

ANDROID STARTER

ART+COM, 2016



BASICS

- Source Code and all documents in english
- Agile Software Development
 - Iteratively
 - Inspect & Adapt
- Version control
 - Git or Mercurial
 - Github
- Android IDE: Android Studio

RESOURCES

- <https://developers.google.com>
- <http://stackoverflow.com/>
- <https://www.udacity.com/course/developing-android-apps--ud853>

- <https://github.com/fabue/location-silencer>
- <http://developer.android.com/training/location/index.html>

APP DEFINITION

- We define a region by geo location and radius
- When entering the region the mobile mutes
- When leaving the region the ring tone is set to normal

For educational and limited time reasons we do not handle

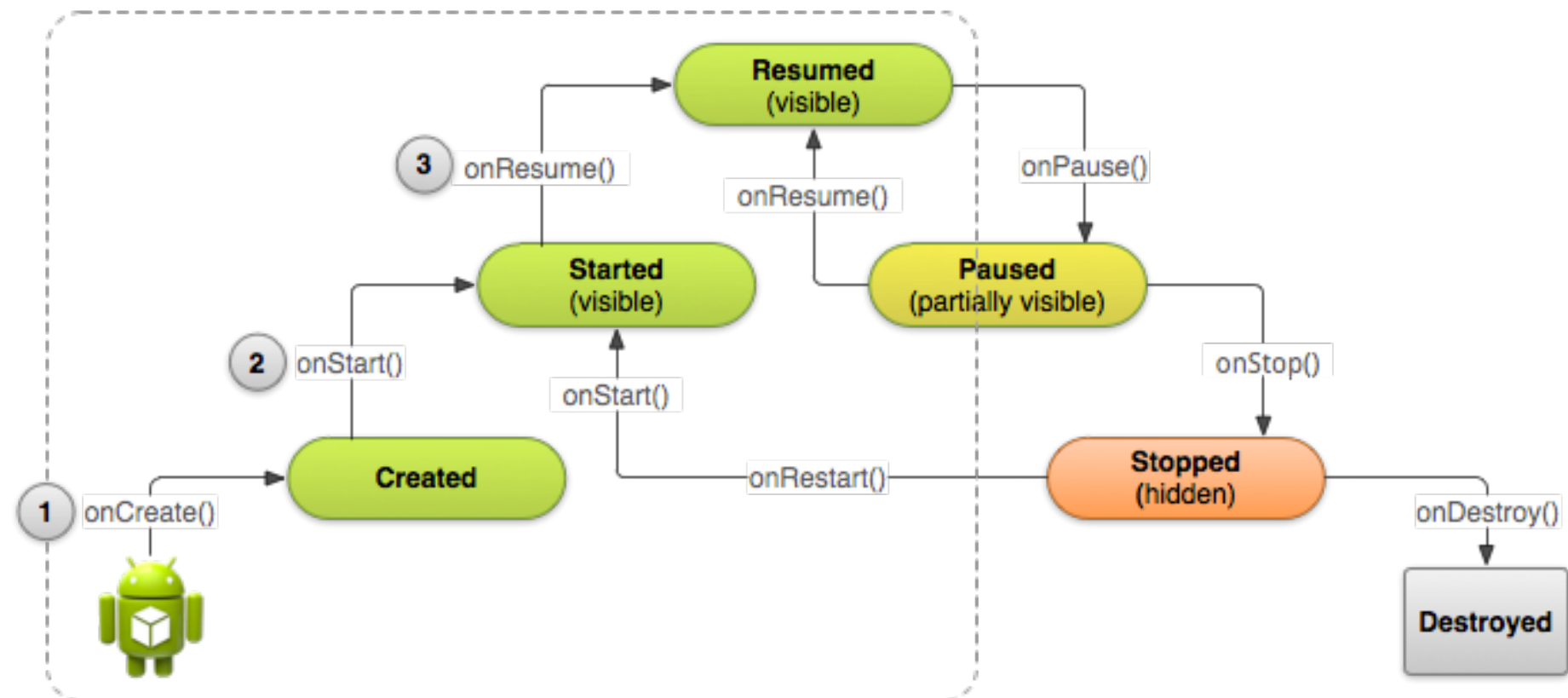
- Permissions
- Safeguard & Edge Case Handling
- Persistence
- Performance
- Orientation Changes
- Backwards Compatibility
- Localization & Internationalization

TASK 1

Setup Project

- Create new Android Studio Project
 - Min SDK 22 & Empty Activity template

- Add logs to lifecycle methods
- Start app on emulator or device
- Observe logs
- Push to GitHub



EXPLANATION

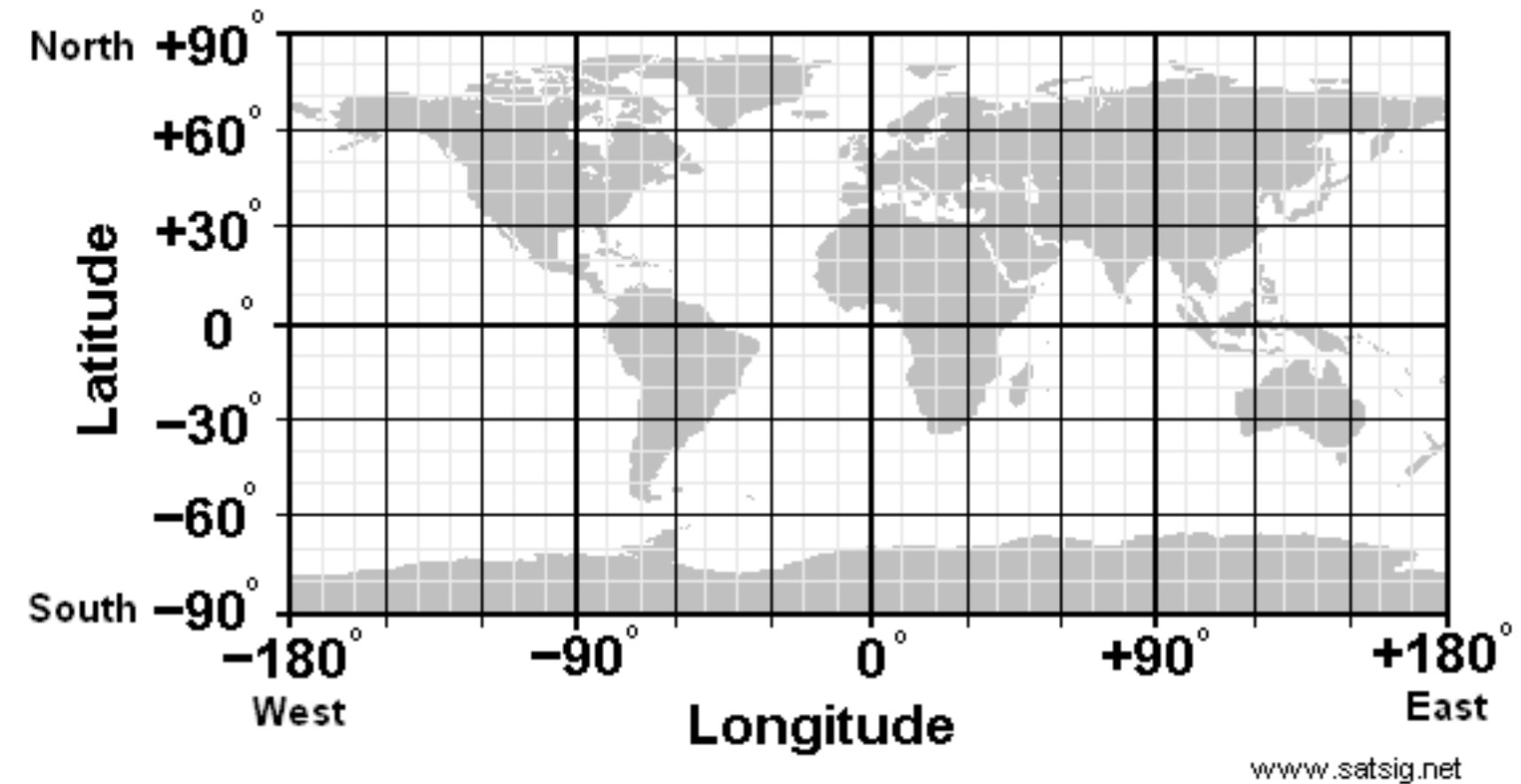
Structure Android Studio

Android Development Files

TASK 2

Simple Button and Labels

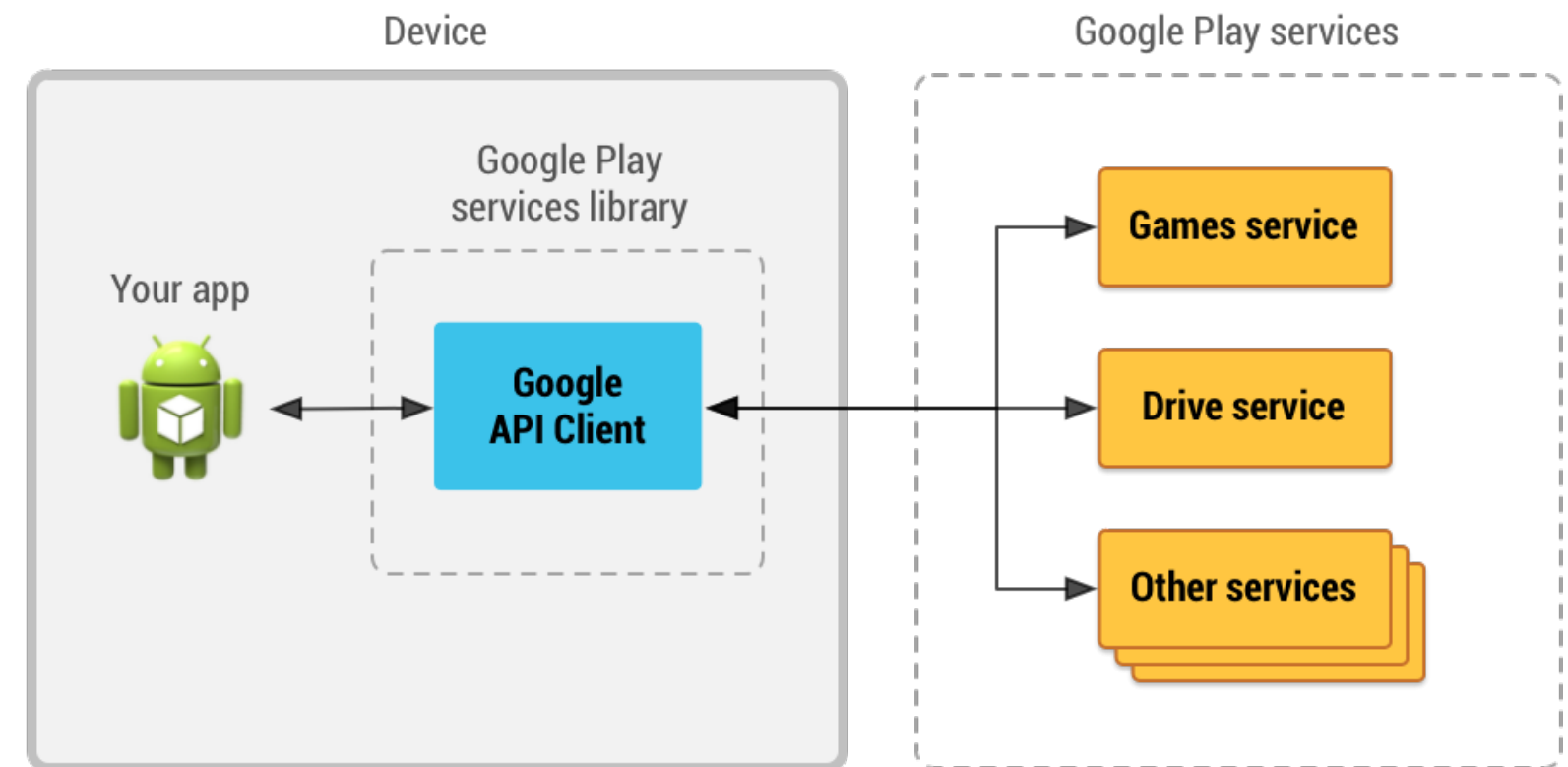
- Add a Button in activity_main.xml
 - Button text: „Get Location Updates“
- Register onClick() handler
- Add logging when button gets clicked
- Add latitude & longitude label TextViews
- Add latitude & longitude value TextViews with default value „not set“
- Set „foo“ and „bar“ as values when the button gets clicked
- Rotate mobile after setting the buttons



TASK 3

Google Play Services Integration for Location API

- Make sure Google Play Services are installed on device
- Download Google Play Services SDK
- Add Google Play Services Location Dependency to build.gradle file
- Initialize Google API Client in on Create()
- Connect client in onStart()
- Disconnect client in onStop()
- Implement required callbacks, add logs



TASK 4

Display Location Updates

- Create a `LocationRequest` when button gets clicked
- Request location updates and implement callbacks
- Add `ACCESS_FINE_LOCATION` permission to Manifest
- Set `targetSdkVersion` in `build.gradle` to 22 to avoid permission topic
- Set latitude and longitude values to received location data in callback

TASK 5

Add a Button to Save Current Location in Member Variable

- Setup a Button and Labels as in Task 2
- On Button click: save current location (with `getLastLocation()`) in member variable
- Set saved latitude & longitude values to last location data

TASK 6

Create an IntentService that Mutes the Mobile

- Create a new class that extends from IntentService
- Register the IntentService in the Manifest
- Add a log to onHandleIntent()
- Start IntentService externally via adb
- Add a Toast to onHandleIntent(), mind the UI Thread
- Test Toast with MainActivity in background, starting it with adb
- Use AudioManager to mute mobile when service starts

TASK 7

Prepare for Geofence: Add a ToggleButton

- Add a ToggleButton in activity_main.xml
 - Button textOn: „Geofence active“
 - Button textOff: „Geofence inactive“
- Get a reference of the ToggleButton in onCreate() with findViewById() to register a setOnCheckedChangeListener() with respective logs
- Disable ToggleButton by default in activity_main.xml
- Enable ToggleButton in MainActivity when location is saved



TASK 8

Use Geofence API in MainActivity to Trigger IntentService

- Create a method that returns a PendingIntent for the IntentService
 - Create a method that returns a Geofence with the Geofence.Builder for enter & exit with our saved location
 - Create a method that returns a GeofencingRequest
 - Create a method activateGeofence() which uses the GeofencingApi.addGeofences() to register the Geofence
 - Create a respective function deactivateGeofence() which uses the GeofencingApi.removeGeofences() to unregister Geofences
-
- Connect with ToggleButton



TASK 9

IntentService Reacts on Geofence Transitions

- Use `GeofencingEvent.fromIntent()` to create `GeofencingEvent` in `onHandleIntent()`
- Get first (and only) `Geofence` from the Event
- Create a String description (Transition & ID) from the `Geofence`
- Put that description into the Toast
- Set `RingerMode` corresponding to `Geofence` transition

Use App Lockito to fake GPS position



Impressum

Der Inhalt dieser Präsentation ist durch das Urheberrecht geschützt.
Alle Rechte an der Präsentation und seinem Inhalt stehen der ART+COM AG zu.

Der Inhalt dieser Präsentation darf ohne vorherige Zustimmung weder ganz noch teilweise kopiert, veröffentlicht, verändert, übersetzt, veräußert oder genutzt werden. Mit der Übergabe, Einsendung oder Übersendung dieser Präsentation ist ausdrücklich keine Einräumung von Nutzungsrechten verbunden.
Die Informationen, Ideen und Inhalte dieser Präsentation sind vertraulich und dürfen ohne Zustimmung nicht weitergegeben werden.

Wir weisen ausdrücklich darauf hin, dass Teile des verwendeten Bildmaterials zu Layoutzwecken dienen und urheberrechtlich nicht abgesichert sind.

© ART+COM AG, 2016



ART+COM AG
Kleiststraße 23–26, 10787 Berlin
Fon: +49 30 21001-0
info@artcom.de, www.artcom.de