Voice Interoperability Initiative Architecture Series Whitepapers

Multi-Agents Wake Words

July 21st, 2021

Contents

Abstract

Customers should have the freedom to choose their preferred service on products that support multiple, simultaneous voice services, each with its own wake word or invocation name – enabling customers to talk to the service of their choice in a secure manner by simply saying its name. This whitepaper provides information for selecting a wake word and tuning wake word models, and describes wake word use cases that require special attention and handling on multi-agent devices. It is one in a series of whitepapers that address topics relevant to building products that support multiple simultaneous voice agents and conform with the Multi-Agent Design Guide. It is intended for technical architects, device maker engineers, and voice agent developers. You will benefit by already having familiarity with the Voice Interoperability Initiative and the design guide for many of the terms and concepts in this whitepaper.

Overview

The Voice Interoperability Initiative (VII) is committed to providing customers choice and flexibility to interact with multiple voice services. In pursuit of this objective, Amazon has published the Multi-Agent Design Guide, which provides guidance on developing products that support multiple simultaneously available agents. This guidance is designed to enhance the user experience while providing assurances for security and privacy. To support you in creating multi-agent experiences that align with the guide, this whitepaper discusses methods for invoking agents, the components of user speech with an agent, wake word detectors, wake word selection, models, types, and use cases, the importance of reducing false accepts and false rejections, and cloud-based wake word verification.

Terminology

These terms are defined in the context of voice-enabled multi-agent devices.

Also refer to terms in the Multi-Agent Design Guide.

Active Agent
The agent that is currently in the listening, thinking or speaking state.

Audio Front End (AFE)
The audio front end is a separate hardware or software component that implements audio algorithms to improve speech content and quality as required by specific agents.

Automatic Speech Recognition (ASR)
The identification and translation of spoken language into text.

False Accept
An accept by a wake word detector of a sound, word, or phrase that was not the wake word.

False Rejection
A rejection by a wake word detector of a correctly spoken wake word.

Applications Processor (AP)
The main processor on the device that runs application programs such as agent client software. It is also known as the host processor or CPU.

Wake Word
A word or phrase a user speaks to “wake up” (invoke) a specific agent, get its attention and have it start listening.

Wake Word Detector (WWD)
One or more wake word engines and wake word models used to detect wake words. It may consist of multiple stages of wake word engines and may include cloud-side verification.

Wake Word Engine (WWE)
A component that is responsible for detecting wake words in a voice audio stream. A WWE can be embedded in a separate processor, such as a digital signal processor or co-processor with/without a neural accelerator, or run on the applications processor.

Wake Word Model
A neural network or machine learning data representation of one or more wake words.

Invoking Agents

A dialog between a user and an agent may be initiated in several ways. A user may initiate a dialog with an agent by tapping or pressing a button, or speaking a distinct wake word that invokes a specific agent. Agents may also initiate a dialog based on a prior user setting or timed events such as reminders, or on behalf of another agent via an agent-to-agent transfer, where an agent that is unable to fulfill a user request initiates a dialog with another capable agent. How agents get invoked requires special consideration on multi-agent devices as users may be more prone to error when presented with multiple interfaces and choices.

By Touch

An agent may be invoked by pressing a physical button or tapping a touch-sensitive button or screen. Touch-sensitive screens may display and utilize soft buttons or use gestures to invoke an agent.

Multi-agent devices supporting action buttons should include a distinct Action button for each agent to initiate a new voice interaction. Action buttons are useful for interrupting responses and media output from any and all agents or for stopping a sounding alert. Accessory devices, such as remote controls, may also provide tap-to-talk (TTT) or push-to-talk (PTT) buttons for invoking agents. The Multi-Agent Design Guide recommends not overloading buttons to avoid inadvertently engaging the incorrect agent and having to remember the pattern (e.g., long vs short press).

By Voice

An agent may be invoked by speaking one of its wake words. When a wake word is spoken, the corresponding agent “wakes up” and enters the listening state.

A wake word or wake phrase may consist of a localized salutation followed by the wake word. Common salutations include words like Hey, OK, or Hi. For example, “Hey Siri”, “OK Google”, “Hi Bixby”, and “Hey Portal”. Not all wake words follow this format, such as “Alexa” or “Cortana”.

Proactive Agent Inquiries and Notifications

Agents may initiate a dialog with the user to ask a question or notify them. For example, a health agent may ask the user whether they’ve taken their medication or an automobile agent may automatically notify the driver an oil change is needed.

Prescheduled Routine or Skill

Agents may initiate a dialog with the user based on a pre-scheduled routine or skill. For example, a user may have scheduled a time of day where an agent is to perform a prescheduled

routine or start a custom skill that asks the user a question. The multi-agent device would then enter its listening state for a specific agent to hear the user's response.

Agent Transfers

An agent transfer is where one agent invokes another agent to fulfill a user request or inquiry that it is unable to fulfill. The agent transfer is performed via a software interface on the device. Voice data are not automatically routed from one agent to the other agent. The user does not need to speak the transferee agent's wake word, but is required to repeat the utterance.

Multi-Agent Wake Words

A device maker must consider several aspects of wake words when making them simultaneously available on a multi-agent device. The following sections provide insight into techniques that may be used to aide distinct wake word detection and ensure that the correct agent is invoked when a wake word is spoken.

Wake Words and Utterances

Users interact with voice enabled smart devices by invoking the agent and following it with speech, such as a question or command. A common method for invoking an agent is to say its wake word or wake phrase, termed more generally as the wake word or key word. The portion of user speech that follows the wake word is known as the utterance. Generally speaking, an utterance is a phrase or sentence a user speaks to an agent to complete a task or ask a question. The components and structure of wake word-based user-initiated dialogs are:

```
[Wake Word], [Utterance]
```

Examples of wake words and utterances for fictitious agents Kathryn, Katherine, and Alecia are:

```
Kathryn, what time is it?

Katherine, please remind me to pick up my dry cleaning at 5pm today.

Hey Alecia, give me driving directions to get home.
```

The device's attention system should convey which agent has been invoked through a distinct visual, such as a unique LED pattern and color scheme. When the device enters the listening state to capture the utterance, an audio cue may also be played at the beginning and end of the listening state to let the user know their voice data are being streamed to the agent's cloud services.

An agent may also be capable of a follow-up mode, where the user is not required to repeat the wake word after the first response from the agent. The device attention system shows the device re-entering the listening state for the same agent and the user may conveniently speak another utterance to it. The user may end a follow-up mode by saying “Stop” or “Cancel.”

Wake Word Detector

A wake word detector (WWD) examines the audio input for a pattern that resembles a wake word and continually performs this examination to support user barge-in such as when an agent is responding. A multi-agent device may utilize a single WWD or several to support all of the agents wake words. When all wake words can be supported on a single WWD, processor overhead and memory requirements may be lower and device software implementation may be easier. WWD technologies vary as do their performance characteristics. They may vary in utilized processing power and memory and may require different input data formats. Lightweight WWDs consume less power and are ideal for portable devices, but may support fewer wake words and may be slightly less accurate.

WWDs may also be implemented as multi-stage detectors on the device followed by another stage in the cloud. A wake on a first-stage, low power WWD may result in additional power being provided to the applications processor (AP) to run a second-stage WWD with a greater capacity for accurately detecting the wake word. Doing so limits higher power consumption to occur only after a wake word has been detected in the first stage.

Wake Word Selection

The earlier wake word and utterance examples listed *Kathryn* and *Katherine* as agent wake words. These are pronounced “Kath-rin” and “Kath-er-in” respectively, varying slightly in their pronunciations. Kathryn has two syllables, whereas Katherine has three, although some people unconsciously elide a syllable when speaking the latter, pronouncing it as “Kath-rin”, identical to the pronunciation for *Kathryn*. When wake words such as these are simultaneously supported on a multi-agent device, they are problematic because they have linguistic similarities and people may pronounce them differently. Even when correctly pronounced, WWDs and ASR may not distinguish them, especially when background noise is prevalent. ASR is relevant because a user may speak a wake word in the utterance. Examples of wake words in utterances for fictitious agents Kathryn and Katherine are:

> Katherine, what do you know about the Kathryn voice assistant?
>
> Kathryn, do you spell Katherine with a K or a C?

The Multi-Agent Design Guide provides guidance on wake word selection and states they should be easy to say and not sound similar. What considerations does this entail?

First, avoid selecting a wake word that contains a subset of the other's phoneme sequence. For example, "Alecia" (Uh-lee-sya) competing with "Alesha" (Uh-lee-shuh) would not do well because they share two phonemes in sequence. Second, try to avoid words that rhyme. Lastly, try to use wake words of different lengths (number of syllables). As a general rule of thumb, more easily distinguishable wake words usually have 3 or 4 syllables to provide wake word detectors with enough acoustic information for distinct recognition.

Another example of potentially problematic wake words when simultaneously supported on a multi-agent device are *Ryan* and *Brian*, where the pronunciation for *Ryan* is fully contained inside the pronunciation for *Brian*. These two names sound very different to people, but not necessarily so for WWD. Similarly, "OK You" and "K You" (pronounced "Kay-you"), a portion of the wake word "OK YouVision," may trigger a wake word beginning with "OK You…." A wake word detector's inability to distinguish two wake words may be as subtle as two wake words being one sound apart for their neutral vowel, such as for 'o' vs 'e' in "Hey Johnny" and "Hey Jenny."

It is also important to avoid using wake words that are common words or phrases in a spoken language. For example, "Hey You" is common in spoken English and when used as an agent's wake word, may have unintended consequences for causing a nearby smart device to enter its listening state.

Wake words with distinctive acoustic signatures are likely to perform better and are more robust in noise. To increase the acoustical energy of a wake word, include phonemes with high resonance in the oral or nasal cavity (sonority), consonants that include movement of the place of articulation (affrication), consonant compounds, and voicing (vibration in the vocal tract). Examples of English phonemes with high acoustical energy are shown in Table 1.

Phoneme/letter in English	**Type**	**Example**
y	sonorant	**y**ellow
ng	sonorant	si**ng**
all vowels	sonorant	**a**rm, c**a**t, **ai**r, s**ay**, m**e**t, ci**n**em**a**, **eye**, s**ee**, r**o**ck, f**oo**d, f**ou**r, l**u**ck, t**u**rn
ch	affricate	**ch**urch
j	affricate	bu**dg**e

Phoneme/letter in English	Type	Example
q [kw]	consonant compound	**qu**ick
pt	consonant compound	attem**pt**
x [ks]	consonant compound	**x**-ray
d	voicing	**d**og
b	voicing	**b**ig
th	voicing	**th**is

Table 1 - English Phonemes with High Acoustical Energy

Even with this guidance, testing and verification will be valuable to ascertain whether two wake words are sufficiently distinguishable.

Wake Word Model

A WWD uses a machine learning model trained from many occurrences of detected wake words to score a match with the input voice data. If the confidence score exceeds the model's threshold, the wake word is considered detected and the WWD triggers. The wake word model may incorporate multiple wake words and also be trained to support multiple languages.

False Rejections and False Accepts

Developing a robust WWD is a significant undertaking and requires planning, data collection, training, and testing. Background noise and user distance from the device's microphones also affect a WWD's ability to accurately detect wake words. While wake words may sound different to a person, the environmental audio that is received by the WWD may be treated differently after having been processed by the audio front end, which reduces background noise and may not completely eliminate it.

The accuracy of wake word detection is measured with two parameters: false rejection rate (FRR) and false acceptance rate (FAR). A false rejection is when a user properly speaks a wake word, but the corresponding WWD fails to detect it. A false accept (FA) is when a user does not speak the wake word, but the WWD detects a sound similar to the wake word and triggers.

There are two general categories of False Accepts: user-generated and media-generated. Both need to be mitigated. User-generated FA events can be generated by a combination of user speech and background noise, including background conversations. Words known as near phrases can create a user observable FA event. An example of a near phrase is a phoneme sequence that is one or two phonemes away from the desired wake word.

Media (TV/Radio/Media players) generated FA events, such as from a nearby TV, where a wake word in media content is rendered through speaker output, can include the exact wake word that the WWD has been modeled to accept. Media content may also include a combination of

words or sounds where a portion resembles a wake word. These are considered media-generated FA events because the user did not initiate the wake word event.

Reducing False Rejections and False Accepts

Reducing false rejections and false accepts is important in order to provide an optimal user experience. Reducing false rejections improves the user experience by reducing the number of times a user may have to repeat the wake word to invoke the agent. Conversely, reducing false accepts improves the user experience by reducing the number of times a wake word triggers due to near phrases, background speech or noise.

A WWD applies a decision-making algorithm as to whether a pattern of sound resembles the wake word. A detection threshold determines whether the sound pattern is accepted or rejected as the wake word, and FRR and FAR can be used as metrics to set it. For any given model, a lower detection threshold increases the likelihood of unintentional wake word triggers. Conversely for any given model, a higher detection threshold increases the likelihood the wake word detector fails to trigger when intended.

False rejection and false acceptance rates can be lowered through improvements in wake word models with additional training as well as algorithms and WWD technology. Plotting the detection error tradeoff (DET) curves for multiple WWDs algorithms on a single graph allows for performance comparisons. Figure 1 shows a smaller area under the DET curve for hypothetical agent ‘B’, indicating superior performance.

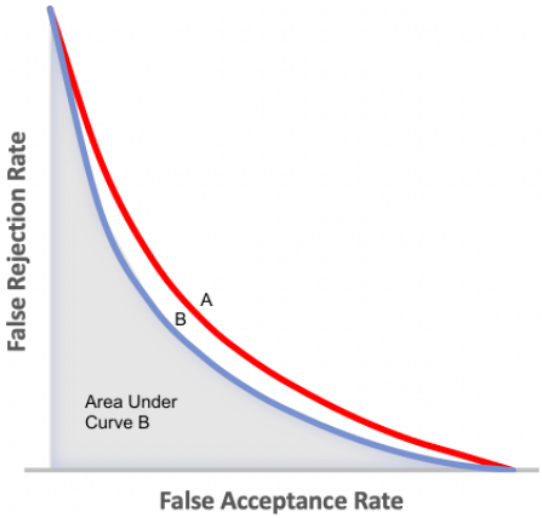

Figure 1 - DET Curves for 2 WWDs

In scenarios involving multiple WWDs, detection thresholds are independently adjusted for FR and FA performance. When multiple WWDs are used on a multi-agent device and testing reveals that more than one WWD occasionally triggers during the span of a spoken wake word, it is an indication that the corresponding wake words may be too similar or that the detection threshold is too low for one or more of them. When simultaneous triggers occur across multiple

WWDs, additional logic may be used in the device firmware to determine whether to select one of the few that were triggered and ensure that only one agent enters the listening state.

Cloud-Based Wake Word Verification

An agent may also use an additional stage to verify that the triggered wake word was not falsely triggered by using cloud-based wake word verification (CBWWV). Cloud compute resources provide considerably more horsepower and memory than a multi-agent device, thereby enabling greater accuracy using more compute-intensive approaches.

The use of CBWWV differs from the utilization of a device-based WWD in that the corresponding agent is already in its listening state due to the trigger of the wake word on the device. When CBWWV verifies the wake word as being valid, cloud-based ASR processing of the utterance ensues. Whereas if the wake word verification in CBWWV does not meet the threshold criteria, the agent's cloud services send a signal back to the agent's client software on the device to return to the idle state. The agent's client software then updates the attention system to indicate the agent invocation was invalid using a distinct visual.

Wake Word Use Cases

A wake word is spoken to begin an interaction with an agent. The subsequent voice utterance data are streamed to the agent's cloud services for natural language processing to fulfill the user request and formulate a text-to-speech (TTS) response that is streamed back to the device and output by the speakers.

A wake word may also be spoken in the utterance, such as when a user is asking an agent about another agent. A WWD may detect the interim wake word and trigger. When device software does not coordinate agent listening and speaking states on a multi-agent device and wake word triggers are not masked while an agent is in the listening state, unintended wakes may result in a very confusing user experience.

The following use cases illustrate desired and undesired behaviors that may result when detections occur and listening and speaking states are not coordinated on a multi-agent device. Two fictitious agents, Alecia and Alesha, are used as examples. Their wake words are intentionally similar to illustrate how this may potentially result in confusing user experiences. Alecia supports meeting schedules, whereas Alesha does not. The first set of use cases are for a multi-agent device with a single WWD that supports both wake words, where only one way word can trigger at any point in time. Whereas, the second set of use cases are for a multi-agent device with two WWDs, each supporting one wake word, where both may trigger at the same time. Mistaken accepts by Alecia when "Alesha" is spoken, also a possibility when wake words are this similar, are not covered to keep things simpler. Wake words are underlined.

Table 2 shows desirable and undesirable behaviors that may occur for a single WWD supporting multiple wake words. When the WWD triggers for Alecia, as intended by the user, Alesha does

not mistakenly accept as only one may trigger at a time on a single WWD. Likewise, when the WWD mistakenly accepts Alesha for "Alecia" because of their similarities, Alecia does not wake. Also, when Alesha detects "Alesha" in the utterance and wakes in use case 2, it receives only the subsequent part of the utterance.

Use Case	User Speech	Desired Behavior	Undesired Behavior
1	Alecia, when is my next meeting?	Alecia triggers, receives the full utterance and responds with the meeting time.	Alesha mistakenly accepts on "Alecia", receives the full utterance and responds with "I do not understand."
2	Alecia, is it true that agent Alesha is better for driving directions?	Alecia triggers on "Alecia", receives the full utterance, and responds with "Yes, it is true."	Alesha mistakenly accepts on "Alecia", receives the full utterance, and responds with "Compared to who?" (Alesha also triggers on "Alesha" and continues listening on behalf of the prior FA.) Or, Alesha only triggers for "Alesha", receives "is better for driving directions?" and responds with "I do not understand."

Table 2 - Wake Words and Utterance Use Cases – Single WWD

Table 3 shows desirable and undesirable behaviors that may occur for a dual WWD. In these examples, Alecia triggers and Alesha mistakenly accepts at the same time for the opening "Alecia" as the two WWDs operate independently and software on this multi-agent device does not arbitrate simultaneous wakes.

Use Case	User Speech	Desired Behavior	Undesired Behavior
1	Alecia, when is my next meeting?	Alecia triggers, receives the full utterance and responds with the meeting time.	Alesha mistakenly accepts on "Alecia", receives the full utterance and responds with "I do not understand." **at the same time that Alecia responds with the meeting time** (as Alecia correctly triggered on "Alecia").
2	Alecia, is it true that agent Alesha is better for driving directions?	Alecia triggers on "Alecia", receives the full utterance, and responds with "Yes, it is true."	Alesha mistakenly accepts on "Alecia", receives the full utterance, and responds with "Compared to who?" **while Alecia is responding for the desired behavior**. (Alesha also triggers on "Alesha", but does not affect Alesha already being in the listening state). Or, Alesha does not mistakenly accept on "Alecia", triggers for "Alesha", receives "is better

			for driving directions?" and responds with "I do not understand" **at the same time that Alecia responds for the desired behavior.**

Table 3 - Wake Words and Utterance Use Cases - Two WWDs

As can be seen, when false accepts occur or agent wake words are embedded in the utterance and agents do not coordinate their listening states, confusing user experiences may result. These undesirable outcomes are the result of agent collisions for listening and speaking states. They illustrate the importance of carefully selecting an agent's wake word and properly tuning WWDs to reduce false accepts. They also emphasize the need for device middleware to aide in the coordination and handling of agent listening and speaking states. For example, such middleware may enforce a single agent listening state, where only one agent may be in the listening state at any time.

Conclusion

Multi-agent devices provide customers delightful experiences for engaging with a diversity of capabilities and features offered by agents and their services. Building multi-agent devices involves careful selection of a wake word, understanding the importance of reducing false accepts and false rejections, and being aware of use cases that require special handling of wake word triggers to avoid confusing user experiences.

Contributors

The contributors to this document are:

- Robert Mars, Principal Solutions Architect, Alexa Voice Services
- Joe Murphy, Vice President Marketing, Sensory, Inc.

Additional Resources

- Voice Interoperability Initiative: https://developer.amazon.com/en-US/alexa/voice-interoperability
- Multi-Agent Design Guide: https://build.amazonalexadev.com/rs/365-EFI-026/images/VII_Multi_Agent_Design_Guide.pdf
- VII Architecture Best Practices – Foundational Concepts: https://m.media-amazon.com/images/G/01/vii/VII_Architecture_Best_Practices_Foundational_Concepts_Whitepaper.pdf

Document Revisions

Date	Description
July 2021	First publication