

# BERECHTIGUNGEN

---

Präsentation von Jacqueline Menke

# GLIEDERUNG

- Einleitung
- Arten von Berechtigungen
- Berechtigungen anfordern
- Runtime Permissions
- Live Coding
- Implementation
- Berechtigungsgruppen
- Der richtige Zeitpunkt
- Zusammenfassung



WhatsApp Messenger

Version 2.11.432 can access



In-app purchases

- Google Play billing service



Device & app history

- retrieve running apps



Identity

- find accounts on the device
- add or remove accounts
- read your own contact card



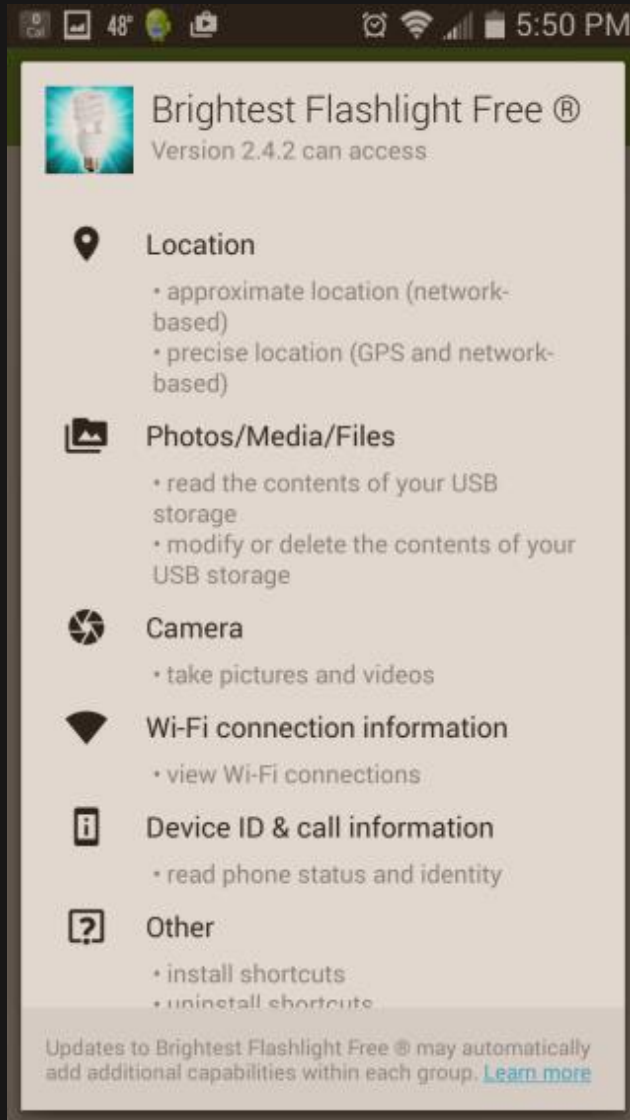
Contacts/Calendar

- read your contacts
- modify your contacts

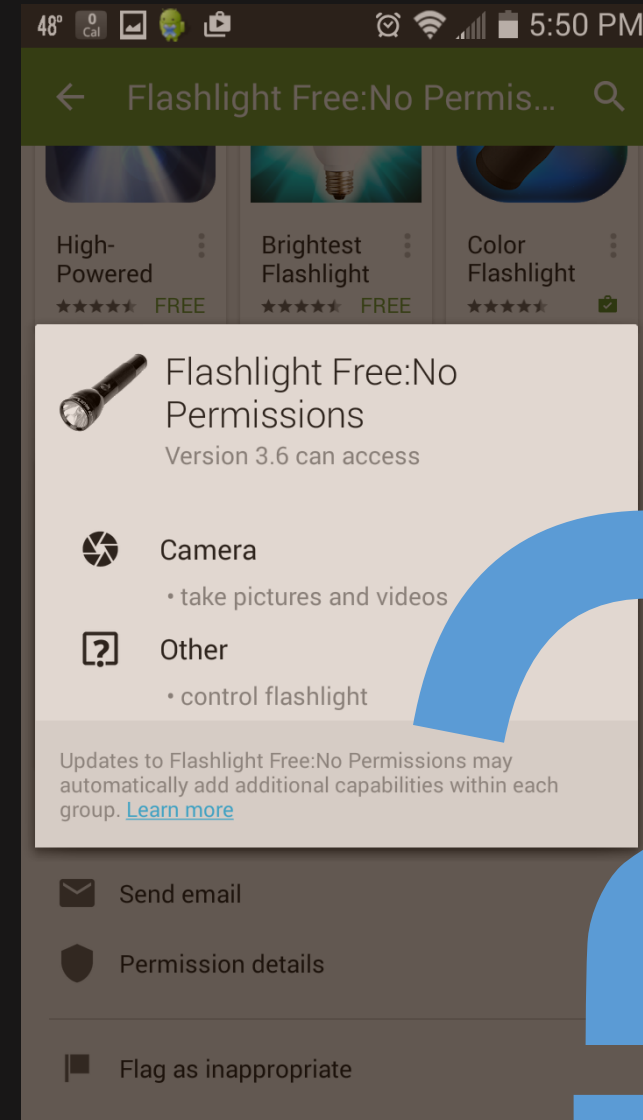


Location

- approximate location (network-based)
- precise location (GPS and network-based)



VS.



# WAS SIND BERECHTIGUNGEN?

Android-Apps werden in einer Sandbox ausgeführt

→ Zugriffe auf andere Systemkomponenten  
oder fremde Anwendungen erfordern

**Berechtigungen**

---

Man unterscheidet **normale** und **gefährliche** Berechtigungen

# NORMALE BERECHTIGUNGEN

Zugriff auf Daten/Ressourcen außerhalb der eigenen Sandbox **ohne großen Einfluss** auf die Privatsphäre

Werden beim Anfordern  
**automatisch** gewährt

## Beispiele

Zugriffe auf das Internet

Setzen der Zeitzone

Setzen eines Wallpapers

Benachrichtigung über Abschluss des Boot-Vorgangs

# GEFÄHRLICHE BERECHTIGUNGEN

Zugriff auf kritische Daten/Ressourcen außerhalb der eigenen Sandbox **mit Einfluss auf die Privatsphäre**

Müssen vor Nutzung **explizit** vom Anwender gewährt oder verweigert werden

## Beispiele

lesender oder schreibender Zugriff auf Kontakt- und Kalenderdaten

Nutzung von Kamera oder Mikrofon

Versenden oder Empfangen von SMS

Laden und Speichern von Dateien

# BERECHTIGUNGEN ANFORDERN

Deklaration von normalen und gefährlichen Berechtigungen in der Manifestdatei, z.B.:

```
<uses-permission  
    android:name="android.permission.RECEIVE_SMS" />
```

# BERECHTIGUNGEN ANFORDERN

Bis einschließlich Android 5 (Lollipop): Abfrage während der Installation

---

Ab Android 6 (Marshmallow): **Runtime Permissions**,  
Berechtigungsabfrage zur Laufzeit

Selektives Zustimmung und nachträgliches  
Entziehen/Gewähren von Berechtigungen möglich

Voraussetzung: **targetSdkVersion** in build.gradle **mind. 23**



# LIVE CODING

---

App zur Anzeige der Anzahl der verpassten Anrufe

# IMPLEMENTATION

```
import android.Manifest;
```

```
import android.content.pm.PackageManager;
```

`checkSelfPermission()` - prüfen, ob eine Berechtigung gegeben ist

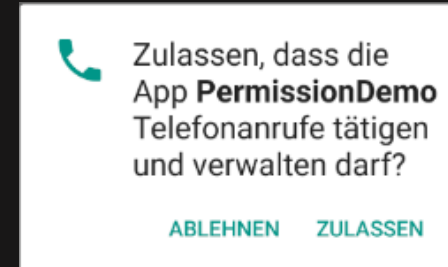
`requestPermission()` - Berechtigung anfordern

`shouldShowRequestPermissionRationale()` – soll den User über den Grund für eine benötigte Berechtigung informieren, wenn dieser schon **mindestens einmal** diese Berechtigung **verweigert** hat

`onRequestPermissionsResult()` – wird automatisch nach `requestPermission` aufgerufen, enthält Ergebnis der Usereingabe

# BERECHTIGUNGSGRUPPEN

```
<uses-permission  
    android:name=  
        "android.permission.READ_CALL_LOG"  
/>
```

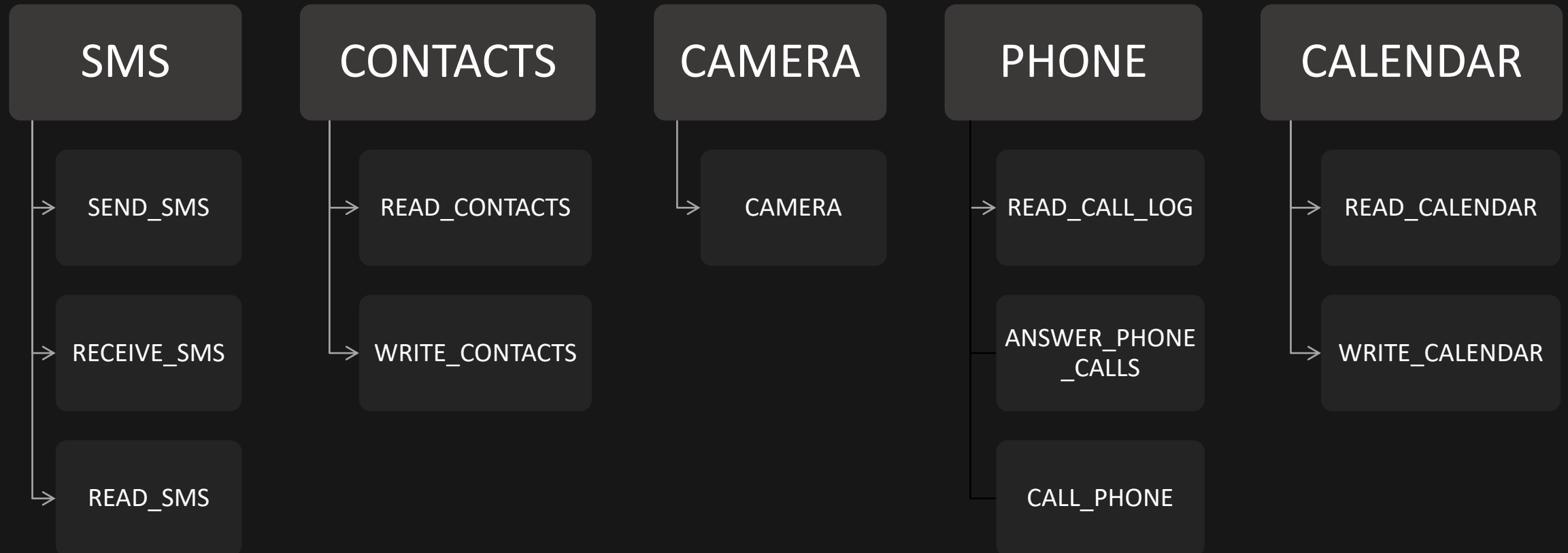


Android fasst Berechtigungen zu **Berechtigungsgruppen** zusammen

D.h., dass **kein Meldungsdialog** für neue Berechtigungen erscheint, die zu einer bereits gewährten Gruppe gehören!

# BERECHTIGUNGSGRUPPEN

Einige Beispiele:



# DER RICHTIGE ZEITPUNKT FÜR BERECHTIGUNGEN

*onCreate()* – nicht optimal, wird nur aufgerufen wenn die Activity noch nie gestartet oder zerstört wurde

*onStart()* – optimal ab Android 7, darunter problematisch: wenn Benutzer Bildschirm dreht, während ein Berechtigungsdialog offen ist, öffnet sich ein weiterer

→ Mögl. Lösung: Shared Preferences: abspeichern, dass Antwort auf eine Berechtigungsanfrage aussteht

# ZUSAMMENFASSUNG

- Gefährliche Berechtigungen müssen explizit gewährt werden
- Definition ALLER Berechtigungen in Manifest-Datei
- Runtime Permissions ab Android 6
- Berechtigungen werden in Gruppen zusammengefasst
- Berechtigungen üblicherweise in onStart() anfordern

---

Viele Apps fordern zu viele und unpassende Berechtigungen an, deswegen genau prüfen, welche Berechtigungen wirklich benötigt werden und nur diese anfordern!

# ENDE

---

## Quellen:

Android 7, Thomas Künnet, Rheinverlag

<http://www.seobook.com/designing-for-privacy>, 6.11.17

<https://developer.android.com/guide/topics/permissions/requesting.html#normal-dangerous>, 9.11.17

<https://stackoverflow.com/questions/27288167/create-permission-screen-like-playstore>, 9.11.17