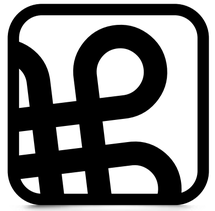


Macoun

Network Extensions

Klaus Rodewig und Mark Zimmermann

Experte

Professionelle Fazialpalmierung

SecureContact Pro

appnö

Ablauf

- Bestandteile der NetworkExtensions
- Entwicklungsschritte
- NEHotSpotHelper in Aktion
- Was ändert sich mit iOS 11?
- Sicherheitskritische Betrachtung

Bestandteile der NetworkExtension

Die einzelnen Komponenten

- Personal VPN
- Network Tunnel Protocol Client
- On-Device Network Content Filter
- WiFi-Hotspot Authentication

Personal VPN

(NEVPNManager API)

- Öffentliche WiFi-Hotspots sind nicht immer Vertrauenswürdig, eine sichere Kommunikation kann über ein persönliches VPN ermöglicht werden.
- Die NEVPNManager API bietet Apps die Möglichkeit, eine persönliche VPN-Konfiguration auf iOS und macOS zu erstellen und zu verwalten.
- Personal VPN Konfiguration erfolgt über eine eigene "Captive App". Ein Regelwerk, zur (De-)Aktivierung der VPN Verbindung, kann hinterlegt werden.

Network Tunnel Protocol Client

(NETunnelProvider API)

- Die NETunnelProvider-Familie von APIs erlaubt es iOS- und macOS-Geräte mit einem VPN-Server zu verbinden, der ein nicht standardmäßiges Netzwerk-Tunneling-Protokoll verwendet.
- Beispiele
  - auf **IP Ebene** (z.B. UDP Traffic, **Paket Tunnel Provider**) oder
  - auf **App Ebene** (Layer 3, **App Proxy Provider**)
- VPN Konfiguration können über MDM Service konfiguriert werden.
- Kooperative Unterstützung von Personal VPN & Network Tunnel Protocol Konfigurationen (Prio: Network Tunnel Protocol).

On-Device Network Content Filter

(NEFilterProvider API)

- Filtern des Internetzugriffs in lokalen Netzwerken wurde bisher durch On-Site-Content-Filter oder durch Global Proxy/VPN Lösungen (aufwändig) umgesetzt.
- Netzwerkinhalte können auf iOS-Geräten dynamisch gefiltert werden.
- Es wird ein Supervised Gerät benötigt.
- iOS bietet die Anzeige von templatebasierten "Block Pages"

WiFi-Hotspot Authentication

(NEHotspotHelper API)

- Mit iOS9 sind „Captive Network Apps“ möglich und bieten damit eine automatische Einwahl, in (hinterlegte) WiFi Hotspots.

- Diese bieten 2 Realisierungsarten:
  - Anmeldung an Captive Portals
  - Anmeldung an passwortgeschützten WiFis

Entwicklungsschritte

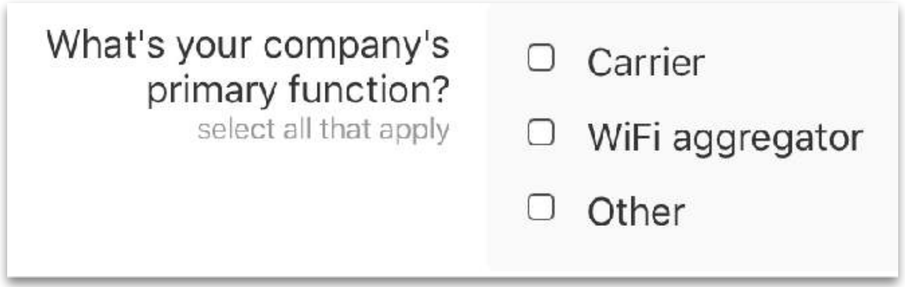
am Beispiel NEHotspotHelper

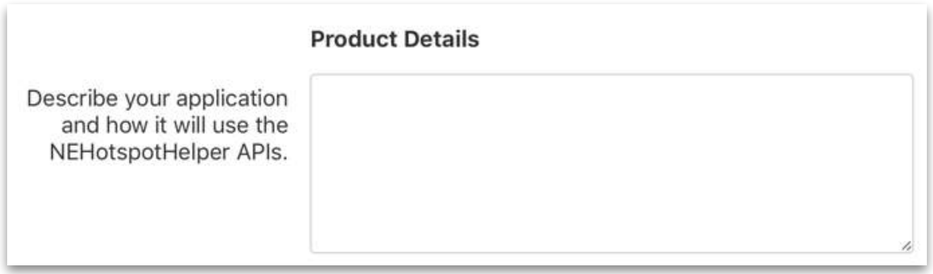
Grundlagen der Verwendung

- Entitlements beantragen
- Konfiguration der App
- Registrierung beim App-Start als Helper
- StateMachine verwaltet Zustände
- Beispiel: Anmeldung an einem Captive Portal

Entitlements beantragen

(https://developer.apple.com/contact/network-extension)

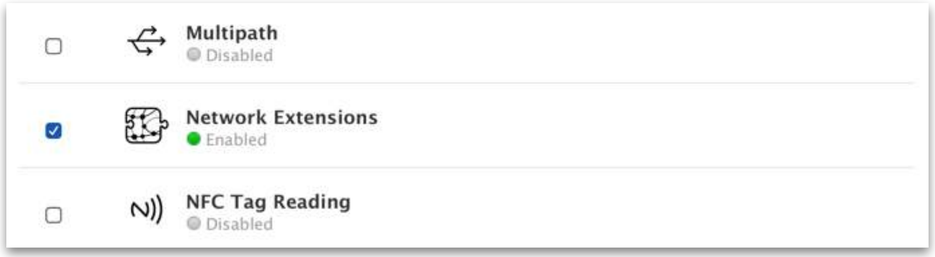




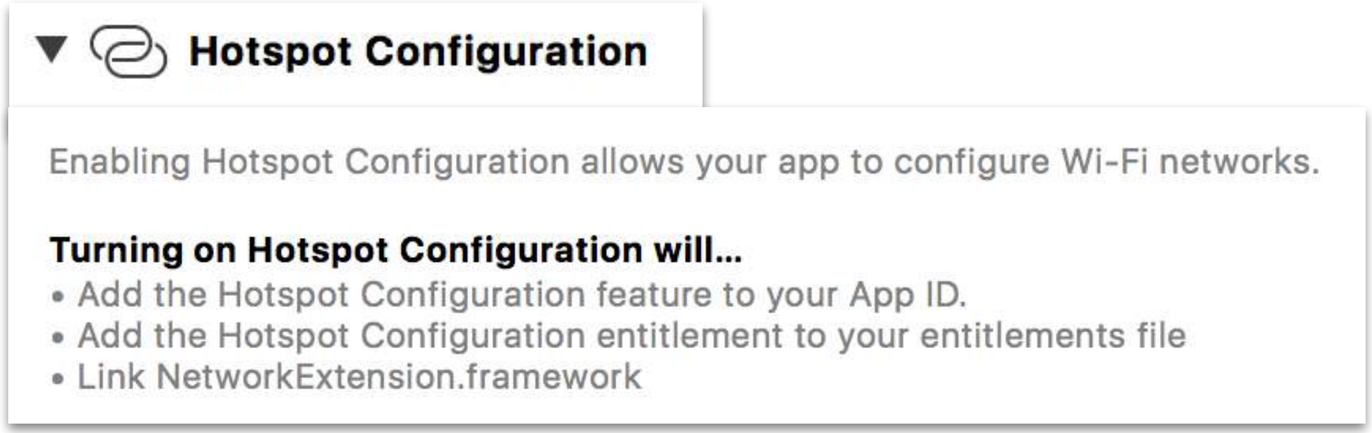
Grundlagen der Verwendung

- Entitlements beantragen ✔
- Konfiguration der App
- Registrierung beim App-Start als Helper
- StateMachine verwaltet Zustände
- Beispiel: Anmeldung an einem Captive Portal

Konfiguration der App



AppID im Portal erstellen / ändern



Entitlements in Xcode konfigurieren

Konfiguration der App

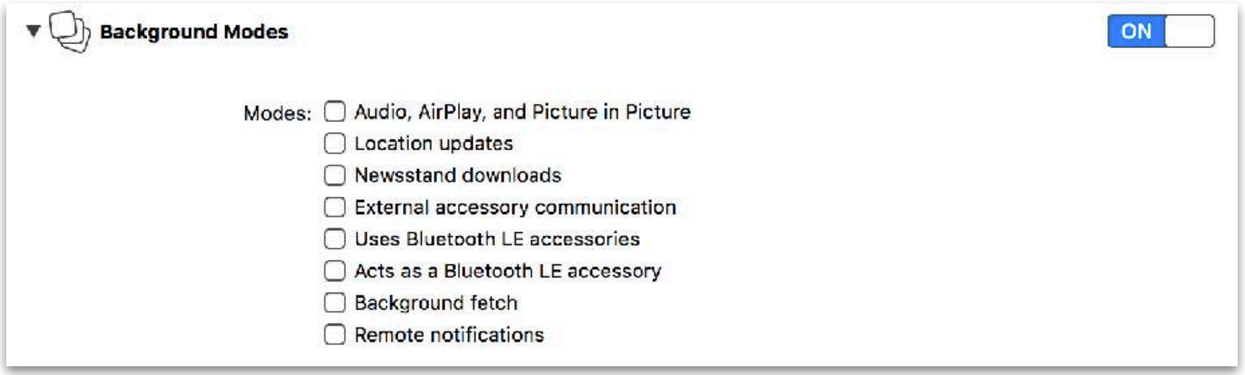
Info.plist

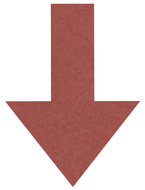
The application's Info.plist must include a UIBackgroundModes array containing network-authentication.

Entitlements

The application must set com.apple.developer.networking.HotspotHelper as one of its entitlements. The value of the entitlement is a boolean set to true.

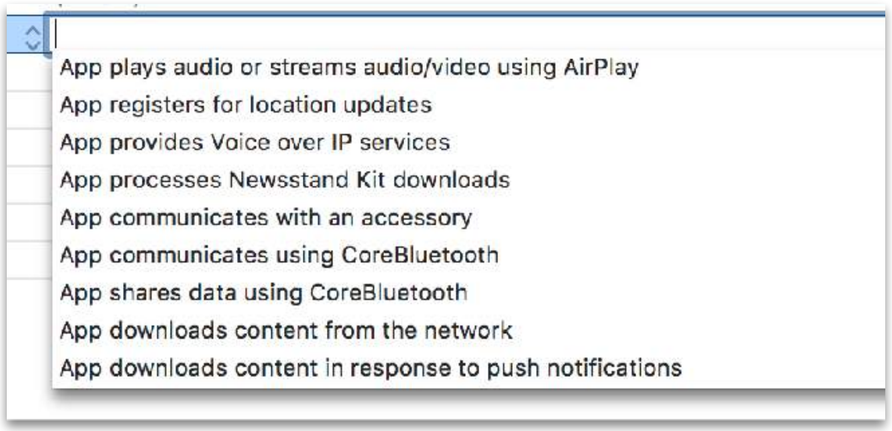
Konfiguration der App





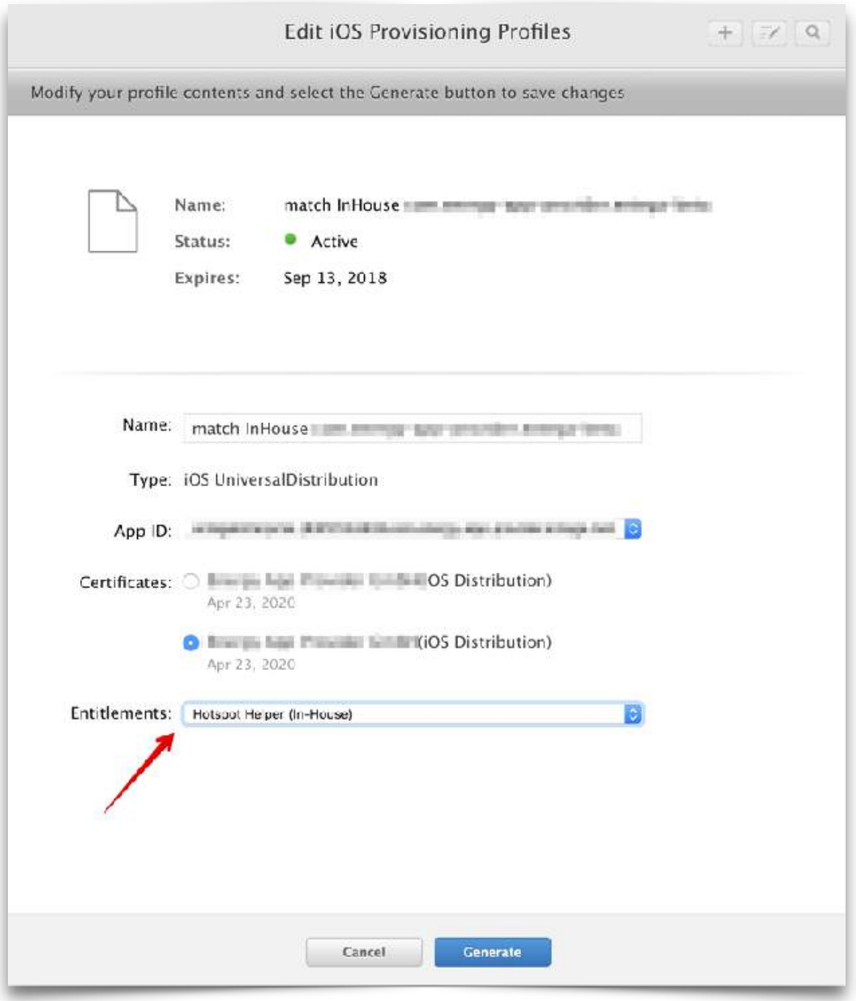
Key	Type	Value
▼ Entitlements File	Dictionary	(4 items)
com.apple.developer.networking.HotspotHelper	Boolean	YES

Konfiguration der App



▼ Required background modes	Array	(1 item)
Item 0	String	network-authentication

Provisionsprofil mit Entitlement



Grundlagen der Verwendung

- Entitlements beantragen ✔
- Konfiguration der App ✔
- Registrierung beim App-Start als Helper
- StateMachine verwaltet Zustände
- Beispiel: Anmeldung an einem Captive Portal

Registrierung als HotspotHelper

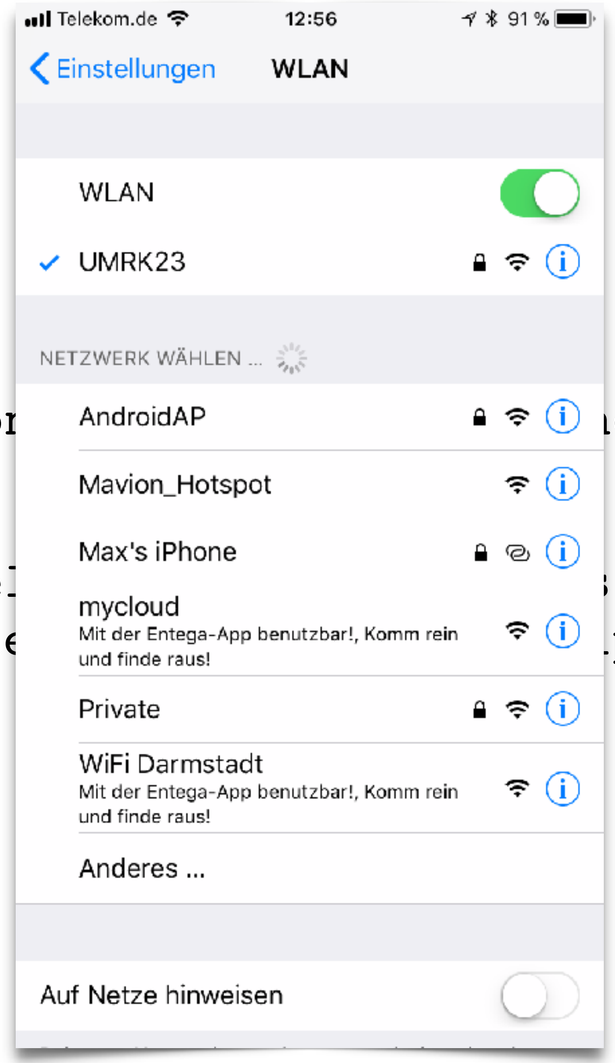
```
let options = [
    kNEHotspotHelperOptionDisplayName: NSString(string: "SOME NAME")
]

let success = NEHotspotHelper.register(options: options, queue:
DispatchQueue.main, handler: processHotspotHelperCommand)
```

Registrierung als HotspotHelper

```
let options = [
    kNEHotspotHelperOptio                    g(string: "SOME NAME")
]

let success = NEHotspotHel                   : options, queue:
DispatchQueue.main, handle                   perCommand)
```



Grundlagen der Verwendung

- Entitlements beantragen ✔
- Konfiguration der App ✔
- Registrierung beim App-Start als Helper ✔
- StateMachine verwaltet Zustände
- Beispiel: Anmeldung an einem Captive Portal

State Machine

```
@available(iOS 9.0, *)
public enum NEHotspotHelperCommandType : Int {
    case none
    case filterScanList
    case evaluate
    case authenticate
    case presentUI
    case maintain
    case logoff
}
```

Anfragen bearbeiten

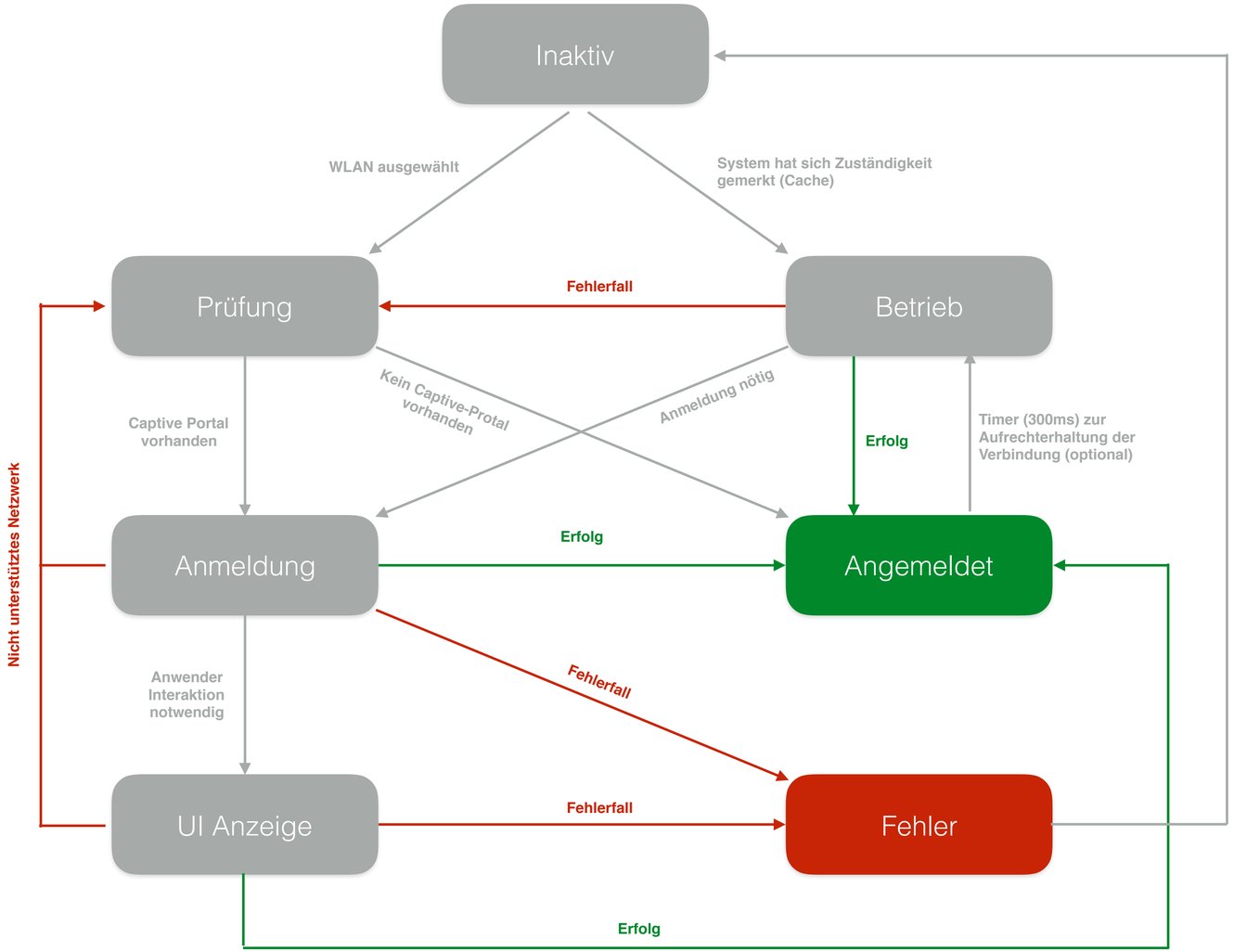
Netzwerkinterface setzen

```
func bind(to command: NEHotspotHelperCommand) -> RequestBuilder {
    request.bind(to: command)
    return self
}
```

Confidence setzen

```
public enum NEHotspotHelperConfidence : Int {
    case none
    case low
    case high
}
```

State Machine

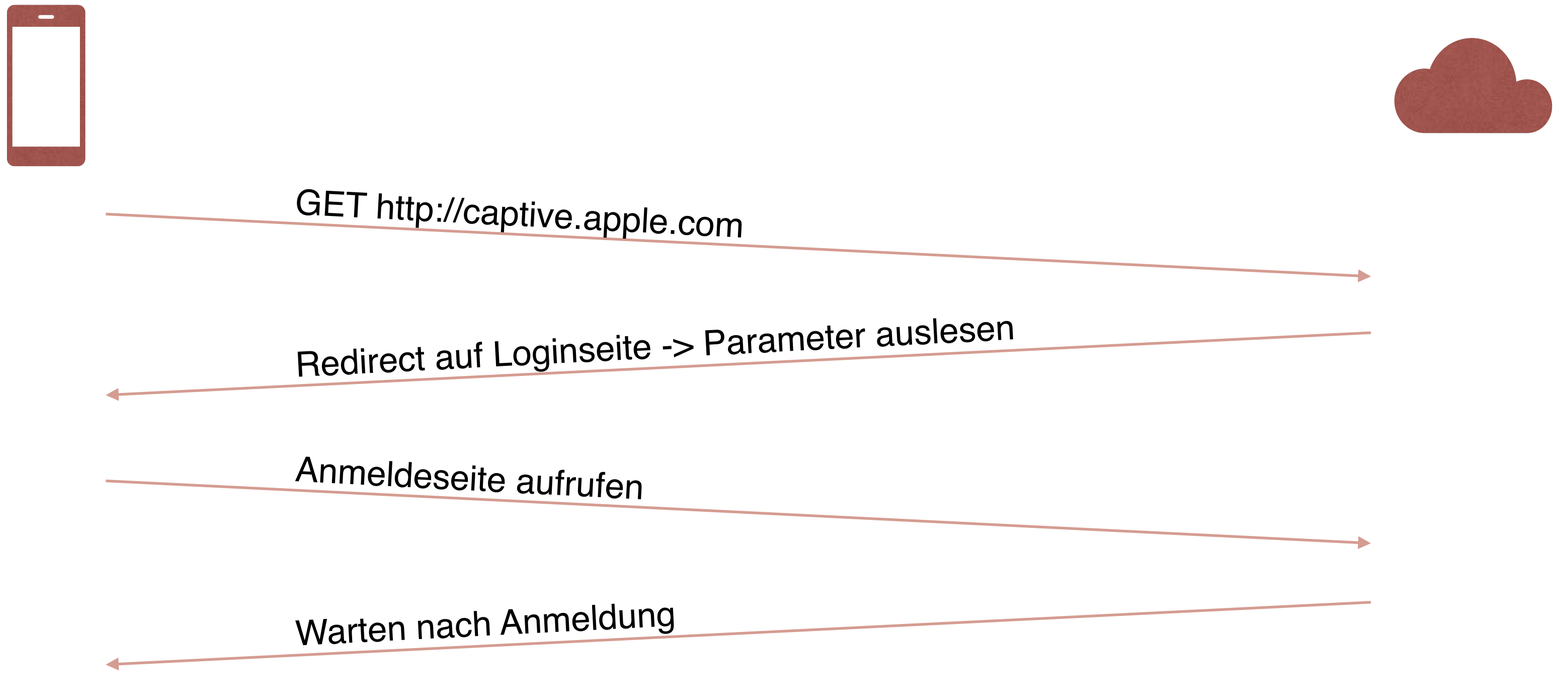


Weitere Übergänge sind möglich
Speziell: Von fast allen Status kann man zu Fehler kommen

Grundlagen der Verwendung

- Entitlements beantragen ✔
- Konfiguration der App ✔
- Registrierung beim App-Start als Helper ✔
- StateMachine verwaltet Zustände ✔
- Beispiel: Anmeldung an einem Captive Portal

Login beim Captive Portal



Demo

Was ändert sich mit iOS 11?

Neue APIs

- DNS Proxy App Extension
- Hotspot Configuration

DNS Proxy App Extension

(NEDNSProxyProvider)

- Mit iOS 11 erhält die Extension alle DNS Anfragen zur weiteren Verarbeitung und kann eigene DNS-Resolver-Betreiber aktivieren
- Unterstützt 2 Verfahren
  - DNS über HTTP
    Beispieleinsatz: Zugriffe mit ständig wechselnder IP-Adresse können können trotzdem einem DNS Namen entsprechen (siehe dynamischem DNS Anbieter)
  - DNS über TLS
    Beispieleinsatz: DNS-Anfragen geben einen Schatz an Metadaten preis, aus denen zum Beispiel DNS-Resolver-Betreiber einiges über das Verhalten eines Nutzers auslesen können: Welche Seiten besucht er im Internet, welche Mail-Server verwendet er und von wem bezieht er gerade den Schlüssel für die nachfolgende verschlüsselte Kommunikation. DNS über TLS adressiert dies.

Hotspot Configuration

(NEHotSpotConfiguration)

- Mithilfe dieser App-Extension ist eine App in der Lage mit einem separaten WLAN-Router/Gerät temporären oder permanent Kontakt aufzunehmen.
- Extension unterstützt zur Authentifizierung
  Open, WEP, WPA, EAP und Hotspot 2.0
- Einsatzbeispiel: App Verbindet sich per WIFI mit einer Kamera

Sicherheitskritische Betrachtung

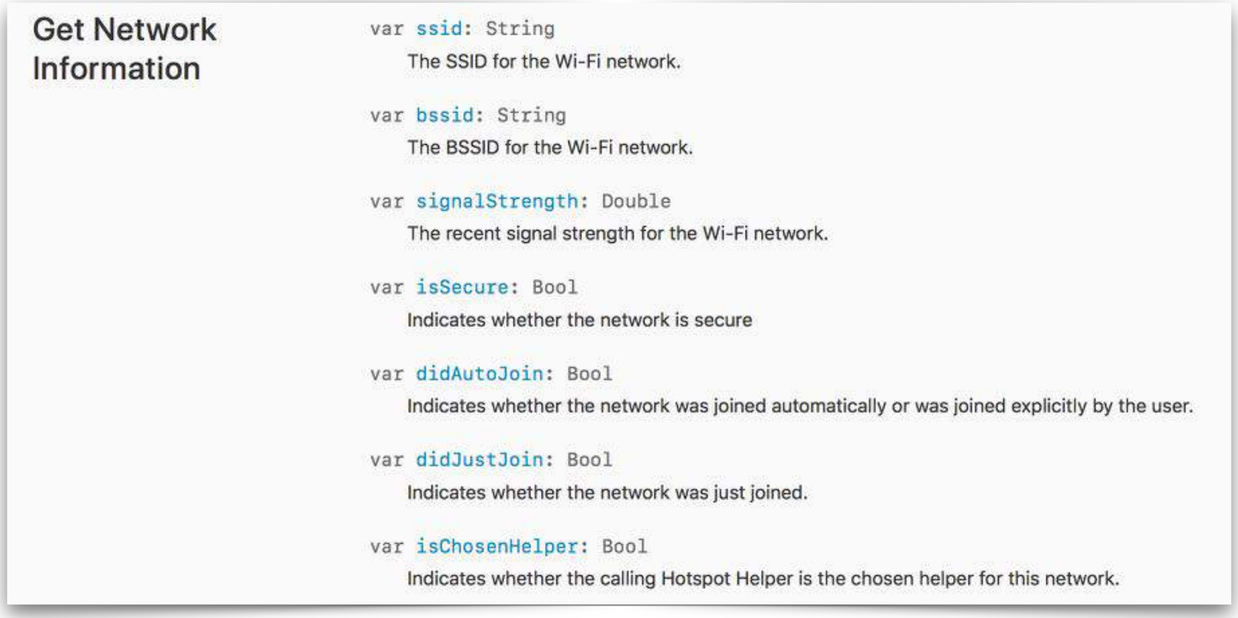
🤡 Spaß mit HotSpots 🤡

Datensammlung

- Die nette App von nebenan:
  Was kann schon passieren ?

Datensammlung

- Die nette App von nebenan: Was kann schon passieren ?
- 🤡 Danke an die 45sek Rechenzeit in der StateMachine: filterScanList und den reichhaltigen Informationen…



Get Network Information

`var ssid: String`
The SSID for the Wi-Fi network.

`var bssid: String`
The BSSID for the Wi-Fi network.

`var signalStrength: Double`
The recent signal strength for the Wi-Fi network.

`var isSecure: Bool`
Indicates whether the network is secure

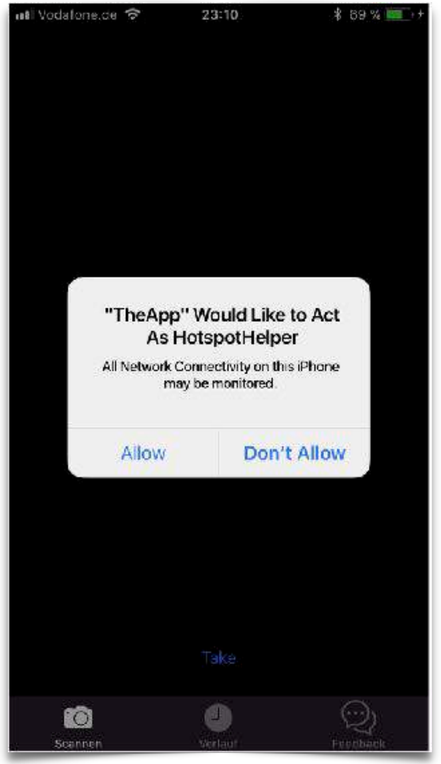
`var didAutoJoin: Bool`
Indicates whether the network was joined automatically or was joined explicitly by the user.

`var didJustJoin: Bool`
Indicates whether the network was just joined.

`var isChosenHelper: Bool`
Indicates whether the calling Hotspot Helper is the chosen helper for this network.

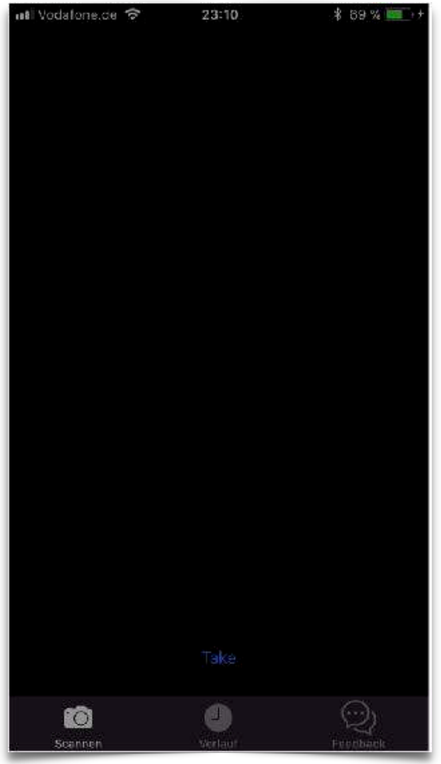
Datensammlung

- Die nette App von nebenan: Was kann schon passieren ?
- 🤡 Danke an die 45sek Rechenzeit in der StateMachine: filterScanList
- Moment, hätte ich nicht „gewarnt“ werden müssen ?



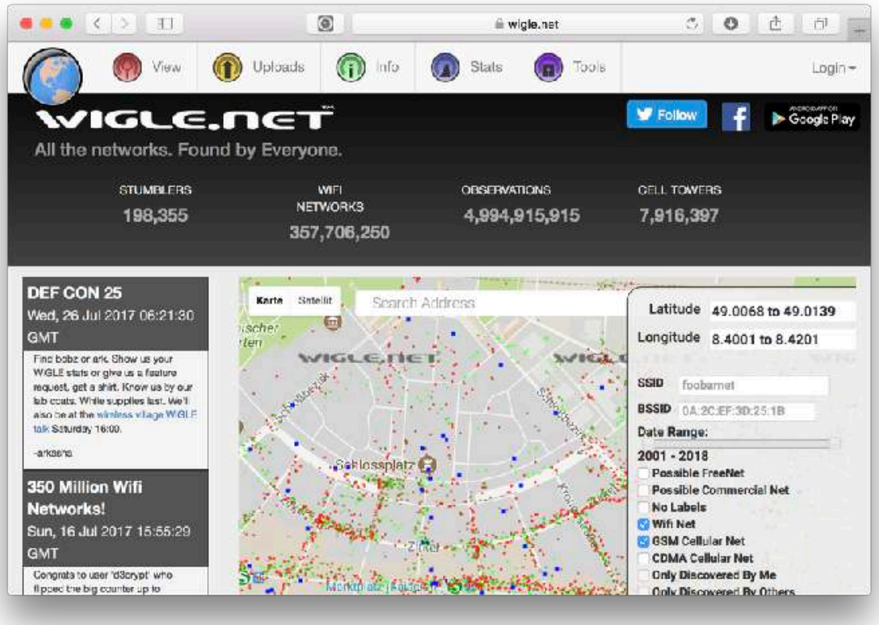
Datensammlung

- Die nette App von nebenan: Was kann schon passieren ?
- 🤡 Danke an die 45sek Rechenzeit in der StateMachine: filterScanList
- Moment, hätte ich nicht „gewarnt“ werden müssen ?
  Ja schon … aber **gibt es nicht** ! 🤡



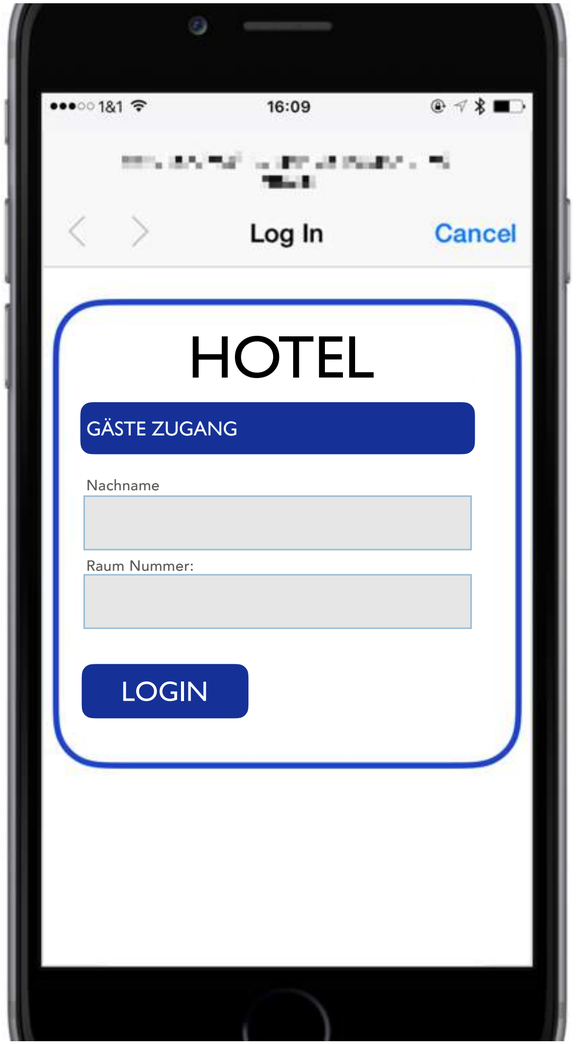
Datensammlung

- Die nette App von nebenan: Was kann schon passieren ?
- 🤡 Danke an die 45sek Rechenzeit in der StateMachine: filterScanList
- Moment, hätte ich nicht „gewarnt“ werden müssen ?
  Ja schon … aber **Nein** ! 🤡
- …dann Mappen wir das ganze, was im Hintergrund geloggt wird, einfach mal auf eine Karte…. 🤡



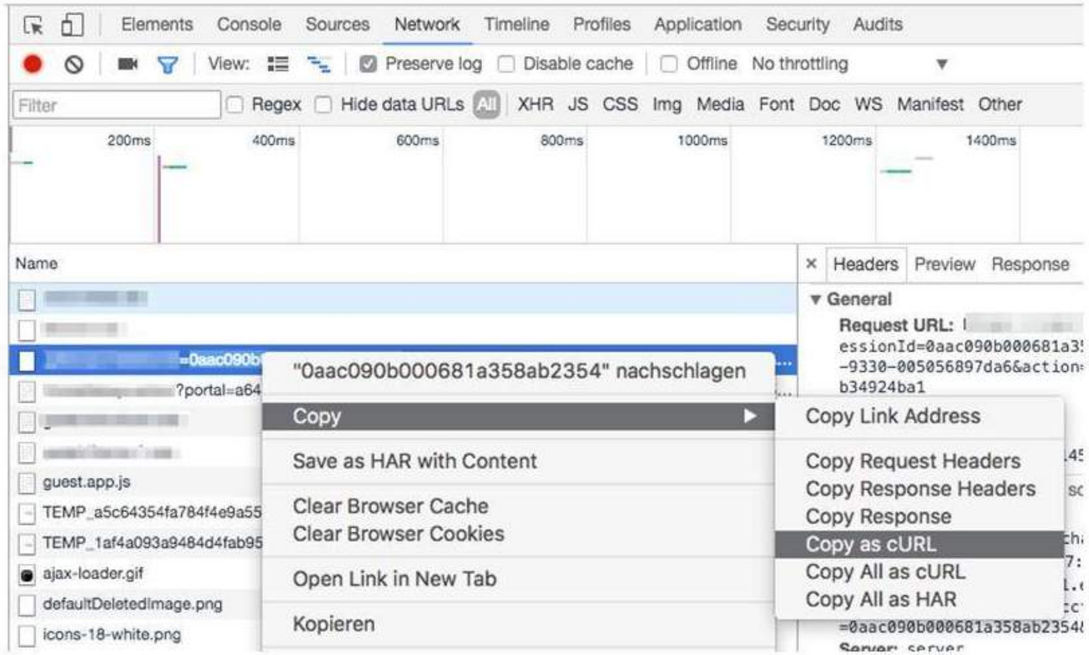
Datensammlung

- Alleine im Hotel: Was kann schon passieren ?



- Alleine im Hotel: Was kann schon passieren ?
- …mit ein wenig Analyse (Google Chrome)

```
Form item: “last_name“ = “NACHNAME“
Form item: “room“ = “000“
Form item: “submit“ = “Login“
```



Datensammlung

- Alleine im Hotel: Was kann schon passieren ?
- …mit ein wenig Analyse (Google Chrome)
- …🤡

Nachname	Zimmer		
Müller	265	192	187
Schmidt	284	170	
Fischer	179	155	
Meyer	161		
Becker	248	173	
Schulz	221		
Schäfer	-		
Koch	-		
...			
Köhler	137		
Schröder	128		
...			

$$\frac{(399\text{-}100)\times 119\text{ms}}{1000\text{ms}} = 35sec$$

$$\frac{67\text{ms}}{1000\text{ms}} = 0{,}067sec$$

$$\frac{15\times 35sec + 13\times 0067sec}{60sec} = 8\,\text{min}$$

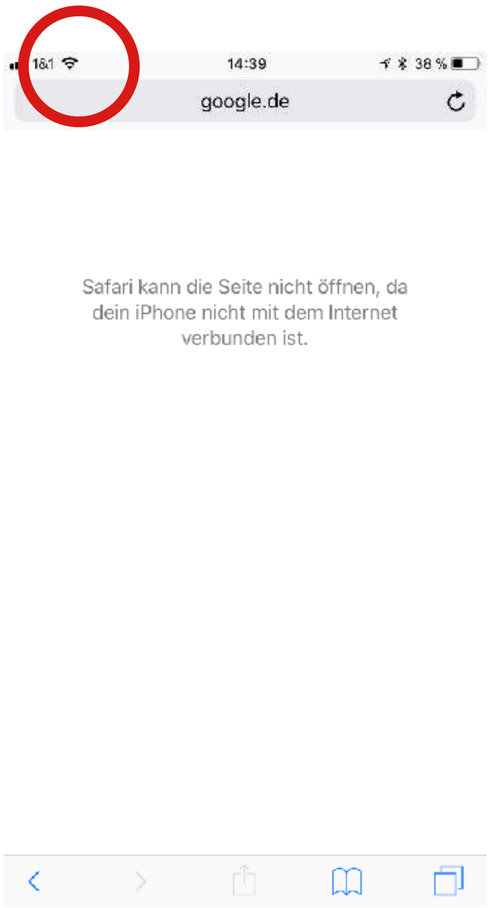
$$\frac{8\text{min}\times 60sec}{45sec} \approx 11$$

Störung der Kommunikation

- Die nette App von nebenan (noch eine andere):
  Was kann schon passieren ?

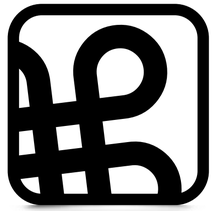
Störung der Kommunikation

- Die nette App von nebenan (noch eine andere): Was kann schon passieren ?
- 🤡 Danke an die StateMachine: authenticate
- Meldet der App-Code in der StateMachine, einen erfolgreichen Login, kommt es zu einem invaliden Netzwerkzustand (kein Internetzugang, dennoch WLAN „an")



Fragen?

Vielen Dank



Macoun