

Comparison of Voice Assistant SDKs for Embedded Linux

Leon Anavi

Konsulko Group

leon.anavi@konsulko.com

leon@anavi.org

ELCE 2018

Konsulko
Group

- Services company specializing in Embedded Linux and Open Source Software
- Hardware/software build, design, development, and training services
- Based in San Jose, CA with an engineering presence worldwide
- <http://konsulko.com>

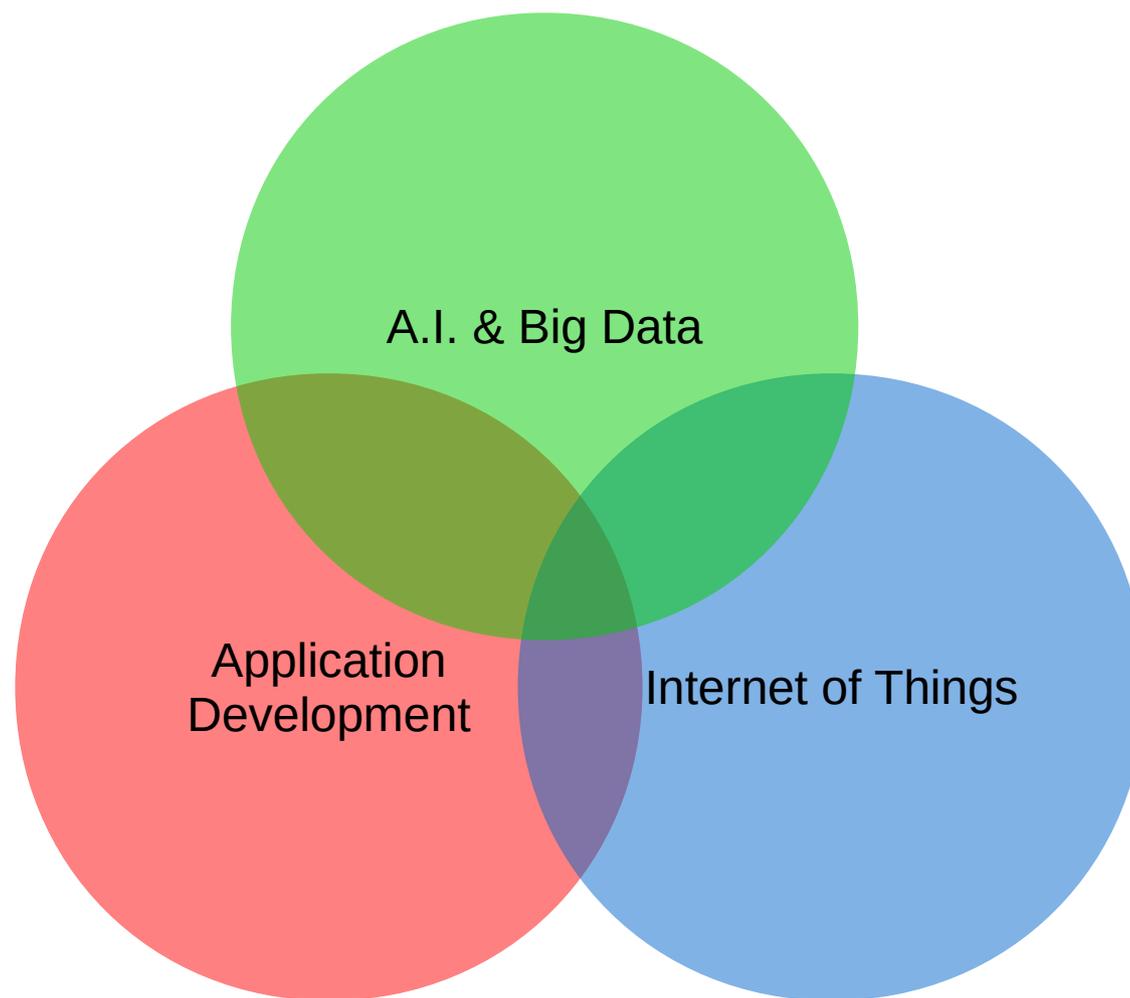
Agenda

- Introduction to smart speakers with voice assistants
- Overview of Amazon Alexa, Google Assistant and Mycroft SDK for integration in embedded Linux devices
- Showcases and conclusions

Virtual assistants

- AliGenie
- Amazon Alexa
- Yandex Alice
- Samsung Bixby
- Braina
- Clova
- Microsoft Cortana
- Google Assistant
- Mycroft
- Apple Siri
- Voice Mate
- More ...

Technologies in Smart Speakers

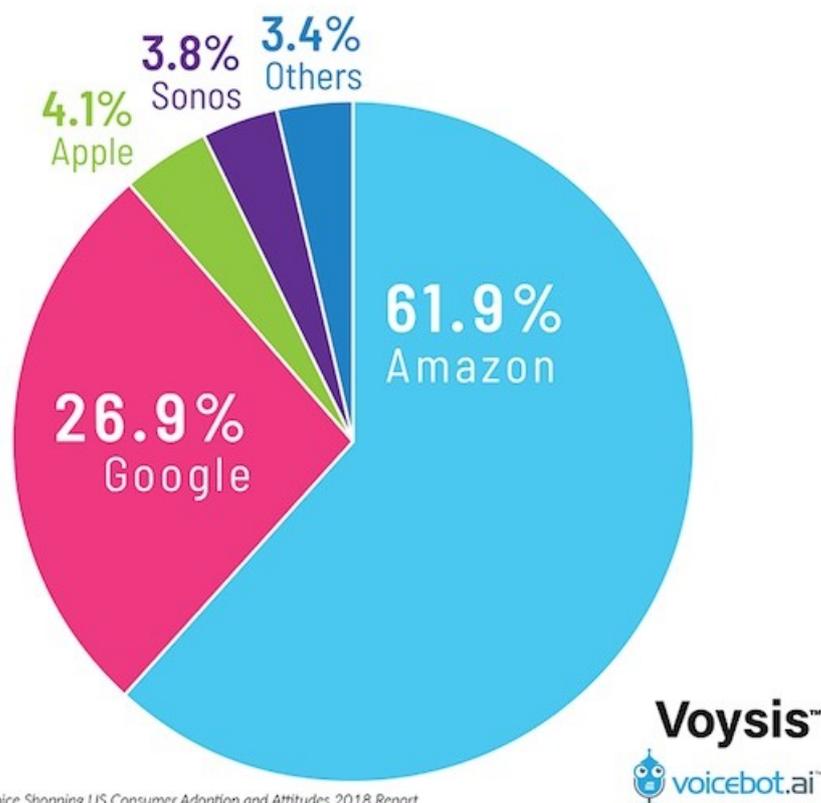


Key Software Ingredients

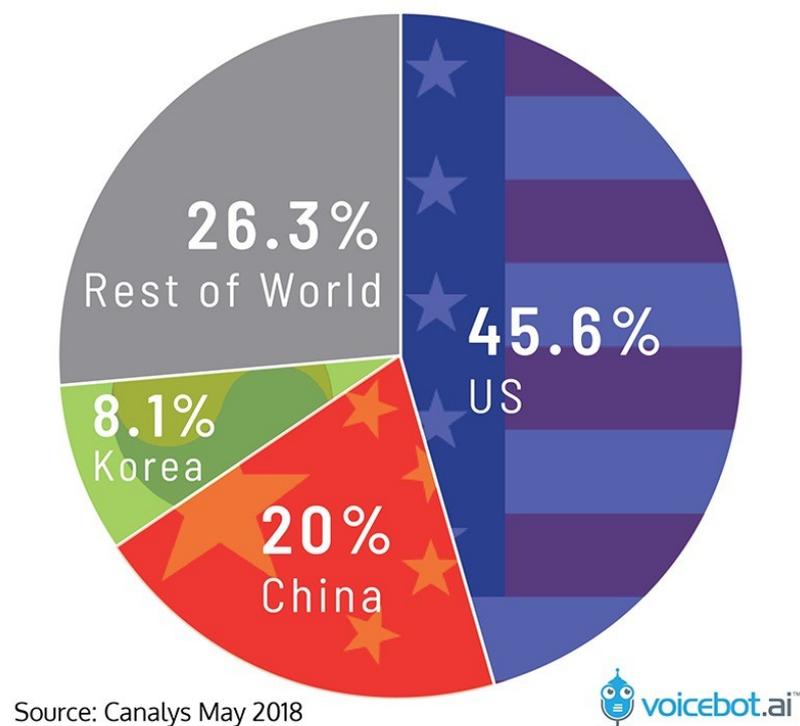
- Artificial Intelligence & Big Data
- Wake word detection
- Text to speech (TTS)
- Speech to text (STT)
- Board bring-up
- 3rd party applications

Smart Speaker Market

U.S. Smart Speaker Market Share - May 2018



Global Smart Speaker Sales Q1 2018



Public statistics from <https://voicebot.ai/>

Amazon Alexa

Amazon Alexa

- Virtual assistant powered by AI and developed by Amazon
- Available for Fire OS, iOS and Android
- Powers Amazon devices such as Echo smart speakers
- Initial release November 2014
- Requires Amazon Alexa app on a smartphone to setup the smart devices

Alexa Features

- Multilingual support
- Supports several wake words (limited number of options)
- Voice profile for personalized experience
- Alexa to Alexa calling and messaging
- Mobile or landline calls in the US, Canada, and Mexico

Developer opportunities with Amazon Alexa

- Integration of Amazon Alexa in custom hardware devices using an **SDK**
- Extend the functionality of Alexa compatible devices by developing software applications called **Skills**

Amazon Smart Speakers

- Amazon Echo
- Amazon Echo Dot
- Amazon Echo Spot
- Amazon Echo Tap
- Amazon Echo Show
- Amazon Echo Plus



3rd Party Devices with Alexa

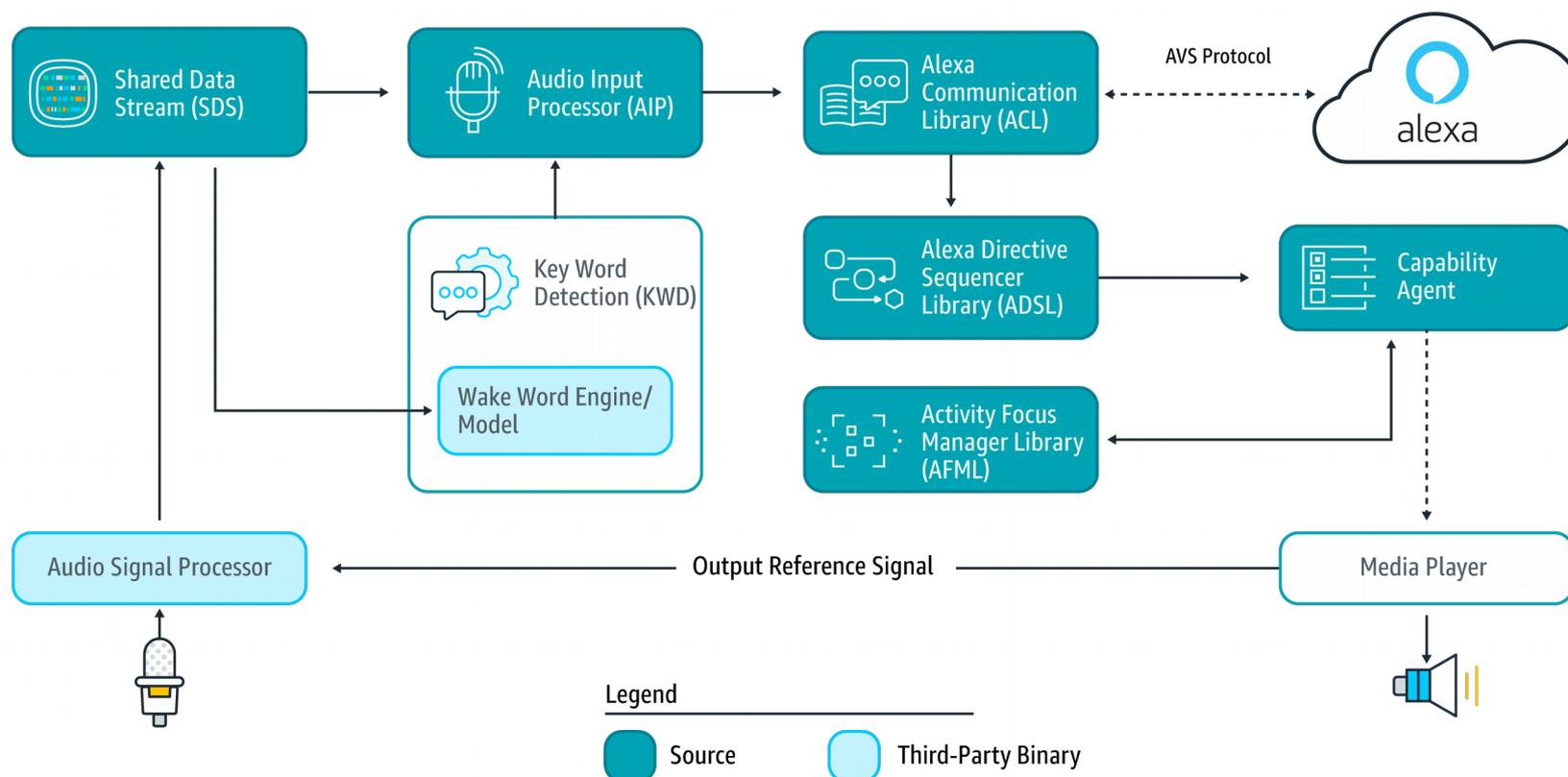
- Ecobee4 Smart Thermostat
- Element EL4KAMZ17 (Amazon Fire TV Edition)
- Eufy Genie
- Fabriq Chorus and Riff
- Garmin Speak
- C by GE Sol Smart Lamp
- iHome AVS16 Alarm Clock
- LG Hub Robot
- More ...

AVS Device SDK

- Alexa Voice Service (AVS)
- An easy-to-use SDK for commercial device makers to integrate Alexa into connected products
- Available at GitHub under Apache License 2.0
<https://github.com/alexav/avs-device-sdk>
- 69 commits, 24 releases, 23 contributors
- Provides C++-based (11 or later) libraries
- Compatible with Android, macOS, Windows 64-bit, GNU/Linux distributions such as Ubuntu, Raspbian

How does AVS SDK work?

■ SDK Architecture



AVS on Raspberry Pi

- Assemble Raspberry Pi with mic and a speaker
- Install Raspbian OS on microSD card and boot it
- Download the AVS Device SDK
- Input your AVS credentials
- Build the AVS Device SDK
- Get a Refresh Token to maintain a connection with the cloud
- Run the sample application
- Details at:
<https://developer.amazon.com/docs/alexa-voice-service/set-up-raspberry-pi.html>

3rd Party Dev Kits

Recommended 3rd party development kits for prototyping:

- XMOS VocalFusion 4-Mic Kit
- Synaptics AudioSmart 2-Mic Dev Kit for Amazon AVS with NXP SoC
- Intel Speech Enabling Developer Kit
- Amlogic A113X1 Far-Field Dev Kit for Amazon AVS
- Allwinner SoC-Only 3-Mic Far-Field Dev Kit for Amazon AVS
- DSPG HDClear 3-Mic Development Kit for Amazon AVS

Launching Commercial Devices with AVS on the Market

- Agree to AVS Terms and Agreements
- Product testing with self-test checklists to ensure your product delivers a familiar Alexa experience to customers
- Alexa Built-in badge for compliance with security and testing requirements
- Details at:
<https://developer.amazon.com/alexa-voice-service/launch>

Google Assistant

Konsulko
Group

Google Assistant

- Virtual assistant powered by AI and developed by Google
- Available for numerous platforms, mobile and smart home devices
- Initial release 18 May 2016
- Written in C++
- Requires Google Home app on a smartphone to setup a smart speaker with Google Assistant

Google Assistant Features

- Multilingual support
- Six different voice options (including both male and female)
- Continued conversation for follow-up questions without repeating the activation word
- Voice match feature to setup up to 6 users of the smart speaker
- Google Duplex extension for accomplishing real-world tasks through natural conversations over the phone

Developer opportunities with Google Assistant



- Integration of Google Assistant in custom hardware devices using the **SDK**
- Extend the functionality of Google Assistant by developing software applications called **Actions**

Google Assistant SDK

- **Google Assistant Library** written in **Python** and supported on devices with **linux-armv7l** and **linux-x86_64** architectures
- **Google Assistant Service** for full control over the integration through API binding on all platforms supported by the modern open source high performance RPC framework **gRPC**

Compatibility and feature support

	Library	Service
Supported architectures	linux-armv7l and linux-x86_64	All gRPC platforms
Supported languages	Python	All gRPC languages
Hands-free activation (Ok Google)	Yes	No
Audio capture and playback	Built in	Reference code is provided
Conversation state management	Built in	Reference code is provided
Timers and alarms	Yes	No
Playback of podcasts and news	Yes	No
Broadcast voice messages	Yes	No
Visual output (HTML5) of Assistant responses	No	Yes

Google Smart Speakers

- **Google Home** - released on 4 November 2016
- **Google Home Mini** - announced on October 4, 2017, released on the market on October 19, 2017
- **Google Home Max** - announced on 4 October 2017, released on the market on 11 December 2017
- **Google Home Hub** - announced on 9 October 2018



3rd Party Devices with Google Assistant

- Panasonic GA10
- Sony LF-S50G
- TicHome Mini
- Polk Assist
- Hogar Milo
- LG ThinQ WK7
- JBL Link
- Lenovo Smart Display
- More...

Building Your Own Device (1/5)

Low cost hardware options using off-the-shelf components for proof of concept demo:

- Google Voice Kit for Raspberry Pi:
<https://aiyprojects.withgoogle.com/voice/>
- Raspberry Pi, Adafruit I2S MEMS Microphone Breakout – SPH0645LM4H and Adafruit I2S 3W Class D Amplifier Breakout – MAX98357A
<https://www.adafruit.com/product/3421>
<https://www.adafruit.com/product/3006>
- OrangePi Zero Set 6 (includes a case and an expansion board with mic and audio input) + Speaker
<http://www.orangepi.org/orangepizero/>

Building Your Own Device (2/5)

- Create new project in Google Platform Console
- Enable Google Assistant API
- Create credentials for OAuth Client ID and download JSON file
- On Debian distribution install Python:

```
export LC_ALL="en_US.UTF-8"  
export LC_CTYPE="en_US.UTF-8"  
sudo dpkg-reconfigure locales  
sudo apt-get update  
sudo apt-get install -y python3-dev python3-venv  
python3 -m venv env  
env/bin/python -m pip install --upgrade pip setuptools
```

Building Your Own Device (3/5)

- Activate virtual Python environment and install Google Assistant SDK:
`python -m pip install --upgrade google-assistant-library`

- Install and run Google authorization tool:

```
python -m pip install --upgrade google-auth-oauthlib[tool]  
google-oauthlib-tool --client-secrets  
~/client_secret_xxxx.apps.googleusercontent.com.json --scope  
https://www.googleapis.com/auth/assistant-sdk-prototype --  
save --headless
```

- Start Google Assistant:
`google-assistant-demo --device_model_id "my-speaker"`

Building Your Own Device (4/5)

- Create systemd service **/etc/systemd/system/google-assistant-demo.service** to launch Google Assistant automatically at startup:

```
[Unit]
Description=google assistant service
After=network.target ntpdate.service
[Service]
Type=simple
Environment=VIRTUAL_ENV=/home/pi/env/
Environment=PATH=/home/pi/env/bin:/usr/local/sbin:/usr/local
ExecStart=/home/pi/env/bin/google-assistant-demo --device_model_id "orangeypi"
WorkingDirectory=/home/pi
StandardOutput=inherit
StandardError=inherit
Restart=always
User=pi
[Install]
WantedBy=multi-user.target
Alias=google-assistant.service
```

Building Your Own Device (5/5)

- Enable the systemd service and start it automatically at boot:

```
sudo systemctl daemon-reload  
sudo systemctl enable google-assistant-demo.service  
sudo systemctl start google-assistant-demo.service
```

Launching Commercial Devices with Google Assistant on the Market



- Commercial products with Google Assistant requires certification by Google
- Get started by filling in a form with basic information about the technical specifications of the product:
<https://developers.google.com/assistant/sdk/interest>

Mycroft

Konsulko
Group

- Entirely open source project for a voice assistant
<https://mycroft.ai/>
- Git repositories in GitHub
<https://github.com/MycroftAI>
- Open source software available under Apache License 2.0
- Open source hardware available under CERN Open Hardware Licence
<https://github.com/MycroftAI/hardware-mycroft-mark-1>
- Certified open source hardware UID US000049

Mycroft AI Inc.

- US start-up company from Kansas City founded in 2015 by Joshua Montgomery
- Initially products were crowdfunded through Kickstarter and IndieGoGo
- Currently is accepting investments at StartEngine

Mycroft Pulse

Mycroft Core, the Mycroft Artificial Intelligence platform
<https://github.com/MycroftAI/mycroft-core>

- Written in Python
- 2820 commits, 90 releases, 89 contributors
- 8 contributors with more than 100 commits

Skills, repository for sharing and collaboration for 3rd party Mycroft skills development

<https://github.com/MycroftAI/mycroft-skills>

- 965 commits, 65 contributors (as of 17 October)

Mycroft Features

- Officially available only in English, community support for other languages
- Supports extension of the functionality by developing software applications called skills
- Mycroft Skills Manager (msm) and a repository with 3rd party skills
- Optional device and account management system known as Mycroft Home
- Allows using devices without Mycroft Home service

Mycroft Wake Words

- Default wake words “Hey Mycroft”
- Support for custom wake words
- Precise - default wake word listener since March 2018, written in Python, available at GitHub under Apache License 2.0:
<https://github.com/MycroftAI/mycroft-precise>
- PocketSphinx the previous default wake word listener, now available as an alternative in Mycroft, written in C, available at GitHub under BSD-like license:
<https://github.com/cmusphinx/pocketsphinx>

Mycroft STT engines

The following Speech to Text (STT) engines are available:

- Google STT (default)
- IBM Watson Speech to Text (username and password required)
- wit.ai Speech to Text (wit.ai API key required)
- DeepSpeech (work in progress as part of the OpenSTT initiative in partnership with Mozilla)

DeepSpeech

- Open source Speech-to-Text (STT) engine developed by Mozilla
- Available at GitHub under Mozilla Public License 2.0
<https://github.com/mozilla/DeepSpeech>
- Written in C++, Python and shell scripts, uses Google's TensorFlow to simplify the implementation
- 50 contributors, 1220 commits, 19 releases

Mycroft TTS engines

The following Text to Speech (TTS) engines are available:

- Mimic (default)
- eSpeak
- MaryTTS
- Google TTS
- FATTS

Mimic

- Fast, light-weight Text to Speech (TTS) engine developed by Mycroft.AI and VocaliD
- Based on Carnegie Mellon University's FLITE software
- Available at GitHub under BSD-like copyright (as FLITE): <https://github.com/MycroftAI/mimic>
- 348 commits, 6 releases, 14 contributors (as of 21 October)
- Written in C
- Works on GNU/Linux distributions, Mac OS X and MS Windows

Mycroft Devices

- Mycroft Mark 1 (crowdfunded in 2015, shipped in 2017)
- Mycroft Mark 2 (expected in December 2018)
- DIY smart speakers with Raspberry Pi 2 & 3 (expected support for 3 B+) with PiCroft GNU/Linux distribution based on Raspbian Jessie Lite

Showcases

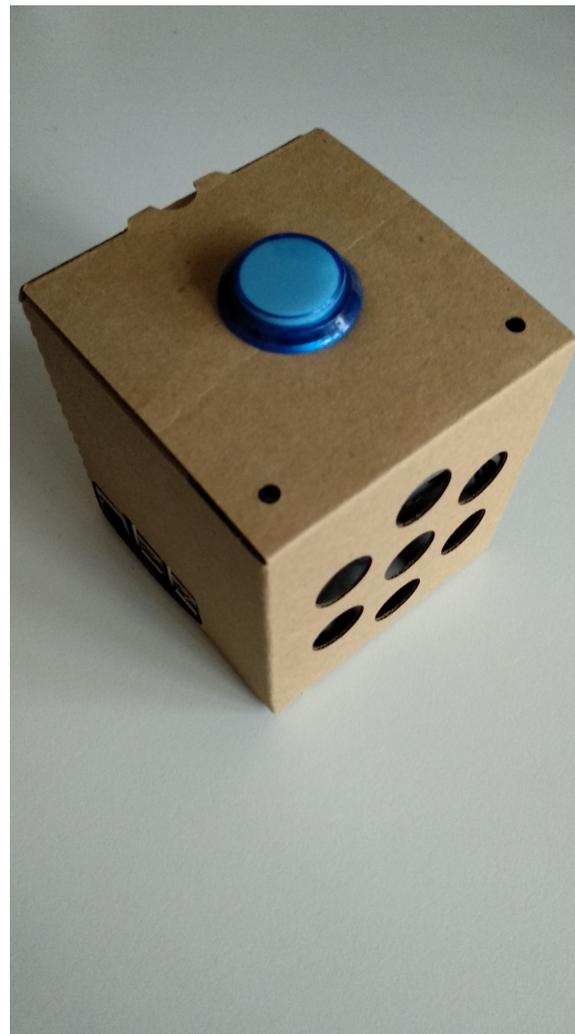
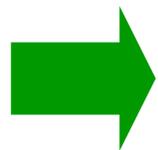
Konsulko
Group

Google Voice Kit for Raspberry Pi

- Do-it-yourself artificial intelligence voice for Raspberry Pi
<https://aiyprojects.withgoogle.com/voice/>
- Two versions, the first was distributed free with the MagPi magazine
- Cardboard case

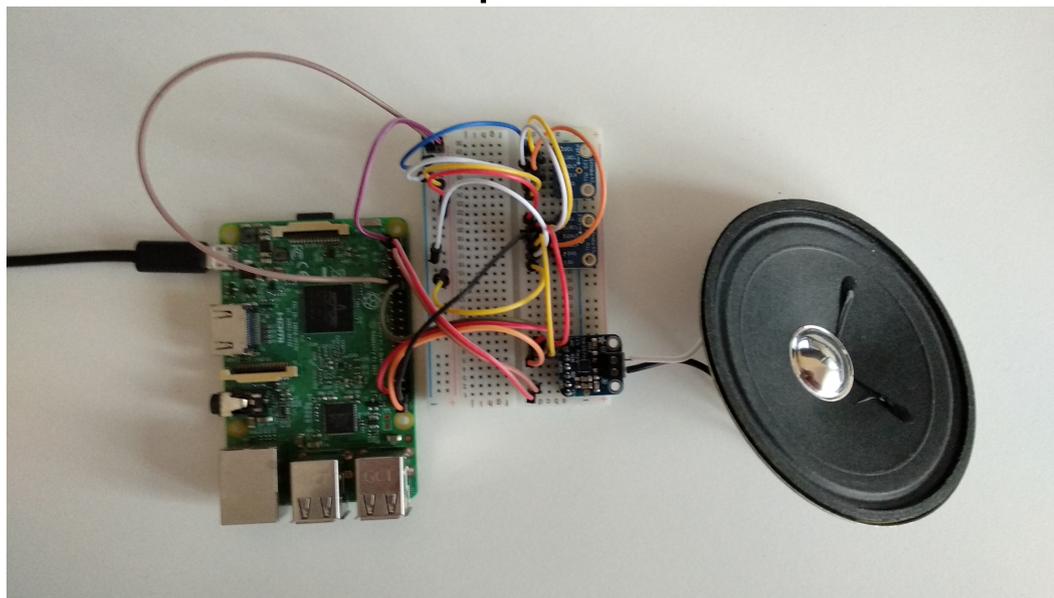


Google Voice Kit for Raspberry Pi



Raspberry Pi + Mic + Speaker

- Raspberry Pi
- Adafruit I2S MEMS Microphone Breakout - SPH0645LM4H
<https://www.adafruit.com/product/3421>
- Adafruit I2S 3W Class D Amplifier Breakout - MAX98357A
<https://www.adafruit.com/product/3006>



Orange Pi Zero Set6

- Orange Pi Zero with Allwinner H2 SoC and 512MB RAM
- Expansion board with audio input, mic, IR receiver and two additional USB ports
- Speaker
- Case
- 5V microUSB power supply
- Armbian GNU/Linux distribution
<https://www.armbian.com/orange-pi-zero/>



Home Assistant

- Open-source home automation platform running on Python 3
- Perfect to run on a Raspberry Pi
- More than 950 components for integration with popular Internet of Things such as IKEA Trådfri, Philips Hue, Google Assistant, Alexa / Amazon Echo, Nest, KODI, etc.
- Started in 2013 by Paulus Schoutsen
- Huge community, more than 830 contributors
- Source code available at GitHub under Apache 2.0 license
- <https://home-assistant.io/>



Simple Voice Control with Alexa

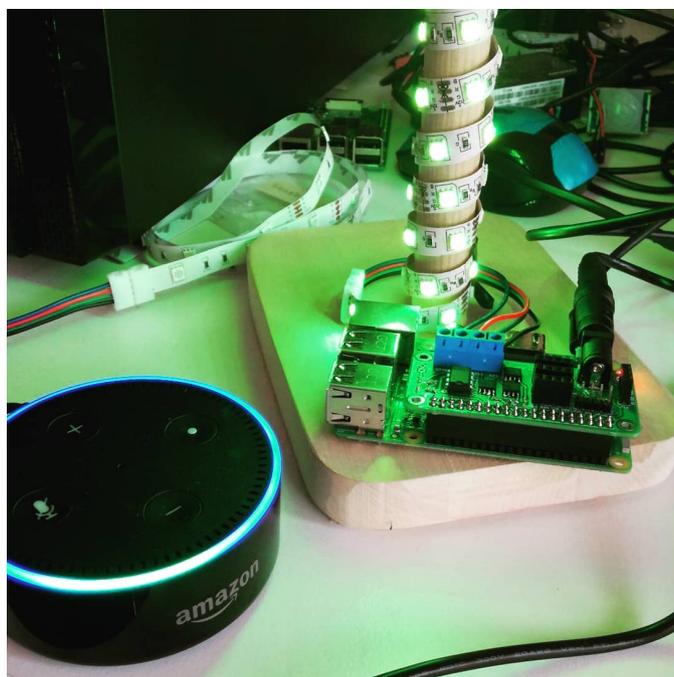
- Home assistant is compatible with Alexa and Amazon Echo
- Basic integration using the Emulated Hue Bridge component of Home Assistant
- Emulated Hue Bridge allows non-Philips Hue devices to be controlled though with voice the built-in support of Amazon Echo



```
emulated_hue:  
  type: alexa  
  expose_by_default: true
```

Simple Voice Control with Alexa

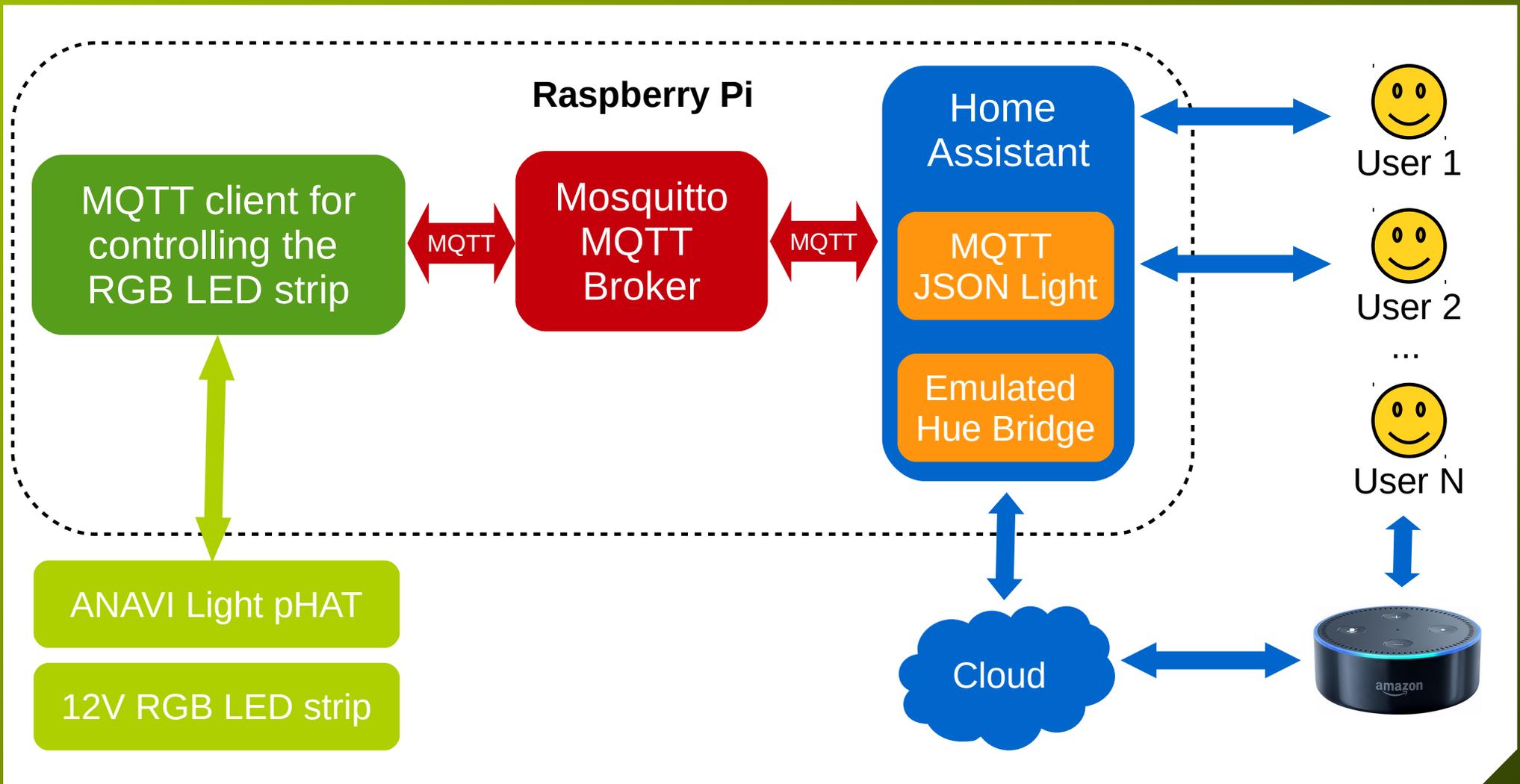
- Example voice commands for MQTT JSON Light component configured with name “ANAVI Light pHAT” in **configuration.yaml**:



Alexa, turn **ON** ANAVI Light pHAT

Alexa, turn **OFF** ANAVI Light pHAT

How Does it Work?



Conclusions

Conclusions

- There is a huge demand on the market for integrating AI and voice assistants in end-consumer devices as well as for development of 3rd software applications and services for them
- The market leaders Amazon and Google provide turn-key solutions for integration in embedded Linux devices but require difficult certifications for end-consumer devices
- Mycroft is an entirely open source voice assistant that combines open source hardware with free and open source software
- In practice all reviewed solutions require Internet connection to the cloud

Thank You!

Useful links:

- <https://developer.amazon.com/alexa/devices>
- <https://developer.amazon.com/alexa-skills-kit>
- <https://developer.amazon.com/docs/ask-overviews/build-skills-with-the-alexa-skills-kit.html>
- <https://www.youtube.com/watch?v=F5DixCPJYo8>
- <https://developers.google.com/actions/>
- <https://developers.google.com/assistant/sdk/>
- <https://www.youtube.com/watch?v=FBXRwu6hgy8>
- <https://mycroft.ai/>
- <https://www.armbian.com/>

