

# Speech Recognition API

Understanding what your users say

Session 509

Henry Mason Siri Speech

# Speech Recognition

What is it?

# Speech Recognition

What is it?

Highly accurate

# Speech Recognition

What is it?

Highly accurate

State of the art

# Speech Recognition

What is it?

Highly accurate

State of the art

Easy to use

# Speech Recognition

What is it?

Highly accurate

State of the art

Easy to use

Fast

# Speech Recognition

What is it?

Highly accurate

State of the art

Easy to use

Fast

Many languages

# Speech Recognition

What is it?

Highly accurate

State of the art

Easy to use

Fast

Many languages

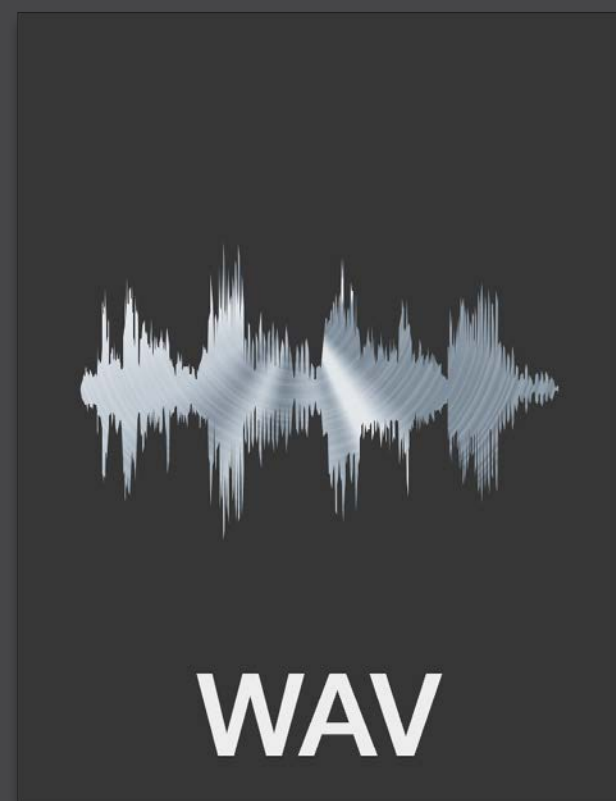
Respects user privacy



# Speech Recognition

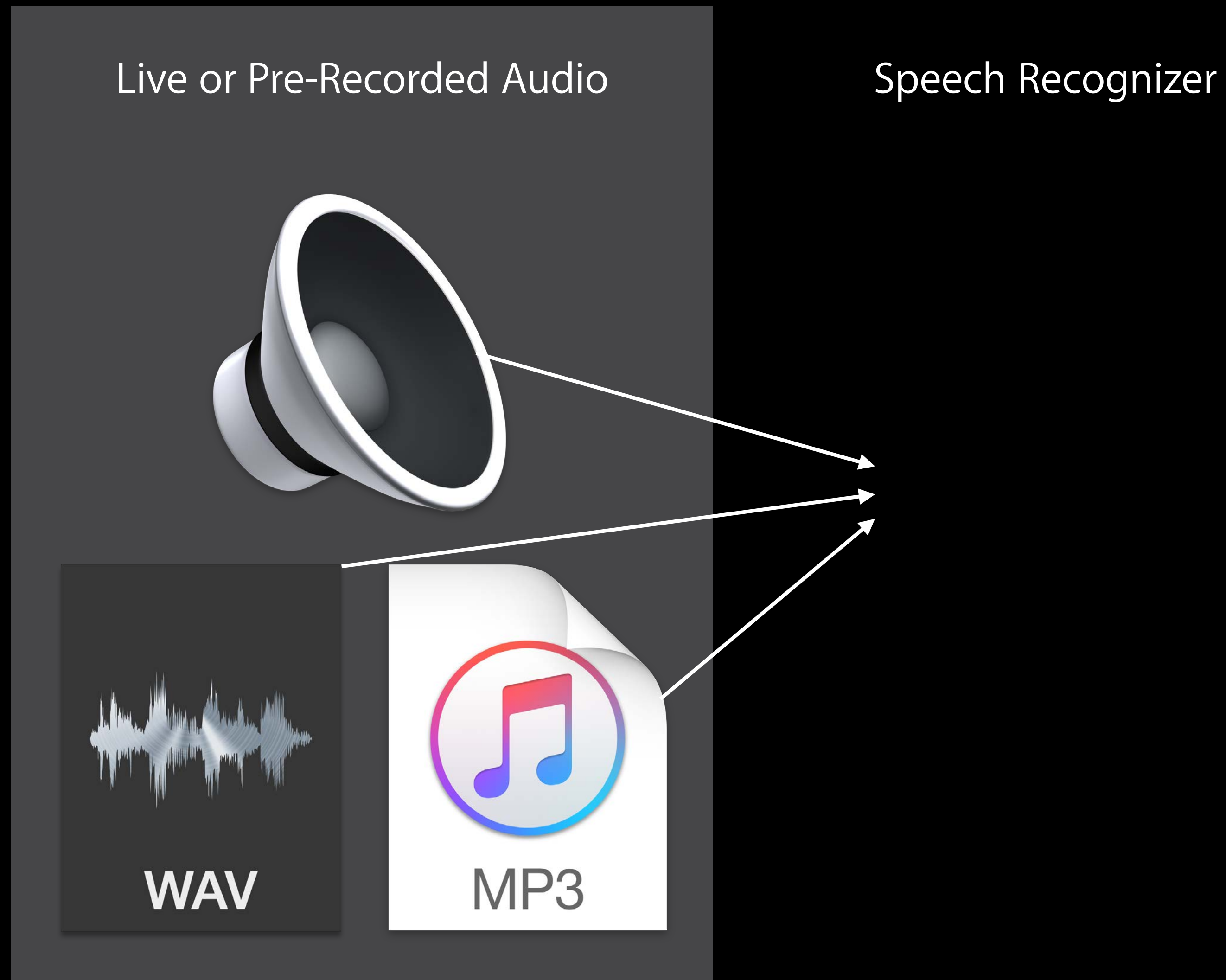
What is it?

Live or Pre-Recorded Audio



# Speech Recognition

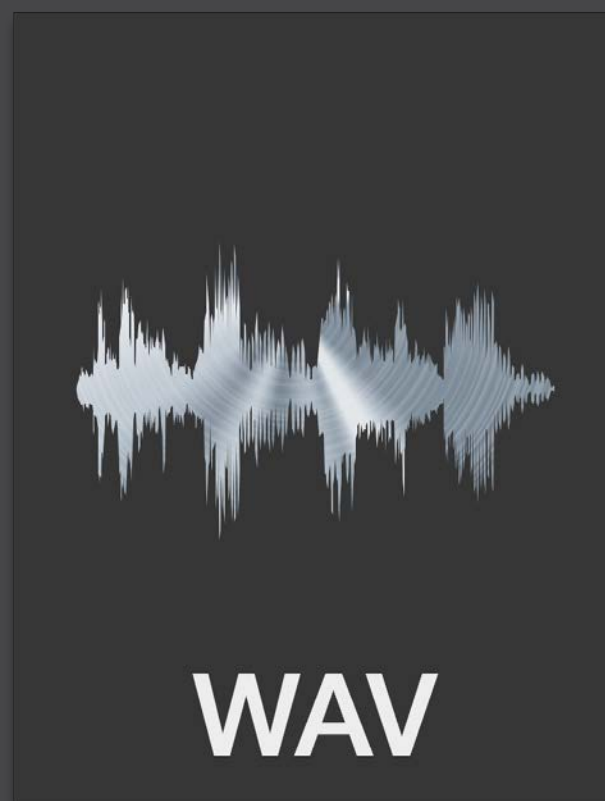
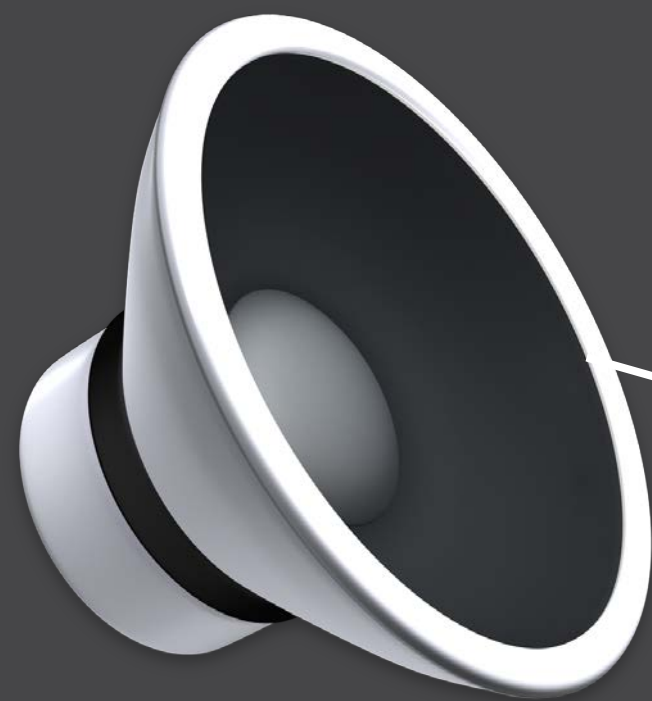
What is it?



# Speech Recognition

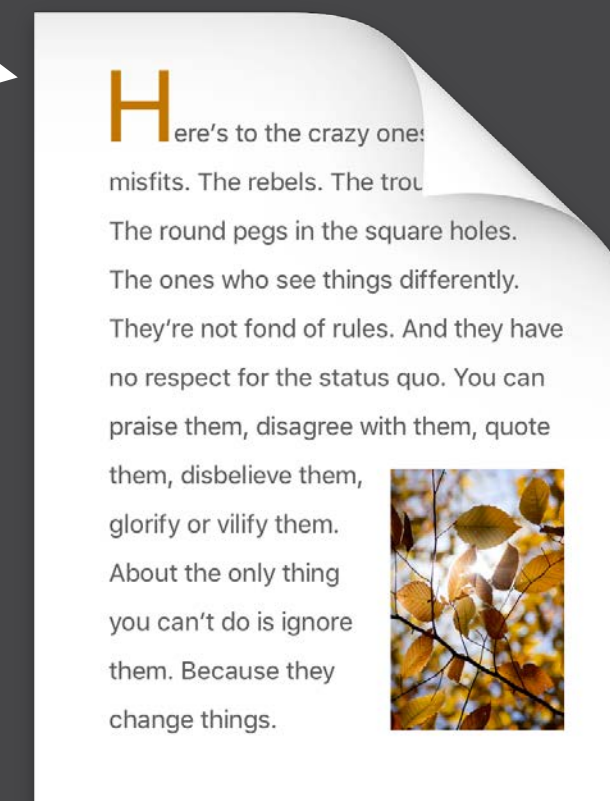
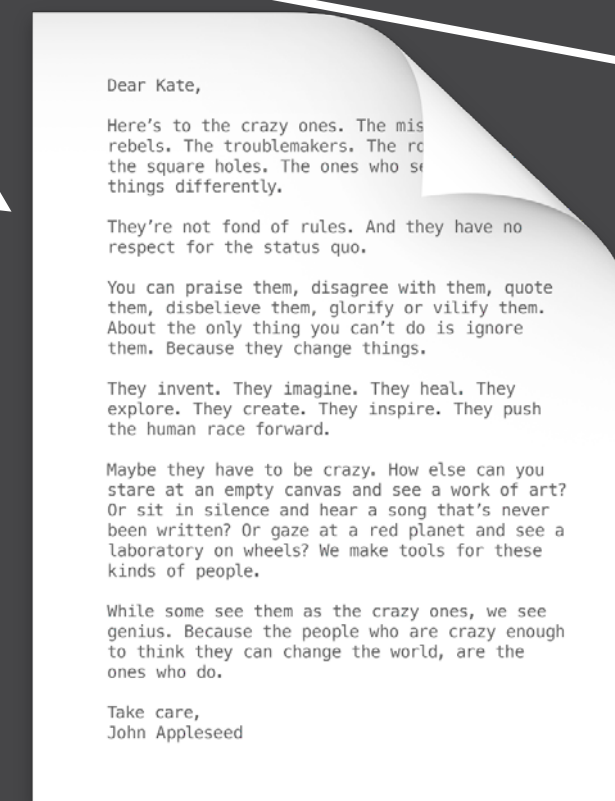
What is it?

Live or Pre-Recorded Audio



Speech Recognizer

Transcribed Text



# iOS Keyboard Dictation

Available since 2011

# iOS Keyboard Dictation

Available since 2011

# iOS Keyboard Dictation

Available since 2011

Widely used

# iOS Keyboard Dictation

Available since 2011

Widely used

- ~65,000 apps per day

# iOS Keyboard Dictation

Available since 2011

Widely used

- ~65,000 apps per day
- About a third of all iOS dictation is your apps!



# iOS Keyboard Dictation

Available since 2011

Widely used

- ~65,000 apps per day
- About a third of all iOS dictation is your apps!

May still be your best option

# iOS Keyboard Dictation

Available since 2011

Widely used

- ~65,000 apps per day
- About a third of all iOS dictation is your apps!

May still be your best option

- Available since iOS 5

# iOS Keyboard Dictation

Available since 2011

Widely used

- ~65,000 apps per day
- About a third of all iOS dictation is your apps!

May still be your best option

- Available since iOS 5
- Handles audio recording

# iOS Keyboard Dictation

Available since 2011

Widely used

- ~65,000 apps per day
- About a third of all iOS dictation is your apps!

May still be your best option

- Available since iOS 5
- Handles audio recording
- No need for special permissions

# iOS Keyboard Dictation

...but limited

# iOS Keyboard Dictation

...but limited

# iOS Keyboard Dictation

...but limited

Limitations

# iOS Keyboard Dictation

...but limited

## Limitations

- Requires presenting a keyboard



# iOS Keyboard Dictation

...but limited

## Limitations

- Requires presenting a keyboard
- Live audio input only

# iOS Keyboard Dictation

...but limited

## Limitations

- Requires presenting a keyboard
- Live audio input only
- Can't customize language

# iOS Keyboard Dictation

...but limited

## Limitations

- Requires presenting a keyboard
- Live audio input only
- Can't customize language
- No way to tell availability

# iOS Keyboard Dictation

...but limited

## Limitations

- Requires presenting a keyboard
- Live audio input only
- Can't customize language
- No way to tell availability
- Can't customize audio recording

# iOS Keyboard Dictation

...but limited

## Limitations

- Requires presenting a keyboard
- Live audio input only
- Can't customize language
- No way to tell availability
- Can't customize audio recording
- "Shallow" results—no timing or confidence information

Speech Recognition API

# Speech Recognition API

More power

NEW

# Speech Recognition API

More power

New framework for iOS

NEW



# Speech Recognition API

More power

New framework for iOS

Same speech technology as Siri  
and Dictation

NEW

# Speech Recognition API

NEW

More power

New framework for iOS

Same speech technology as Siri  
and Dictation

- State of the art accuracy

# Speech Recognition API

NEW

More power

New framework for iOS

Same speech technology as Siri  
and Dictation

- State of the art accuracy
- Fast—results as your users speak

# Speech Recognition API

NEW

More power

New framework for iOS

Same speech technology as Siri  
and Dictation

- State of the art accuracy
- Fast—results as your users speak
- Automatically adapts to the user

# Speech Recognition API

NEW

More power

New framework for iOS

Same speech technology as Siri  
and Dictation

- State of the art accuracy
- Fast—results as your users speak
- Automatically adapts to the user

Rich transcriptions

# Speech Recognition API

NEW

More power

New framework for iOS

Same speech technology as Siri  
and Dictation

- State of the art accuracy
- Fast—results as your users speak
- Automatically adapts to the user

Rich transcriptions

Flexible—recording and pre-recorded audio

# Speech Recognition API Availability

Far and wide, with permission

# Speech Recognition API Availability

Far and wide, with permission

Over 50 languages and dialects



# Speech Recognition API Availability

Far and wide, with permission

Over 50 languages and dialects

Any device running iOS 10

# Speech Recognition API Availability

Far and wide, with permission

Over 50 languages and dialects

Any device running iOS 10

*Usually* requires an Internet connection

# Speech Recognition API Availability

Far and wide, with permission

Over 50 languages and dialects

Any device running iOS 10

*Usually* requires an Internet connection

- (except some some languages and device models)

# Speech Recognition API Availability

Far and wide, with permission

Over 50 languages and dialects

Any device running iOS 10

*Usually* requires an Internet connection

- (except some some languages and device models)
- Use Speech availability API

# Speech Recognition API Availability

Far and wide, with permission

Over 50 languages and dialects

Any device running iOS 10

*Usually* requires an Internet connection

- (except some some languages and device models)
- Use Speech availability API

*Always* requires user permission

# Using Speech Recognition

Explain, authorize, request

# Using Speech Recognition

Explain, authorize, request

Explain why in your Info.plist

# Using Speech Recognition

Explain, authorize, request

Explain why in your Info.plist

- If your app “Phromage” would like to access Speech Recognition...



# Using Speech Recognition

Explain, authorize, request

Explain why in your Info.plist

- If your app “Phromage” would like to access Speech Recognition...
- Usage Description— *This will allow you to take a photo just by saying “cheese.”*

# Using Speech Recognition

Explain, authorize, request

Explain why in your Info.plist

- If your app “Phromage” would like to access Speech Recognition...
- Usage Description— *This will allow you to take a photo just by saying “cheese.”*

Request authorization using `SFSpeechRecognizer.requestAuthorization`

# Using Speech Recognition

Explain, authorize, request

Explain why in your Info.plist

- If your app “Phromage” would like to access Speech Recognition...
- Usage Description— *This will allow you to take a photo just by saying “cheese.”*

Request authorization using `SFSpeechRecognizer.requestAuthorization`

Create recognition request

# Using Speech Recognition

Explain, authorize, request

Explain why in your Info.plist

- If your app “Phromage” would like to access Speech Recognition...
- Usage Description— *This will allow you to take a photo just by saying “cheese.”*

Request authorization using `SFSpeechRecognizer.requestAuthorization`

Create recognition request

- Pre-recorded on disk, use `SFSpeechURLRecognitionRequest`

# Using Speech Recognition

Explain, authorize, request

Explain why in your Info.plist

- If your app “Phromage” would like to access Speech Recognition...
- Usage Description— *This will allow you to take a photo just by saying “cheese.”*

Request authorization using `SFSpeechRecognizer.requestAuthorization`

Create recognition request

- Pre-recorded on disk, use `SFSpeechURLRecognitionRequest`
- From live audio or memory, use `SFSpeechAudioBufferRecognitionRequest`

# Using Speech Recognition

Explain, authorize, request

Explain why in your Info.plist

- If your app “Phromage” would like to access Speech Recognition...
- Usage Description— *This will allow you to take a photo just by saying “cheese.”*

Request authorization using `SFSpeechRecognizer.requestAuthorization`

Create recognition request

- Pre-recorded on disk, use `SFSpeechURLRecognitionRequest`
- From live audio or memory, use `SFSpeechAudioBufferRecognitionRequest`

Give Request to Recognizer

# Using Speech Recognition

Explain, authorize, request

Explain why in your Info.plist

- If your app “Phromage” would like to access Speech Recognition...
- Usage Description— *This will allow you to take a photo just by saying “cheese.”*

Request authorization using `SFSpeechRecognizer.requestAuthorization`

Create recognition request

- Pre-recorded on disk, use `SFSpeechURLRecognitionRequest`
- From live audio or memory, use `SFSpeechAudioBufferRecognitionRequest`

Give Request to Recognizer

- Optionally hold onto `SFSpeechRecognitionTask`

```
// Asking Permission

import Speech
import UIKit

public class MyViewController: UIViewController {

    public func askPermission() {
        SFSpeechRecognizer.requestAuthorization { (authStatus) in
            NSOperationQueue.main().addOperation {
                switch authStatus {
                case .authorized:
                    // Good to go
                    break
                case .denied:
                    // User said no
                    break
                case .restricted:
                    // Device isn't permitted
```



```
// Asking Permission

import Speech
import UIKit

public class MyViewController: UIViewController {

    public func askPermission() {
        SFSpeechRecognizer.requestAuthorization { (authStatus) in
            NSOperationQueue.main().addOperation {
                switch authStatus {
                case .authorized:
                    // Good to go
                    break

                case .denied:
                    // User said no
                    break

                case .restricted:
                    // Device isn't permitted
```

```
// Asking Permission

import Speech
import UIKit

public class MyViewController: UIViewController {

    public func askPermission() {
        SFSpeechRecognizer.requestAuthorization { (authStatus) in
            NSOperationQueue.main().addOperation {
                switch authStatus {
                case .authorized:
                    // Good to go
                    break

                case .denied:
                    // User said no
                    break

                case .restricted:
                    // Device isn't permitted
```

```
// Asking Permission

import Speech
import UIKit

public class MyViewController: UIViewController {

    public func askPermission() {
        SFSpeechRecognizer.requestAuthorization { (authStatus) in
            NSOperationQueue.main().addOperation {
                switch authStatus {
                case .authorized:
                    // Good to go
                    break

                case .denied:
                    // User said no
                    break

                case .restricted:
                    // Device isn't permitted
```



```
// Asking Permission
```

```
    switch authStatus {  
    case .authorized:  
        // Good to go  
        break  
    case .denied:  
        // User said no  
        break  
    case .restricted:  
        // Device isn't permitted  
        break  
    case .notDetermined:  
        // Don't know yet  
        break
```

```
    }
```

```
  }
```

```
}
```

```
}
```

```
// Recognizing pre-recorded audio

import Speech

func recognizeFile(url: NSURL) {
    guard let recognizer = SFSpeechRecognizer() else {
        // Not supported for device's locale
        return
    }
    if !recognizer.isAvailable {
        // Not available right now
        return
    }
    let request = SFSpeechURLRecognitionRequest(url: url)
    recognizer.recognitionTask(with: request) { (result, error) in
        guard let result = result else {
            // handle error
            return
        }
    }
}
```

```
// Recognizing pre-recorded audio
```

```
import Speech
```

```
func recognizeFile(url: NSURL) {
```

```
    guard let recognizer = SFSpeechRecognizer() else {  
        // Not supported for device's locale  
        return  
    }
```

```
    if !recognizer.isAvailable {  
        // Not available right now  
        return  
    }
```

```
    let request = SFSpeechURLRecognitionRequest(url: url)
```

```
    recognizer.recognitionTask(with: request) { (result, error) in
```

```
        guard let result = result else {  
            // handle error  
            return  
        }
```

```
    }
```

```
// Recognizing pre-recorded audio
```

```
import Speech
```

```
func recognizeFile(url: NSURL) {
```

```
    guard let recognizer = SFSpeechRecognizer() else {
```

```
        // Not supported for device's locale
```

```
        return
```

```
    }
```

```
    if !recognizer.isAvailable {
```

```
        // Not available right now
```

```
        return
```

```
    }
```

```
    let request = SFSpeechURLRecognitionRequest(url: url)
```

```
    recognizer.recognitionTask(with: request) { (result, error) in
```

```
        guard let result = result else {
```

```
            // handle error
```

```
            return
```

```
        }
```



```
// Recognizing pre-recorded audio

import Speech

func recognizeFile(url: NSURL) {
    guard let recognizer = SFSpeechRecognizer() else {
        // Not supported for device's locale
        return
    }
    if !recognizer.isAvailable {
        // Not available right now
        return
    }
    let request = SFSpeechURLRecognitionRequest(url: url)
    recognizer.recognitionTask(with: request) { (result, error) in
        guard let result = result else {
            // handle error
            return
        }
    }
}
```

```
// Recognizing pre-recorded audio

import Speech

func recognizeFile(url: NSURL) {
    guard let recognizer = SFSpeechRecognizer() else {
        // Not supported for device's locale
        return
    }
    if !recognizer.isAvailable {
        // Not available right now
        return
    }
    let request = SFSpeechURLRecognitionRequest(url: url)
    recognizer.recognitionTask(with: request) { (result, error) in
        guard let result = result else {
            // handle error
            return
        }
    }
}
```

```
// Recognizing pre-recorded audio

    return
}
if !recognizer.isAvailable {
    // Not available right now
    return
}
let request = SFSpeechURLRecognitionRequest(url: url)
recognizer.recognitionTask(with: request) { (result, error) in
    guard let result = result else {
        // handle error
        return
    }
    if result.isFinal {
        print("File said \(result.bestTranscription.formattedString)")
    }
}
}
}
```

```
// Recognizing pre-recorded audio

    return
}
if !recognizer.isAvailable {
    // Not available right now
    return
}
let request = SFSpeechURLRecognitionRequest(url: url)
recognizer.recognitionTask(with: request) { (result, error) in
    guard let result = result else {
        // handle error
        return
    }
    if result.isFinal {
        print("File said \(result.bestTranscription.formattedString)")
    }
}
}
```

```
// Recognizing pre-recorded audio

    return
}
if !recognizer.isAvailable {
    // Not available right now
    return
}
let request = SFSpeechURLRecognitionRequest(url: url)
recognizer.recognitionTask(with: request) { (result, error) in
    guard let result = result else {
        // handle error
        return
    }
    if result.isFinal {
        print("File said \(result.bestTranscription.formattedString)")
    }
}
}
}
```

```
// Recognizing live audio

import AVFoundation
import Speech

let audioEngine = AVAudioEngine()
let speechRecognizer = SFSpeechRecognizer()!
let request = SFSpeechAudioBufferRecognitionRequest()
var recognitionTask: SFSpeechRecognitionTask?

func startRecording() throws {
    // Setup Audio Session
    let node = audioEngine.inputNode!
    let recordingFormat = node.outputFormat(forBus: 0)
    node.installTap(onBus: 0, bufferSize: 1024, format: recordingFormat) { (buffer, _) in
        recognitionRequest.append(buffer)
    }

    audioEngine.prepare()
    try audioEngine.start()
    recognitionTask = speechRecognizer.recognitionTask(with: request) { /* ... */ }
```

```
// Recognizing live audio

import AVFoundation
import Speech

let audioEngine = AVAudioEngine()
let speechRecognizer = SFSpeechRecognizer()!
let request = SFSpeechAudioBufferRecognitionRequest()
var recognitionTask: SFSpeechRecognitionTask?

func startRecording() throws {
    // Setup Audio Session
    let node = audioEngine.inputNode!
    let recordingFormat = node.outputFormat(forBus: 0)
    node.installTap(onBus: 0, bufferSize: 1024, format: recordingFormat) { (buffer, _) in
        recognitionRequest.append(buffer)
    }

    audioEngine.prepare()
    try audioEngine.start()
    recognitionTask = speechRecognizer.recognitionTask(with: request) { /* ... */ }
```

```
// Recognizing live audio

import AVFoundation
import Speech

let audioEngine = AVAudioEngine()
let speechRecognizer = SFSpeechRecognizer()!
let request = SFSpeechAudioBufferRecognitionRequest()
var recognitionTask: SFSpeechRecognitionTask?

func startRecording() throws {
    // Setup Audio Session
    let node = audioEngine.inputNode!
    let recordingFormat = node.outputFormat(forBus: 0)
    node.installTap(onBus: 0, bufferSize: 1024, format: recordingFormat) { (buffer, _) in
        recognitionRequest.append(buffer)
    }

    audioEngine.prepare()
    try audioEngine.start()
    recognitionTask = speechRecognizer.recognitionTask(with: request) { /* ... */ }
```



```
// Recognizing live audio

    let recordingFormat = node.outputFormat(forBus: 0)
    node.installTap(onBus: 0, bufferSize: 1024, format: recordingFormat) { (buffer, _) in
        recognitionRequest.append(buffer)
    }
    audioEngine.prepare()
    try audioEngine.start()
    recognitionTask = speechRecognizer.recognitionTask(with: request) { /* ... */ }
}

func stopRecording() {
    audioEngine.stop()
    request.endAudio()
}

func cancelRecording() {
    audioEngine.stop()
    recognitionTask?.cancel()
}
```

```
// Recognizing live audio

    let recordingFormat = node.outputFormat(forBus: 0)
    node.installTap(onBus: 0, bufferSize: 1024, format: recordingFormat) { (buffer, _) in
        recognitionRequest.append(buffer)
    }
    audioEngine.prepare()
    try audioEngine.start()
    recognitionTask = speechRecognizer.recognitionTask(with: request) { /* ... */ }
}

func stopRecording() {
    audioEngine.stop()
    request.endAudio()
}

func cancelRecording() {
    audioEngine.stop()
    recognitionTask?.cancel()
}
```

```
// Recognizing live audio

    let recordingFormat = node.outputFormat(forBus: 0)
    node.installTap(onBus: 0, bufferSize: 1024, format: recordingFormat) { (buffer, _) in
        recognitionRequest.append(buffer)
    }
    audioEngine.prepare()
    try audioEngine.start()
    recognitionTask = speechRecognizer.recognitionTask(with: request) { /* ... */ }
}

func stopRecording() {
    audioEngine.stop()
    request.endAudio()
}

func cancelRecording() {
    audioEngine.stop()
    recognitionTask?.cancel()
}
```

```
// Recognizing live audio

    let recordingFormat = node.outputFormat(forBus: 0)
    node.installTap(onBus: 0, bufferSize: 1024, format: recordingFormat) { (buffer, _) in
        recognitionRequest.append(buffer)
    }
    audioEngine.prepare()
    try audioEngine.start()
    recognitionTask = speechRecognizer.recognitionTask(with: request) { /* ... */ }
}

func stopRecording() {
    audioEngine.stop()
    request.endAudio()
}

func cancelRecording() {
    audioEngine.stop()
    recognitionTask?.cancel()
}
```

# Best Practices

# Resources

Be responsible

# Resources

Be responsible

Speech Recognition is free, but not unlimited

# Resources

Be responsible

Speech Recognition is free, but not unlimited

- Per-devices, per day recognition limits



# Resources

Be responsible

Speech Recognition is free, but not unlimited

- Per-devices, per day recognition limits
- Per-app limits

# Resources

## Be responsible

Speech Recognition is free, but not unlimited

- Per-devices, per day recognition limits
- Per-app limits
- Be prepared to handle failures

# Resources

## Be responsible

Speech Recognition is free, but not unlimited

- Per-devices, per day recognition limits
- Per-app limits
- Be prepared to handle failures
- If you're hitting limit routinely, talk to us

# Resources

## Be responsible

Speech Recognition is free, but not unlimited

- Per-devices, per day recognition limits
- Per-app limits
- Be prepared to handle failures
- If you're hitting limit routinely, talk to us

Speech recognition isn't cheap

# Resources

## Be responsible

Speech Recognition is free, but not unlimited

- Per-devices, per day recognition limits
- Per-app limits
- Be prepared to handle failures
- If you're hitting limit routinely, talk to us

Speech recognition isn't cheap

- Requires power and data

# Resources

## Be responsible

Speech Recognition is free, but not unlimited

- Per-devices, per day recognition limits
- Per-app limits
- Be prepared to handle failures
- If you're hitting limit routinely, talk to us

Speech recognition isn't cheap

- Requires power and data
- Maximum utterance duration—  
about one minute

# Privacy and Usability

Transparency



# Privacy and Usability

## Transparency

- ✔ Make sure users know they're being recorded
  - Show something in your UI





# Privacy and Usability

## Transparency

- ✓ Make sure users know they're being recorded
  - Show something in your UI
- ✗ Some speech is not appropriate for recognition
  - Passwords
  - Sensitive speech



# Privacy and Usability

## Transparency

- ✔ Make sure users know they're being recorded
  - Show something in your UI
- ✘ Some speech is not appropriate for recognition
  - Passwords
  - Sensitive speech
- ✔ Show recognition results to the user before acting on them
  - Helps users deal with recognition errors



# Summary

# Summary

iOS app developers now have access to fast, accurate, and flexible speech recognition

# Summary

iOS app developers now have access to fast, accurate, and flexible speech recognition  
It's important to gracefully handling speech recognition not being available

# Summary

iOS app developers now have access to fast, accurate, and flexible speech recognition

It's important to gracefully handling speech recognition not being available

As will all user-facing features, show the user what your app is doing

# Summary

iOS app developers now have access to fast, accurate, and flexible speech recognition

It's important to gracefully handling speech recognition not being available

As will all user-facing features, show the user what your app is doing

We can't wait to see what you do with this!

More Information

<https://developer.apple.com/wwdc16/509>



# Related Sessions

---

Introducing SiriKit

Presidio

Wednesday 6:00PM

---

Extending Your Apps with SiriKit

Nob Hill

Thursday 1:40PM

---

# Labs

---

Speech Recognition Lab

Graphics, Games,  
and Media Lab D

Wednesday 1:00PM

---

Speech Recognition Lab

Graphics, Games,  
and Media Lab D

Friday 1:00PM

---

SiriKit Lab

Location

Thursday 3:00PM

---

SiriKit Lab

Location

Friday 9:00AM

---

---

---

---

---

WWDC 16 