



Sign in with Apple

**SICHER, SCHNELL UND EINFACH ZUM NEUEN
ACCOUNT**

ABOUT ME

- » Stefan Herold • @blackjacxxx 
- » Mobile Dev seit 2009
- » Seit 2017 bei ioki im Herzen Frankfurts

IOKI

- » Demand Responsive Transport
- » Leuchtturmpunkt mit HVV Hamburg
- » Whitelabel Passenger App • iOS & Android
- » Driver App • React Native
- » Backend • Ruby

SIGN IN WITH APPLE

- » Registrierung und Login
- » schnell, sicher und privat
- » kein Tracking durch Apple
- » App erhält:
 - » stabile, eindeutige userID
 - » Vor- und Nachname
 - » Email Adresse

SIGN IN WITH APPLE

PLATFORM

iOS 13+

iPadOS 13+

tvOS 13+

watchOS 6+

macOS Catalina 10.15+

Web

Windows

Android

APPLE ID

iCloud Nutzer

beliebige Apple ID

SICHERHEIT

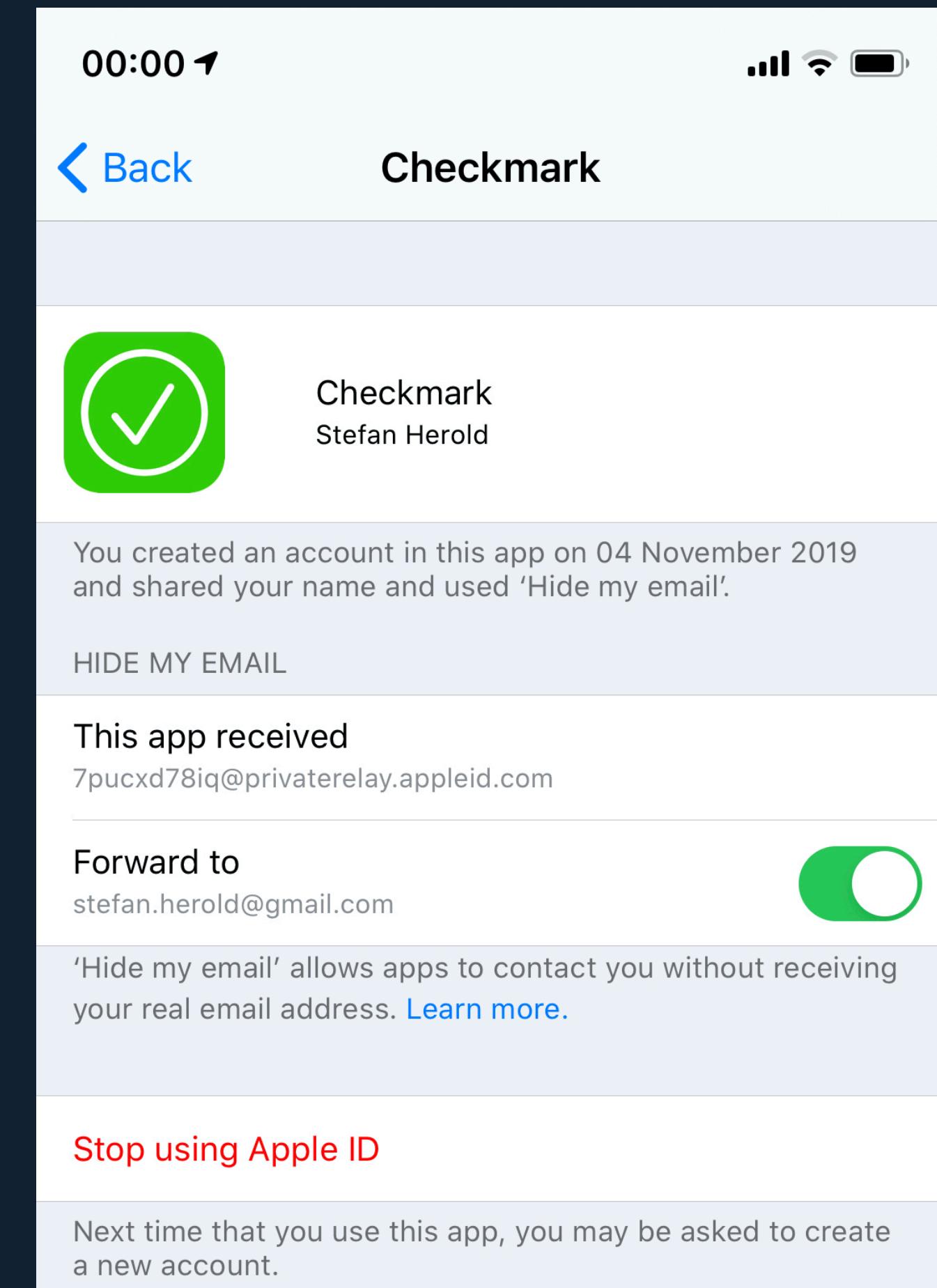
- » Kein Passwort
- » Zwei-Faktor-Authentisierung
- » Anti-Fraud

PRIVATSPHÄRE

Eindeutige, zufällige Email
privaterelay.appleid.com

App sieht nur diese Email

- » Kommunikation mit genau einem Developer
- » Zwei-Wege-Kanal
- » Kein speichern von Emails
- » Weiterleitung deaktivierbar



WER'S BRAUCHT

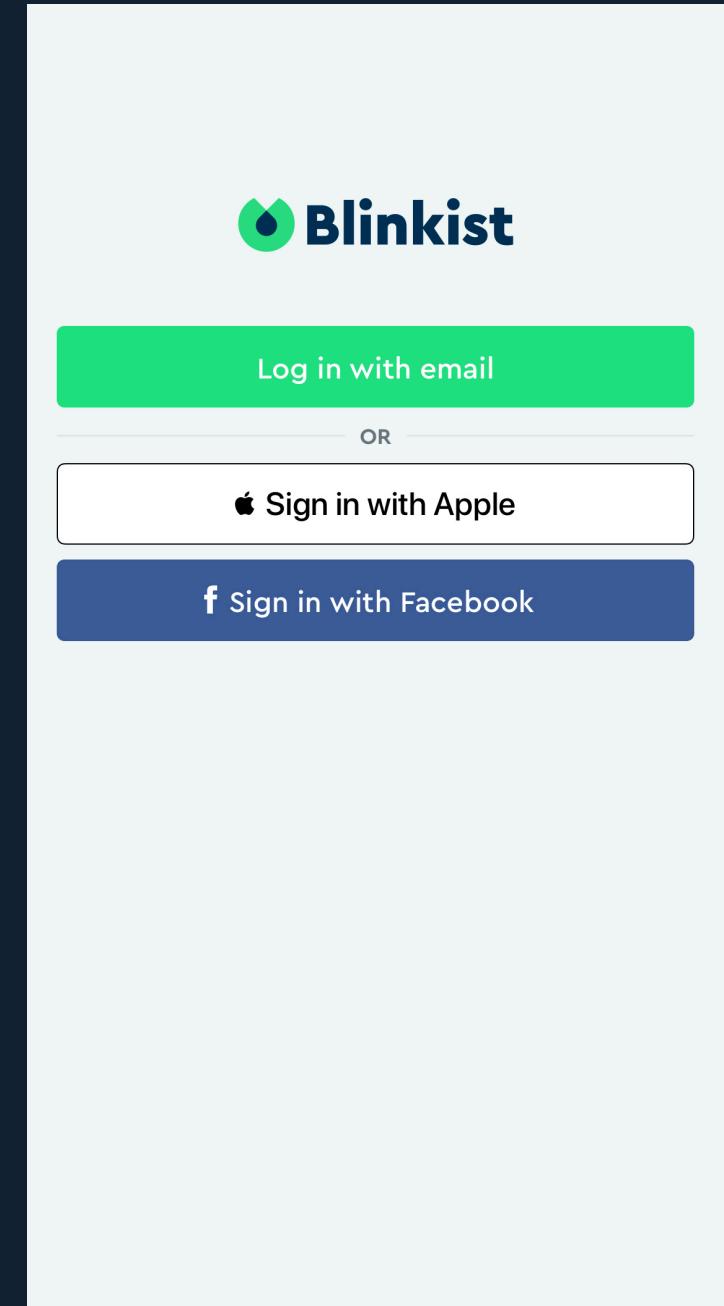
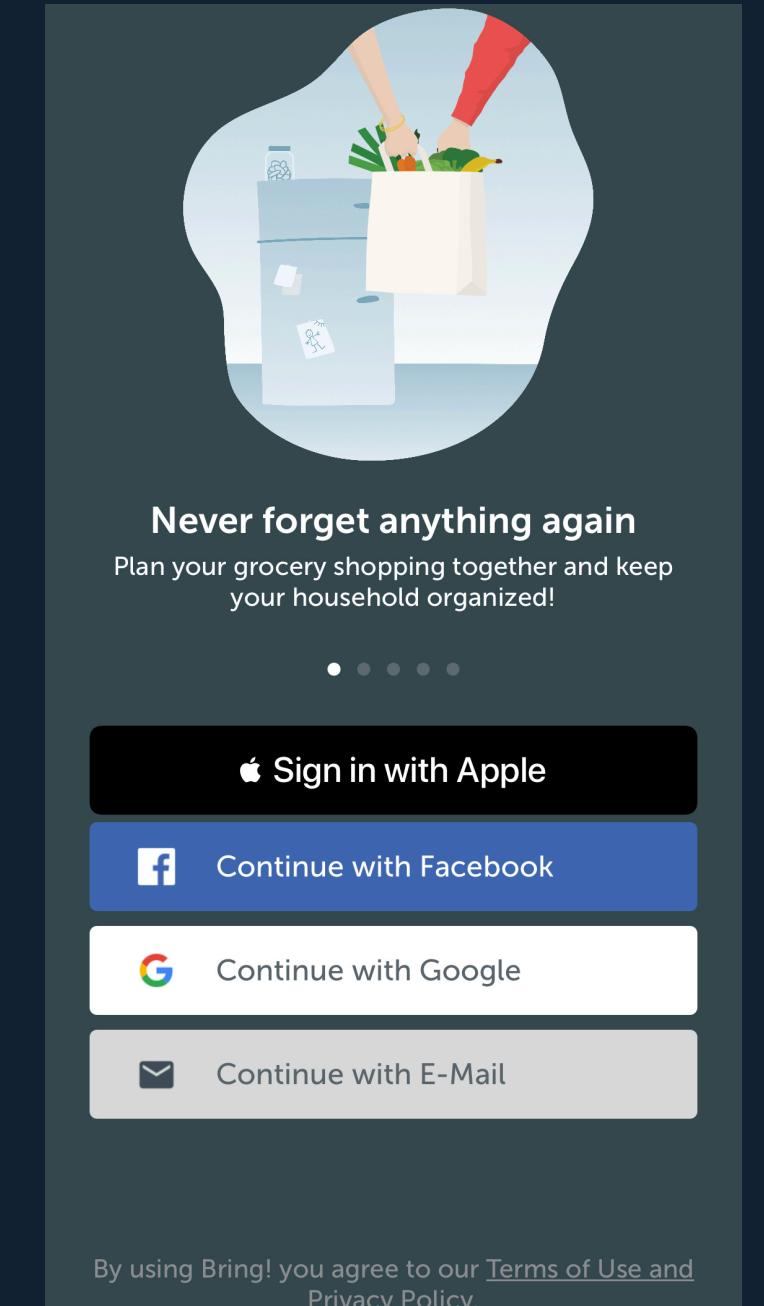
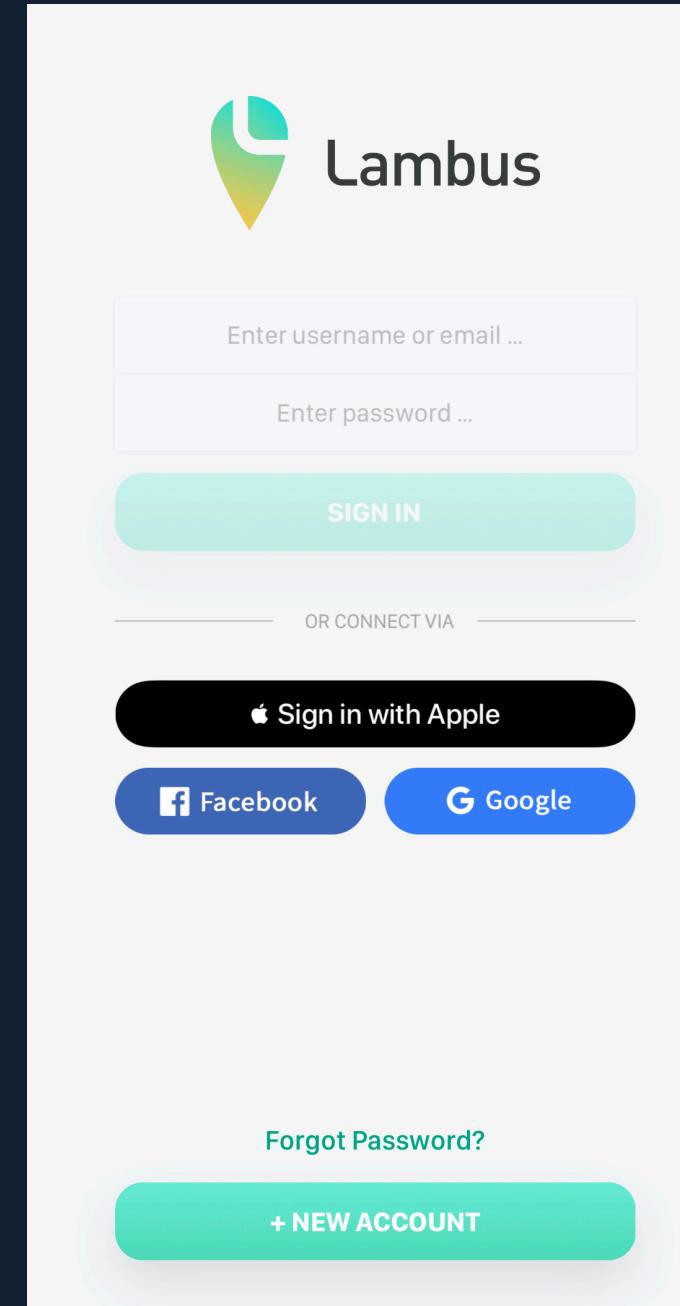
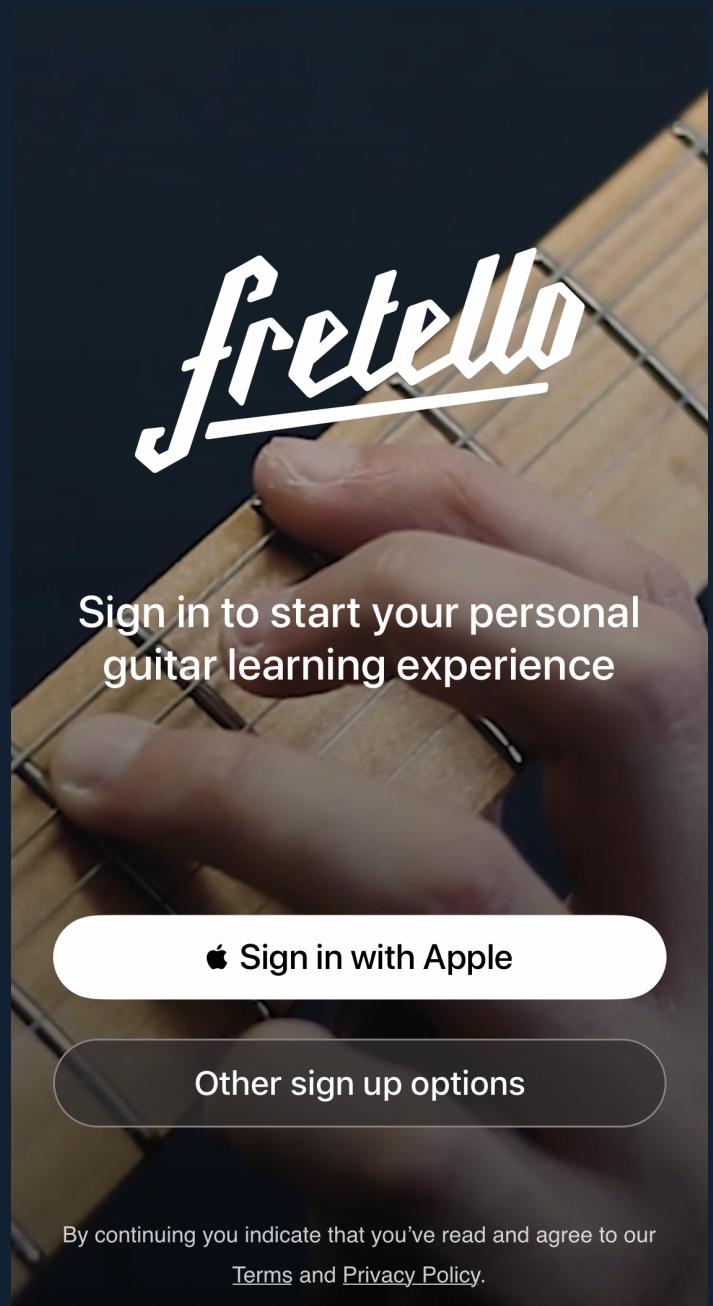
Apps die exklusiv 3rd-Party Login-Services nutzen

- » Facebook, Google, Twitter, Linked-In, Amazon, ...
- » Pflicht für neue Apps: ab heute
- » Pflicht für existierende Apps: ab April 2020

WER NICHT

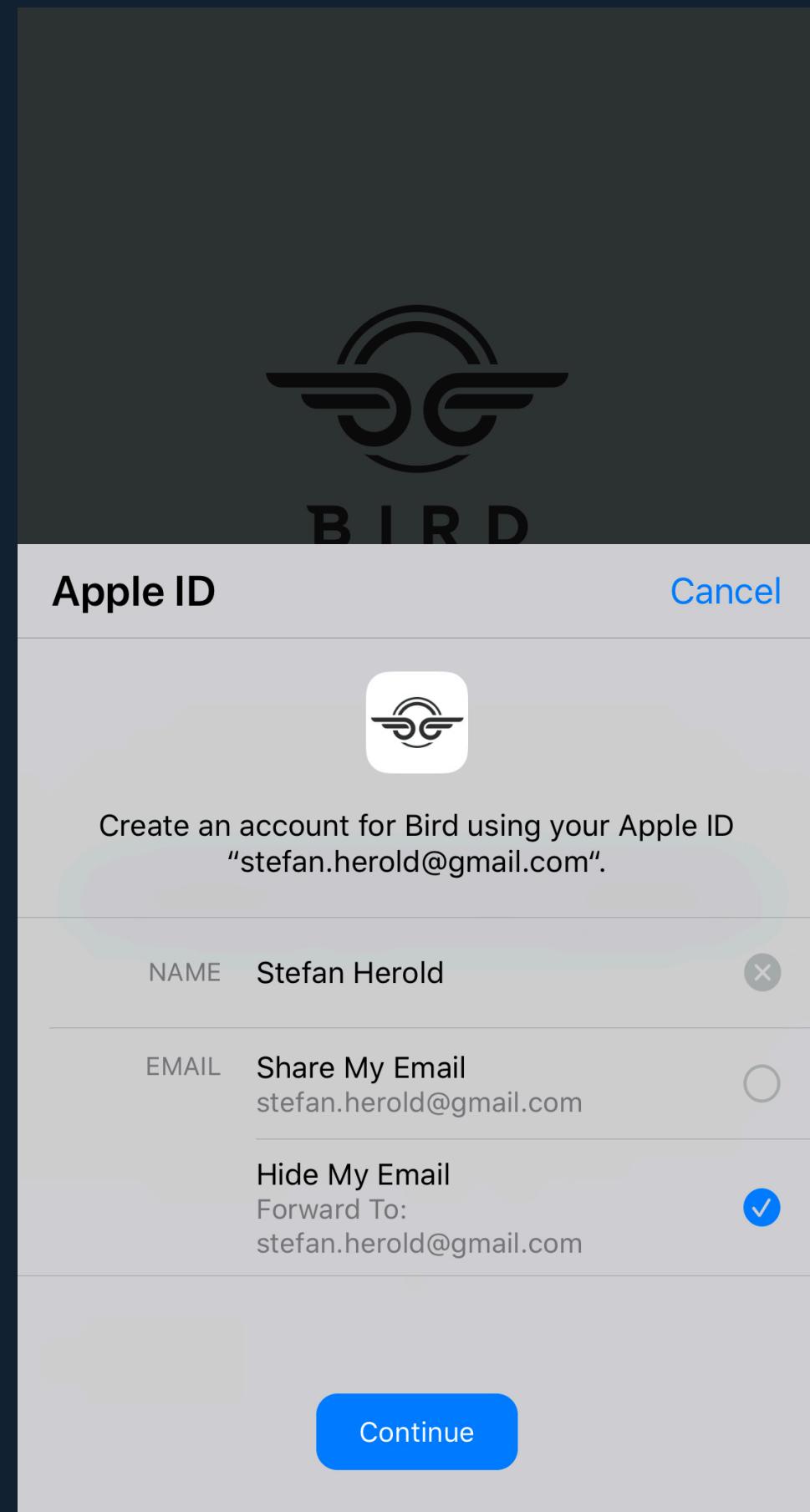
- » App nutzt firmeneigenes Login-System
- » App nutzt ausweisbasiertes Login-System
- » App ist Client für 3rd party / social service
- » Bildungs-, Enterprise- oder Business-App mit existierendem Firmen-Account

APPS



REGISTRIERUNG

APP	EMAIL	NAME
Bird	✓	✓
Lambus	✓	✓
Bring	✓	✓
Blinkist	✓	✗
Parcel	✗	✗



REGISTRIERUNG

Folgendes bekommen wir:

- » UserID • eindeutig, stabil über alle Geräte mit gleicher AppleID
- » Identity Token • JWT zur Nutzerverifizierung • 10 min
- » Auth Code • Refresh Token
- » Verifizierte Email • entfällt beim Onboarding
- » Vor- und Zuname • PersonNameComponents
- » Real User Indicator • Enum: likelyReal, unknown, unsupported

DEMO



VIELEN DANK FÜR'S ZUHÖREN



⚠ Demo App Code & Slides

<https://github.com/Blackjacx/SignInWithApple>

📺 Introducing Sign In with Apple - Session Video Notes

<https://github.com/Blackjacx/WWDC#introducing-sign-in-with-apple>

💻 Apple Docs

<https://developer.apple.com/sign-in-with-apple>

💡 Review Guidelines

<https://developer.apple.com/app-store/review/guidelines/#sign-in-with-apple>

🌐 REST API

<https://developer.apple.com/documentation/signinwithapplerestapi>

🐦 Twitter

@blackjacxxx

MEHR LINKS

- » Answers to your burning questions
<https://techcrunch.com/2019/06/07/answers-to-your-burning-questions-about-how-sign-in-with-apple-works>
- » Token Handling im Backend
<https://blog.curtisherbert.com/so-theyve-signed-in-with-apple-now-what>
- » Ray Wenderlich Tutorial mit SwiftUI
<https://www.raywenderlich.com/4875322-sign-in-with-apple-using-swiftui>
- » Sign in with Apple für Web
<https://developer.okta.com/blog/2019/06/04/what-the-heck-is-sign-in-with-apple>
- » 9To5Mac Artikel
<https://9to5mac.com/2019/10/15/how-to-use-sign-in-with-apple-iphone-ipad-mac>
- » How to integrate Sign In with Apple in your iOS app
<https://benoitpasquier.com/how-to-integrate-sign-in-with-apple-ios/>
- » Auth0 - Sign In With Apple
<https://auth0.com/docs/quickstart/native/ios-swift-siwa/00-login>
- » Why sign-in with apple may take you more than 5 minutes and how it works?
<https://dev.to/michalrogowski/why-sign-in-with-apple-may-take-you-more-than-5-minutes-and-how-it-works-55p6>
- » Erzeugen des Private Keys
<https://developer.apple.com/account/resources/authkeys/add>
- » Konfiguration erlaubter Email-Adressen
<https://developer.apple.com/account/resources/services/configure>
- » Why isn't the user data being returned every time?
<https://forums.developer.apple.com/thread/119826>

BACKUP

IOKI

- » Autonomous Driving
- » Erster fahrerloser Service Deutschlands
- » Bad Birnbach • Bayern
- » 2 km • Stadtzentrum - Bahnhof

REGISTRIERUNG

```
func didPressSignInWithApple(_ sender: UIButton) {  
  
    let provider = ASAuthorizationAppleIDProvider()  
    let request = provider.createRequest()  
    request.requestedScopes = [.email, .fullName] // optional - only request what's required  
  
    let controller = ASAuthorizationController(authorizationRequests: [request])  
    controller.delegate = self  
    controller.presentationContextProvider = self  
    controller.performRequests()  
}
```

LOGIN

Beim Appstart:

```
let provider = ASAuthorizationAppleIDProvider()  
  
provider.getCredentialState(forUserID: userId) { (state, error) in  
    // evaluate state  
}
```

State-Änderungen:

```
let name = ASAuthorizationAppleIDProvider.credentialRevokedNotification  
center.addObserver(forName: name, object: nil, queue: nil) { [weak self] _ in  
    self?.performSignOut()  
}
```