300 Seconds and Single Concurrent Session	**Intel i5-1135G7**	**AMD Ryzen 3 4300U**	**Intel i7-1165G7**	**AMD Ryzen 7 4700U**
Average CPU Utilization %	18.02	18.304	6.28	7.3331
Average Memory Utilization	2.2GB	1.4GB	2.1GB	1.7GB

Average response time (in seconds)	0.9791	0.8539	0.8466	1.8767

Observations:

Faster Response Time: When either ASR or NLP was loaded, the Intel i7-1165G7 outperformed the AMD 4700U in response time.

In the same scenario, the Intel i5-1135G7 with OpenVINO Optimization outperformed the Nvidia NX and AGX. It was also observed that OpenVINO Optimization led to increase in performance in Intel devices.

The tests were repeated with both ASR and NLP workloads on a single device.

ASR and NLP Loaded, Single User

Table 18.1: Basic, Pure Edge Setup – ASR and NLP Both Loaded on Single Device, Single User, Models are not OpenVINO optimized, these numbers in the below table are the outcome of the benchmarking tests conducted during phase 1

300 Seconds and Single Concurrent Session	**Intel i5-1135G7**	**AMD Ryzen 3 4300U**	**Intel i7-1165G7**	**AMD Ryzen 7 4700U**
Average CPU utilization %	18.73	24.89	17.79	8.81
Average memory utilization	2.2GB	2.6GB	3.7GB	3.4GB
Average response time (in seconds)	1.9124	1.118	1.7601	2.9685

Table 18.2: Basic, Pure Edge Setup – ASR and NLP Both Loaded on Single Device, Single User, these numbers in the below table are the outcome of the benchmarking tests conducted during phase 2.

300 Seconds and Single Concurrent Session	**Intel i5-1135G7 (Standard - Non OpenVINO)**	**Intel i5-1135G7 (OpenVINO)**	**Jetson AGX**	**Jetson NX***
Average CPU utilization %	4.37	6.67	11.70	8.19
Average memory utilization	2.10GB	3.30GB	4.00GB	3.60GB
Average response time (in seconds)	1.10852	1.04151	1.04220	1.14274

ASR and NLP Loaded, Two Concurrent Users

Table 19: Basic, Pure Edge Setup – ASR and NLP Both Loaded on Single Device, Two Users, Models not optimized for OpenVINO, these numbers in the below table are the outcome of the benchmarking tests conducted during phase 1

300 Seconds and Two Concurrent Sessions	**Intel i5-1135G7**	**AMD Ryzen 3 4300U**	**Intel i7-1165G7**	**AMD Ryzen 7 4700U**
Average CPU utilization %	26.07	54.86	11.93	8.45
Average memory utilization	3.2GB	2.7GB	4.5GB	3.4GB
Average response time (in seconds)	3.1937	6.7587	2.1043	13.9032

Observations:

- **Both Components Loaded, Single User, Fastest Average Response Time:** Intel i7-1165G7 responded faster than the AMD 4700U.
- **Both Components Loaded, Single User, Fastest Average Response Time:** Performance of Intel i5-1135G7 running OpenVINO Optimized Models are faster than Nvidia NX and has the comparable performance as Nvidia AGX. It was also observed that OpenVINO Optimization led to increase in performance in Intel devices.
- **Both Components Loaded, Two Users, Best Overall Performance:** For 2 concurrent users, Intel devices performed better than the AMD counterparts.
-

Lastly, benchmarking was carried out for a pure edge setup that contained all audio pipeline components. A display monitor was connected to the devices to load Avatar + Digital Signage.

All Components Loaded, Single Concurrent Users

Table 20: Basic, Pure Edge Setup - All Components, Single Concurrent Users, Models not optimized for OpenVINO, these numbers in the below table are the outcome of the benchmarking tests conducted during phase 1

300 Seconds and Single Concurrent Session	**Intel i5-1135G7**	**AMD Ryzen 3 4300U**	**Intel i7-1165G7**	**AMD Ryzen 7 4700U***
Average CPU Utilization %	70.35	93.34	22.39	90.433
Average Memory Utilization	4.8GB	5.9GB	5.2GB	5.8GB
[Wake] Average response time (in seconds)	0.6220	0.7366	0.6414	0.647

[ASR+NLP] Average response time (in seconds) **End to-End Audio-Pipeline Response Time**	2.97	5.3703	2.2074	7.3737
Avatar FPS	30-35	25-30	30-35	20-25
Usability Score	High	Medium	High	Low

- Test results updated on 07th Nov 2022

Observations:

- **Audio pipeline Response Time:** Intel i7 1165G7 outperform AMD 4700U by 29% (5.1663s).
- **Complete Load:** The Intel devices exhibit more CPU headroom than the AMD devices.
- *Please note that benchmark tests for all components were not conducted in the Nvidia device ecosystem due to limitation in capturing GPU stats. Please follow this section for any updates.*

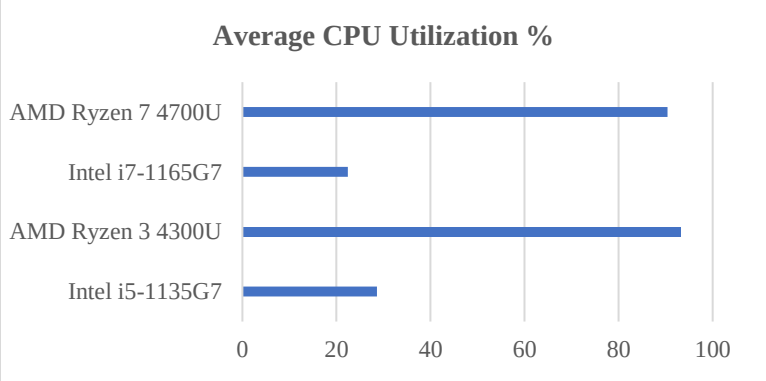


Figure 3. Basic Profile - Average CPU Utilization

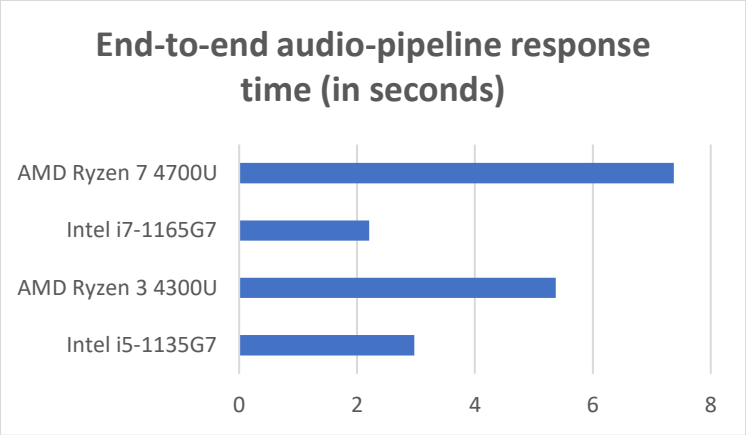


Figure 4. Basic Profile - Response Time

Hybrid Setup

In the hybrid setup, Avatar and WAKE components ran on the local devices, Intel i5-11 and AMD-4300U, at the edge. The remaining components ran in the cloud in AWS EC2.

NOTE: There was no resource restrictions in cloud devices. The cloud device is a higher specification device, containing:

- vCPU: 8
- RAM: 32GB
- SSD: 120GB.

Table 21: Basic Hybrid Setup

	Intel i5-1135G7	**AMD Ryzen 3 4300U**
[Local] Average CPU%	18.65	94.01
[Local] Average Memory	2.5GB	3.2GB
[Wake] Average response time (in seconds)	0.6289	0.6857
[ASR+NLP] Average response time (in seconds) **End to-End Audio-Pipeline Response Time**	3.7742	4.7098
Avatar FPS	30-35	25-30
Usability Score	High	Medium

Observations:

- **Hybrid Response Times, Slower Response Compare with Pure Edge:** The initial load occurs on the local device, while the rest of the workloads are processed in the cloud. A hybrid device took 70% more time to respond, at 1.5668 seconds, than the pure edge device.
- **CPU Utilization:** The AMD device utilized more than 90% CPU. The Intel device utilized less than 20% of the CPU, creating the opportunity for provisioning other tasks and workloads.

Cloud Setup

In the cloud setup, the avatar ran on a local Intel i3 device while all other components ran on the cloud instance AWS EC2.

NOTE: The cloud device is a higher specification device, containing:

- vCPU: 8
- RAM: 32GB
- SSD: 120GB.

NOTE: The i3-11 device resource utilization was not recorded as the avatar was the only device loaded.

Table 22. Basic Cloud Setup

	Intel i3 Device + AWS EC2
[Wake] Average response time (in seconds)	0.7678
[ASR+NLP] Average response time (in seconds) **End to-End Audio-Pipeline Response Time**	4.8967
Usability Score	Medium

Observations:

- **Delayed Response Time:** Most operations in this test happen in the cloud. In the cloud, the average response time of an audio pipeline is longer than the response time in hybrid setup.
- **Network Latency:** The end-to-end response time in hybrid and cloud setup depends on the network latency.

Intermediate Profile Benchmarking Results

Components: Avatar, Speech (WAKE + ASR), NLP, Ordering App, Facial Detection

This profile adds a facial detection component, which generates an additional image-based workload.

Summary and Recommendations

In the intermediate profile, while the response time in both audio pipeline and image inference this performance advantage is not significant enough to offset the price advantage in a pure edge setup. Hence for this Intermediate profile, a pure-edge setup with multi-device deployment is suggested for better performance and a high usability score.

See the response time for all setups below in Figure 5 and Figure 6.

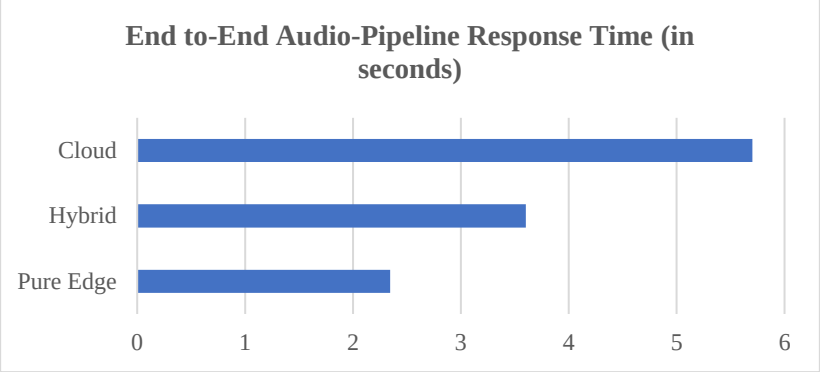


Figure 5: Intermediate – End to End Audio Pipeline Response Time

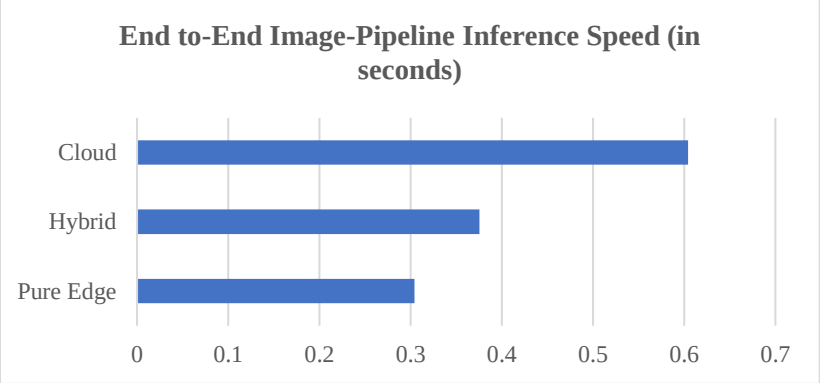


Figure 6: Intermediate – End to End Audio Pipeline Inference Time

Details

Pure Edge Setup

DaveAI benchmarked with:

- devices connected locally through LAN or WLAN
- a single user and two concurrent users
- component-level tests and entire audio pipeline tests

Before the pure edge setup, individual tests, with different FPS rates, were executed on the facial detection component to determine its performance.

Table 23. Intermediate Pure Edge - Facial Detection, 1 FPS

1FPS **IMAGE [Face]**	**Intel i5-1135G7**	**AMD Ryzen 3 4300U**	**Intel i7-1165G7**	**AMD Ryzen 7 4700U**
Average CPU Utilization %	55.94	90.33	55.93	58.3
Average Memory Utilization	1.3GB	1.0GB	1.3GB	1.1GB
[IMAGE] Average inference speed (in seconds)	0.32	0.3301	0.2911	0.3533

Table 24. Intermediate Pure Edge - Facial Detection, 2 FPS

2FPS **IMAGE [Face]**	**Intel i5-1135G7**	**AMD Ryzen 3 4300U**	**Intel i7-1165G7**	**AMD Ryzen 7 4700U**
Average CPU Utilization %	58	91.7	58.72	60.91
Average Memory Utilization	1.2GB	1004.8MB	1.2GB	1.1GB
[IMAGE] Average inference speed (in seconds)	0.3423	0.392	0.3274	0.4513

When all components ran on a single device, it caused device overload. Devices with weaker specification features, Intel i5-1135G7 and AMD Ryzen 3 4300U, were judged unsuitable for testing. Tests were re-run with a single and multi-device setup using the stronger platforms, Intel i7-1165G7 and AMD Ryzen 7 4700U.

Table 25. Intermediate Pure Edge Setup - Single Device Setup

Single device setup	**Intel i7-1165G7**	**AMD Ryzen 7 4700U**
Average CPU Utilization %	75.72	95.58
Average Memory Utilization	6.4GB	6.2GB
[WAKE] Average response time (in seconds)	0.7452	0.7735

[ASR+NLP] Average response time (in seconds) **End to-End Audio-Pipeline Response Time**	6.5658	7.6597
[IMAGE] Average inference speed (in seconds) **End to-End Image-Pipeline Response Time**	0.3761	0.6518
Avatar FPS	25-35	20-25
Usability Score	Medium	Low

Table 26. Intermediate Pure Edge Setup - Multi-device Setup

Multi-device setup	**Intel i5-1135G7:** WAKE + Avatar **Intel i7-1165G7:** ASR + NLP + Image Processing (Face)	**AMD 4300U:** WAKE + Avatar **AMD 4700U:** ASR + NLP + Image Processing (Face)
Average CPU Utilization %	59.08	64.83
Average Memory Utilization	4.3GB	4.0GB
[WAKE] Average response time (in seconds)	0.7428	0.6865
[ASR+NLP] Average response time (in seconds) **End to-End Audio-Pipeline Response Time**	2.344	2.4051
[IMAGE] Average inference speed (in seconds) **End to-End Image-Pipeline Inference Speed**	0.3043	0.3635
Avatar FPS	25-35	20-25
Usability Score	High	High

Observations:

- **Fully Loaded Setup:** The Intel i7-1165G7 outperformed the AMD device with a 14% faster response time.
- **Image Processing and Inference Speed:** Intel also outperformed AMD's image processing and inference speed.
- **Multi-device Setup:** The overall performance in both Intel and AMD devices improved predictably when workloads were spread over multiple devices. In this setup, Intel performed with a relatively better response time in the audio pipeline and a better inference speed in image-pipeline.

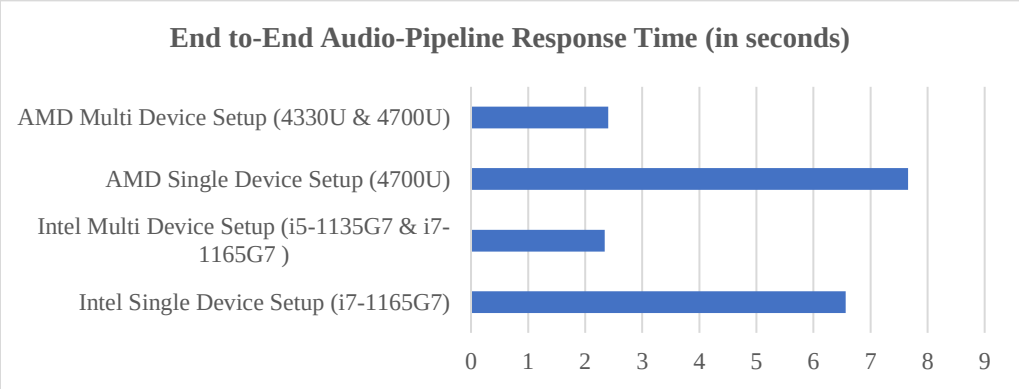


Figure 7 Intermediate Profile – Audio pipeline Response Time

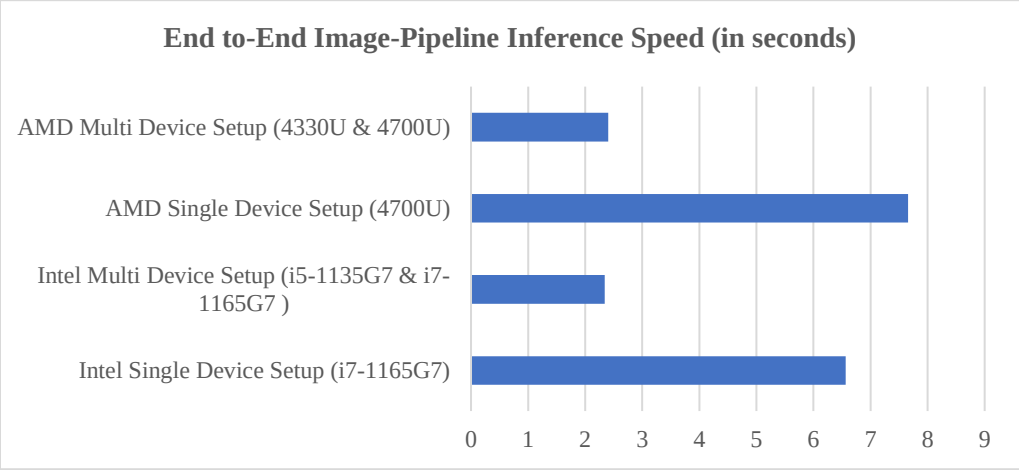


Figure 8: Intermediate Profile - Inference Speed

Hybrid Setup

In the hybrid setup, Avatar and WAKE components ran on the local devices, Intel i5-11 and AMD-4300U, at the edge. The remaining components ran in the cloud in AWS EC2.

Table 27. Intermediate Hybrid Setup

	Intel i5-1135G7	**AMD Ryzen 3 4300U**
[Local] Average CPU%	15.9	93.24
[Local] Average Memory	2.4GB	3.0GB
[WAKE] Average response time (in seconds)	0.6521	0.71158
[ASR+NLP] Average response time (in seconds) **End to-End Audio-Pipeline Response Time**	3.6031	6.8387
[IMAGE] Average inference speed (in seconds) **End to-End Image-Pipeline Inference Speed**	0.3753	0.4334
Avatar FPS	30-35	25-30
Usability Score	High	Medium

Observations:

- **Hybrid Response Times:** The initial load occurs on the local device, and the remaining workloads are processed in the cloud. The average response time of the audio pipeline in pure edge is 83% lower than the hybrid setup's response time. The inference speed for image processing had minimal impact.
- **Usability Score:** Usability score was similar in both Intermediate and Pure Edge profiles
- **CPU Utilization:** While the AMD device utilized more than 90% CPU, Intel utilized less than 20% CPU, allowing plenty of processing power for additional workloads.

Cloud Setup

In the cloud setup, the avatar ran on a local Intel i3 device while all other components ran on the cloud instance AWS EC2, including face detection component.

NOTE: The cloud device is a higher specification device, containing:

- vCPU: 8
- RAM: 32GB
- SSD: 120GB.

NOTE: The i3-11 device resource utilization was not recorded as the avatar was the only device loaded.

Table 28: Intermediate Cloud Setup

	Intel i3 Device + AWS EC2
[WAKE] Average response time (in seconds)	0.8797
[ASR+NLP] Average response time (in seconds) **End to-End Audio-Pipeline Response Time**	5.7016
[IMAGE] Average inference speed (in seconds) **End to-End Audio-Pipeline Inference Speed**	0.60415
Usability Score	Medium

Observations:

- **Response Time, Inference Speed:** In the cloud setup, most of the operations are leveraging the cloud compute infrastructure, the exception being the Avatar. At 2.0985 seconds, the average response time of the audio pipeline in a cloud setup is 58% higher than that of hybrid setup. The inference speed for image processing was severely affected. Cloud setup took 60% (0.22885 seconds) more time to respond than hybrid setup.
- **Usability Score:** A hybrid setup achieved a higher usability score than cloud setup.
- **Network Latency:** The end-to-end response time and inference speed in hybrid and cloud setup depends on the network latency.

Comprehensive Profile Benchmarking Results

Components: Avatar, Speech (WAKE + ASR), NLP, Ordering App, Facial Detection, Vehicle License Number Recognition

In the comprehensive profile, image processing workloads include both the face detection and vehicle license number recognition classes.

Summary Recommendations

For a comprehensive profile, the hybrid setup outperforms pure-edge and cloud setup in response times for the audio pipeline. The hybrid setup's inference speed is 200 milliseconds more than pure-edge. For overall performance, the hybrid setup has a high usability score compared to pure-edge setup.

Figure 9 shows the faster response time of the hybrid setup.

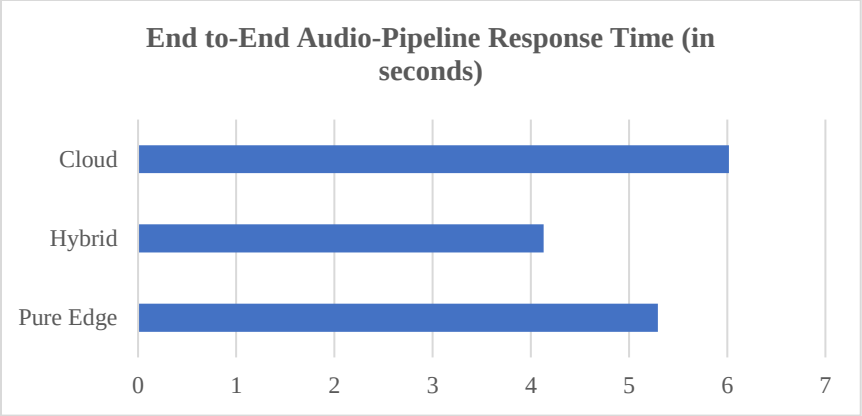


Figure 9: Comprehensive Profile - End-to-End Audio Response Time

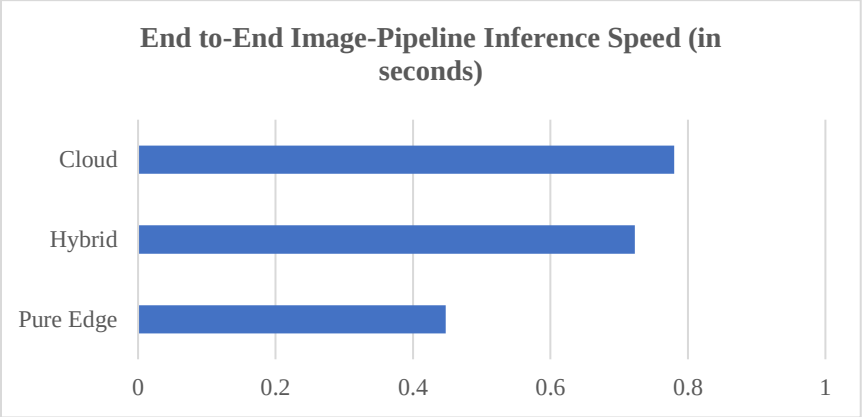


Figure 10: Comprehensive Profile - End-to-End Inference Speed

The hybrid setup fared best in audio pipeline response time while the pure edge had the best inference speed. Considering the cost, the pure-edge setup is less pricey, has a low TCO, compared hybrid setup. For this profile, DaveAI recommends the less pricey pure-edge setup. The response time for pure-edge is just a second slower than hybrid setup's response time. In the next phase, the audio pipeline for pure-edge will be optimized for the faster response time.

Details

Pure Edge Setup

DaveAI benchmarked with:

- devices connected locally through LAN or WLAN
- a single user and two concurrent users
- component-level tests and entire audio pipeline tests

As described earlier, loading all the components on a single device overloaded the device. Therefore, tests were conducted only on the higher configuration device options, Intel i7-1165G7 and AMD Ryzen 7 4700U.

Before the pure edge setup, individual tests, with different FPS rates, were executed on the vehicle recognition component to determine its performance.

Table 29:Comprehensive, Pure Edge - Facial Detection, 1 FPS

1FPS **IMAGE [Vehicle]**	**Intel i5-1135G7**	**AMD Ryzen 3 4300U**	**Intel i7-1165G7**	**AMD Ryzen 7 4700U**
Average CPU utilization %	55.21	91.63	55.11	58.14
Average memory utilization	1.5GB	1.2GB	1.5GB	1.3GB
Average inference speed (in seconds)	0.4356	0.6375	0.3984	0.7814

Table 30: Comprehensive, Pure Edge - Facial Detection, 2 FPS

2FPS **IMAGE [Vehicle]**	**Intel i5-1135G7**	**AMD Ryzen 3 4300U**	**Intel i7-1165G7**	**AMD Ryzen 7 4700U**
Average CPU utilization %	56.91	92.37	56.7	67.01
Average memory utilization	1.4GB	1.2GB	1.5GB	1.3GB
Average inference speed (in seconds)	0.5278	7.04491	0.5043	7.8837

Both Face and Vehicle Image Processing workloads were loaded on each device. Three tests, organized by frames per second, were executed: 1FPS, 2FPS, and 0.5FPS. The tests sent a frame every 2 seconds to both Face Detection and Vehicle Number Recognition.

Table 31: Comprehensive, Pure Edge - Face and Vehicle Image Processing 1 FPS

1FPS **IMAGE** **[Face + Vehicle]**	**Intel i5-1135G7**	**AMD Ryzen 3 4300U**	**Intel i7-1165G7**	**AMD Ryzen 7 4700U**

Average CPU Utilization %	56.28	91.66	56.53	59.51
Average Memory Utilization	1.5GB	1.5GB	1.5GB	1.3GB
Average inference speed (in seconds)	0.4951	0.6154	0.3815	0.7849

Table 32: Comprehensive, Pure Edge - Face and Vehicle Image Processing 2 FPS

2FPS IMAGE [Face + Vehicle]	**Intel i5-1135G7**	**AMD Ryzen 3 4300U**	**Intel i7-1165G7**	**AMD Ryzen 7 4700U**
Average CPU Utilization %	60.77	94.39	60.45	67.62
Average Memory Utilization	1.5GB	1.5GB	1.6GB	1.4GB
Average inference speed (in seconds)	0.6946	5.5505	0.6126	4.3502

Table 33: Comprehensive, Pure Edge - Face and Vehicle Image Processing 0.5 FPS

0.5FPS IMAGE [Face + Vehicle]	**Intel i5-1135G7**	**AMD Ryzen 3 4300U**	**Intel i7-1165G7**	**AMD Ryzen 7 4700U**
Average CPU Utilization %	56.01	91.94	57.29	58.66
Average Memory Utilization	1.5GB	1.2GB	1.4GB	1.2GB
Average inference speed (in seconds)	0.3855	0.47107	0.3462	0.5614

Observations:

- **Intel Performance:** Intel i7-1165G7 performed better in all three cases.

- **Response Time:** The response time of AMD 47700U was slower than the response time of the Intel i7-1165G7.
- With the increase in frame rate, the performance of AMD devices dropped over time. There was no response from the AMD devices at the end of the test, while Intel devices performed rather uniformly throughout the tests despite the increasing frame rate. The AMD devices' image-pipeline became blocked, and real-time delay accumulated.

Next all the DaveAI components were loaded. A multi-device setup was selected for this test, on which Avatar and WAKE were loaded onto one device and the remaining components on another device.

Table 34 Comprehensive, Pure Edge - All Components

	Intel i5-1135G7: WAKE + Avatar **Intel i7-1165G7:** ASR + NLP + Image Processing (Face & Vehicle)	**AMD 4300U:** WAKE + Avatar **AMD 4700U:** ASR + NLP + Image Processing (Face & Vehicle)
[Device 1] Average CPU Utilization %	16.51	94.03
[Device 1] Average Memory Utilization	2.5GB	3.2GB
[Device 2] Average CPU Utilization %	66.98	66.47
[Device 2] Average Memory Utilization	5.8GB	4.3GB
Avatar FPS	25-35	20-30
[WAKE] Average response time (in seconds)	0.7162	0.679
[ASR+NLP] Average response time (in seconds) **End to-End Audio-Pipeline Response Time**	5.2954	4.8184
[IMAGE] Average inference speed (in seconds) **End to-End Image-Pipeline Inference Speed**	0.4474	0.8491

Usability Score	Medium	Medium

Observations:

- **Response Time:** When the system was fully loaded, Intel exhibited a 10% slower audio pipeline response time than that of AMD, but AMD utilized most of the CPU.
- **Fully Loaded Image Processing:** When the system was fully loaded, AMD devices were 89% slower in response to image processing compared to Intel devices.
- Intel performed better for plain audio pipeline setup. Intel's performance was affected when the system was fully loaded. Intel has a 10% slower response for audio pipeline compared to AMD, but it is not suitable for user experience.

Hybrid Setup

Avatar and WAKE components were loaded on local devices, Intel i5-11 and AMD-4300U, and the remaining components ran in the cloud, AWS EC2.

Table 35: Comprehensive Hybrid Setup

	Intel i5-1135G7	**AMD Ryzen 3 4300U**
[Local] Average CPU%	17.71	93.23
[Local] Average Memory	2.5GB	3.0GB
[Local] Avatar FPS	30-35	25-30
[Wake] Average response time (in seconds)	0.7065	0.7121
[ASR+NLP] Average response time (in seconds) **End to-End Audio-Pipeline Response Time**	4.1315	6.7107
[IMAGE] Average inference speed (in seconds) **End to-End Image-Pipeline Inference Speed**	0.7229	0.7468
Usability Score	High	Medium

Observations:

- **Response Time:** Compared to edge setup, the response time in the hybrid setup is slightly faster. The response time of a pure-edge setup is 21% slower than the hybrid setup's response time.
- **Usability Score:** A hybrid setup has a high usability score compared to a pure edge setup.
- **CPU Utilization:** The AMD device utilized more than 90% CPU, while Intel utilized less than 20% CPU. Lower CPU utilization enables more room for other tasks to run.

Cloud Setup

The avatar ran locally on the Intel i3 Device, and the remaining components ran in the cloud, AWS EC2.

The streaming is 1 FPS for this setup. The video frames (images) were sent every second.

NOTE: The cloud device is a higher specification device, containing:

- vCPU: 8
- RAM: 32GB
- SSD: 120GB.

NOTE: The i3-11 device resource utilization was not recorded as the avatar was the only device loaded.

Table 36: Comprehensive Cloud Setup

	Intel i3 Device + AWS EC2
[Wake] Average response time (in seconds)	0.6803
[ASR+NLP] Average response time (in seconds) **End to-End Audio-Pipeline Response Time**	6.0168
[IMAGE] Average inference speed (in seconds) **End to-End Image Inference Speed**	0.78
Usability Score	Medium

Observations:

- **Average Response Time:** The average response time for cloud setup is 45% slower compared to pure-edge setup.
- **Usability Score:** A hybrid setup has a higher usability score than that of the cloud setup.

- **Network Latency:** The end-to-end response time and inference speed in hybrid and cloud setup depends on the network latency.

Summary of Key findings

After the entire benchmarking report, here is a summary of key findings.

1. For a basic profile, a single Intel device is sufficient to handle the entire workload of the audio pipeline as a QSR kiosk, delivering a 2.2s end-to-end audio response time and a high 3D avatar graphic performance (30-35fps). In addition, Intel i7-1165G7 still had more than 70% CPU headroom for other tasks. This single device solution provides a cost-effective option for low quantity and distributed QSR deployment.
2. For an intermediate profile, the pure edge setup, with a faster response time and inference speed for image processing, outperformed the hybrid and cloud setups. The Intel i5-1135G7 as the QSR Kiosk and the Intel i7-1165G7 as the edge server, with 2.3s end-to-end audio response time and high 3D avatar graphics performance (25-35fps), provide a cost-effective multi-device option.
3. For the intermediate profile, the hybrid and cloud setups provide medium usability with an end-to-end audio of more than 3s and more than 5s respectively.
4. In a comprehensive profile, the hybrid setup performed with the best end-to-end response time of 4.1s.
5. In the comprehensive profile, the pure edge setup provided medium usability with an end-to-end audio response time of more than 5s.
6. DaveAI recommends using OpenVINO optimized AI models in all deployments. This is from the evidence captured while benchmarking ASR workloads on Intel i5 devices, which outperformed other configurations.

Total Cost of Ownership (TCO) Analysis

DaveAI conducted a TCO analysis for different deployment types. Table 37 presents the TCO calculation for all three profiles with three workloads at a larger scale. A detailed overview of TCO analysis considerations is available in the Annexure.

Table 37. Total Cost of Ownership Analysis Results

TCO Calculation at Scale = 100				
Profile	**Workloads**	**Pure Edge TCO**	**Hybrid TCO**	**Cloud TCO**
Basic	Avatar + Speech + NLP + Ordering Application	$246,755.93	$342,503.26	$345,862.30

Intermediate	Avatar + Speech + NLP + Ordering Application + Image (Face)	$318,775.93	$502,879.21	$506,238.25
Comprehensive	Avatar + Speech + NLP + Ordering Application + Image (Face + Vehicle)	$388,481.93	$552,145.08	$555,504.13

Note: Deployment & support cost, Network Infrastructure costs are not included in this analysis. Assumptive indication of these costs are outlined in the Annexure, incase the reader wishes to compare costs including the same. Inclusion does not however create any change in comparative observations. Pricing obtained is as on March 1, 2022.

Annexure

Place an Order – Details and Variations

There are five technology components that enable a seamless QSR ordering experience:

A. Computer Vision to detect vehicle numbers and details to identify the customer
B. Facial Detection Module to detect a customer at the kiosk & if consent is available to detect other parameters
C. Automated Speech Recognition to accept Speech Inputs enabling contactless and convenient ordering
D. Natural Language Processing Engine to understand the intent from natural language inputs
E. Digital Signage workloads to show relevant content to users

The following list presents a more detailed description of a potential customer order:

1. Figure 1 presents the home page or the idle state of the kiosk as it showcases all the trending and popular dishes of the restaurant.
2. The customer enters the drive-through area.
    a. A camera-enabled device detects customer presence.
    b. The Kiosk detects the vehicle plate number and performs facial recognition.
3. The greetings exchange occurs between the customer and kiosk. This may occur in a variety of ways:
    a. The kiosk issues a stock greeting, such as “I am Dave. How can I help you?”
    b. The customer issues a greeting into the microphone using a wake-up word, such as “Hi Dave.”
    c. The customer initiates interactions by clicking buttons on the digital screen.

4. The customer indicates:
    a. **menu items:** The customer taps the touch screen to consider menu items and then taps **Add to Cart** to choose an item, or the customer indicates the choice verbally (e.g., “I’d like an order of the Macaron Cookies and two mocha coffees.”).
    b. **menu categories**: The screen presents food categories and groupings, such as **Top Offers**, to filter the user’s experience and enable quick finds.
5. The customer may see cart options by using the **Show Cart** option.
6. The cart page presents a **Did you Forget** section that suggests more options. Users may add additional dishes as illustrated in the user interface design here. Customers will be asked to specify the quantity.
7. Finally, customers are taken to the review page This is the final page where customers review and finalize the order. It includes all the dishes that were added to the cart and an option to modify the cart using menu selection or voice.
8. Customers select their choice of payment and complete the order.

For more details, refer to link.