



Application Development for Mobile and Ubiquitous Computing

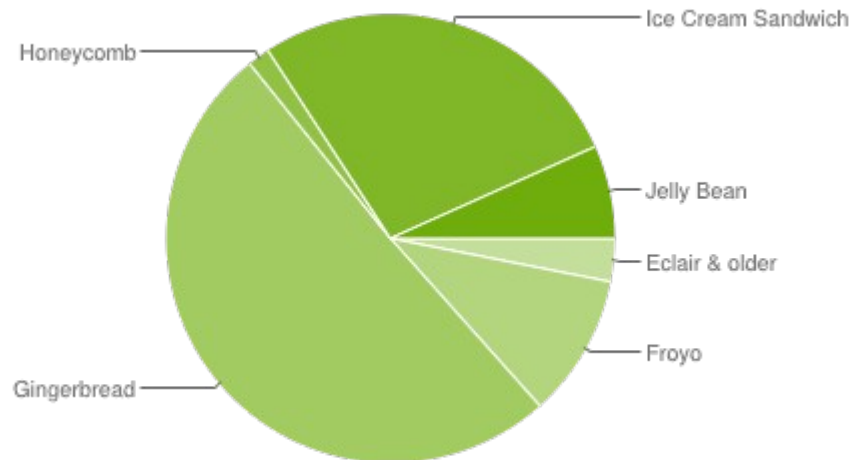
Seminar Task Second Presentation Application: **MeetUp!**

GroupNo. 08
Team: Fabian Göbel and Falk Morus

- Meet “MeetUp!” again
- Technologies
- Brief architecture overview
- Application scenario
- Challenges
- Next steps

- We're building a **location based** chat client for Android
- You can create and join so called **chat zones** on a map
- **Within** you can anonymously chat with all members of that **themed** zone
- A theme can be a topic of interest or name of a location

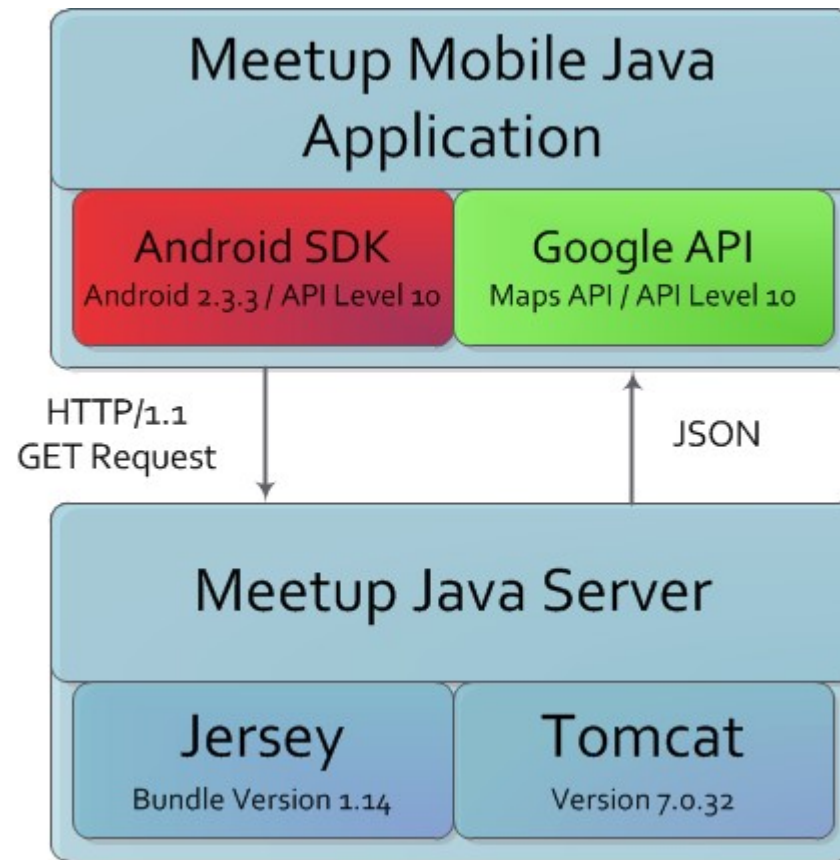
- Android Platform 2.3.3, API level 10, Gingerbread
- Thus at least 50,6% of all Android-based devices are supported^[1]



- Map implementation uses Google Maps APIv1 level 10
- Test device is a virtual 5.1" WVGA Android / ARM system
- Server is RESTful Java service running on Tomcat 7.0.32

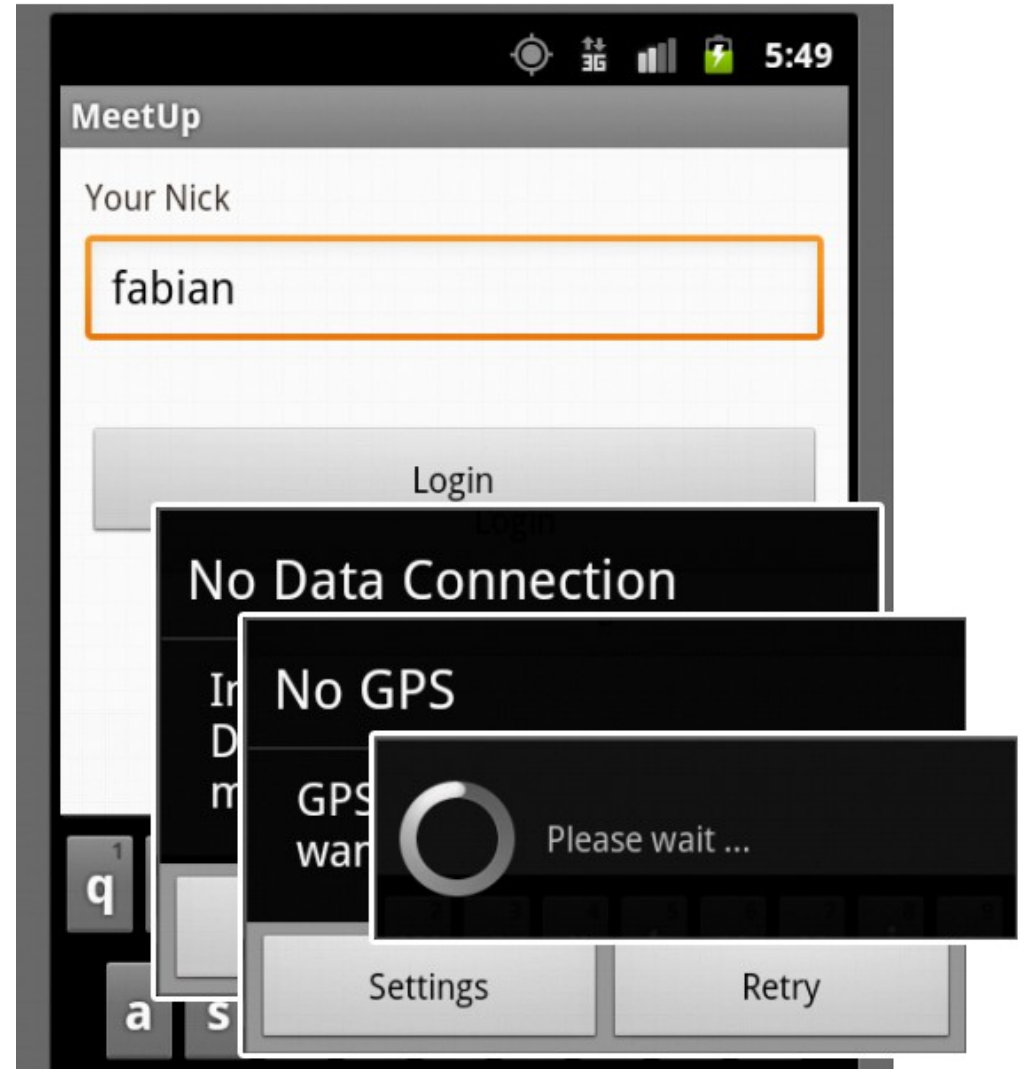
[1] <http://developer.android.com/about/dashboards/index.html>

- Overview of the project's architecture



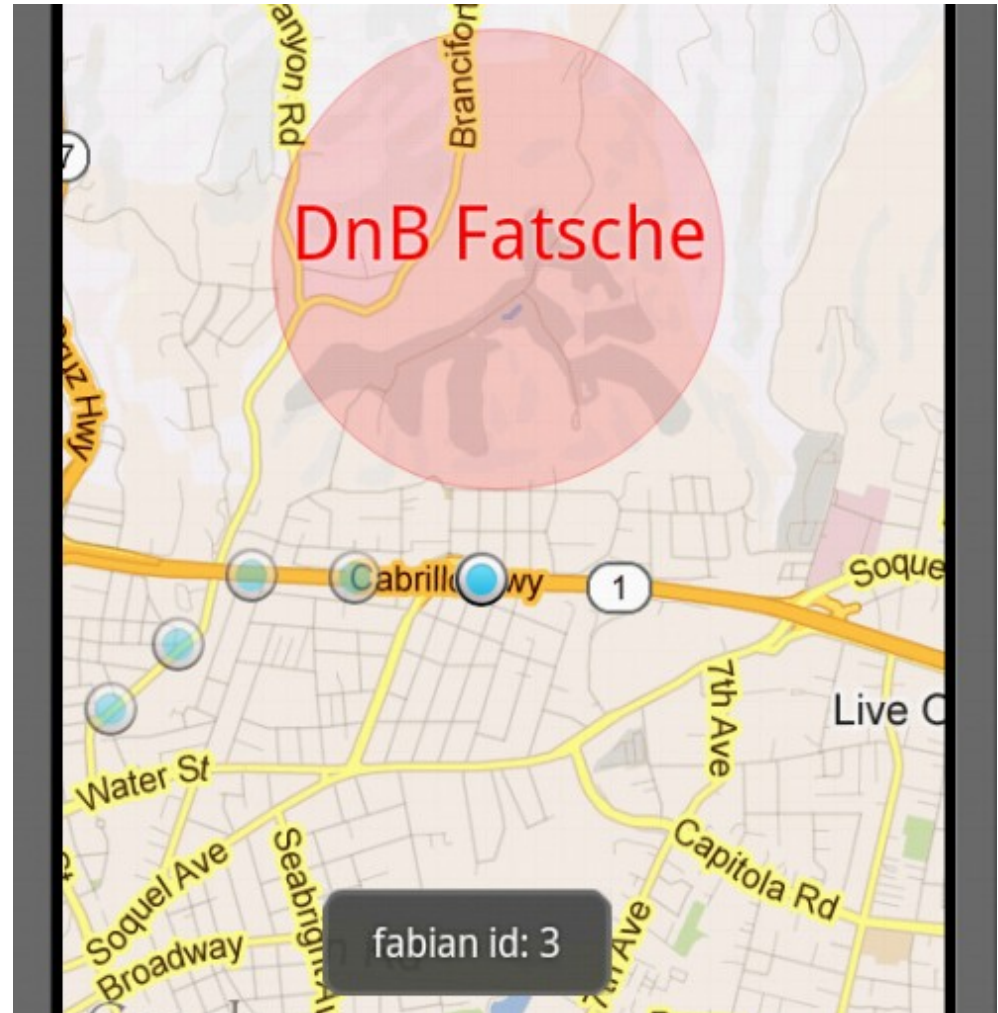
Login / Error handling

- Check if all necessary technologies are available
- If so, User is registered with the server and can continue to the **Map View**



Map View

- Once logged in, users can see their current location and chat zones nearby
- When within a zone the user can join



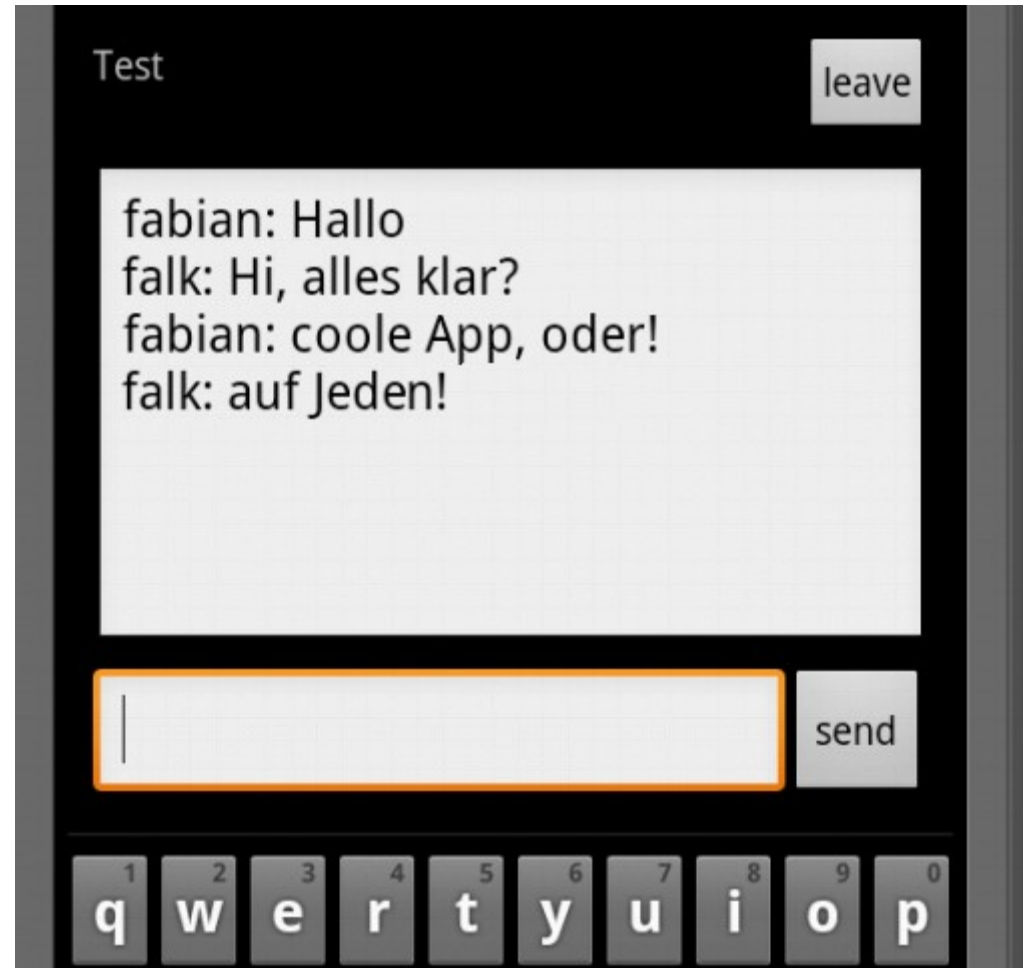
Create Zone

- User chooses title and size of the new zone
- When created User is directed to the **Chat Room**



Chat Room

- User chooses title and size of the new zone
- When created User is directed to the **Chat Room**



- Connectivity – how to cope with **data loss** and how to **maintain sessions**?
 - If no or defect data is received by the application a new request is sent again, responsible task canceled and new one is started
 - Session maintained by server / user authentication takes place in every request
- Battery life – **location updates** are expensive
 - Polling for location postponed as long as possible
- Async tasks – show **user related feedback** during background processing
 - Show user progress dialog and a “still processing” indicator (e.g. while loading map tiles)

- Optional: Adopting Google API v2
 - Addition: Polygons and Polylines which allow us to draw marker and objects more efficiently
 - Vector based tiles resulting in faster loading and consuming less bandwidth
 - Difficulties with testing on emulator
- Fancy chat room layout “bubble-style”
- Join zone around you
- Present message when about to leave current zone
- Possibility to whisper to other users
- Big multi user test on different virtual machines
- Followed by test with actual devices and online MeetUp! service

Thank you for your attention!

Are there any questions?