



# Sign In with Apple

Takeshi Ihara

2019/06/19

# Takeshi Ihara

---



- 2018.09.01 AbemaTV Join
- ボルダリングと格闘ゲームが趣味
- <https://github.com/Nonchalant>
- <https://twitter.com/nonchalant0303>

# 関連セッション

---

- Introducing Sign In with Apple (Session 706)
- What's New in Authentication, Safari, and WebKit (Session 504)

# 今日話すこと

---

- Sign In with Apple とは
- Sign In with Apple の構造
- Labで質問してきたこと

# 今日話すこと

---

- Sign In with Apple の実装方法
  - <https://developer.apple.com/documentation/authenticationservices>
  - <https://developer.apple.com/documentation/signinwithapplejs>

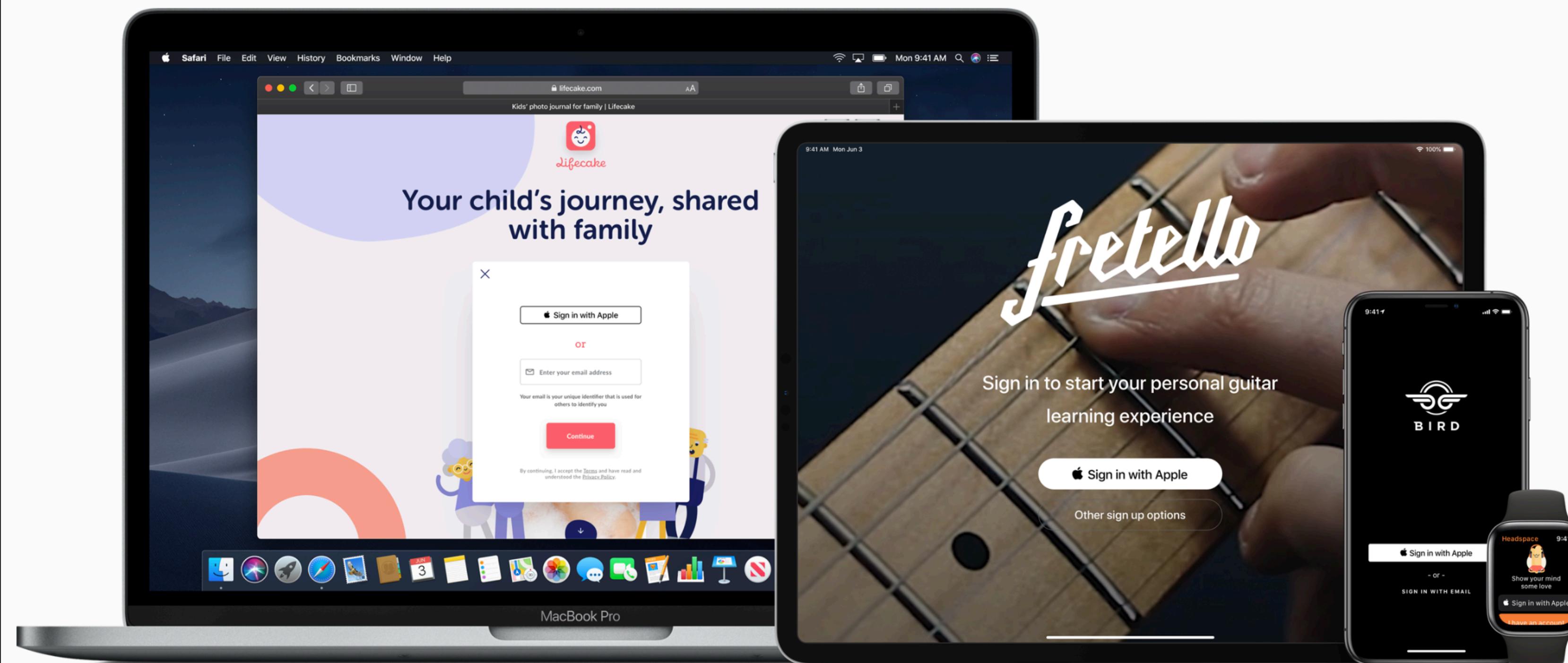
1

Sign In with Apple とは



# The fast, easy way to sign in to apps and websites.

Sign In with Apple makes it easy for users to sign in to your apps and websites using their Apple ID. Instead of filling out forms, verifying email addresses, and choosing new passwords, they can use Sign In with Apple to set up an account and start using your app right away. All accounts are protected with two-factor authentication for superior security, and Apple will not track users' activity in your app or website.

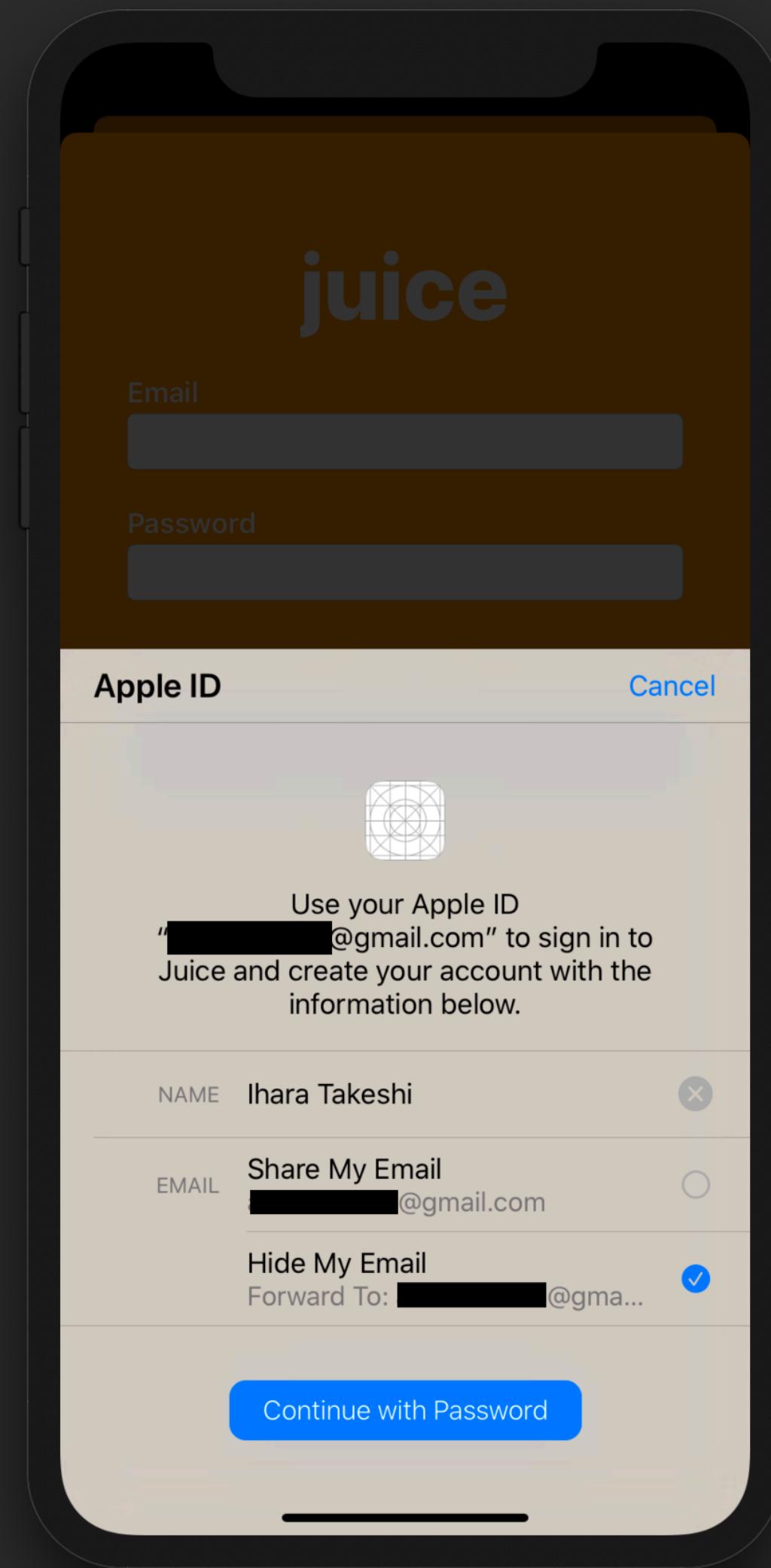


# Sign In with Apple とは

---

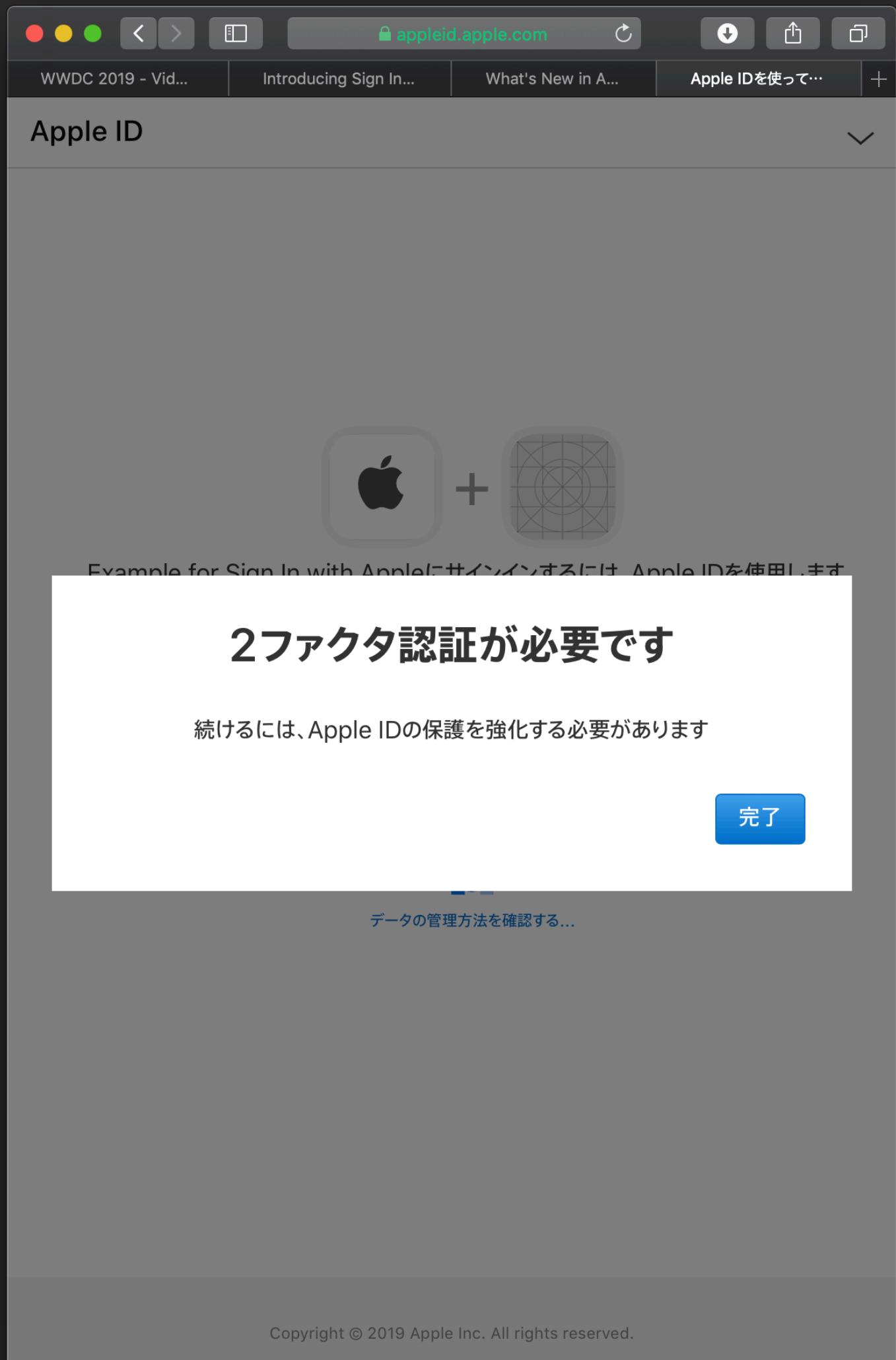
- Respect for privacy

- 名前とメールアドレスしか取得できない
- サービスごとにPrivateなメールアドレスを発行可能
- Appleはサービス内の行動を追跡しない



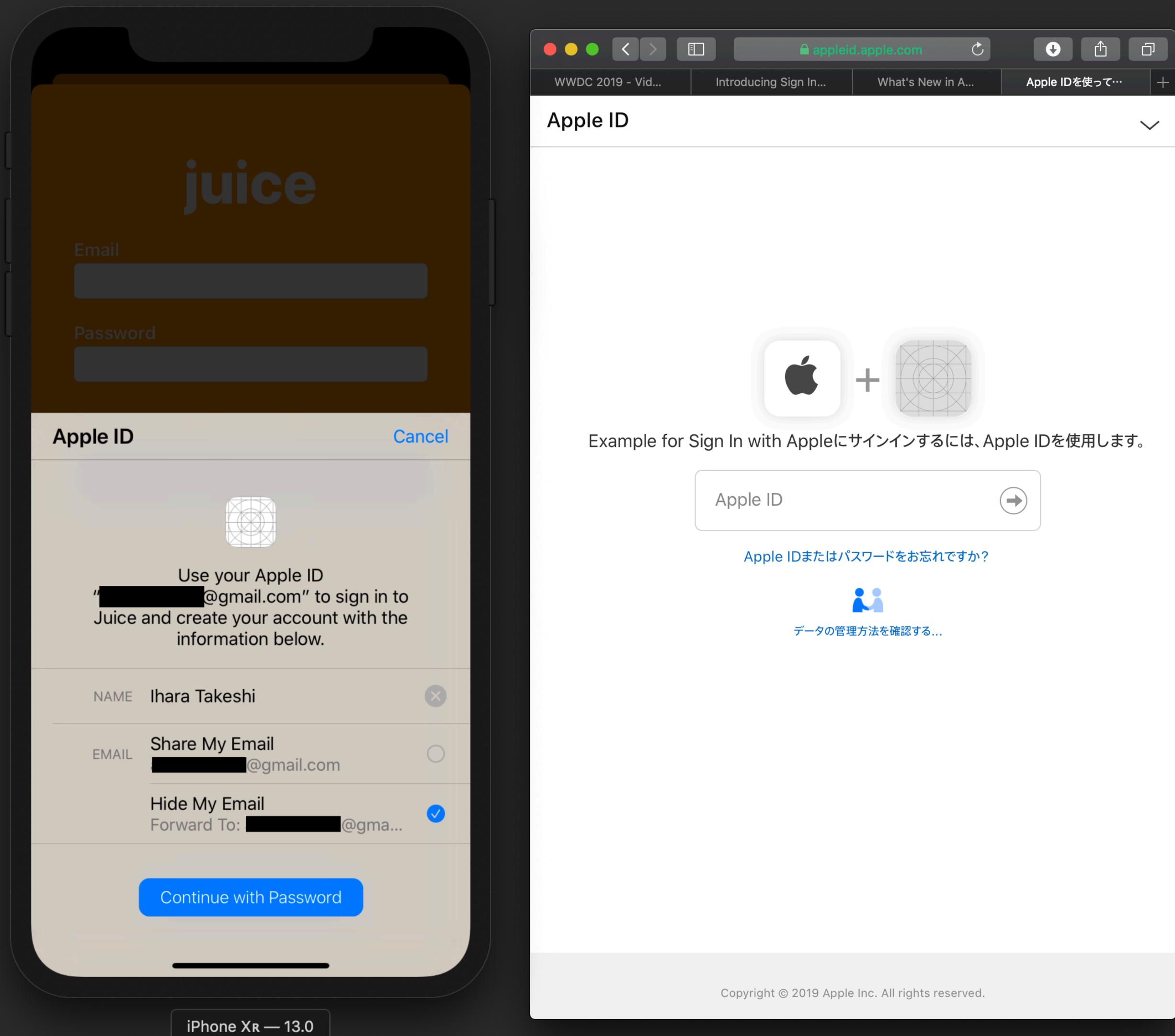
# Sign In with Apple とは

- Security Built in
  - 2段階認証を設定しているApple IDのみ利用可能
  - AppleデバイスではFace ID or Touch IDでサインイン可能



# Sign In with Apple とは

- Works everywhere
  - iOS, macOS, tvOS, watchOS,  
ブラウザで動作可能



# Sign In with Apple とは

- Antifraud
  - デバイス上の機械学習やその他の情報によって、新しいユーザーが人かBotかを判別する
  - レスポンスに含まれている情報を参考にできる

Enumeration

## ASUserDetectionStatus

Possible values for the real user indicator.

SDKs

iOS 13.0+ Beta

macOS 10.15+ Beta

tvOS 13.0+ Beta

watchOS 6.0+ Beta

Framework

AuthenticationServices

On This Page

[Declaration](#)

[Topics](#)

[See Also](#)

Declaration

```
enum ASUserDetectionStatus : Int
```

### Topics

User Status

case [likelyReal](#)

The user appears to be a real person.

case [unknown](#)

The system hasn't determined whether the user might be a real person.

case [unsupported](#)

The system can't determine this user's status as a real person.

iOS 13 のリリースと同時に利用可能  
(2019.09)

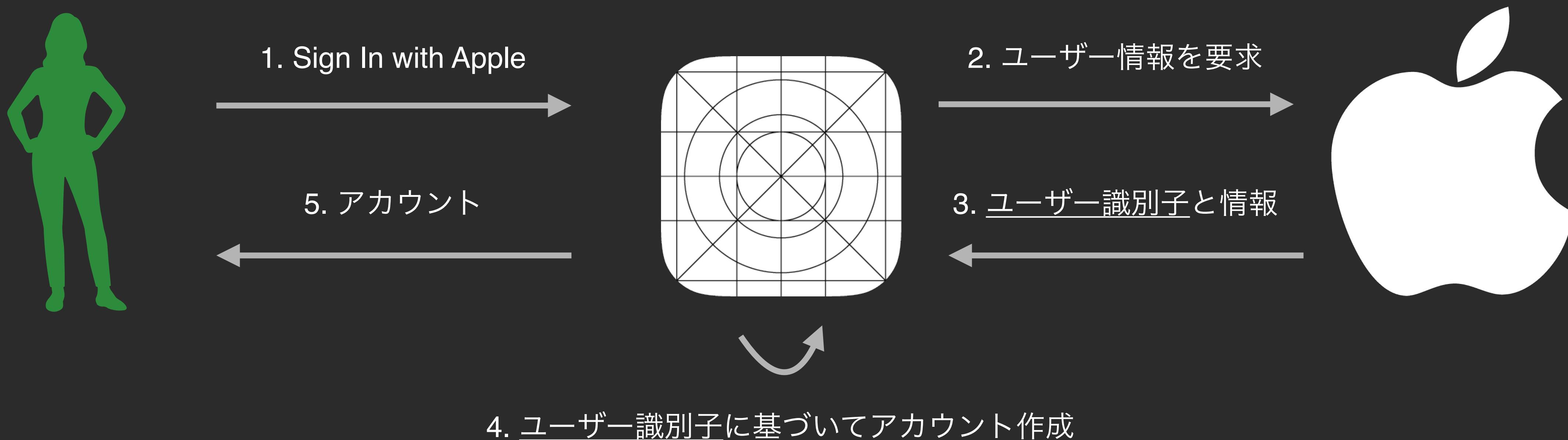
2

# Sign In with Apple の構造



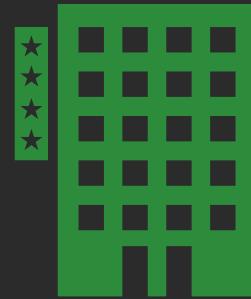
# Sign In with Apple の構造

---



# Sign In with Apple の構造





## Apple Developer Team

- 共通のUser識別子
- PrivateなEmailアドレスに送信可能なドメインまたは個別アドレス
- OAuthやOpenID Connectのような認証規格のKey群



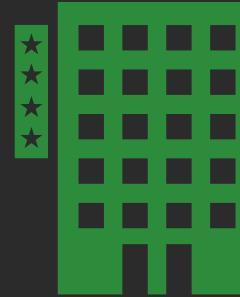
App ID



App ID



App ID



## Apple Developer Team

- 共通のUser識別子
- PrivateなEmailアドレスに送信可能なドメインまたは個別アドレス
- OAuthやOpenID Connectのような認証規格のKey群



Service ID



Primary App ID



App ID



ブラウザ  
(登録不要)



App ID

- 単一のAppの場合はService ID不要
- PrivateなEmailアドレス

- 共通のService識別子
- Service識別子を用いてブラウザでサインイン可能
- Service毎に共通のPrivateなEmailアドレス

# Sign In with Apple の構造

- Apple Developer (旧 iOS Dev Center)

The screenshot shows the 'Certificates, Identifiers & Profiles' section of the Apple Developer portal. The 'Identifiers' tab is selected. A table lists one identifier:

NAME	IDENTIFIER
Example for Sign In with Apple 1 2	com.nonchalant.group.signinwithapple

A dropdown menu titled 'Services IDs' is open, showing various ID types:

- App IDs
- Services IDs
- Pass Type IDs
- Website Push IDs
- iCloud Containers
- App Groups
- Merchant IDs
- Music IDs
- Maps IDs

# 3

Lab で質問してきたこと



# Lab で質問してきたこと

---

- Privateなアドレスにメールを送信する方法
- PrivateなアドレスのRevoke
- Sign In with Apple JS でのユーザー情報の取り方
- 3rd Partyログインのボタンより上に置かない場合はリジェクト対象か？
- 3rd Partyログインをサポートしているアプリで、Sign In with Apple をサポートしない場合はリジェクト対象か？
- AuthenticationService.ASFoundationとはなにか？

# Lab で質問してきたこと (akatsuki174)

---

- ・アカウントを複数持てるサービスではどう対応すればよいか
- ・Sign In with Appleでサインインした後に年齢などを追加で取得するのはアリなのか？

# Privateなアドレスにメールを送信する方法

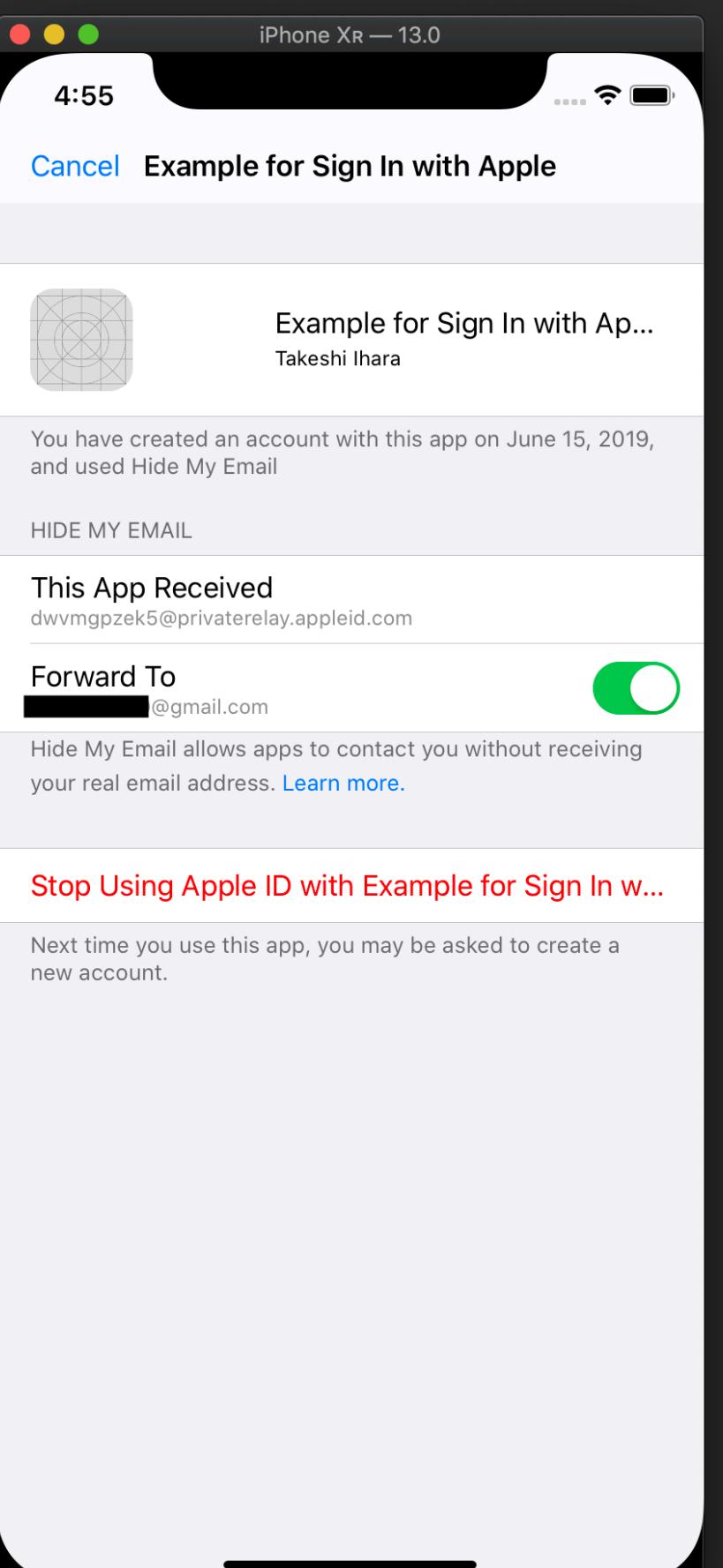
---

- Apple Developer で事前に登録したドメインまたはアドレスで送れる
- 送信者と受信者が同一だと送れない
- Sign In with Apple を複数したときに、最初に生成されたアドレスのみ有効
  - 毎回同じアドレスが生成されるように修正されてました (2019.06.19)

# PrivateなアドレスのRevoke

- ・設定アプリまたはブラウザからService単位でRevoke可能

- ・<https://appleid.apple.com>



**Example for Sign In with Apple 3**

2019年06月15日にこのAppでアカウントを作成し、名前を共有して「メールを非表示」を使用しました

「メールを非表示」にする  
このAppが受信: dwvmpzek5@privaterelay.appleid.com

メールをコピー

転送先: [REDACTED]@gmail.com  
「メールを非表示」にすることで、実際のメールアドレスを受け渡すことなくAppと連絡することができます。[詳しい情報](#)

オフにする

このAppでApple IDの使用を停止すると、次回Appを使用する際、アカウントを新しく作成するように指示される場合があります

Example for Sign In with Apple 3でApple IDの使用を中止

戻る 完了

# Sign In with Apple JS でのユーザー情報の取り方

---

- まだ提供していない (2019.06.19)

# 3rd Partyログインのボタンより上に置かない場合は リジェクト対象か？

---

- In a stacked layout, place the Sign In with Apple button above the other buttons.
  - <https://developer.apple.com/design/human-interface-guidelines/sign-in-with-apple/overview/>
- HIGから上記文言が削除されました (2019.06.19)

# 3rd Partyログインをサポートしているアプリで、 Sign In with Apple をサポートしない場合はリジェクト対象か？

---

- まだ決めていない (2019.06.05)

# AuthenticationServices.ASFoundationとはなにか？

- Foundationとの内部通信クラス
- Jump to Definitionできない

```
1 import·AuthenticationServices.ASAuthorization·
2 import·AuthenticationServices.ASAuthorizationAppleIDButton·
3 import·AuthenticationServices.ASAuthorizationAppleIDCredential·
4 import·AuthenticationServices.ASAuthorizationAppleIDProvider·
5 import·AuthenticationServices.ASAuthorizationAppleIDRequest·
6 import·AuthenticationServices.ASAuthorizationController·
7 import·AuthenticationServices.ASAuthorizationCredential·
8 import·AuthenticationServices.ASAuthorizationError·
9 import·AuthenticationServices.ASAuthorizationOpenIDRequest·
10 import·AuthenticationServices.ASAuthorizationPasswordProvider·
11 import·AuthenticationServices.ASAuthorizationPasswordRequest·
12 import·AuthenticationServices.ASAuthorizationProvider·
13 import·AuthenticationServices.ASAuthorizationProviderExtensionAuthorizationRequest·
14 import·AuthenticationServices.ASAuthorizationRequest·
15 import·AuthenticationServices.ASAuthorizationSingleSignOnCredential·
16 import·AuthenticationServices.ASAuthorizationSingleSignOnProvider·
17 import·AuthenticationServices.ASAuthorizationSingleSignOnRequest·
18 import·AuthenticationServices.ASCredentialIdentityStore·
19 import·AuthenticationServices.ASCredentialIdentityStoreState·
20 import·AuthenticationServices.ASCredentialProviderExtensionContext·
21 import·AuthenticationServices.ASCredentialProviderViewController·
22 import·AuthenticationServices.ASCredentialServiceIdentifier·
23 import·AuthenticationServices.ASFoundation·
24 import·AuthenticationServices.ASPasswordCredential·
25 import·AuthenticationServices.ASPasswordCredentialIdentity·
26 import·AuthenticationServices.ASWebAuthenticationSession·
27 import·AuthenticationServices.ASWebAuthenticationSessionRequest·
28 import·AuthenticationServices.ASWebAuthenticationSessionWebBrowserSessionHandling·
29 import·AuthenticationServices.ASWebAuthenticationSessionWebBrowserSessionManager·
--
```

# アカウントを複数持てるサービスではどう対応すればよいか

---

- ・ひとつのApple IDで複数アカウントの作成をサポートしていない
  - ・サービス側で解決するしかなさそう

# Sign In with Appleでサインインした後に年齢などを追加で取得する のはアリなのか？

- ・サービスとして必要な情報はとってもよい、Facebook連携なども同様
- ・Sign In with Apple はあくまでサインインの手段

Thank you



# iOSDC 2019

レギュラートーク (30分)

動画アプリの投げ銭機能における消耗型課金の仕組みと実装

iOSDC Japan 2019

 Takeshi Ihara  nonchalant0303

近年、ユーザーがクリエイターに直接お金を払ういわゆる投げ銭機能を提供しているサービスが増えてきました。

自分が開発している動画アプリでも投げ銭機能を実装することになり、それに伴い投げ銭するためのアプリ内通貨の購入機能が必要でした。

しかし、まだまだ事例は少なく参考になる情報などが少ない状況です。

プレミアム機能を使うための月額課金プランをすでにネイティブ実装していたので、レシート検証の共通化などを行うためアプリ内通貨の購入機能もネイティブ実装することになりました。

一部の処理は共通化できますが、月額課金プランはAuto-Renewable Subscriptions、アプリ内通貨の購入機能はConsumableと呼ばれる課金方式で実装するため、リトライ処理など課金トランザクションの扱いの差異などを意識して実装する必要があります。

しかしよくネイティブ実装される課金方式は、自動更新型の

- 動画アプリの投げ銭機能における消耗型課金の仕組みと実装

- <https://twitter.com/nonchalant0303> の固定ツイ

