

# **Cloud-based Real Time ADL Recognition Using Wearable Devices**

Robert Baldauf

- Grundlagen “ADL Recognition”
- Offline ADL Recognition
- Cloud-based ADL Recognition
- Ergebnisse
- Diskussion & Ausblick

● ○ ○ ○ ○ Grundlagen “ADL Recognition”

- ➔ **ADL** = Activities of Daily Living (alltägliche Aktivitäten, die eigenständig durchgeführt werden können)
- ➔ **ADL Recognition** = Automatisierte Erkennung von ADLs

Mobilität	Datenerfassung	Datenanalyse
stationär	RGB-Video, Infrarot	Computer Vision, Machine Learning
<b>mobil</b>	<b>Inertial Sensoren (Smartphones, Wearables)</b>	<b>Signal Processing, Machine Learning</b>
stationär / mobil	WIFI Radio Signal	Signal Processing, Machine Learning

## **Fitness-/Health-tracking**

- ➔ Überwachung der eigenen körperlichen Aktivität (Schritte, Laufdistanz, etc.)

## **ADL Erkennung im Pflegebereich**

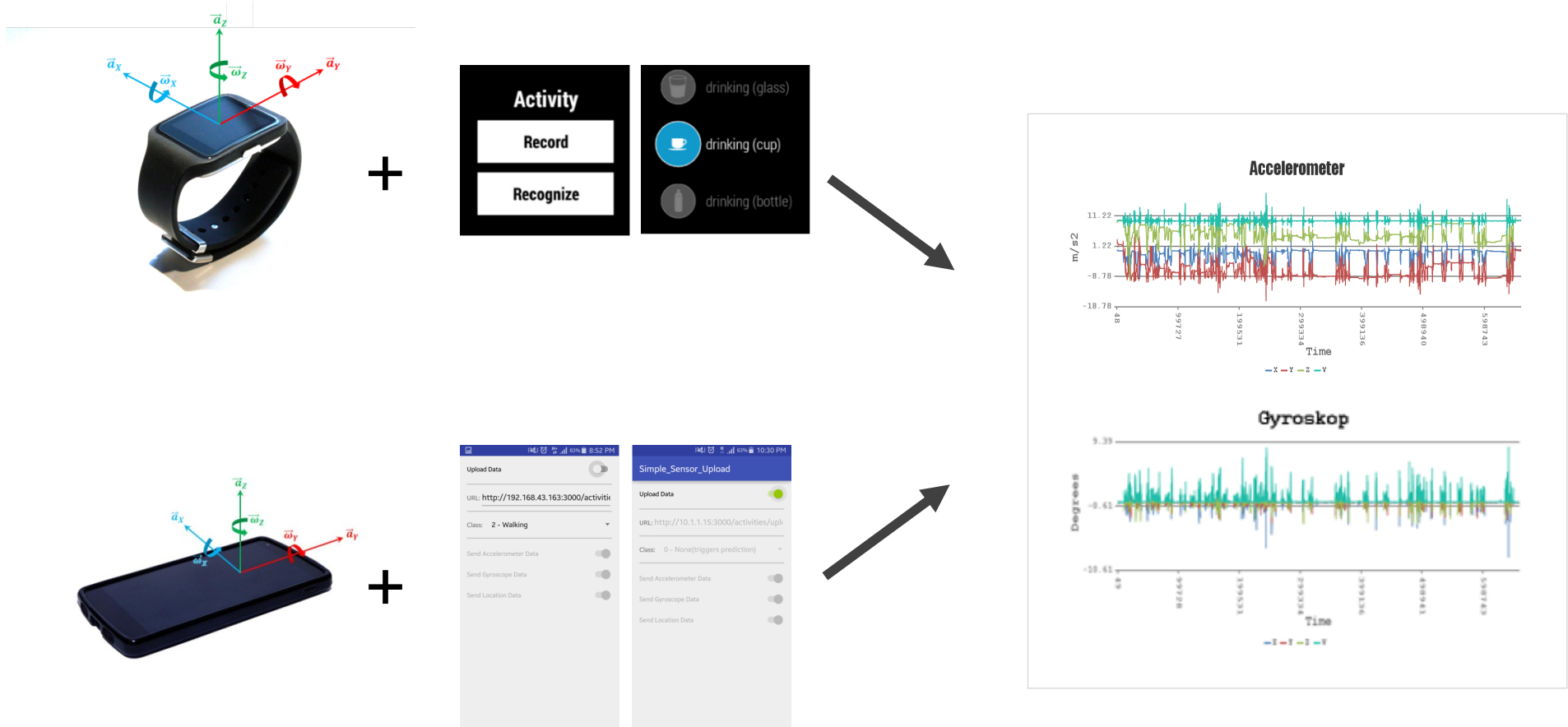
- ➔ Erstellung von “Aktivitätsprofilen”
- ➔ Beurteilung der Fähigkeit zur eigenständigen Ausführung von ADLs
- ➔ Bei kurzzeitiger Verschlechterung: *Erinnerungsfunktion*
- ➔ Bei langfristiger Verschlechterung: *Verständigung von Angehörigen, Medizinischen- / Pflegeeinrichtungen*

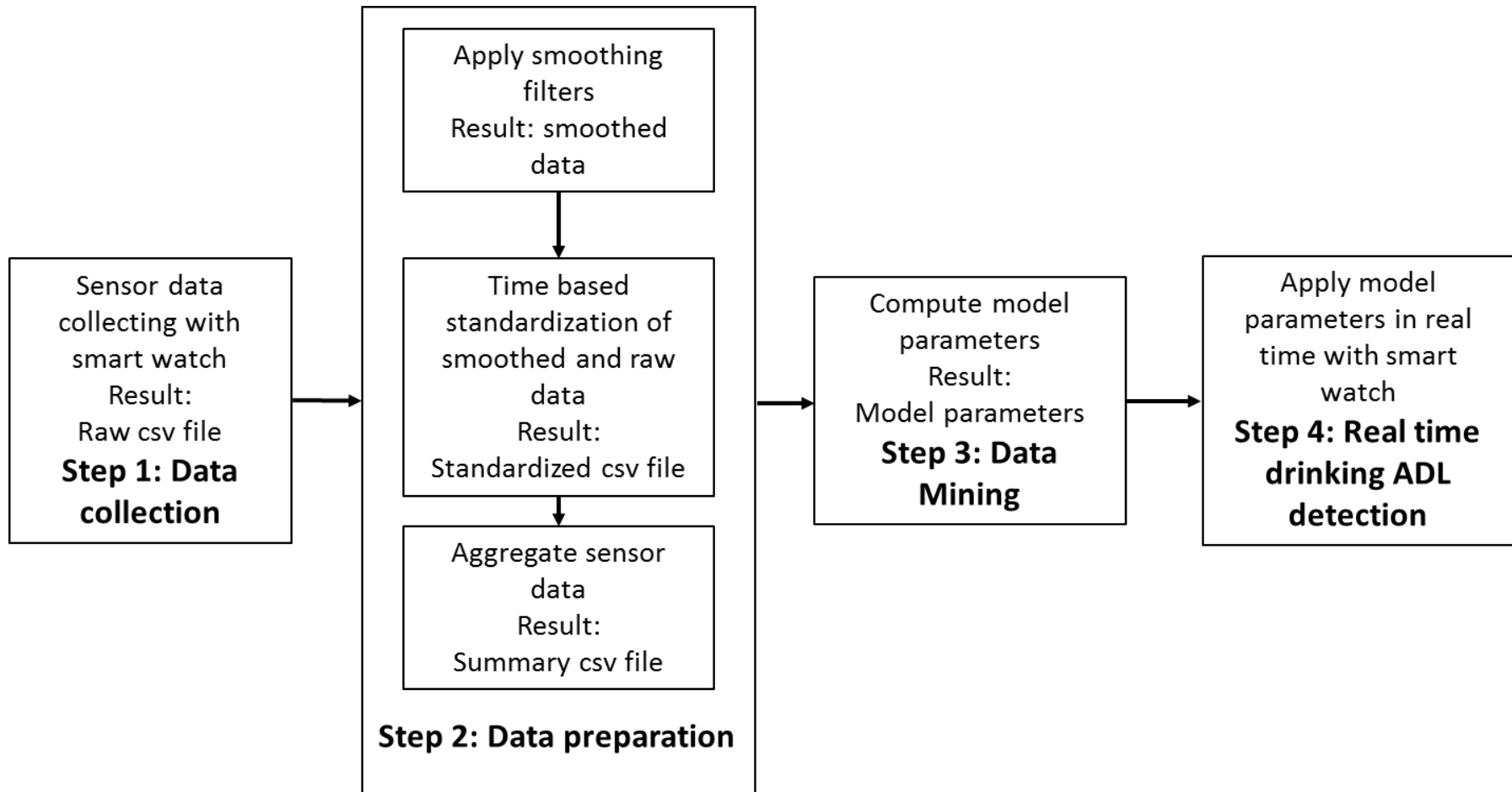
## **ADL Erkennung in Gefahrensituationen**

- ➔ Sturzerkennung
- ➔ Identifikation epileptischer Anfälle

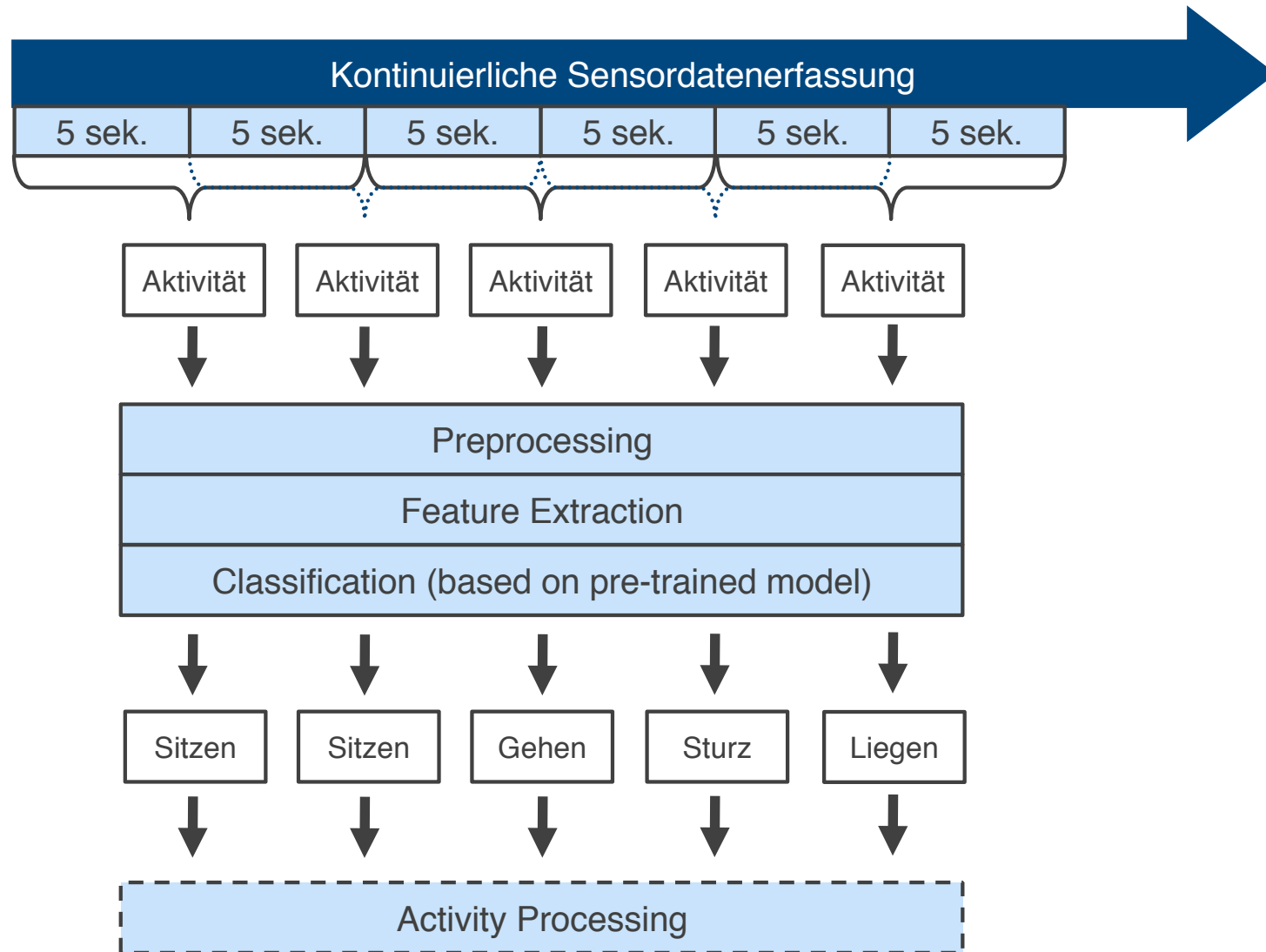
○ ● ○ ○ ○ Offline ADL Recognition

○ ● ○ ○ ○ Offline ADL Recognition – Datenerfassung









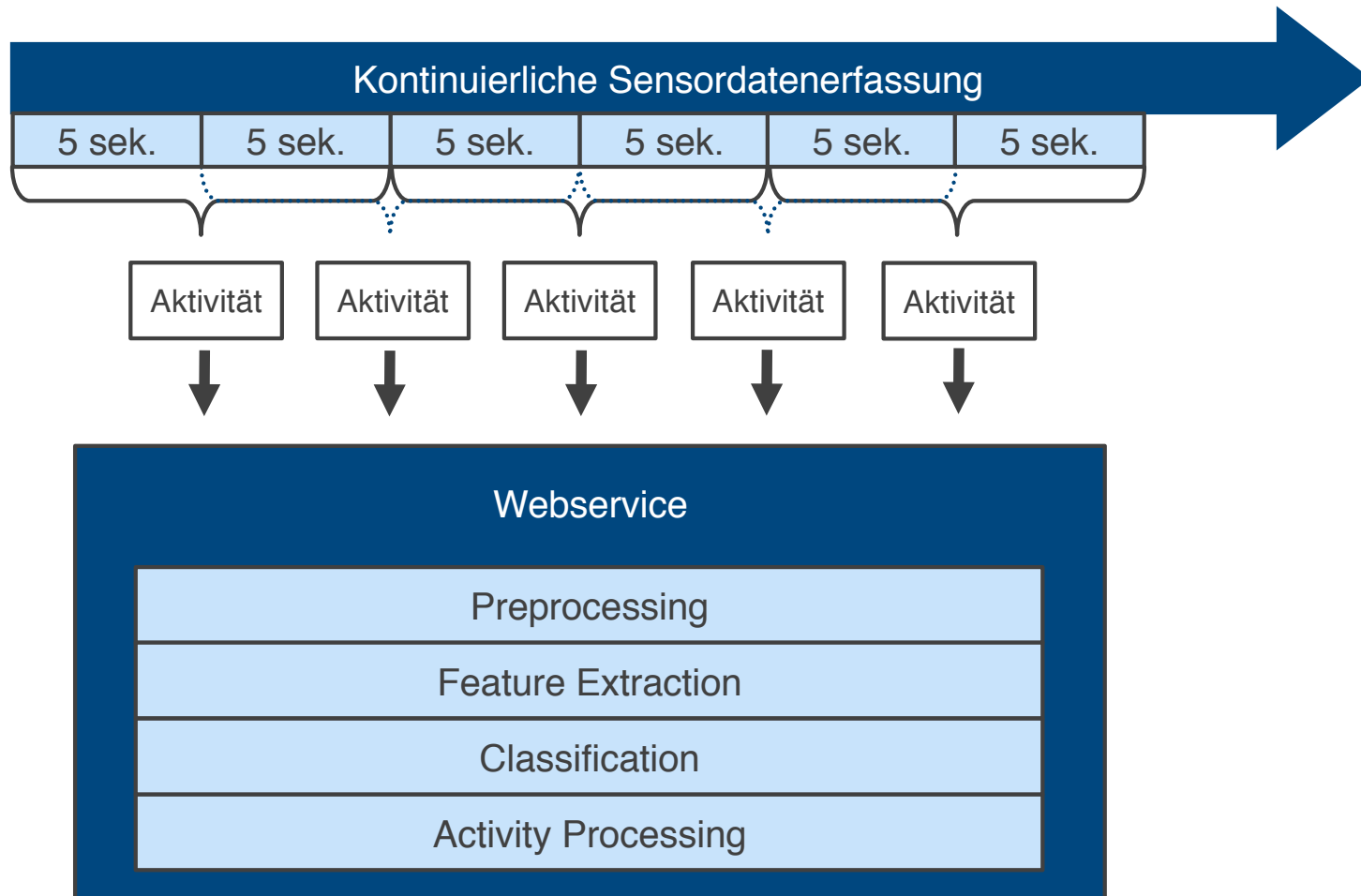


### **Grenzen mobiler Hardwareplattformen bei der ADL Erkennung**

- ➔ Permanente Echtzeiterkennung erzeugt hohe CPU Last
- ➔ Beeinflusst andere Prozesse
- ➔ Reduziert Akkulaufzeit
- ➔ Langfristige Aktivitätshistorie nicht lokal speicherbar
- ➔ Keine Auswertung langfristiger Trends / Veränderungen
- ➔ Erfasste Aktivitäten werden nicht gespeichert und tragen nicht zur Optimierung des Klassifikationsmodells bei

Lösungsansatz: **Verlagerung ressourcenintensiver Prozesse in die Cloud**

○ ○ ● ○ ○ **Cloud-based ADL Recognition**

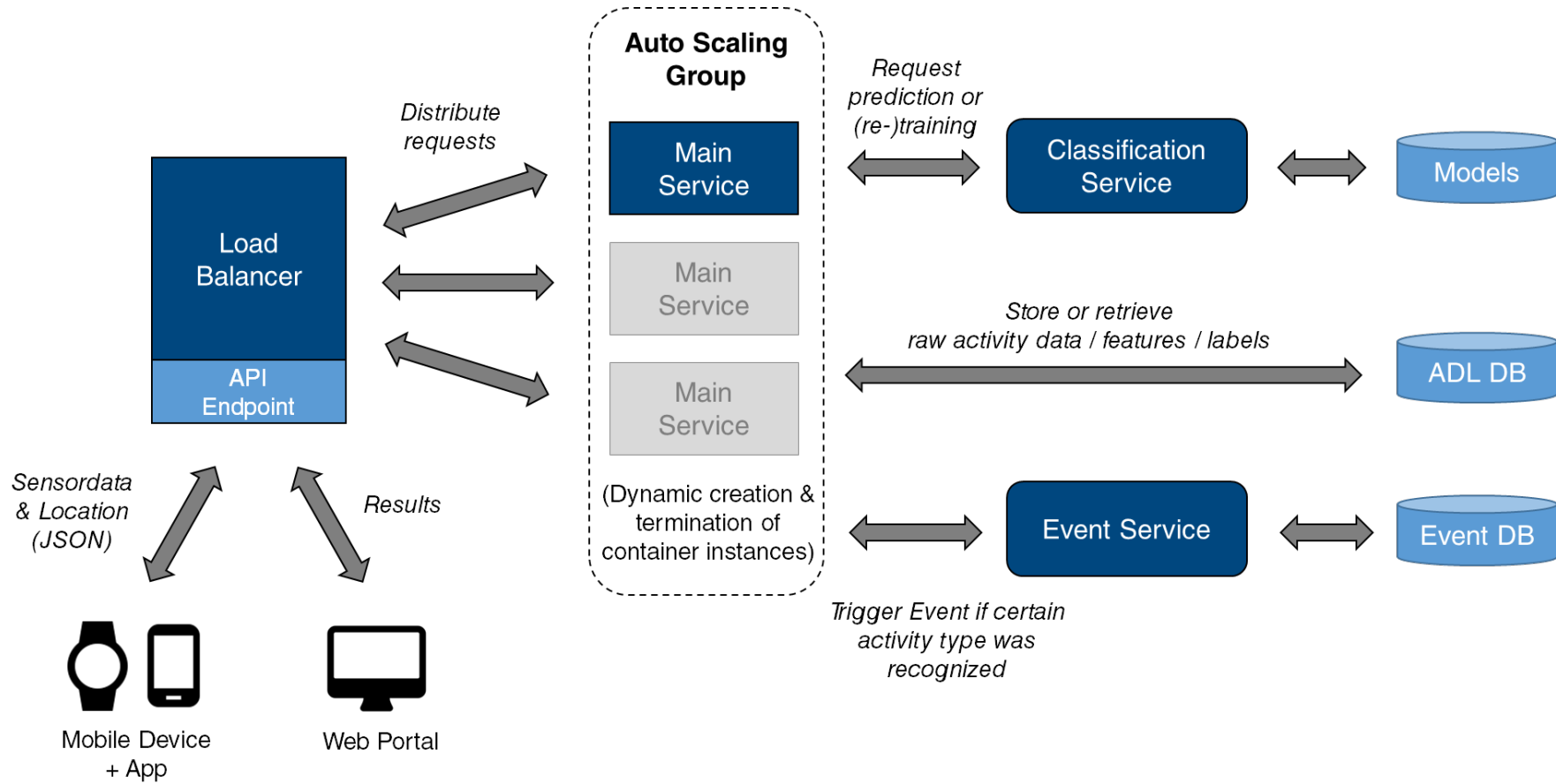


**Ziel:** Entwurf und Prototyping einer Cloud-basierten Architektur für Übertragung, Training und Erkennung von ADLs

**Anforderungen:**

- ➔ Verfügbarkeit
- ➔ Skalierbarkeit (dynamische Allokation zusätzlicher Ressourcen bei Bedarf)
- ➔ geringe Kosten / Risiko
- ➔ Datenschutz/-sicherheit
- ➔ Erweiterbarkeit (z.B.: neue Aktivitäten, Filter, Algorithmen)
- ➔ Portabilität

**Vorgehen:** Exemplarische Validierung der Architektur anhand eines einfachen ADL Erkennungsproblems (Sitzen / Laufen / Sonstige)



Hosting: 

Deployment: 

Komponente	Plattform	Infrastruktur
Main Service	node.js / express.js	AWS EC2 Container Service
Classification Service	flask / sklearn	
Event Service	node.js / express.js	
Activitiy DB / Event DB	MySQL	AWS RDS
Model Storage	-	AWS S3

○ ○ ○ ● ○ Ergebnisse



## Training des Klassifikationsmodells

- ➔ Aufzeichnung der Trainingsdaten von sieben Testpersonen mit fünf verschiedenen Android Geräten
- ➔ 439 Aktivitäten der Klassen „Sitzen“, „Laufen“, „Sonstige“
- ➔ Training von SVM-Klassifikatoren mit versch. Parametern und Cross. Valid. (Beste Ergebnisse:  $C=300000$ ,  $\gamma=0,001$ ,  $\text{kernel}='rbf'$ )

## Validierung der Echtzeiterkennung

- ➔ 85 Aktivitäten von fünf Testpersonen mit fünf realen & zehn virtuellen Android Geräten
- ➔ Cross Validation Accuracy 96,3% (über alle Klassen)
- ➔ Anzahl der EC2-Instanzen skaliert mit Anzahl der verbundenen Endgeräte



## Ergebnisse – Portal zur Auswertung

ADL Monitor Search About Contact

---

Search activities

**Activity location**

**Device tag (optional)**

**Activities (click to enable/disable)**

Start search

**Activity Type:** walking  
**Device:** 00000000-384f-d738-0000-000026612036  
**Timestamp:** 2016-11-02T00:50:06.963Z  
**Latitude:** -27.480268  
**Longitude:** 153.024236

○ ○ ○ ○ ● **Diskussion und Ausblick**



**Die vorgestellte Architektur für Cloud-basierte ADL Erkennung...**

**...wurde erfolgreich implementiert.**

**...erfüllt die vorab definierten Anforderungen**

**...wurde anhand eines einfachen ADL-Erkennungsproblems validiert.**

**Problem:** - Verbrauchtes Datenvolumen bei Übertragung zu hoch

**Lösungsansätze:** - Sensor-Abstraten minimieren  
- JSON Format durch binäre Serialisierung (Google Protobuf) ersetzen  
- Zweistufiges Klassifikationsverfahren einführen

**Nächste Schritte:** - Implementieren einer Cloud-basierten Sturzerkennung  
- Aktivitätsprofile + Anomalieerkennung

● ● ● ● ● Vielen Dank für Ihre Aufmerksamkeit!