



# GreenWatch

Termin 1

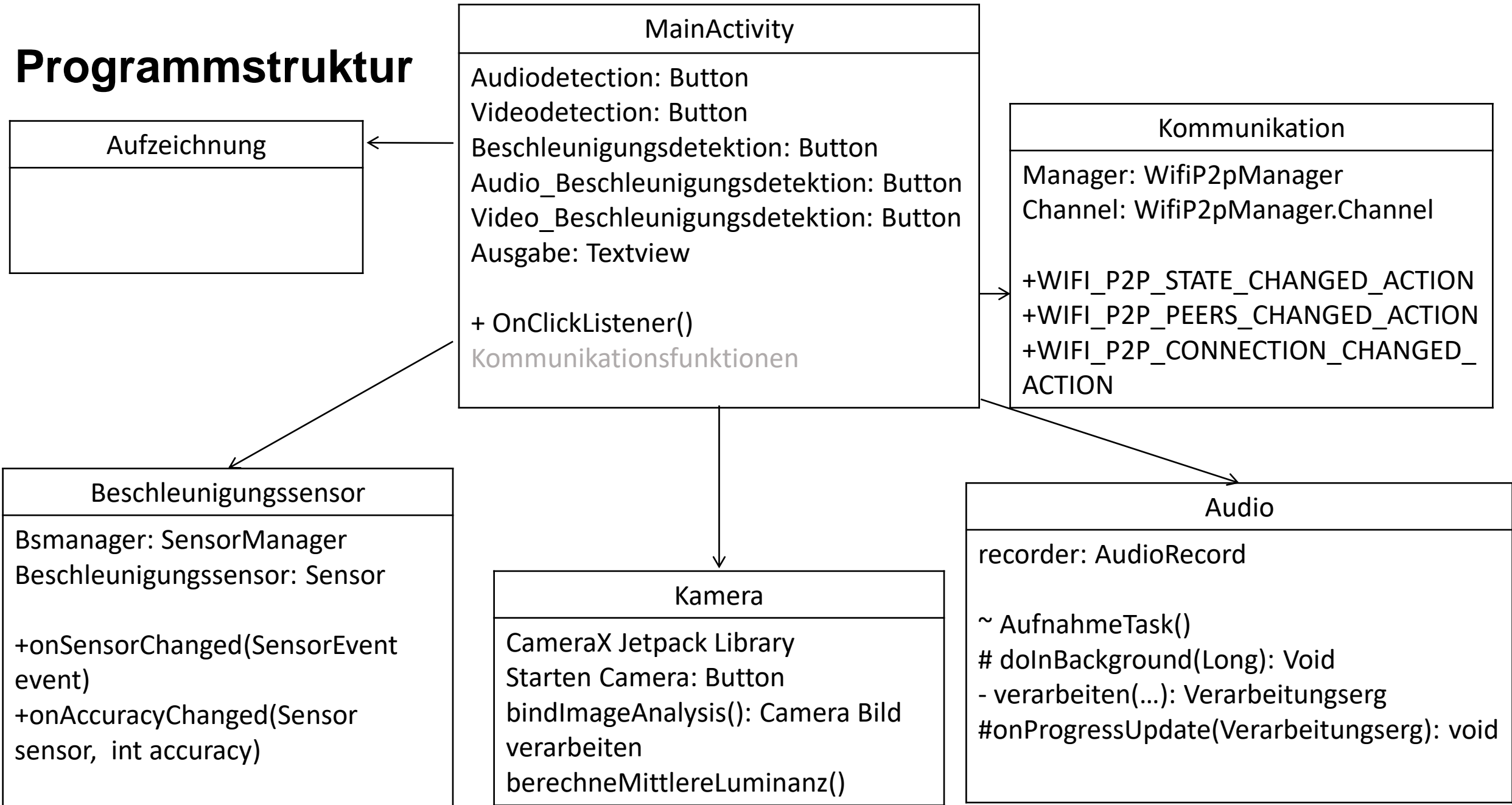
# Einarbeitung in die Aufgabenstellung

- Android Version 6.0
- GitHub einrichten und Grundlegendes Projekt erstellen
- Aufgabendefinitionen und Arbeitspaketeinteilung:
  1. Mikrofon
  2. Kamera
  3. Beschleunigungssensor
  4. Benutzeroberfläche/Datenverwaltung
  5. Kommunikation

Außerdem:

- Aufzeichnung
- Rechteverwaltung

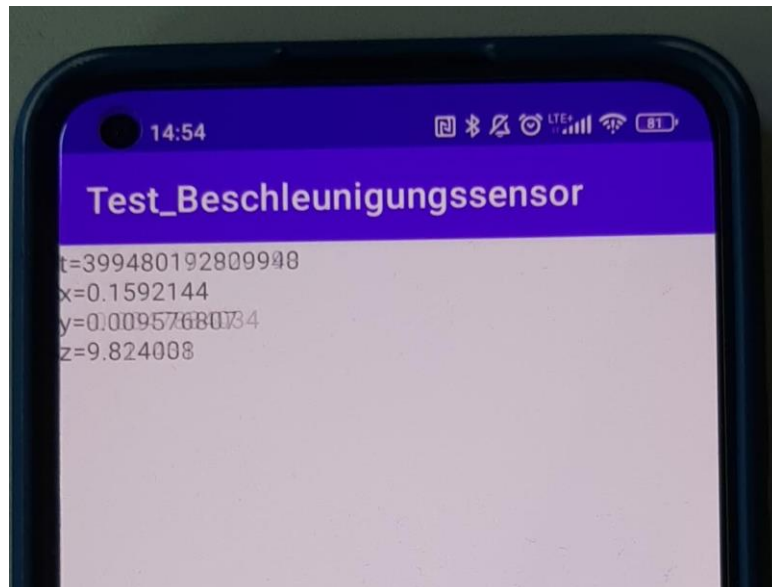
# Programmstruktur



# Beschleunigungssensor

## Aktueller Stand

- Einarbeitung in die Sensorfunktionen
- Programmierung einer Auslesefunktion der Sensordaten mit Warnung bei veränderten Positionsdaten in X-Richtung



## Aufgaben bis zum nächsten Termin

- Problem: Einbettung des Programmcodes in eigene Aktivität
- Warnung als Rückgabewert in einen String (Zeitstempel, Amplitude, etc.)
- Genauigkeit/Accuracy einstellen (bis jetzt gerundet)
- Warnung auf andere Richtungen vervielfältigen

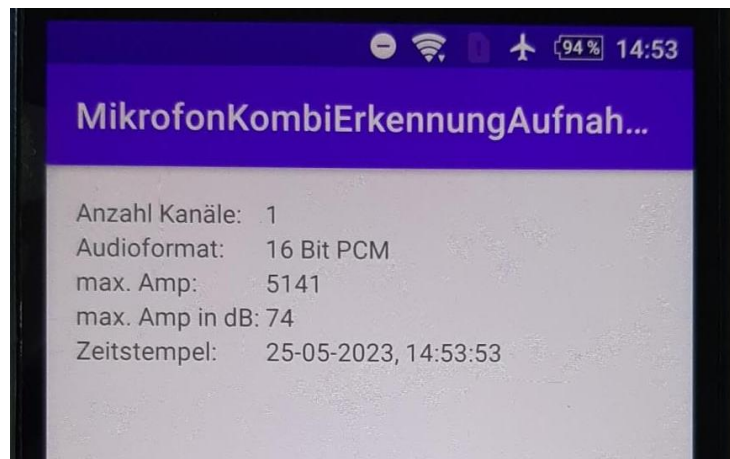
# Mikrofon

## Aktueller Stand

- Einarbeitung in die Audiofunktionen (AudioRecord) und dB-Berechnung
- Programmierung: Auslesefunktion des aktuellen Geräuschpegels in dB und eines Zeitstempels (für Meldungen)  
→ bei Geräuschdetektion Setzen einer booleschen Variable

## Aufgaben bis zum nächsten Termin

- Einbetten in eigene Activity
- Adaptive Schwelle integrieren
- Problem: Aktualisieren des Zeitstempels bei Geräuschdetektion während boolesche Variable true  
→ Flankenerkennung, sodass nur einmaliges Aktualisieren des Zeitstempels



# Kamera

## Aktueller Stand

- Befassen mit dem Codebeispiel aus der Vorlesung DemoCamera2
- Einarbeitung in die Kamerafunktionen von CameraX
- Implementierung der CameraX Funktionen
- Berechtigungen der Kamera in gradle.build und Manifest eingetragen
- Anzeige einer Preview und der Luminanz auf dem Interface zu Testzwecken

## Aufgaben bis zum nächsten Termin

- Probleme beim Starten der Kamera beheben
- Grenzwert für mittlere Luminanz festlegen

# Benutzeroberfläche/Datenverwaltung

## Aktueller Stand

- Anlegen der Activities für die verschiedenen Sensoroptionen
- Überlegung eines Konzeptes für die Konsistenz der Sensordaten
- Erste Implementierung eines Model View Controllers

## Aufgaben bis zum nächsten Termin

- Fertigstellung des Model View Controllers und testen der Funktionalität
- Hinzufügen eines Menüs zur Anzeige der registrierten Sensoren

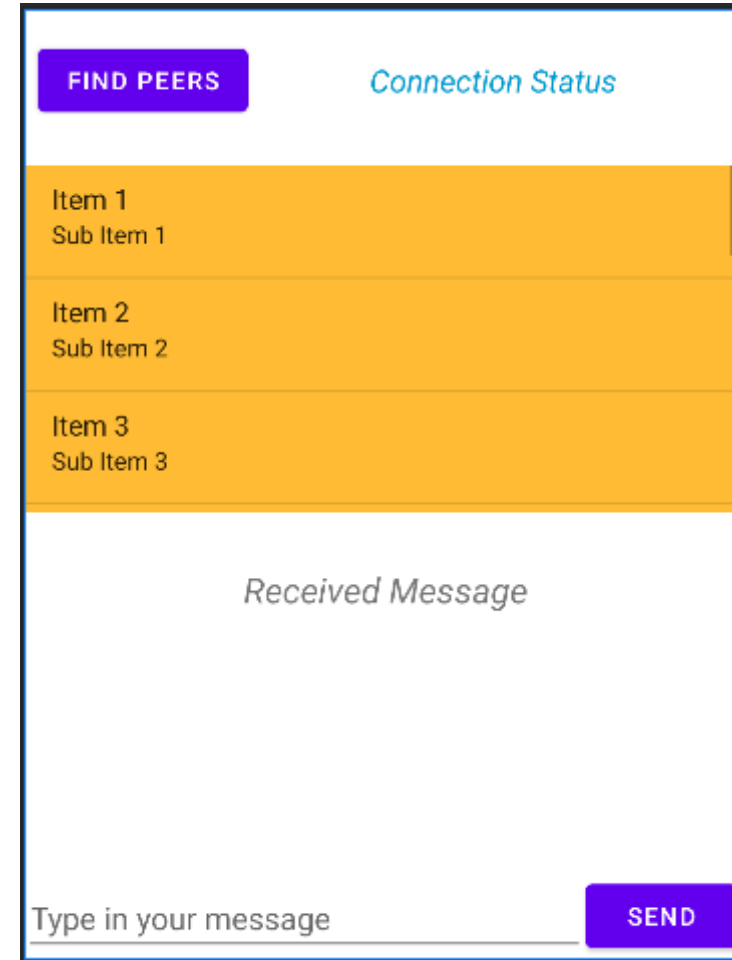
# Kommunikation

## Aktueller Stand

- Findet verfügbare Teilnehmer im Netz
- Zeigt verfügbare Teilnehmer in ListView an
- Stellt Verbindung zwischen 2 Geräten her
- Kann Nachrichten übermitteln

## Aufgaben bis zum nächsten Termin

- Einbetten in eine eigene Klasse
- groupConnection aufbauen
- Mehr als 2 Geräte kommunizieren
- Problem mit Host und Client Zuweisung beheben





# Demo Kommunikation

