

Speech EG's CES 2019 Project

- [Purpose](#)
- [Background](#)
- [Use cases](#)
 - [Supported](#)
 - [Out of Scope](#)
- [Major Tasks](#)
- [External Dependencies](#)
- [Proposed Work Flows](#)
 - [General - At System Start Up,](#)
 - [General - Before user starts speaking,](#)
 - [Domain Specific Flows \(Alexa Commitments\)](#)
 - [Navigation \[To be Decided\]](#)
 - [Weather](#)
 - [Alerts](#)
 - [Phone Call Control](#)
- [CES Demo Integration Meetings](#)
 - [Minutes of Meeting](#)
 - [November 28th, 2018](#)
 - [November 30th, 2018](#)

Purpose

This document outlines the Speech EG's commitment proposal to CES Demo 2019. It also highlights the dependencies on Audio High Level, Application Framework and Native App Integration teams in order to deliver a successful demo that demonstrates our vision of multi agent architecture.

Background

The Speech EG presented the [Voice Services architecture](#) v1.2 at AGL F2F meeting on September 11th, 2018. The architecture was actively reviewed and consensus has been met on many areas. The latest updated version [v1.3](#) incorporates all the comments.

Use cases

Supported

- Multiple voice agents installed but one active voice agent running on the system and triggered through Tap-To-Talk button
- The Amazon team will deliver Alexa voice agent.
- Active voice agent will be selected through the Settings Application Menu on reference AGL platform.

Out of Scope

Wake word detection will not be supported. As of today, AGL Audio 4a, doesn't yet support storing a persistent audio input buffer can be shared between multiple consumers. In this case wake word module and high-level voice services. We discussed in detail about the audio design to support wake word use cases. However there is no further information on the timeline for having this support to be baked into AGL Audio framework.

Major Tasks

#	Component	Ownership
1	Voice Service High Level (VSHL) Development	Amazon to deliver first draft the can be open sourced and submitted to AGL repository.
2	Alexa Voice Agent Development	Amazon
3	QT based App for Template Rendering	Amazon
4	Native App Development and Integration With Voice Service High Level	Linux Foundation / IOT.BZH

5	Audio Input Output Support	Linux Foundation / IOT.BZH
6	Application Framework Support	Linux Foundation / IOT.BZH

External Dependencies

- Applications should be able to launch themselves when they receive intents from Voice Service Interaction Manager.
- Audio High level needs to create 4 audio roles for Alexa to do audio output.
- Audio High Level needs to create 1 audio role for High Level Voice Service to do audio input.
- Speech Chrome Application needs to be implemented to display different dialog states (IDLE, LISTENING, THINKING, SPEAKING) of the voice agent.
- Template Runtime Application to show the templates that are delivered as responses by each voice agents. If we can't standardize the language for this template, then as a workaround, Amazon will implement Alexa UI Template Runtime Application that can render Alexa templates for CES Demo 2019.

Proposed Work Flows

General - At System Start Up,

- Speech Chrome Application subscribes to OnDialogEvent = (IDLE, LISTENING, SPEAKING, THINKING) with Voice Services High Level.
- Voice Services High Level subscribes to OnDialogEvent = (IDLE, LISTENING, SPEAKING, THINKING) with all the voice agents.
- Navigation Application on the system will subscribe to Navigation messages on the Voice Interaction Manager.
- Template Run-time Application will subscribe to Template Run-time messages on the Voice Interaction Manager.

General - Before user starts speaking,

- User selects Alexa from Settings
- User presses Tap to Talk button
- Voice Service High Level is in IDLE state and will listen to tap to talk signal
- Voice Service High Level will automatically signal Alexa Voice Agent.
- After few milliseconds, Alexa Voice Agent will publish OnDialogEvent = LISTENING.
- Voice Service High Level will receive it and propagate the same event OnDialogEvent = LISTENING to Speech Chrome App.
- Speech Chrome App receives the event and displays a UI to indicate that user can start speaking.
- At this point, the user is ready to start speaking.

Domain Specific Flows (Alexa Commitments)

Navigation [To be Decided]

- User starts speaking and says, "Alexa, Navigate me to nearest Star Bucks." or "Navigate me to nearest Star Bucks"
- Alexa Voice Agent will call Voice Interaction Manager's Navigation::Publish API to publish navigation message with the Geo-code of the destination.
- Navigation App will receive the message and launch itself or ask Homescreen to launch it.

Weather

- User starts speaking and says, "Alexa, What's the weather" or "What's the weather."
- Alexa Voice Agent will speak TTS about weather.
- Alexa Voice Agent will publish OnDialogStateEvent = SPEAKING so that the Speech Chrome can show appropriate UI.
- Alexa Voice Agent will call Voice Interaction Manager's TemplateRuntime::Publish API to publish UI template.
- Alexa UI Template Run-time Application will receive the message and launch itself or ask Homescreen to launch show the template.

Alerts

- User starts speaking and says, "Alexa, Set an Alert for next 1 minute" or "Set an Alert for next 1 minute."
- Alexa Voice Agent will prompt TTS that Alert is set.

- Alexa Voice Agent will call Voice Interaction Manager's Alerts::Publish API to publish new Alert state.
- Alexa Voice Agent will play Alerts Audio after one min.

Phone Call Control

- User starts speaking and says "Alexa, Call Bob"
- Alexa Voice Agent will prompt TTS to disambiguate the contact request.
- Alexa Voice Agent will call Voice Interaction Manager's Call:: Publish API to publish a DIAL event.
- Dialer app on the AGL reference platform will pick the event and initiate call based on the event payload.
- Dialer app will call Voice Interaction Manager's Call:: Publish API to publish a CALL_ACTIVATED downstream event for Alexa Voice Agent to update it's context.

CES Demo Integration Meetings

To join : <https://appear.in/agl@iotbzh>

Next Meeting : 8:00 AM PST on November 30th, 2018

IRC : #automotive

Minutes of Meeting

November 28th, 2018

Alexa Voice Agent is tested and working with Microchip array and Fiberdyne at AGL F2F workshop in Germany. Thanks to Shotaro and IOT.bzh. The main purpose of the integration session is well under control.

Action Items:

- IOT.bzh to provide the image and SDK that will be used for CES demo today.
- IOT.bzh to help with running HTDocs on dev machine that connects to VSHL on a Renesas board
- IOT.bzh to work with Naveen & Shotaro to integrate the phone control capability with Dialler app.
- IOT.bzh to look into the VSHL config file to fix the problem with two VSHL icons visible on the homescreen, one for the html app and one for the service.
- Shotaro to host the low-level Alexa voiceagent widget privately and share it with IOT.bzh for testing and integration.
- Naveen to provide a temporary work around for refreshing the auth tokens required for low-level voice agent.
- Naveen to provide an update on our discussions with ICS. The company that we are trying to engage for building Alexa UI and Login Apps.

November 30th, 2018

- IOT.bzh provided the AGL images that will be used for CES.
- IOT.bzh is temporarily unblocked on Alexa Voice Agent authentication. Naveen updated that this is only a temporary solution and we will have a proper CBL auth solution in place before the December 10th SF F2F session.
- Naveen updated IOT.bzh that Alexa apps will be implemented by ICS.

Action Items:

- IOT.bzh to look into the VSHL config file to fix the problem with two VSHL icons visible on the homescreen, one for the html app and one for the service.
- IOT.bzh to integrate Dialer app with phonecontrol messages from VSHL and also integrate the app launching plugin with VSHL.