wit.ai

FACEBOOK

Building Immersive Voice-XR Experiences with Wit.ai

Pan Wangperawong, Facebook Reality Labs

About Me

Pan Wangperawong

Partner Engineer, Facebook Reality Labs

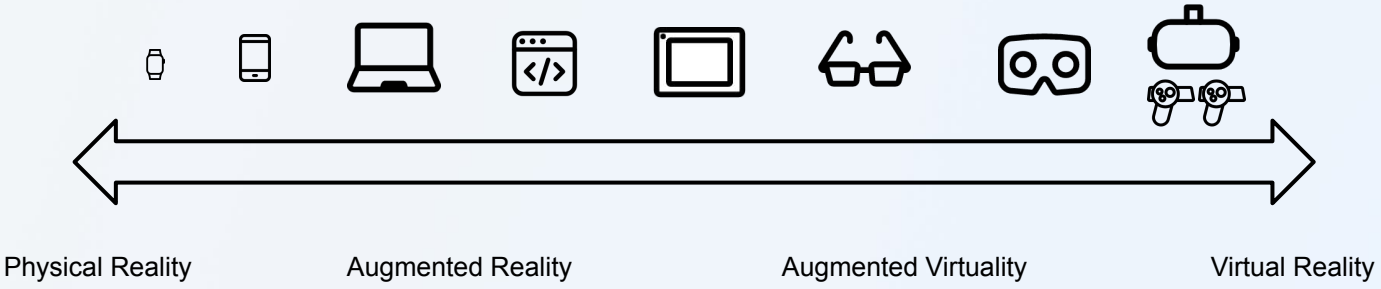
Bio

Over 10 years of experience working at the intersection of product, engineering, and developer experience in the areas of natural user interfaces, Artificial Intelligence, and XReality.

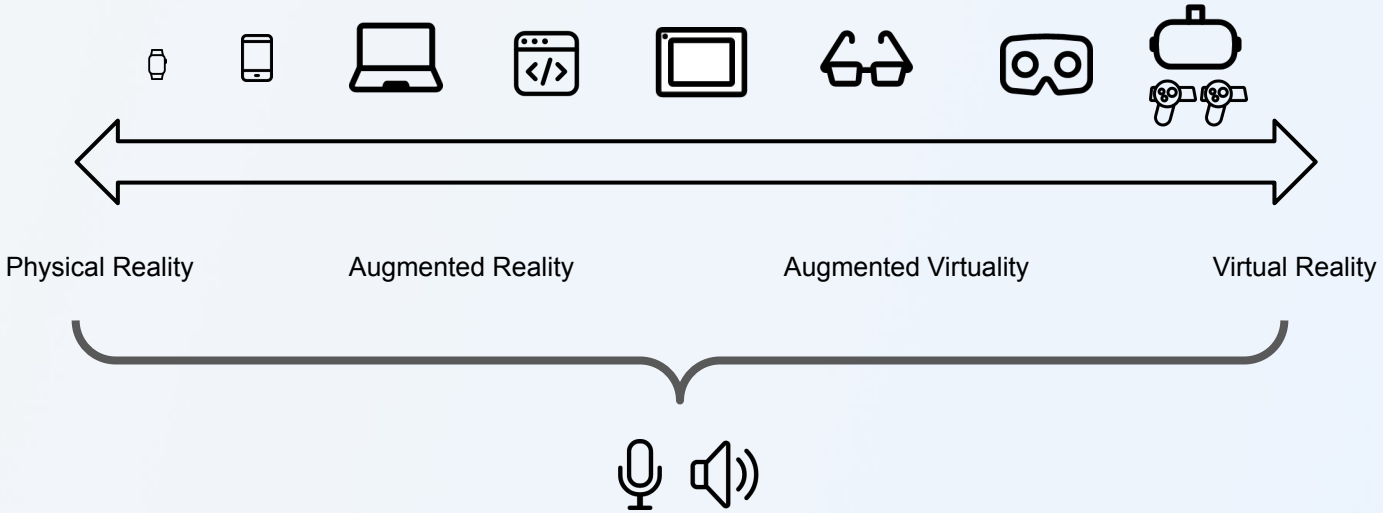
Agenda

1. Understand the Reality-Virtuality Continuum
2. Explore Benefits of Voice User Interfaces (VUIs) and its place in the Reality-Virtuality Continuum
3. Learn how Wit.ai can be used with various technologies to build Voice-XR experiences

The Reality-Virtuality Continuum



The Reality-Virtuality Continuum



Benefits of Using Voice 🗣️

- Lowers the adoption barrier
- Adds interactivity
- Helps with multitasking
- Provides accessibility

Use Cases that Greatly Benefits from Voice

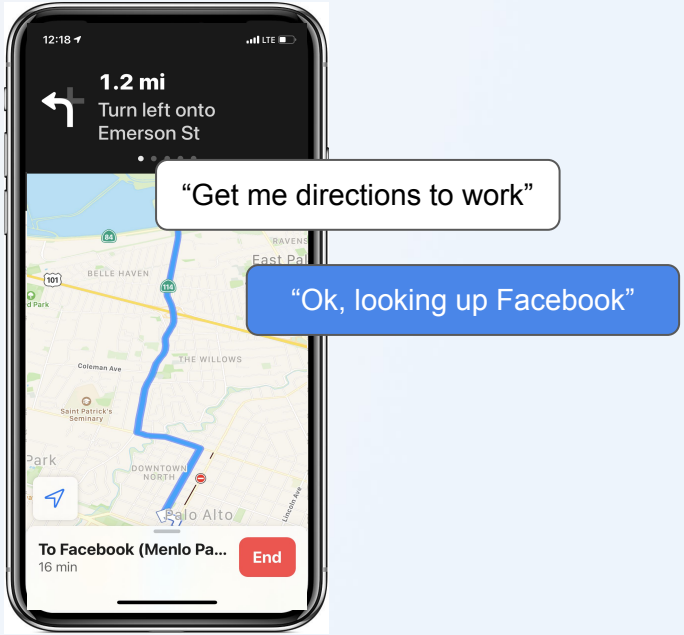
- Quick tasks
  - System and in-app controls e.g. game controls, take a screenshot, media playback
  - Information retrieval e.g. friend's status, knowledge search, weather, stock price, etc.
  - Scheduling e.g. reminders, timers, events
- Interactivity
  - Interactive avatar or game element
  - Multi-turn games e.g. RPG games
  - Guided activities e.g. ordering, workouts, installation
- Accessibility
  - Alternative input option to accommodate different user needs
  - Hands-free and distant independent interaction for people with impaired motor capabilities e.g. voice dictation
  - Visual free interaction for people with visual impairment e.g. screen reader

Physical-Augmented Reality: Areas to Naturally Incorporate a Voice Feature

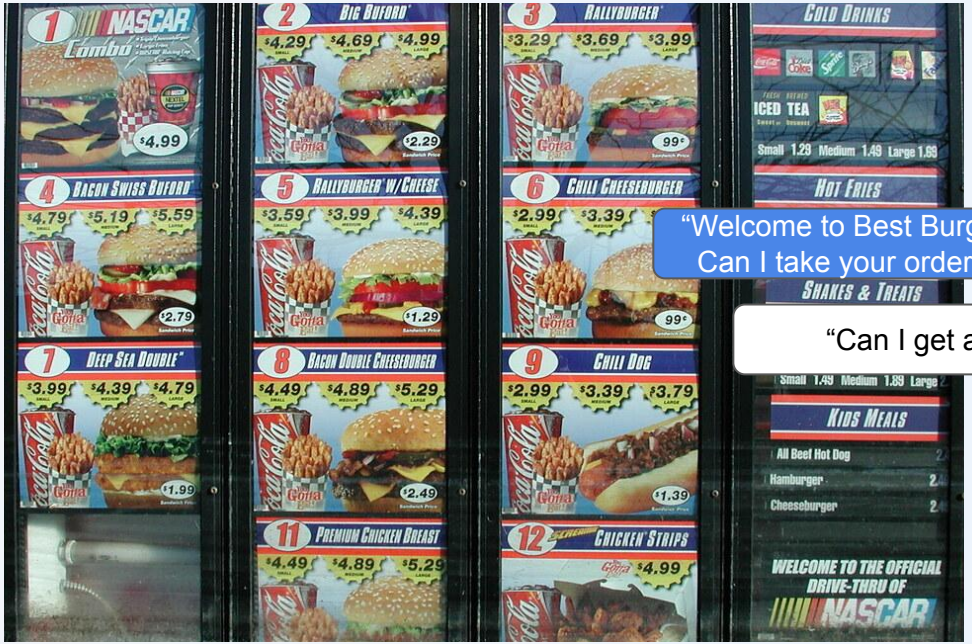
- Guided activities
- Quick tasks
- Freezing temperatures
- Tasks that does not require visual attention
- Tasks where other modalities of interaction are out of reach

Physical-Augmented Reality: Areas to Naturally Incorporate a Voice Feature

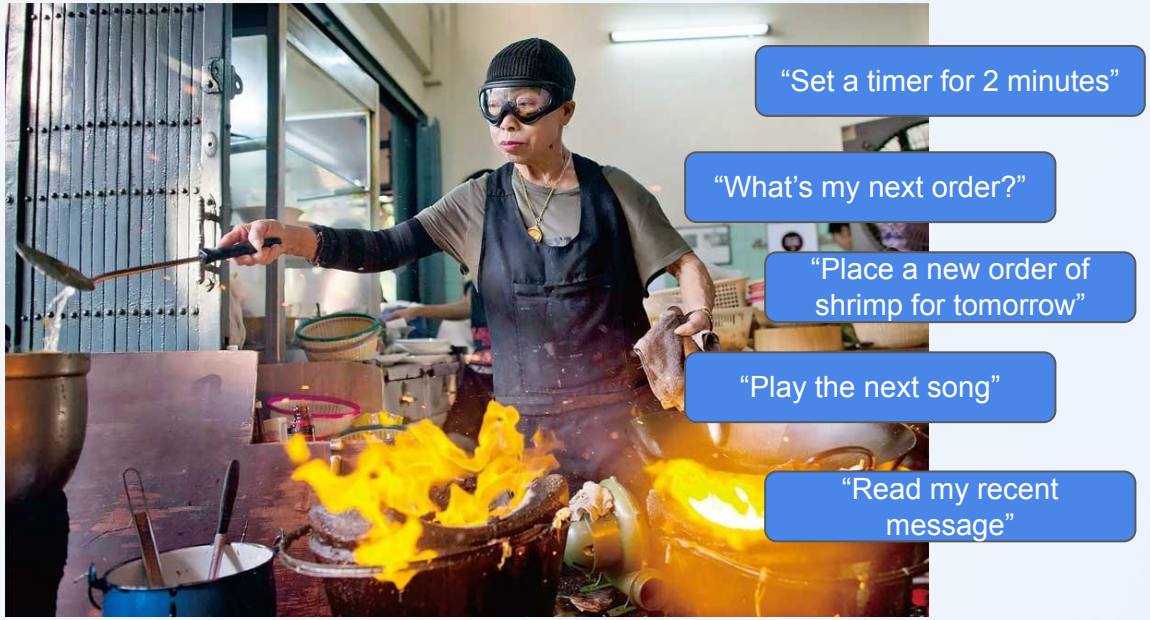
Physical-Augmented Reality: Areas to Naturally Incorporate a Voice Feature



Physical-Augmented Reality: Areas to Naturally Incorporate a Voice Feature



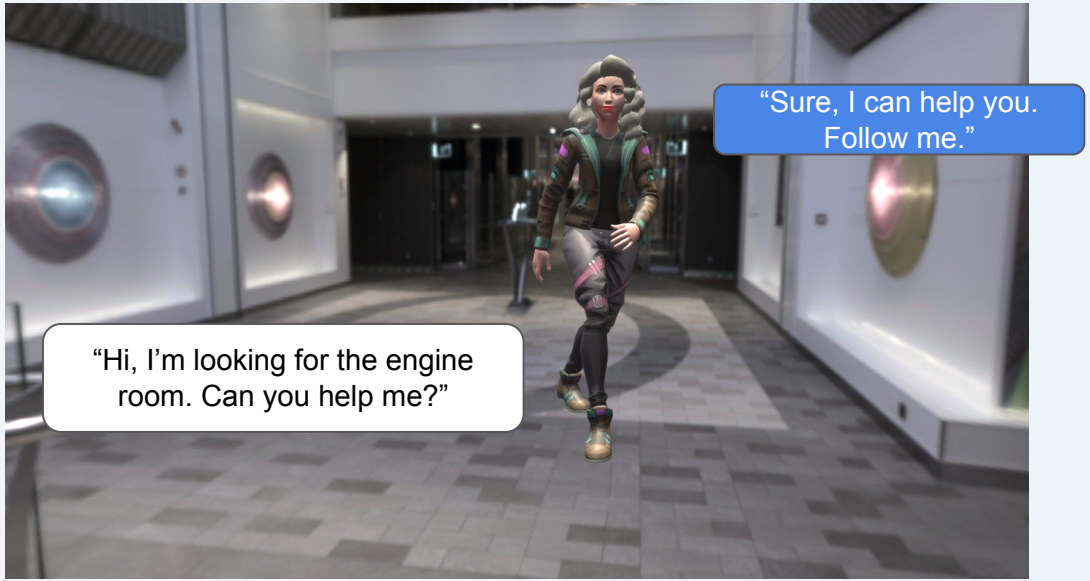
Physical-Augmented Reality: Areas to Naturally Incorporate a Voice Feature



Augmented-Virtual Reality: Areas to Naturally Incorporate a Voice Feature

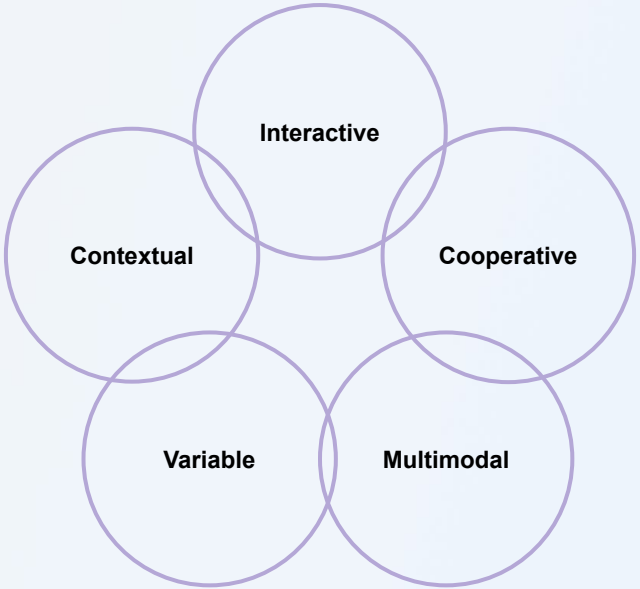
- Mostly everything mentioned previously for the virtual context
- Game controls
- Interactions with virtual characters

Augmented-Virtual Reality: Areas to Naturally Incorporate a Voice Feature



Conversation Design

*A set of ideas on how a computer can socialize naturally and effectively with **people**.*



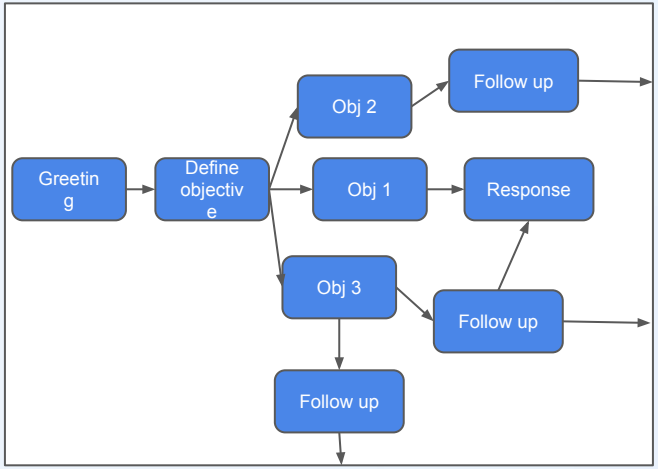
Frameworks for Conversation Design

Script

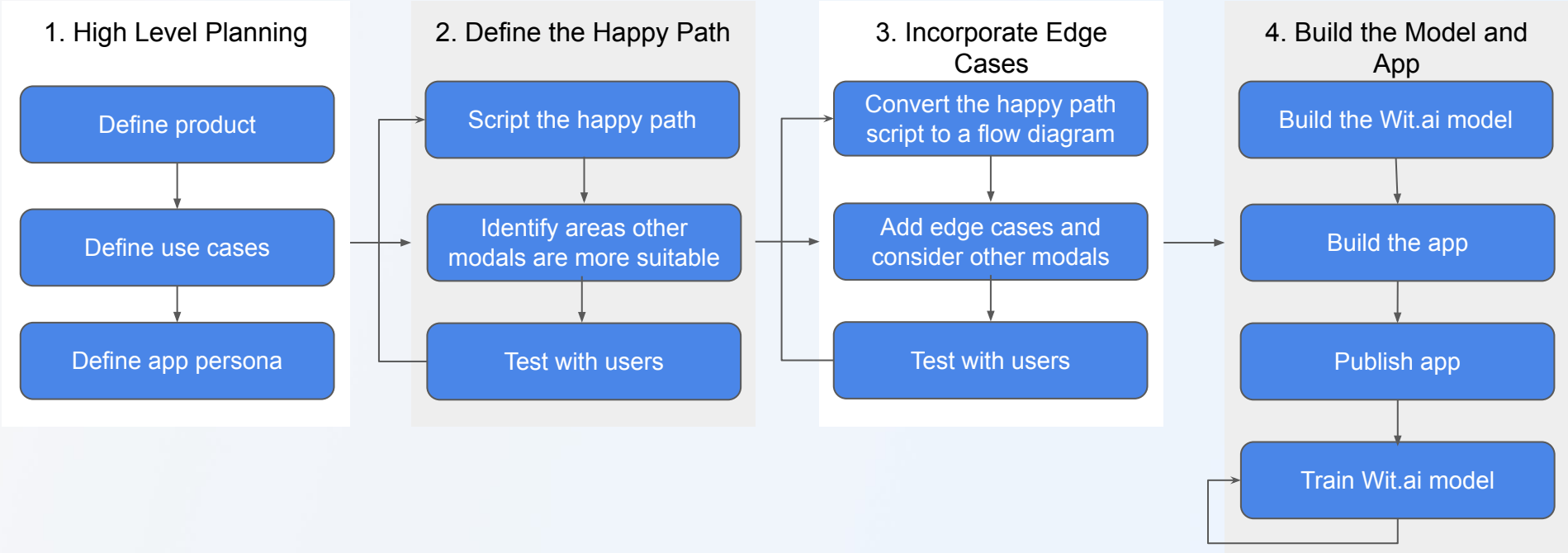
Customer service system:
Hi, welcome to Cool Computers. How can I help you? You can ask me about store hours or to check the status of your order.
Customer:
Check order status
Customer service system:
Since you are calling from a number associated with an order I was able to look up your orders and found one for a Facebook Portal. It is expected to arrive on Wednesday, November 1st at 2pm. Would you like me to email you the tracking information for your reference to pan@mail.com?
Customer:
Yes, that would be great!
Customer service system:
Ok, you should receive an email shortly! Is there anything else I can help you with?
Customer:
No, that's everything!
Customer service system:
Ok, it was a pleasure assisting you. Please get in contact with us again if you have other questions.

Roleplay

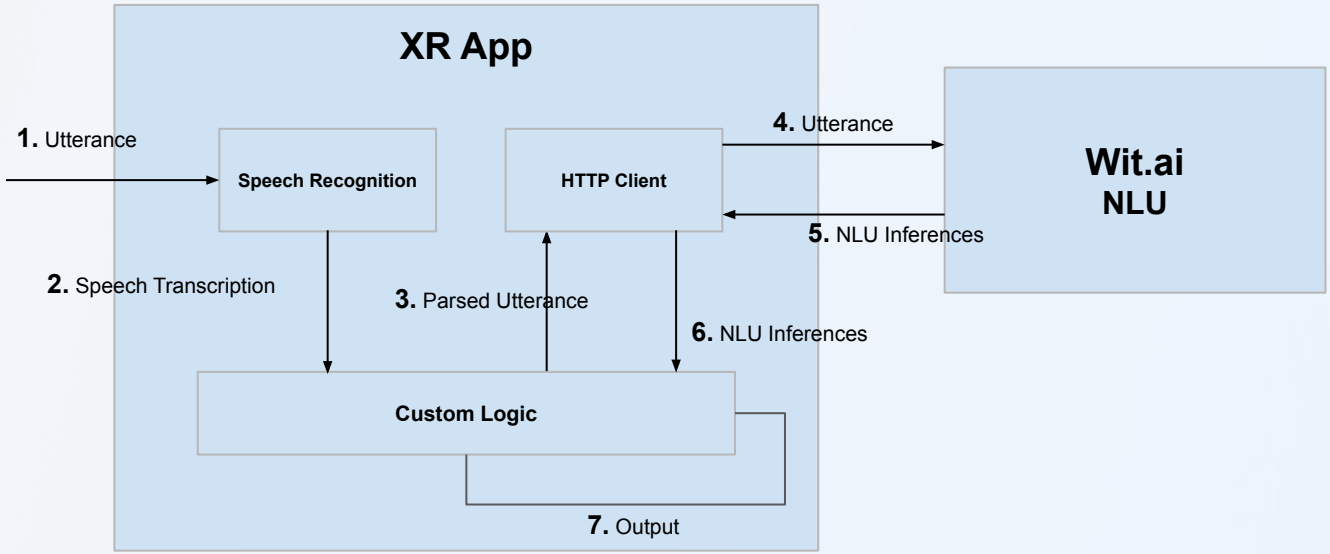
Flow Diagram



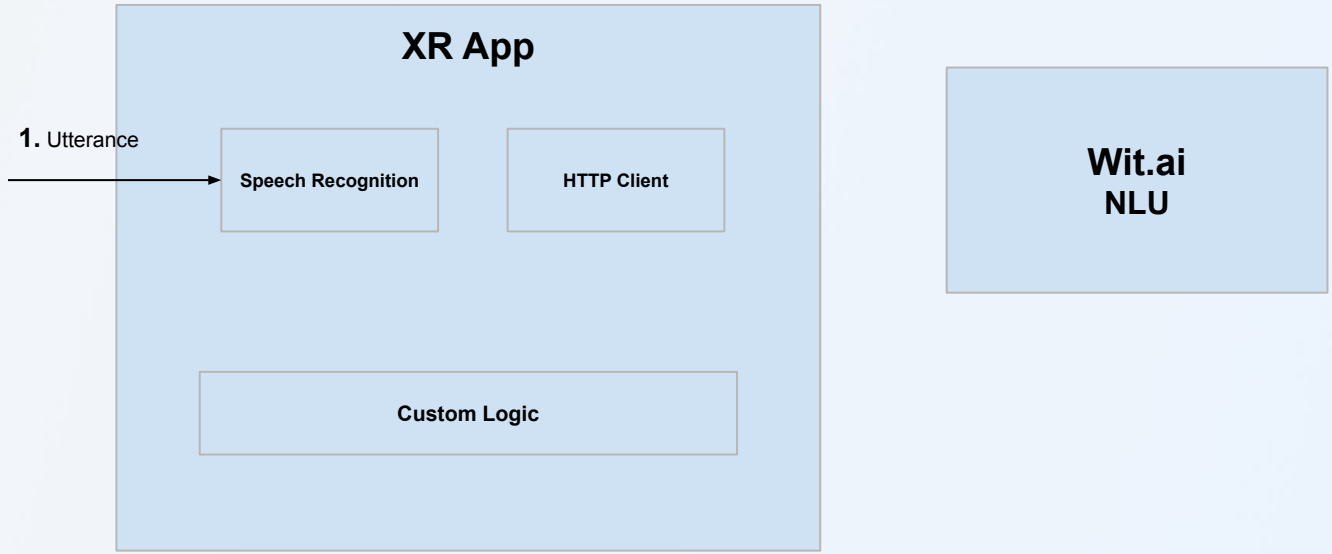
Steps to Build Multimodal Conversational Experiences



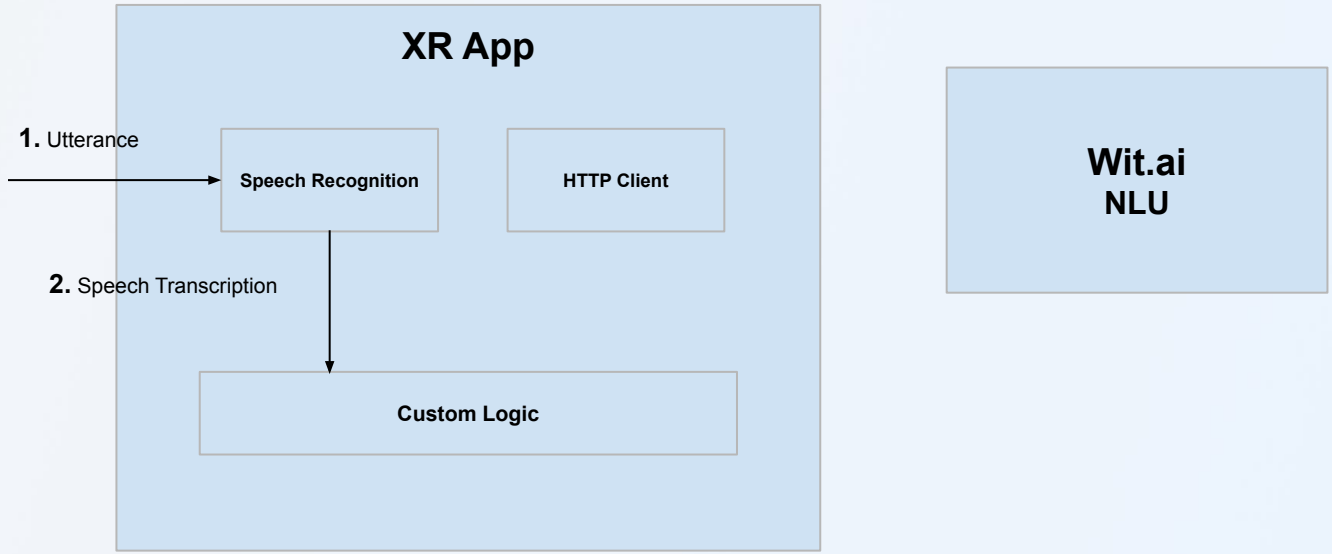
Build VUIs with Wit.ai



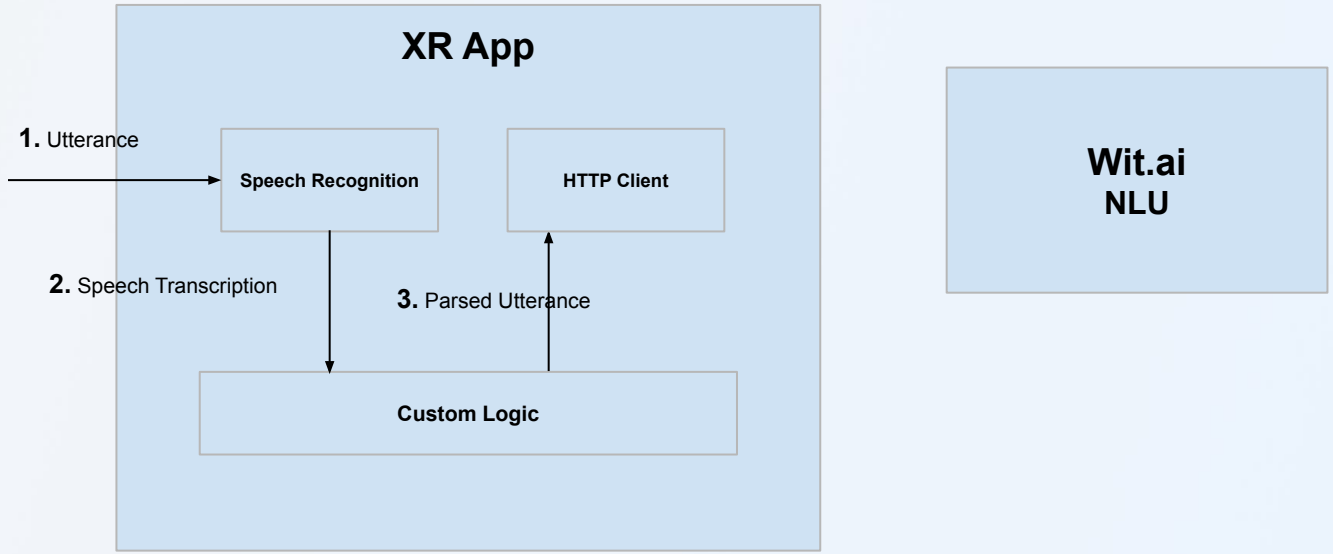
Build VUIs with Wit.ai



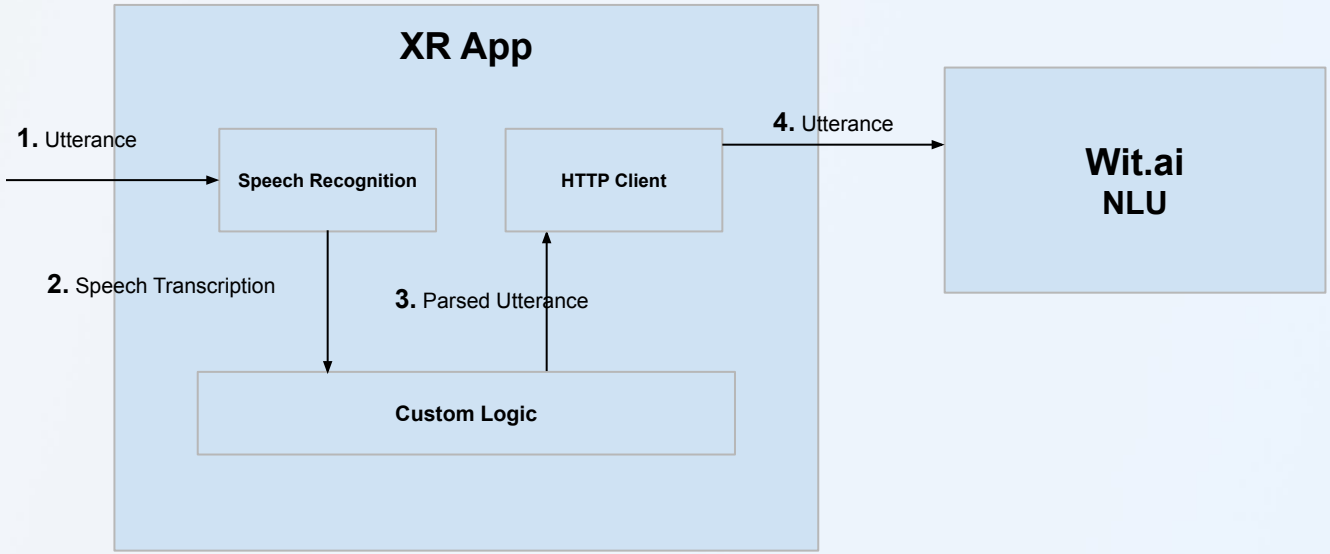
Build VUIs with Wit.ai



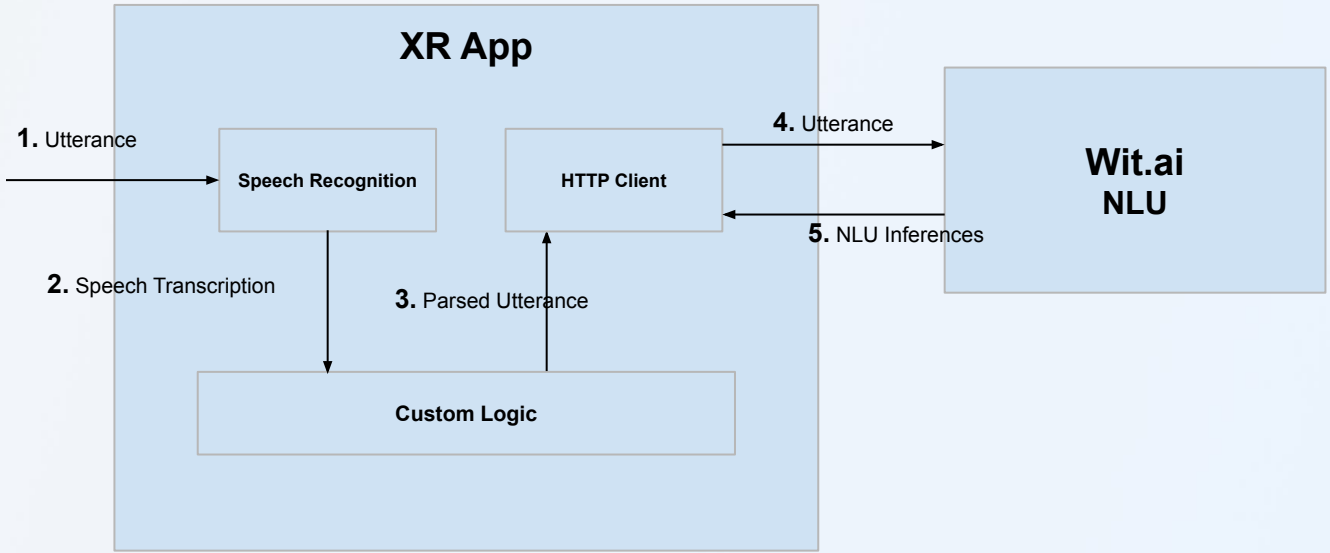
Build VUIs with Wit.ai



Build VUIs with Wit.ai



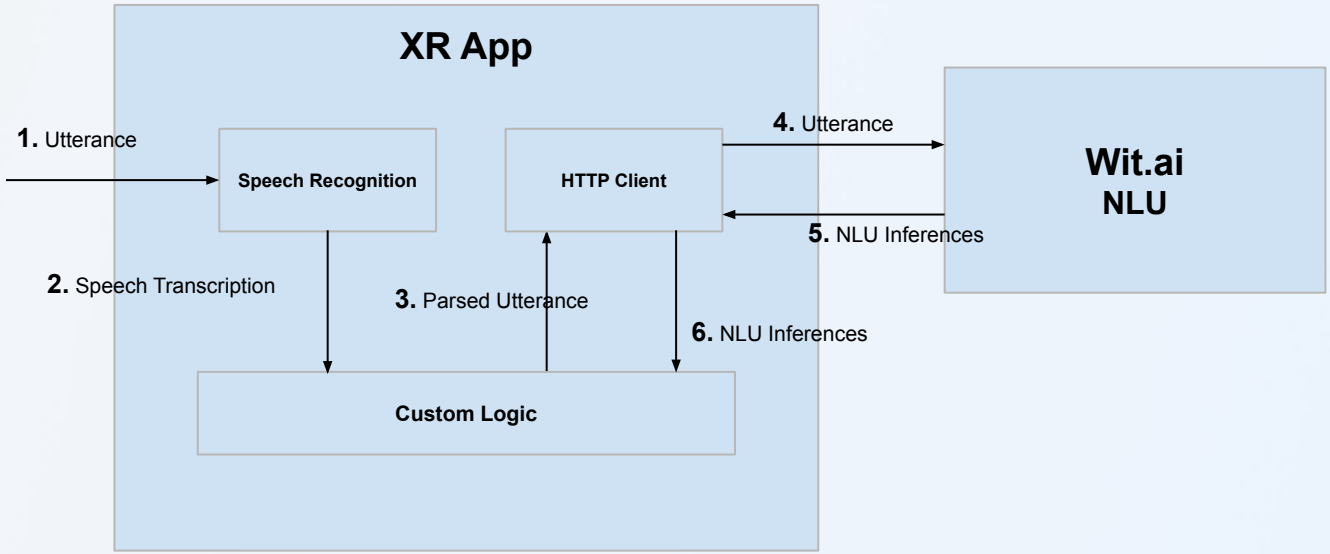
Build VUIs with Wit.ai



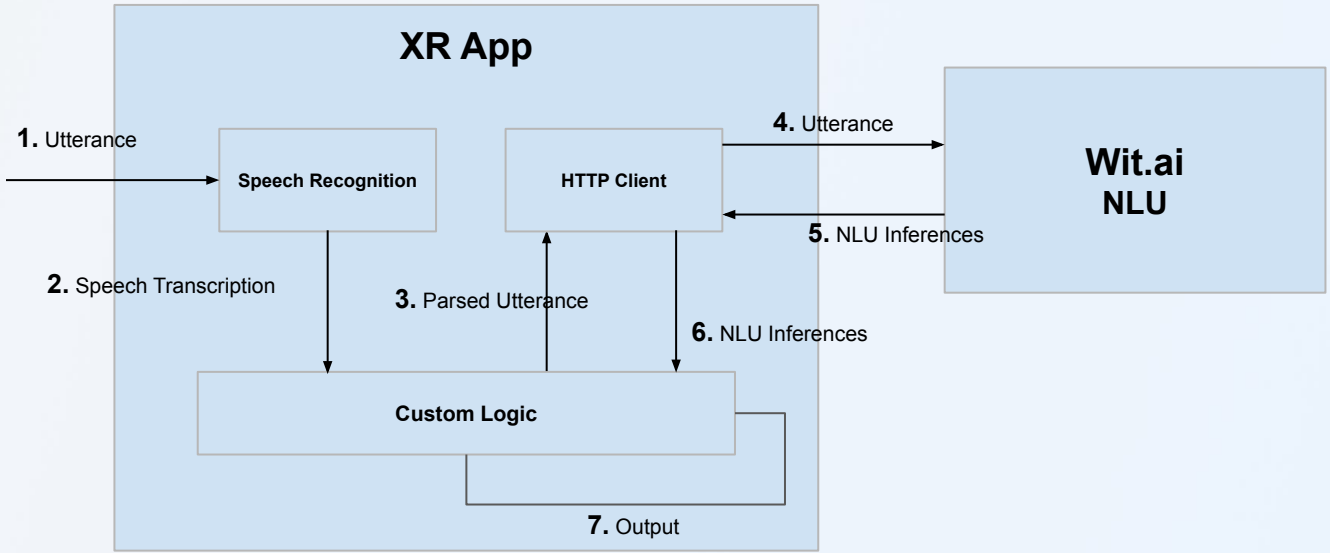
NLU Terminologies

- **Utterances**: Written or spoken message
- **Intent**: Intention of the message
- **Entities**: Key values relevant to the intent of the message

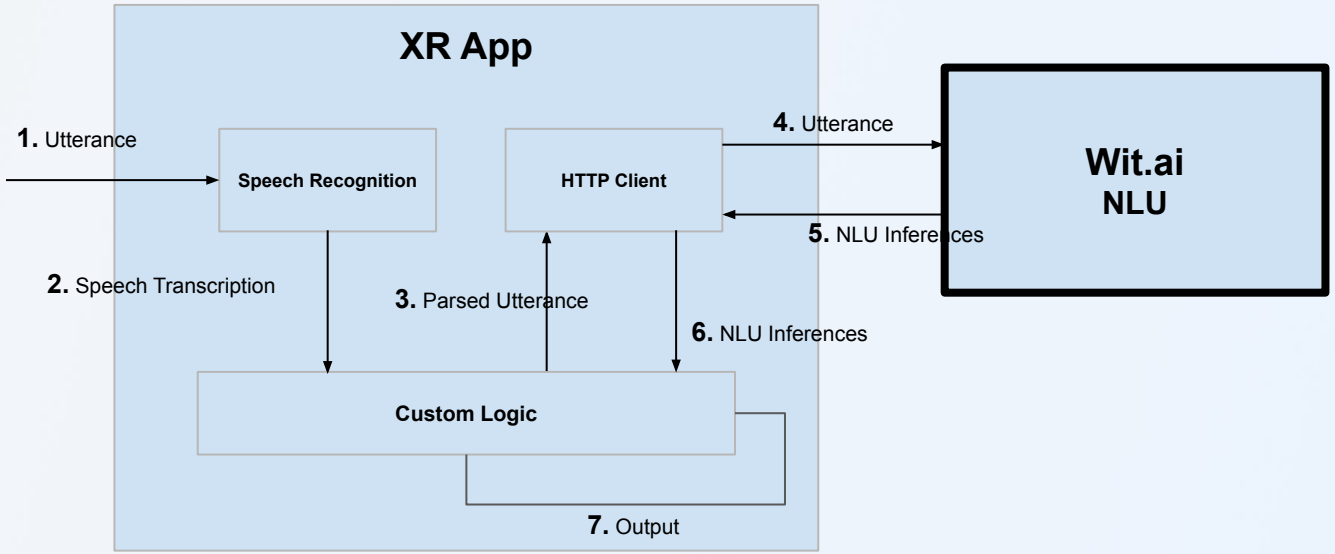
Build VUIs with Wit.ai



Build VUIs with Wit.ai

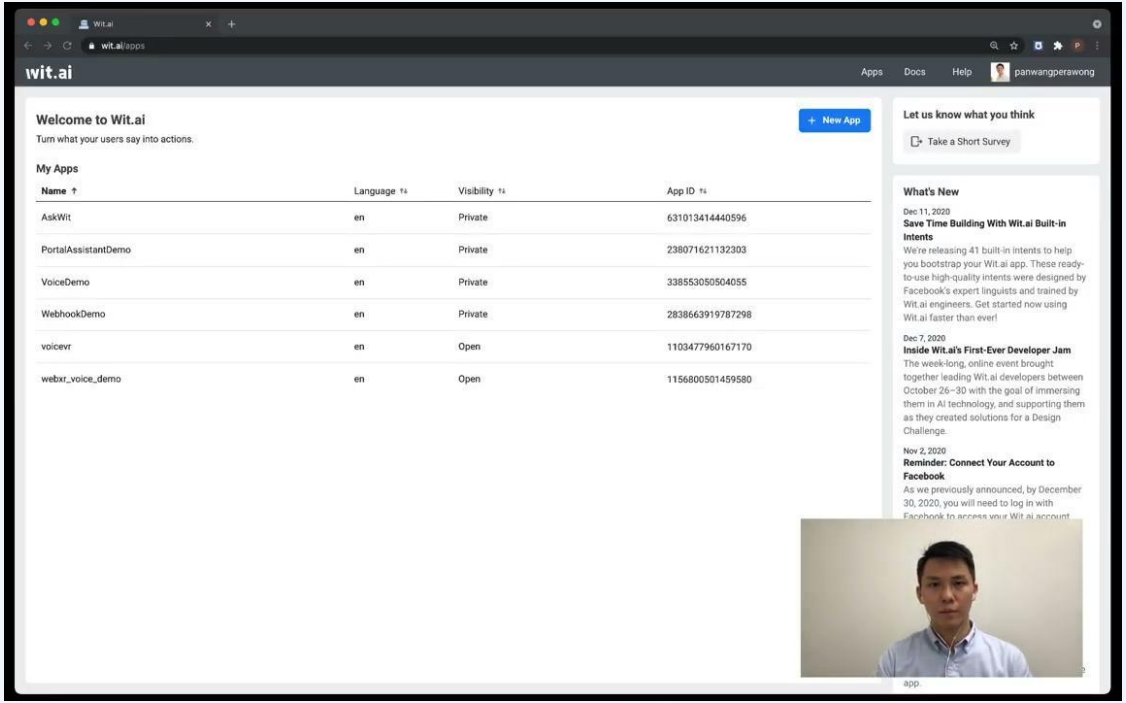


Build VUIs with Wit.ai



Voice-XR Demo

Train a Wit App for NLU



Set the Scene

```
<html>
  <head>
    <script src="https://aframe.io/releases/1.2.0/aframe.min.js"></script>
    <script src="voice.js"></script>
  </head>
  <body>
    <a-scene>
      <a-entity
        id="text-object"
        text="value: Try saying: 'Hey Gizmo, I add a box'; color: #FAFAFA; width: 5; anchor: align"
        position="-2.5 0.2 -3" scale="1.5 1.5 1.5"
        error-message voice-command
      ></a-entity>
      <a-camera>
        <a-cursor></a-cursor>
      </a-camera>
      <a-entity environment="preset: forest; dressingAmount: 500"></a-entity>
    </a-scene>
  </body>
</html>
```

Initialize Web Speech API

```
// Intiatilize an instance of SpeechRecognition from the Web-Speech-API
const SpeechRecognition = window.SpeechRecognition || window.webkitSpeechRecognition;
const recognition = new SpeechRecognition();
recognition.continuous = true;
recognition.lang = 'en-US';
recognition.interimResults = true;
recognition.maxAlternatives = 1;

// Obtain it from your Wit.ai app's Settings page
const CLIENT_TOKEN = "<REPLACE WITH YOUR CLIENT TOKEN>";
```

Use Web Speech API

```
// Register component for voice commands
AFRAME.registerComponent('voice-command', {
  init: () => {
    recognition.start();
    recognition.onresult = (event) => {
      console.log(event.results)
      let utteranceList = event.results;
      let latestUtterance = utteranceList[utteranceList.length-1];
      let speechRecognition = latestUtterance[latestUtterance.length-1];

      // Update text object with speech recognition transcription
      let transcript  = speechRecognition.transcript.toLowerCase();
      let textEl = document.querySelector('#text-object');
      textEl.setAttribute("text", `value:${transcript}`);
    }

    /* Wit.ai integration to be continue ... */
    ...
});
```

Integrate with Wit.ai

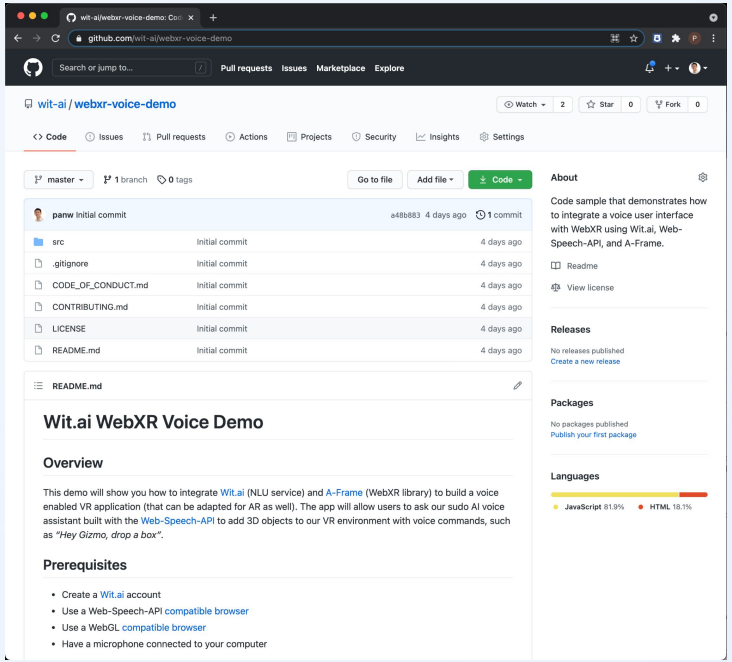
```
// Component to for voice commands
AFRAME.registerComponent('voice-command', {
  init: () => {
    /* Web Speech API recognition */
    ...

    // Extract the utterance from the wake word
    let utterance = transcript.split(`hey ${WAKE_WORD}`)[1];
    // Send the user's utterance to Wit.ai API for NLU inferencing
    fetch(`https://api.wit.ai/message?v=20210414&q=${utterance}`, {
      headers: {Authorization: `Bearer ${CLIENT_TOKEN}`}
    })
    .then(response => response.json())
    .then(json => {
      // Add a 3D object to the scene based on the NLU inferencing result
      let scene = document.querySelector('a-scene');
      let objectType = json["entities"]["object:object"][0].value;
      let object = createObject(objectType);
      scene.append(object);
    });
  }
});
```

Voice-XR Demo

Get the Code Sample

https://github.com/wit-ai/webxr-voice-demo



Takeaways

- Voice is a natural human interface and has a place in all parts of the Reality-Virtuality Continuum
- Many voice scenarios can be modeled from natural everyday interactions and can be creatively extended further in VR
- It is very simple to integrate Wit.ai with any platform and framework to add voice to your app’s experience

Resources

- Build Your First Wit App
- Wit.ai Speech API
- Wit.ai Documentation
- Wit.ai Recipes
- Wit.ai FAQ
- GitHub
- Medium Blog
- Wit.ai Hackers Facebook Group

Thank You