DHBW
Duale Hochschule
Baden-Württemberg
Mannheim

Entwicklung mobiler Applikationen

WS2022/2023

Jan Brodhaecker | November 2022

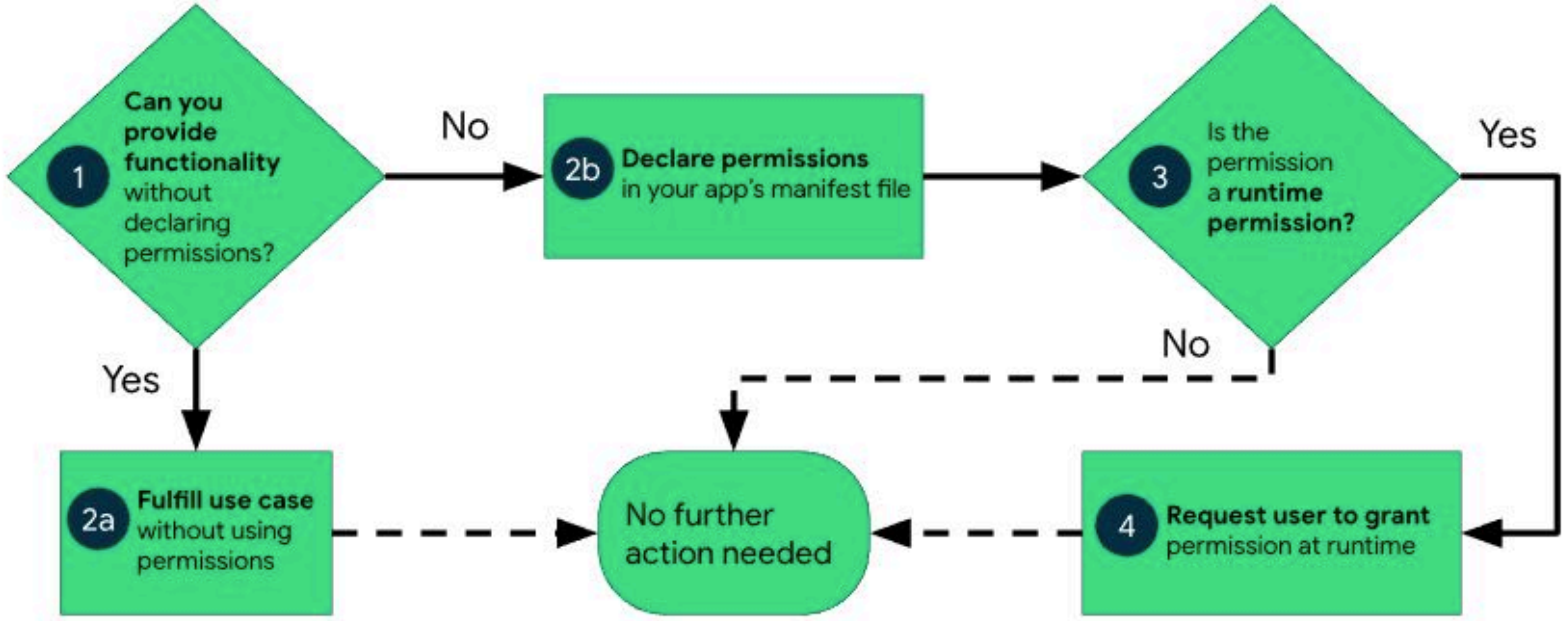
Agenda

- Permissions
- Intents

Permissions

- Android benötigt ein Berechtigungskonzept, welches dabei hilft …
  - Daten zu schützen (bspw. System-Informationen, Kontakte, …)
  - Aktionen zu verhindern (bspw. ungewollte Foto- oder Ton-Aufnahmen, …)

Permissions

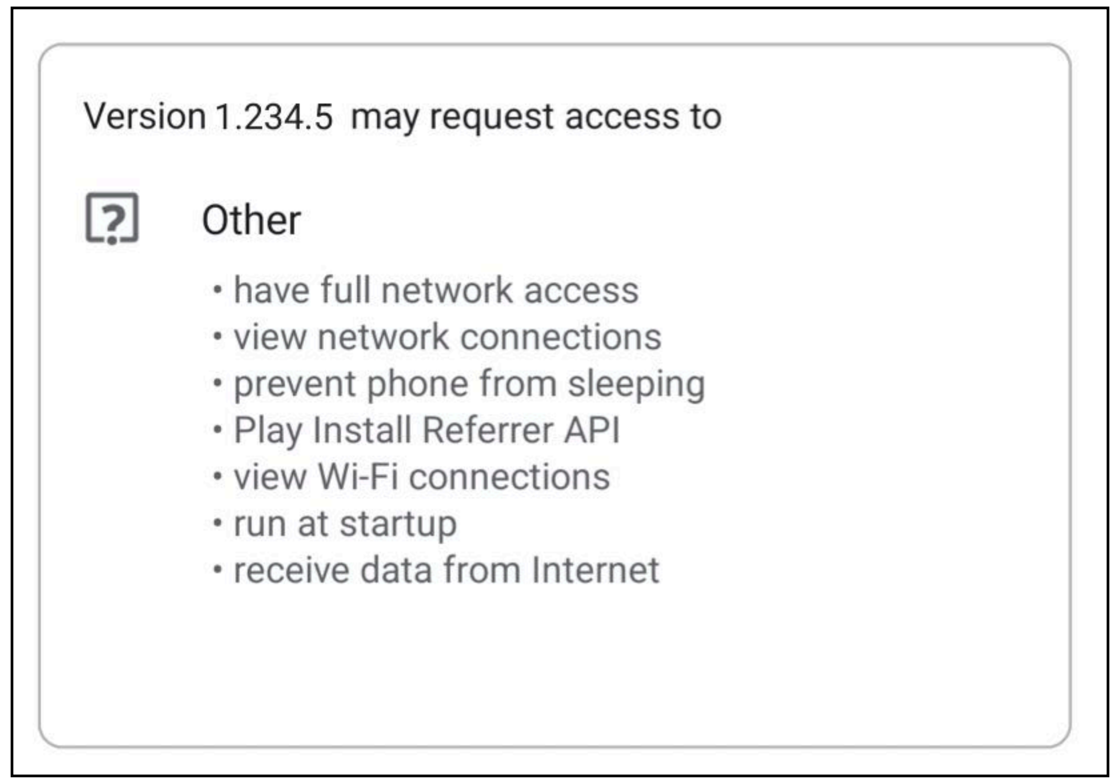

1 Can you provide functionality without declaring permissions?
No
2b Declare permissions in your app's manifest file
3 Is the permission a runtime permission?
Yes
No
Yes
2a Fulfill use case without using permissions
No further action needed
4 Request user to grant permission at runtime

https://developer.android.com/guide/topics/permissions/overview

Permissions

- verschiedene typen von Berechtigungen
  - **Install-time** - auto. mit Installation, werden im Playstore dargestellt
  - **Normal** - minimales Risiko, leicht erweiterter Zugriff außerhalb der Sandbox
  - **Signature** - sofern mit der selben Signatur signiert wurde
  - **Runtime** - werden zur Laufzeit abgefragt (seit API >= 23)
  - **Special** - hauptsächlich für OEM Anbieter, um bestimmte Funktionalität zu schützen

install-time permissions

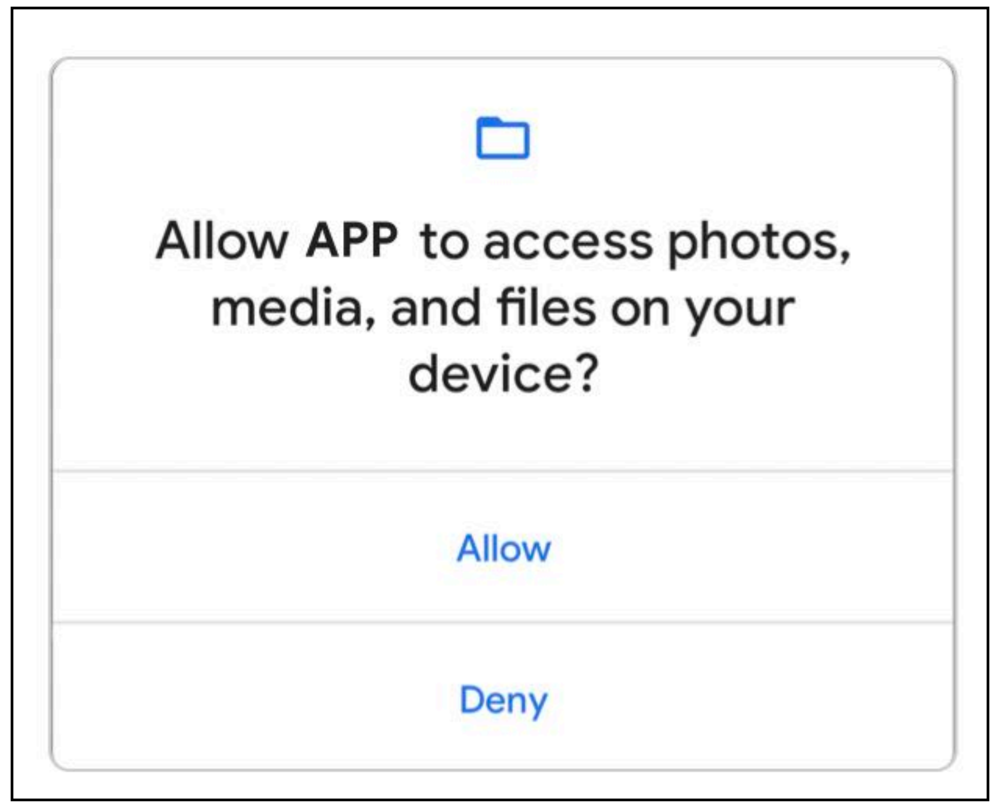

Version 1.234.5 may request access to

Other

- have full network access
- view network connections
- prevent phone from sleeping
- Play Install Referrer API
- view Wi-Fi connections
- run at startup
- receive data from Internet

https://developer.android.com/guide/topics/permissions/overview

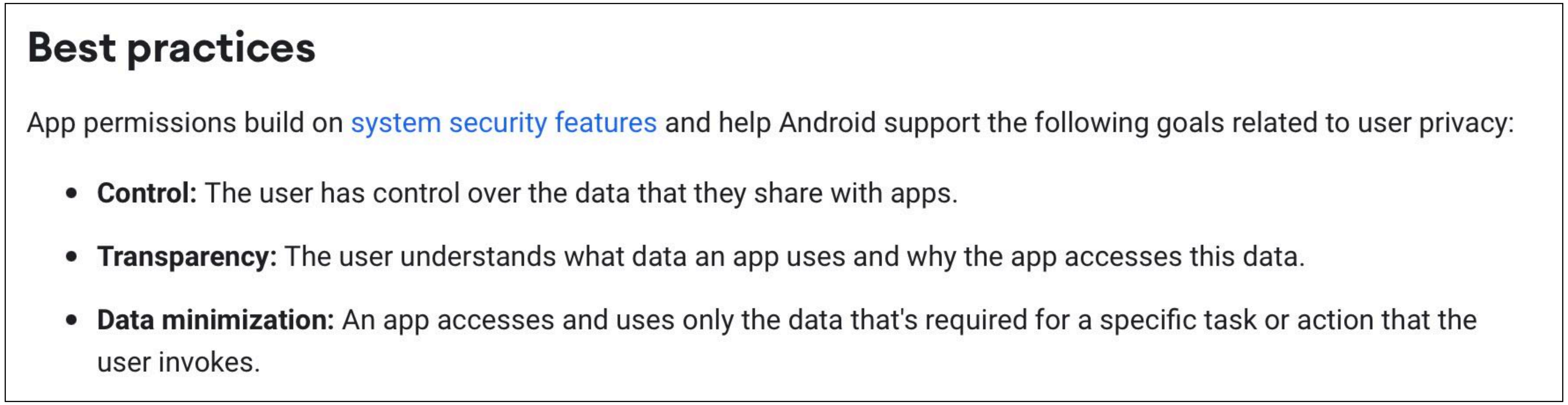
runtime permissions


Allow **APP** to access photos, media, and files on your device?

https://developer.android.com/guide/topics/permissions/overview

Permissions

https://developer.android.com/guide/topics/permissions/overview#best-practices



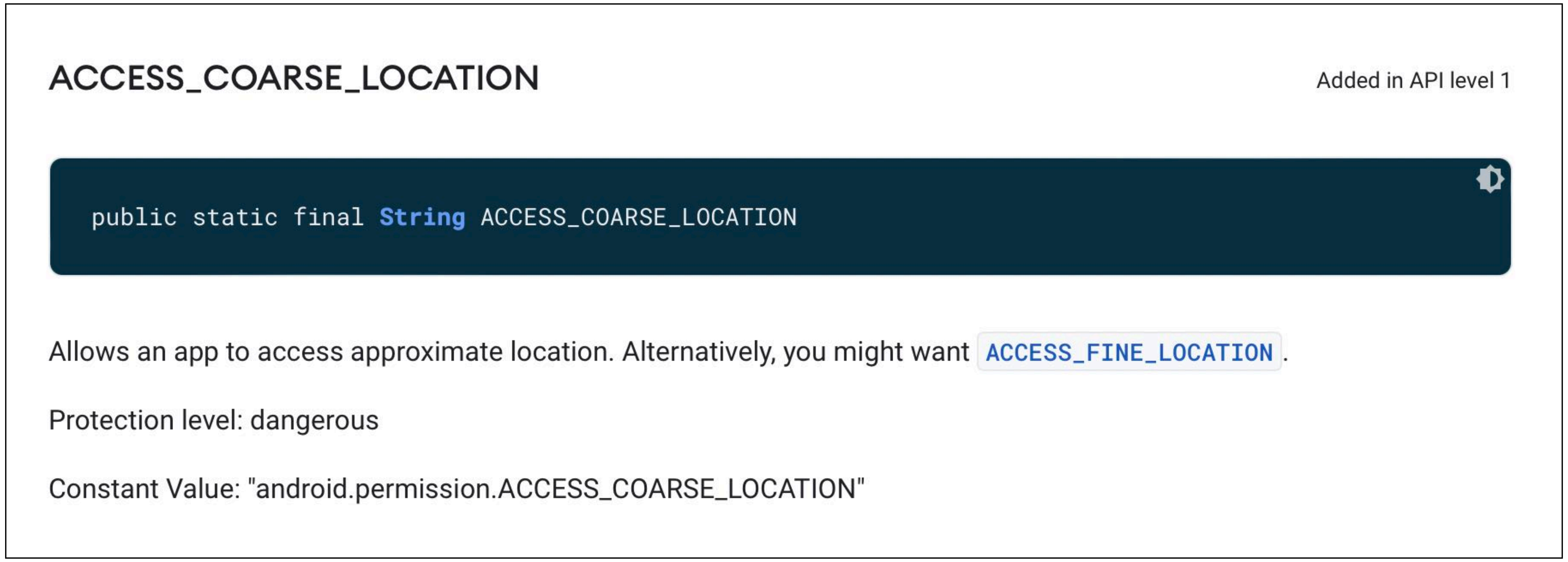
Best practices

App permissions build on system security features and help Android support the following goals related to user privacy:

- **Control:** The user has control over the data that they share with apps.
- **Transparency:** The user understands what data an app uses and why the app accesses this data.
- **Data minimization:** An app accesses and uses only the data that's required for a specific task or action that the user invokes.

- Request a minimal number of permissions
- Associate runtime permissions with specific actions
- Consider your app's dependencies
- Be transparent
- Make system accesses explicit

Permissions


ACCESS_COARSE_LOCATION

Added in API level 1

```
public static final String ACCESS_COARSE_LOCATION
```

Allows an app to access approximate location. Alternatively, you might want `ACCESS_FINE_LOCATION`.

Protection level: dangerous

Constant Value: "android.permission.ACCESS_COARSE_LOCATION"

https://developer.android.com/reference/android/Manifest.permission

Permissions

- Beispiel Standortabfrage
  - Permission(s) in Manifest.xml eintragen
  - Permission(s) zur Laufzeit abfragen
  - FusedLocationProvider verwenden

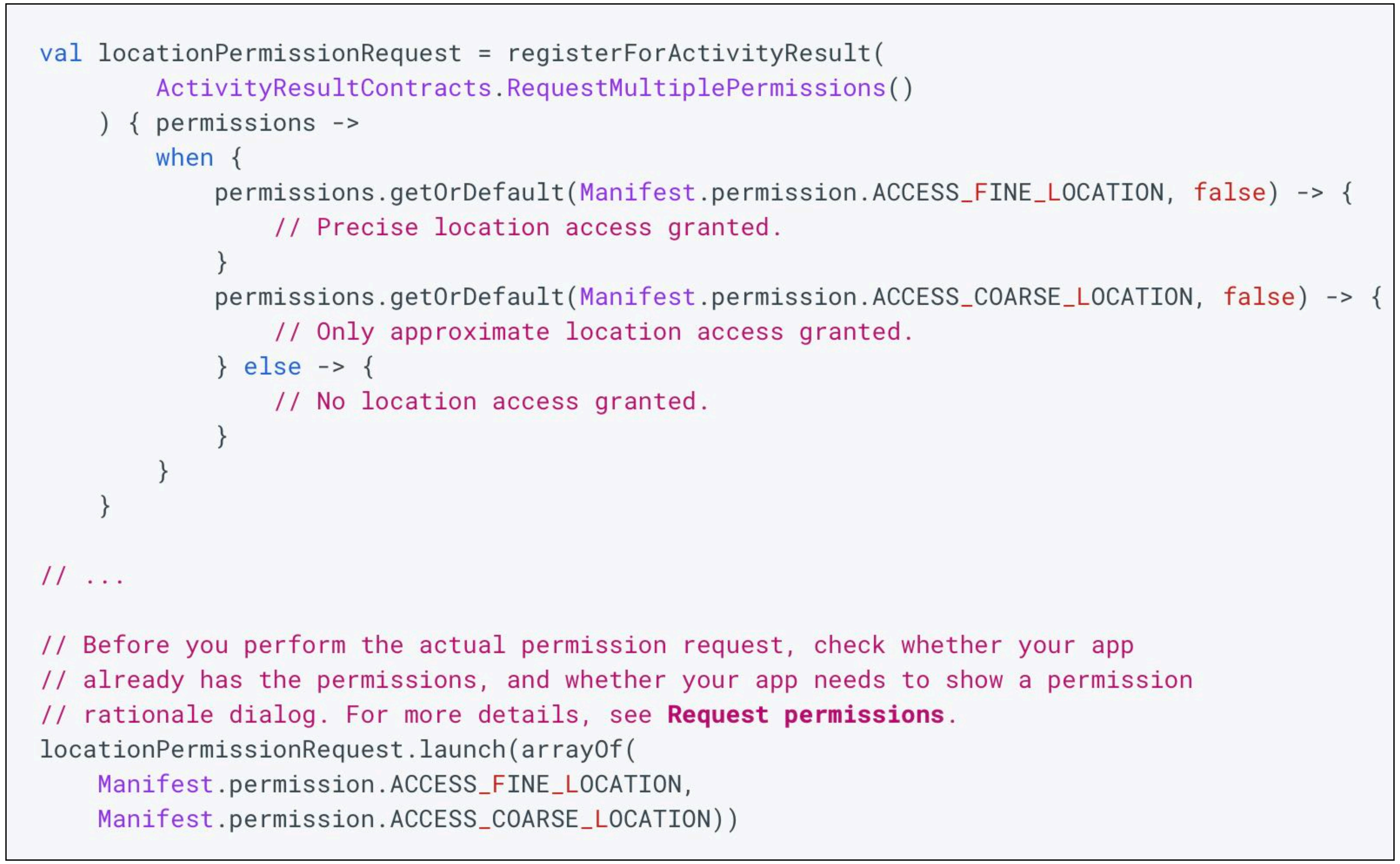
Permissions

```
<manifest ... >
  <!-- Always include this permission -->
  <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />

  <!-- Include only if your app benefits from precise location access. -->
  <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
</manifest>
```

https://developer.android.com/training/location/permissions

Permissions



```kotlin
val locationPermissionRequest = registerForActivityResult(
        ActivityResultContracts.RequestMultiplePermissions()
    ) { permissions ->
        when {
            permissions.getOrDefault(Manifest.permission.ACCESS_FINE_LOCATION, false) -> {
                // Precise location access granted.
            }
            permissions.getOrDefault(Manifest.permission.ACCESS_COARSE_LOCATION, false) -> {
                // Only approximate location access granted.
            } else -> {
                // No location access granted.
            }
        }
    }

// ...

// Before you perform the actual permission request, check whether your app
// already has the permissions, and whether your app needs to show a permission
// rationale dialog. For more details, see Request permissions.
locationPermissionRequest.launch(arrayOf(
    Manifest.permission.ACCESS_FINE_LOCATION,
    Manifest.permission.ACCESS_COARSE_LOCATION))
```

https://developer.android.com/training/location/permissions

Permissions

```
private lateinit var fusedLocationClient: FusedLocationProviderClient

override fun onCreate(savedInstanceState: Bundle?) {
    // ...

    fusedLocationClient = LocationServices.getFusedLocationProviderClient(this)
}
```

```
fusedLocationClient.lastLocation
        .addOnSuccessListener { location : Location? ->
            // Got last known location. In some rare situations this can be null.
        }
```

https://developer.android.com/training/location/retrieve-current#last-known

Intents

- Nachrichten Objekt, welche Aktionen von anderen App Components anfragen kann und die Kommunikation zwischen eben diesen erlaubt
- drei fundamentale Aufgaben
    - Activity starten
    - Service starten
    - Broadcast ausliefern

Intents

Kategorien

- **explizit** - eindeutig spezifiziert, welche Applikation den Intent bedienen soll, für gewöhnlich wird eine (App-)Komponente aus der eigenen App gestartet wird
- **implizit** - keine spezifische Komponente definiert, sondern eine Aktion; welche wiederum von Komponenten einer anderen App bedient werden kann (bspw. zeige aktuelle Position auf Karte: Google Maps, Here, etc.)

https://developer.android.com/guide/components/intents-filters#Types

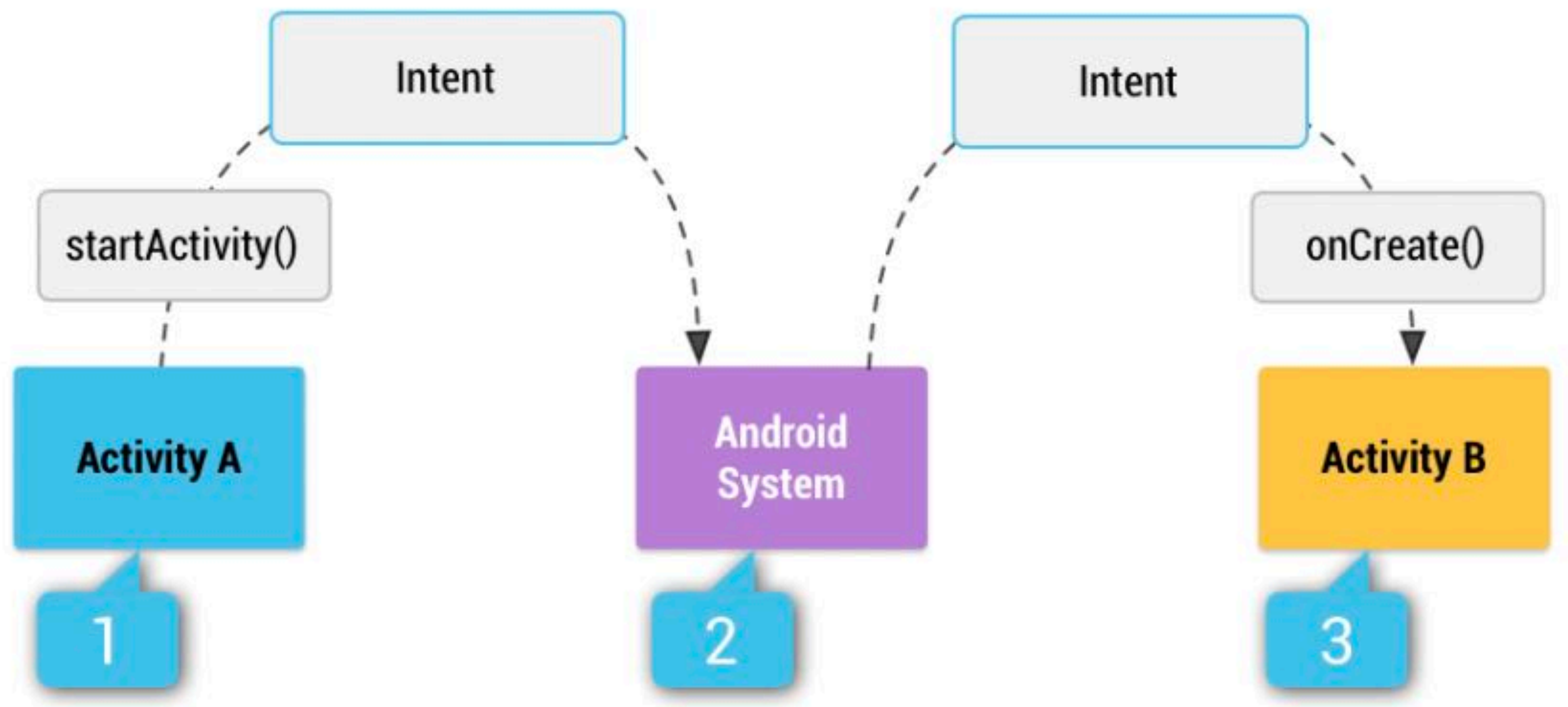


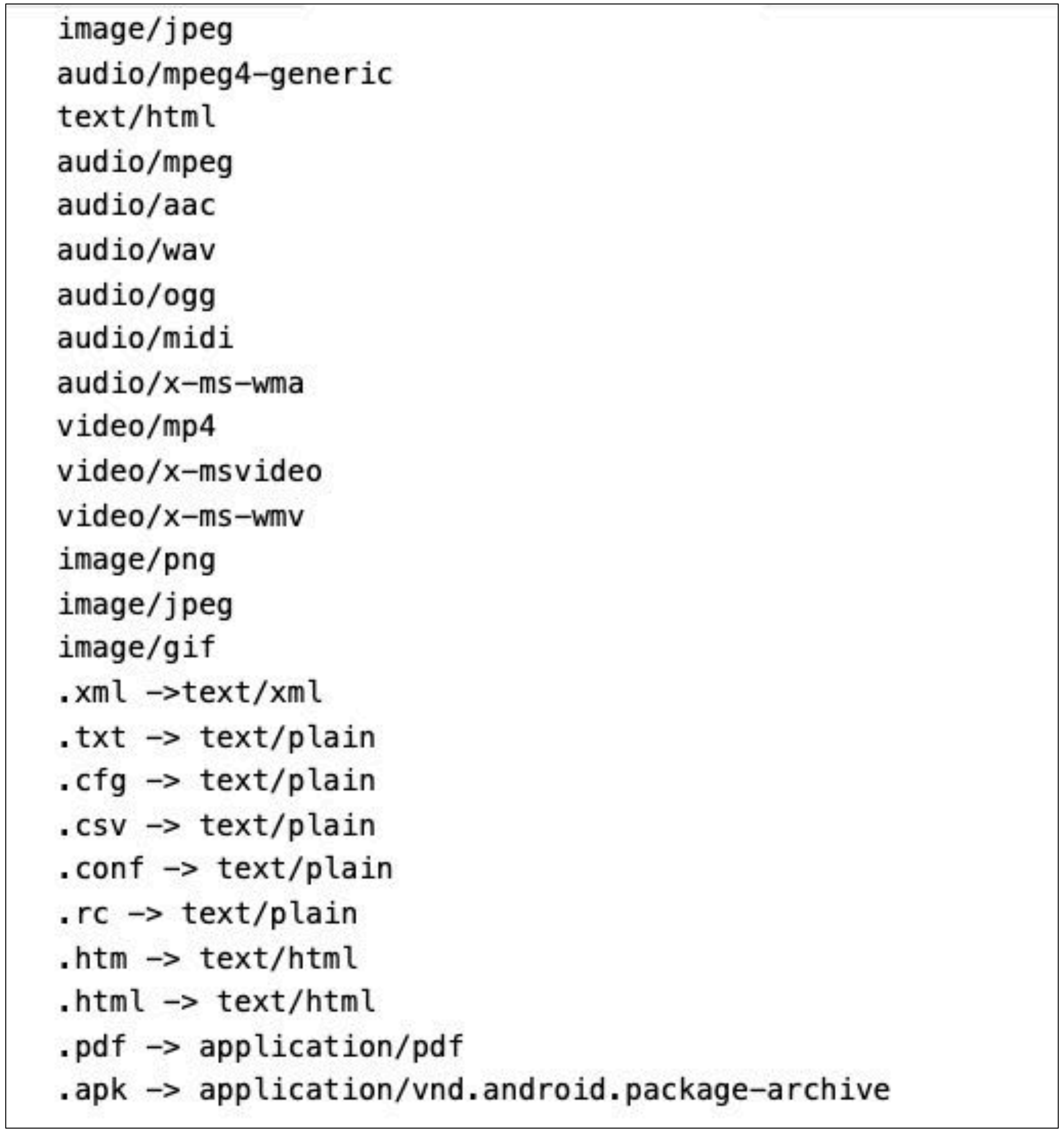
Figure 1. How an implicit intent is delivered through the system to start another activity: **[1]** *Activity A* creates an `Intent` with an action description and passes it to `startActivity()`. **[2]** The Android System searches all apps for an intent filter that matches the intent. When a match is found, **[3]** the system starts the matching activity (*Activity B*) by invoking its `onCreate()` method and passing it the `Intent`.

Intents

Aufbau

Component Name	optional, ohne Namen ist der Intent implizit
Aktion	welche Aktion soll ausgeführt werden
Data	Uri, Datentyp für die Aktion (ACTION_EDIT -> Uri des Dokuments, welches zu bearbeiten ist)
Category	zusätzliche Information, über die Komponente, die den Intent bedienen soll (bspw. BROWSABLE für WebBrowser)
Extras	Key-Value Pairs
Flags	Metadaten (bspw. welche Task gestartete Activity bedienen soll)

Intents

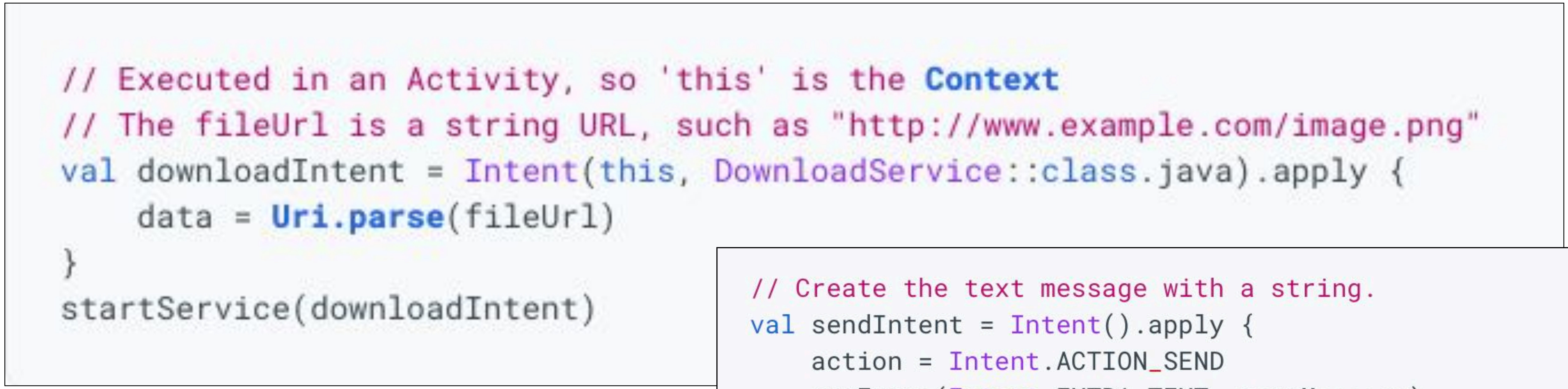


```
image/jpeg
audio/mpeg4-generic
text/html
audio/mpeg
audio/aac
audio/wav
audio/ogg
audio/midi
audio/x-ms-wma
video/mp4
video/x-msvideo
video/x-ms-wmv
image/png
image/jpeg
image/gif
.xml ->text/xml
.txt -> text/plain
.cfg -> text/plain
.csv -> text/plain
.conf -> text/plain
.rc -> text/plain
.htm -> text/html
.html -> text/html
.pdf -> application/pdf
.apk -> application/vnd.android.package-archive
```

https://stackoverflow.com/a/24134677

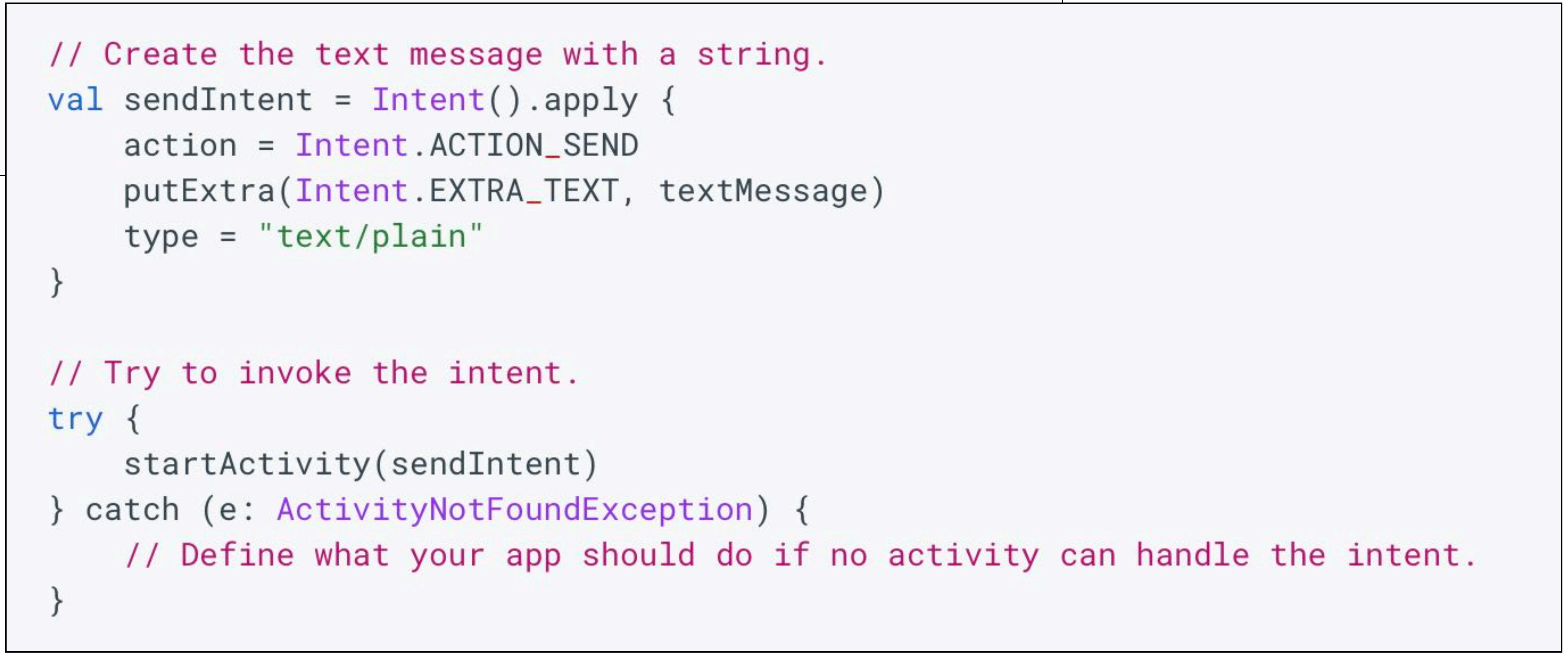
Intents

Beispiele



```kotlin
// Executed in an Activity, so 'this' is the Context
// The fileUrl is a string URL, such as "http://www.example.com/image.png"
val downloadIntent = Intent(this, DownloadService::class.java).apply {
    data = Uri.parse(fileUrl)
}
startService(downloadIntent)
```

https://developer.android.com/guide/components/intents-filters#ExampleExplicit



```kotlin
// Create the text message with a string.
val sendIntent = Intent().apply {
    action = Intent.ACTION_SEND
    putExtra(Intent.EXTRA_TEXT, textMessage)
    type = "text/plain"
}

// Try to invoke the intent.
try {
    startActivity(sendIntent)
} catch (e: ActivityNotFoundException) {
    // Define what your app should do if no activity can handle the intent.
}
```

https://developer.android.com/guide/components/intents-filters#ExampleSend