TECHNISCHE UNIVERSITÄT DRESDEN

Application Development for Mobile and Ubiquitous Computing

Seminar Task
Adaption Concept Presentation

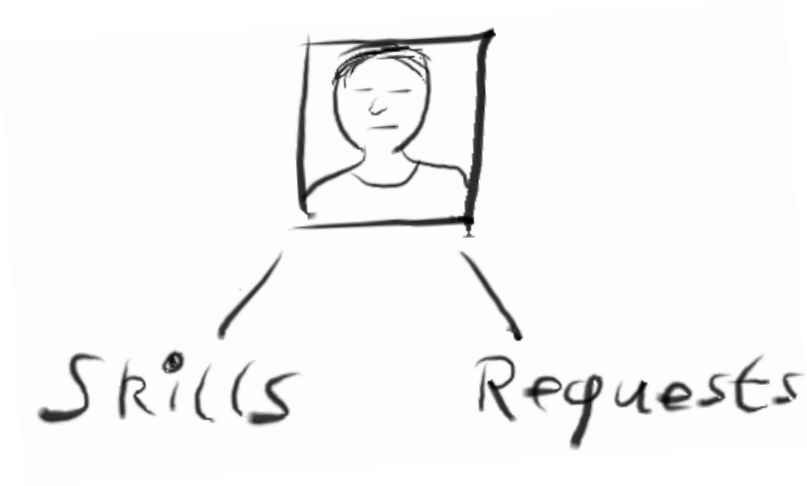
GroupNo. 7

Michael Numair
William Engelmann

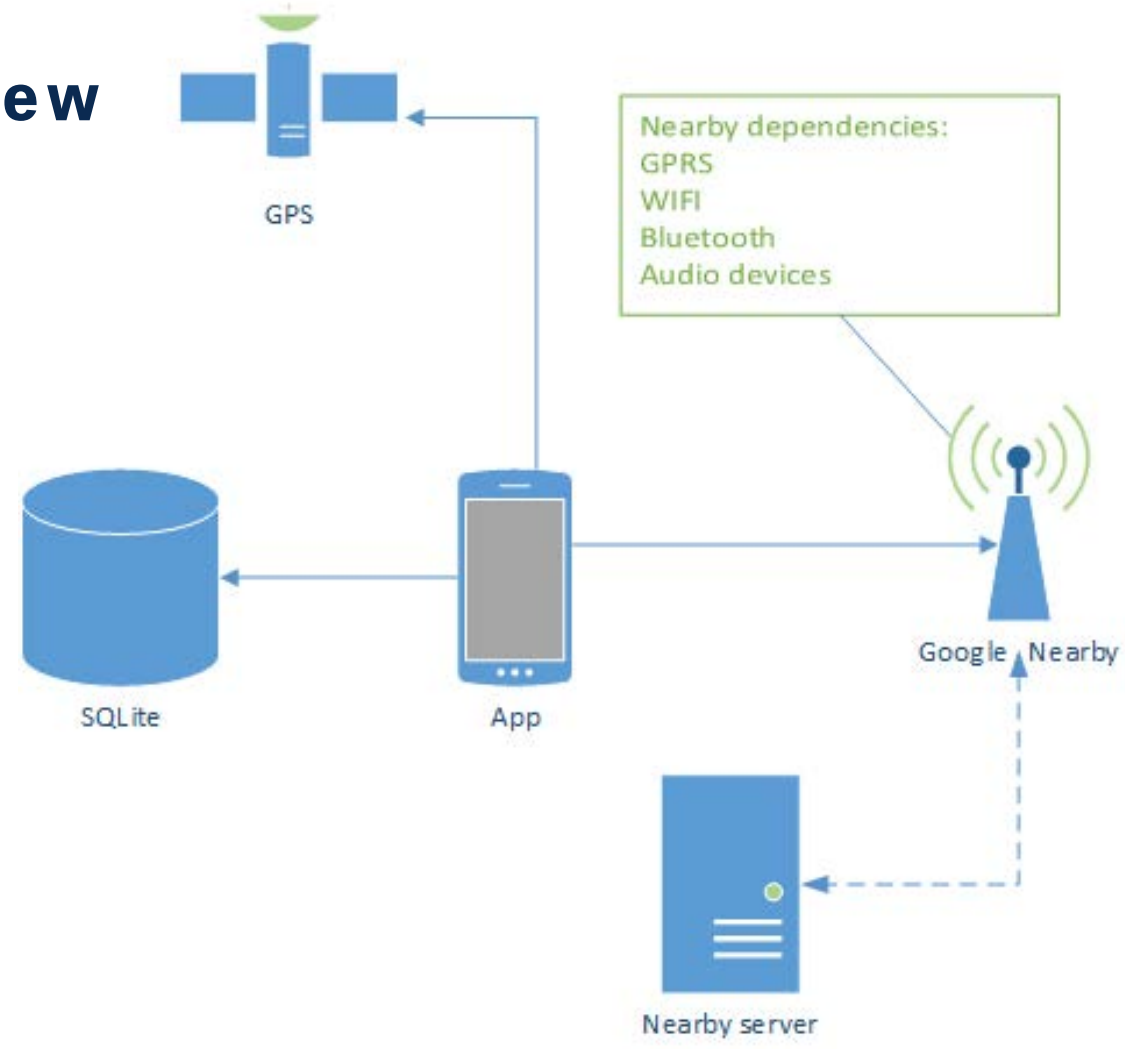
Application Scenario

Application Scenario

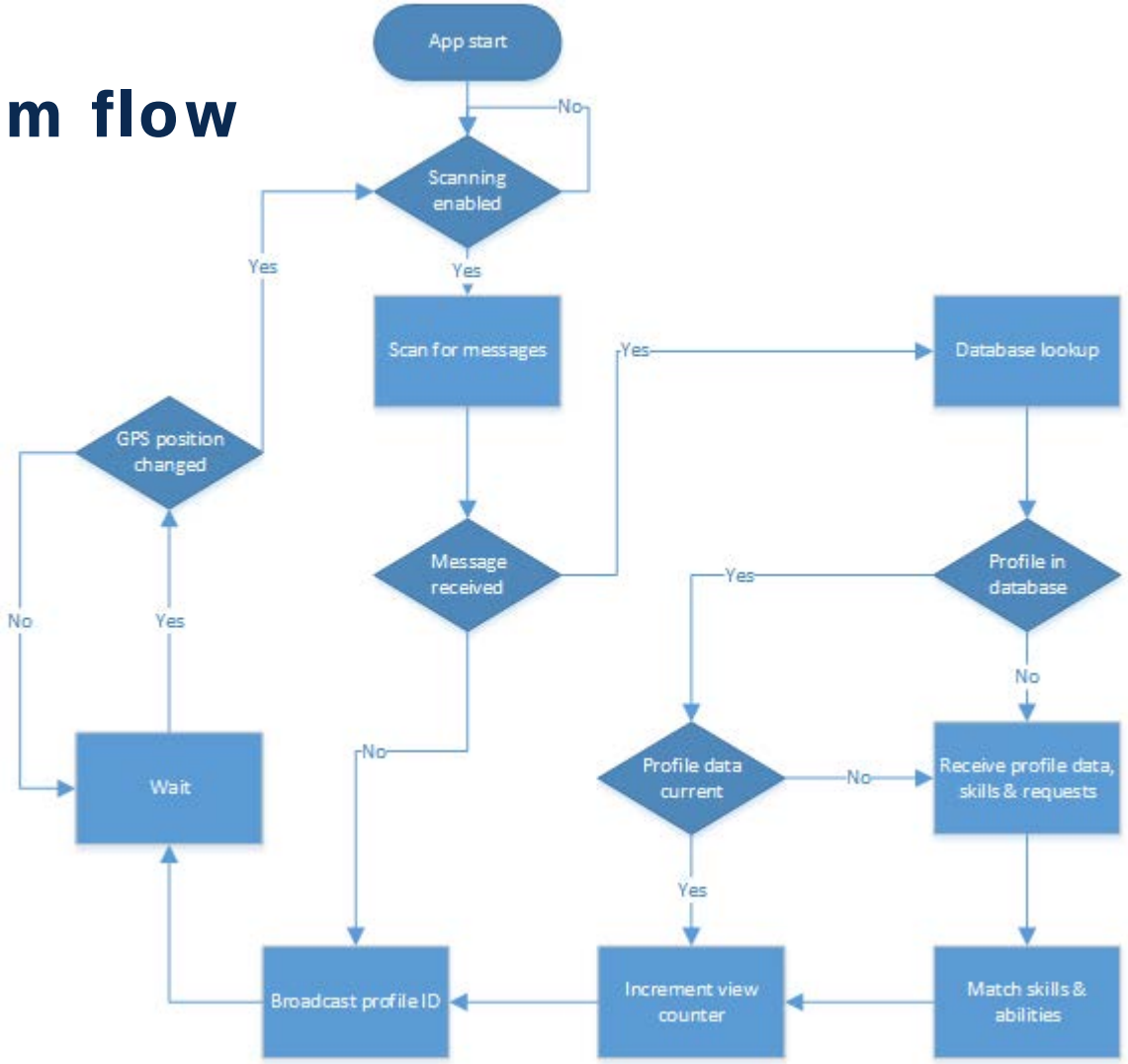
- Help needed for a task
- ‚Employ' people you may already ‚know'
- Scan for users around your position
- Exchange user profiles via Google Nearby Server
- Match peers' skills with user requests



Overview

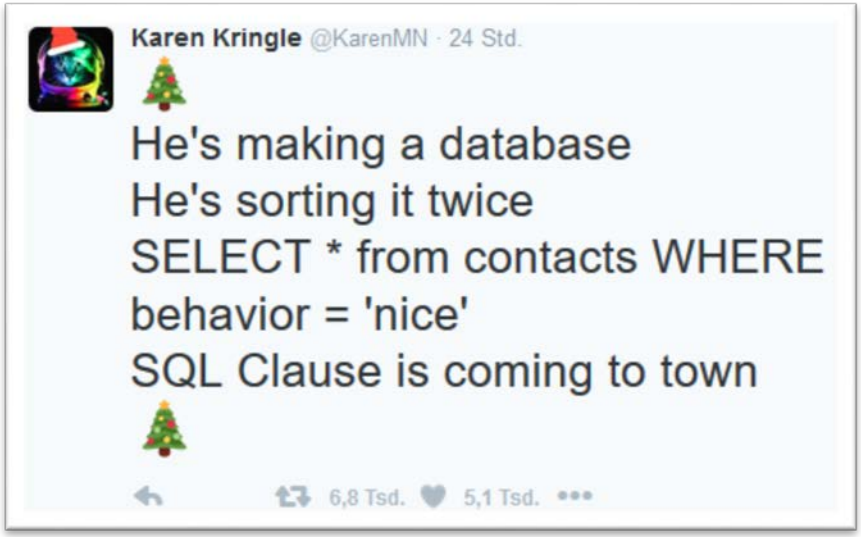

GPS
Nearby dependencies:
GPRS
WIFI
Bluetooth
Audio devices
SQLite
App
Google Nearby
Nearby server

Program flow



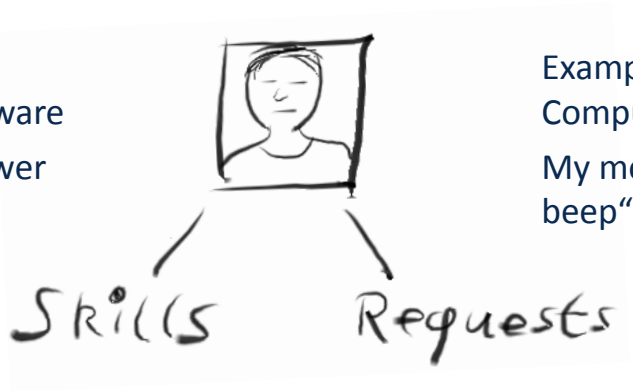
Database Design

- SQLite (built-in)
- Sugar ORM library
  - Simple DB creation
  - Save Java Objects



Example skill:
Computer & Electronics - Hardware

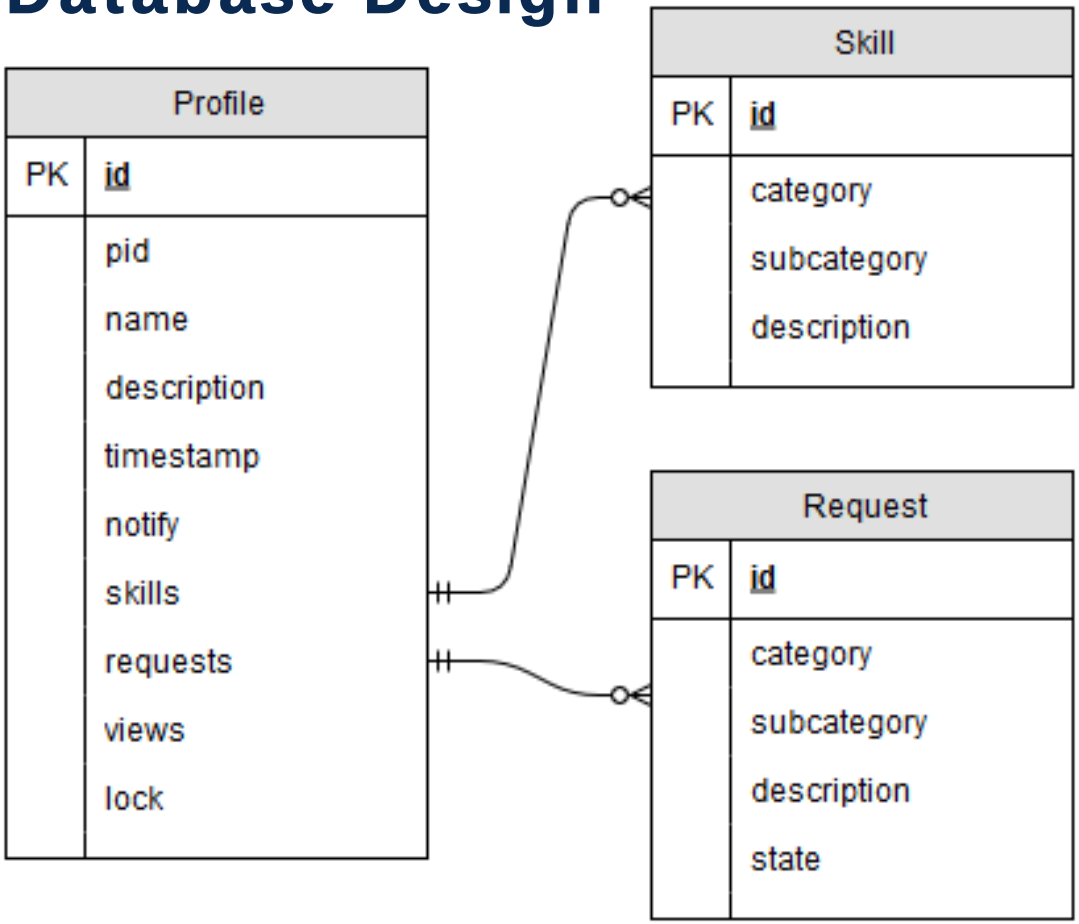
Litrally evrything that haz a power supply. Much knowledge. Very skill. Wow.



Example request:
Computer & Electronics - Hardware

My mom‘s printer goes „beep, beep“ all the time. Plz help!!!111!

Database Design



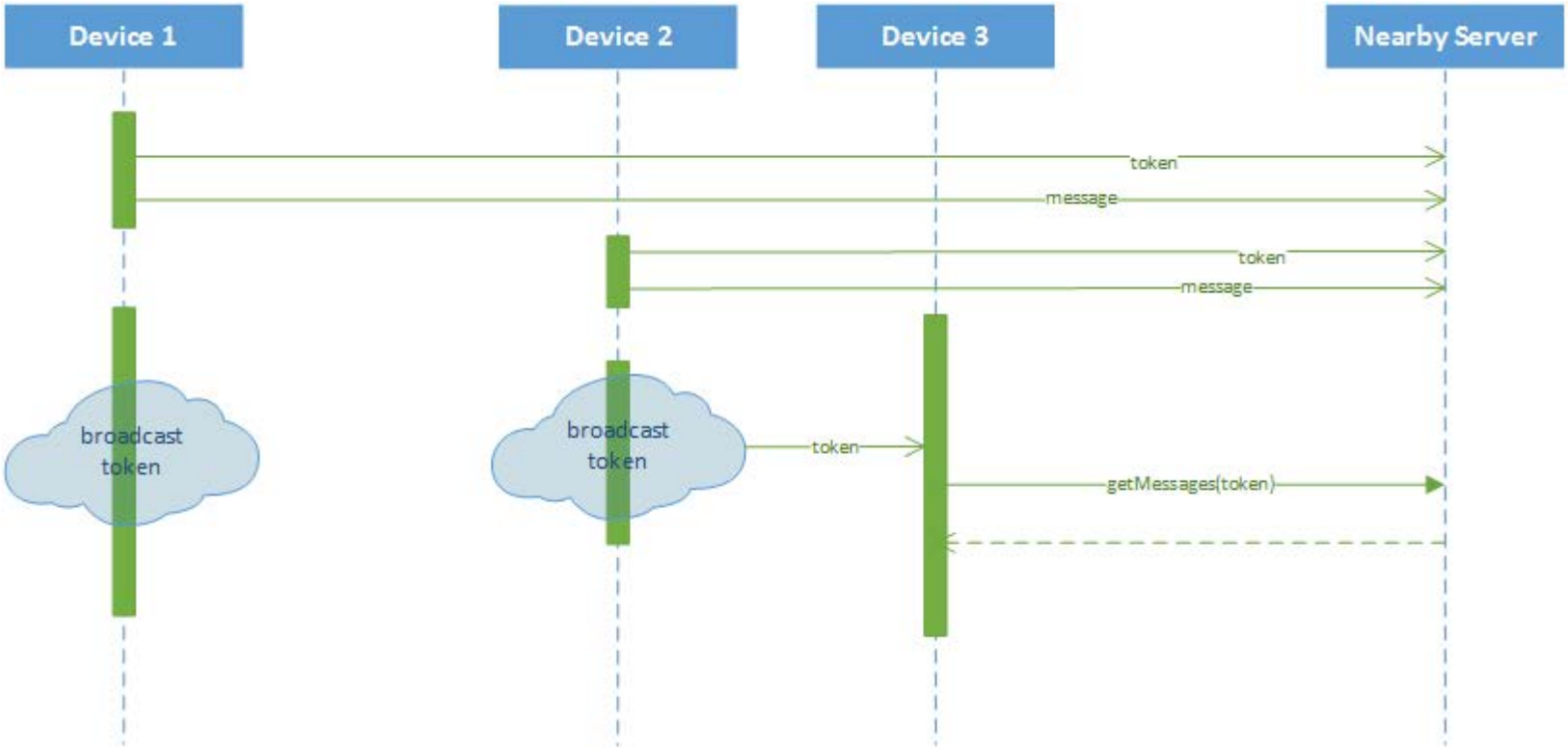
Example skill:
Computer & Electronics - Hardware

Litrally evrything that haz a power supply. Much knowledge. Very skill. Wow.

Example request:
Computer & Electronics - Hardware

My mom‘s printer goes „beep, beep“ all the time. Plz help!!!111!

Google Nearby connection sequence



Challenges

Connectivity

- Profile/request/skill data split into multiple messages to avoid corrupt data on connection abort
- Compression for data sets exceeding recommended message size of 3kb
- Only 2 scanning radius options available

Usability

- Notification on next meeting (opt-in)
- Recognize match – profile picture possible
- Trustworthyness of match – rate users
- Category-based request and ability comparision

Challenges

Energy

- Only scan/send if location changed – onLocationChanged() handler: calculate distance to last location (~10m)
- Disable scanning if battery is almost empty (<20%)
- Stop scanning after configurable idle time

Offline

- Offline detection by Nearby API – view/modify profiles/skills/requests from local DB
- Other people‘s profile data stored locally, updated on next connection (timestamp-based)

Adaption and Context

Transformation

- Coding: Java Object – MsgPack – byte[] - NearbyMsg
- Structure: profile fragments – skills, requests, profile information as independant data sets

Transmission

- Priorisation: profile ID > skills > requests
- Caching: profiles stored in local database

Reduction

- Nearby message size limited – only download profile ID + timestamp, request other information if needed

Work plan

Current State

- Analysis and testing of Nearby API
- Basic UI & functionality, database design

Open Issues

- Time-intensive testing: efficient scanning configuration, message handling
- Profile storage on server