ETH zürich

Distributed Systems 2016 – Project

Subho Shankar Basu
subho.basu@inf.ethz.ch

Project

- Form a new group of 6 members
  - Ideally merge 2 groups
- Choose your own topic, the topic should
  - be a distributed application
  - involve Android
- Register via the submission system
  - Create a new group
  - Add members

Submission (via submission system)

- Project proposal submission due on November 18, 2016 midnight
  - 3-4 pages project proposal

- Final submission due on December 18, 2016 midnight
  - Code
  - 1-minute-madness presentation slides

Project proposal

- Only one project proposal per team (3 – 4 pages)
  - Focus on technical description of your work
  - Problem statement
  - System overview (e.g., architecture)
    - Clearly state the distributed system components
  - Requirements
    - E.g., external libraries, hardware
  - Work packages
    - Planning: Schedule + distribution among team members
- Use Latex (template provided)

Project presentation

- Prepare slides for 1-minute madness
- Focus on selling your idea
  - Make clear what your app does, why someone would need it and what is nice about it
  - Motivation, general idea, interesting technical aspects, results, ...
- 1-minute-madness will take place on 19th December, 2016
- Hint: Prepare in advance with trials
  - Strict **1 minute** limitation

Demo session

- Demo session will follow the 1-minute madness
- Similar to an exhibition booth
- Possibility to discuss with others and answer questions
- Inform us early enough about any special requirements for your demo
  - E.g., need Internet for your demo?
- All students are required to participate to be graded
  - Conflicts: Make sure at least one member can attend and demonstrate the project

Grading

- Project proposal
- 1-minute-madness presentation
- Project
  - Complexity
  - Implementation
  - Innovation

Sample project ideas

- Distributed file update:
  - Each user has a copy of a file
  - They synchronize with each other when they are within wireless range
  - Easier if there is a common server
  - Becomes complicated for P2P
  - Related: Design a P2P system among mobile users, able to update their status (busy, available, should meet, etc.) so that when they are within range they get corresponding notifications.
- Multi-player real time game
- P2P data transfer, video streaming involving service discovery mechanisms

Sample project ideas

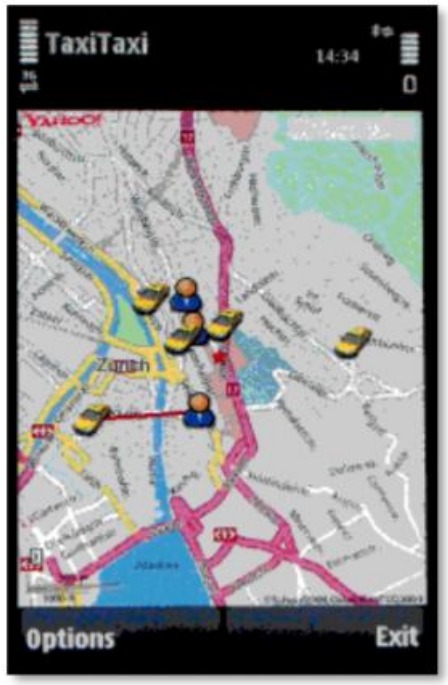
- Distributed storage server with
  - Dynamic load distribution (server addition, removal, maintenance downtime, etc.)
  - Fault tolerance
  - Data replication
  - Client interface to access data

Example projects

- Some selected projects from previous years

Consensus-based Taxi

- Implementation for the consensus problem
- Distributed application to find the optimal cab



djCrowd – Interactive distributed music player

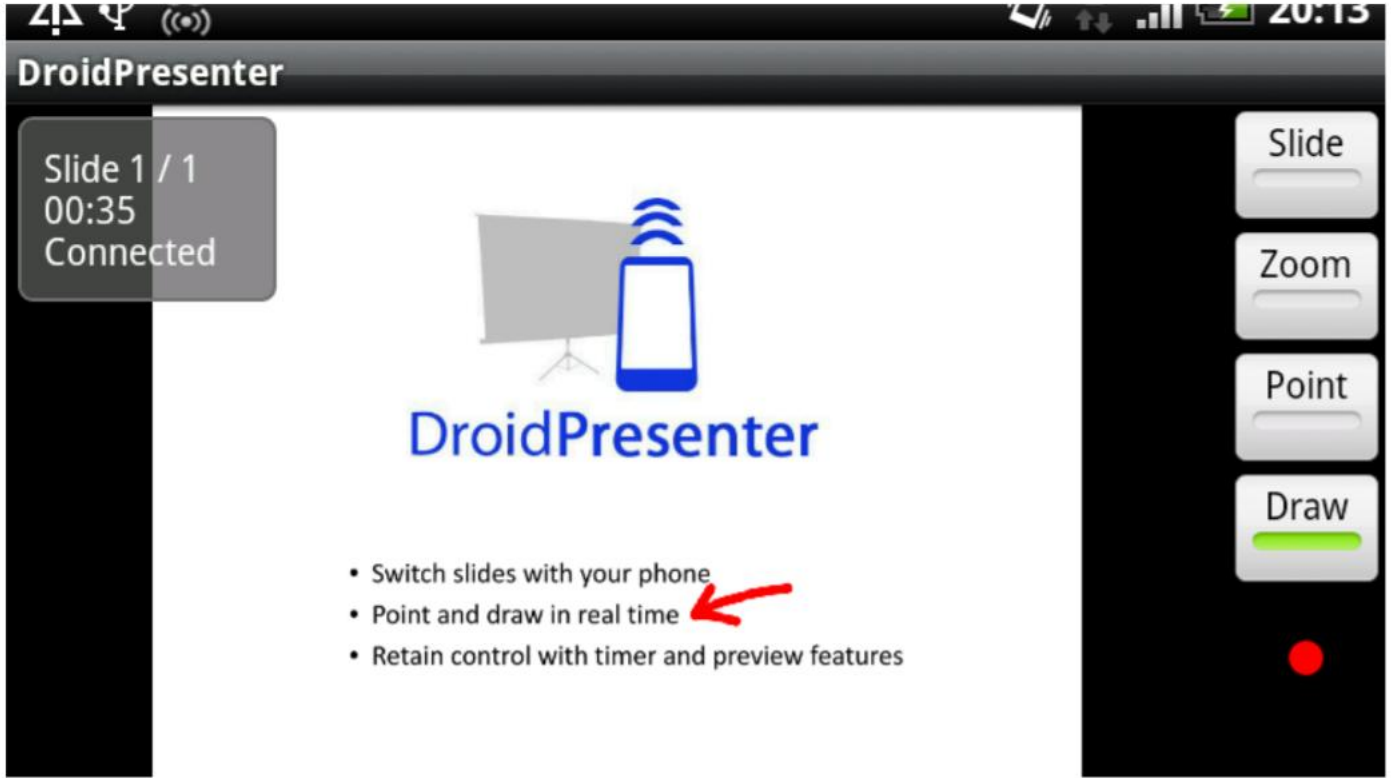
HS10: Luchin Doblies, Alexander Grest, Moritz Hoffmann, Jost Joller, Philipp Schmid, David Stolz

- Start up one phone as server (connected to hi-fi system)

- Your friends can connect to the server
  - Check the song that is currently playing
  - See upcoming songs in the playlist
  - Modify playlist by voting for their preferences
  - Upload songs from their phones
  - + Web interface to provide access for non-Android devices

DroidPresenter – Presentations remote control

HS10: Andreas Tschofen, Leonhard Helminger, Mathias Buerki, Damian Karrer

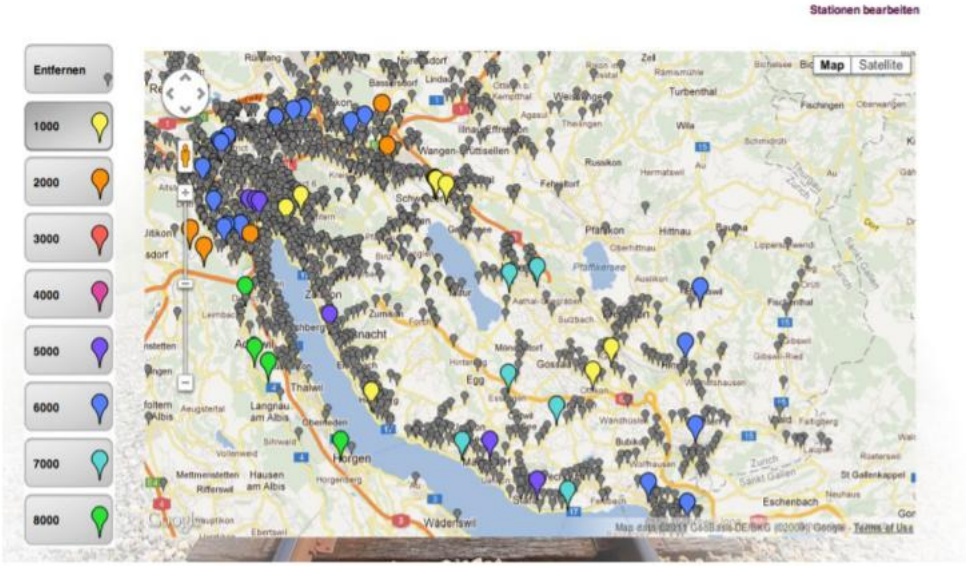


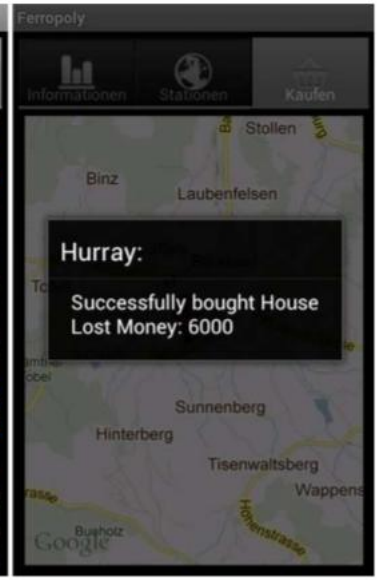
DroidPresenter allows you to draw in, point at, zoom in/out and control your presentation through your smartphone

Ferropoly – Monopoly in the real field

HS11: Ameri Michael, Aras Ersan, Marti, Messmer Stefan

- Emulate Monopoly in the real word
  - Travel across Switzerland and buy train stations
  - Ruby on Rails server
  - REST services with JSON interface

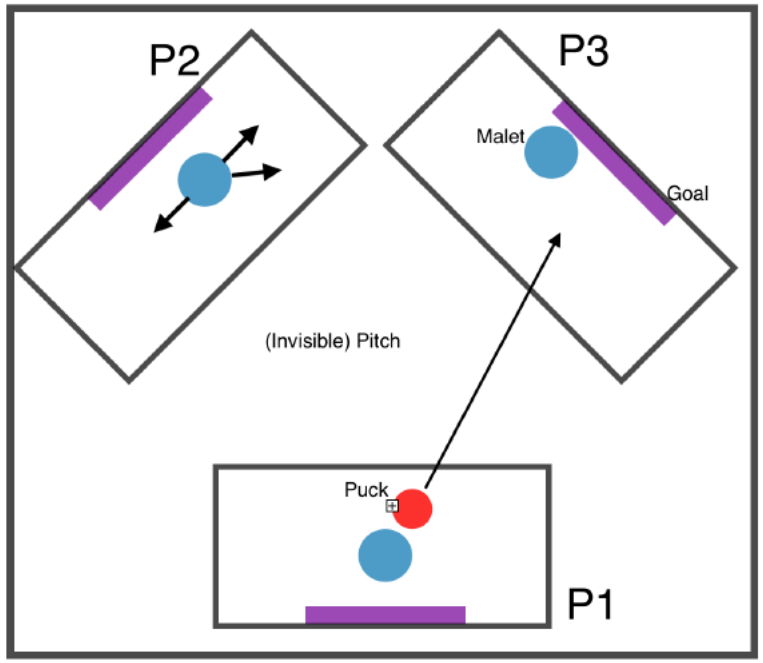




AirHockey 3X

Basile Maret, Philipp Rimle, Etienne de Stoutz, Oliver Butz, Raphael Schnider, Valentin Venzin

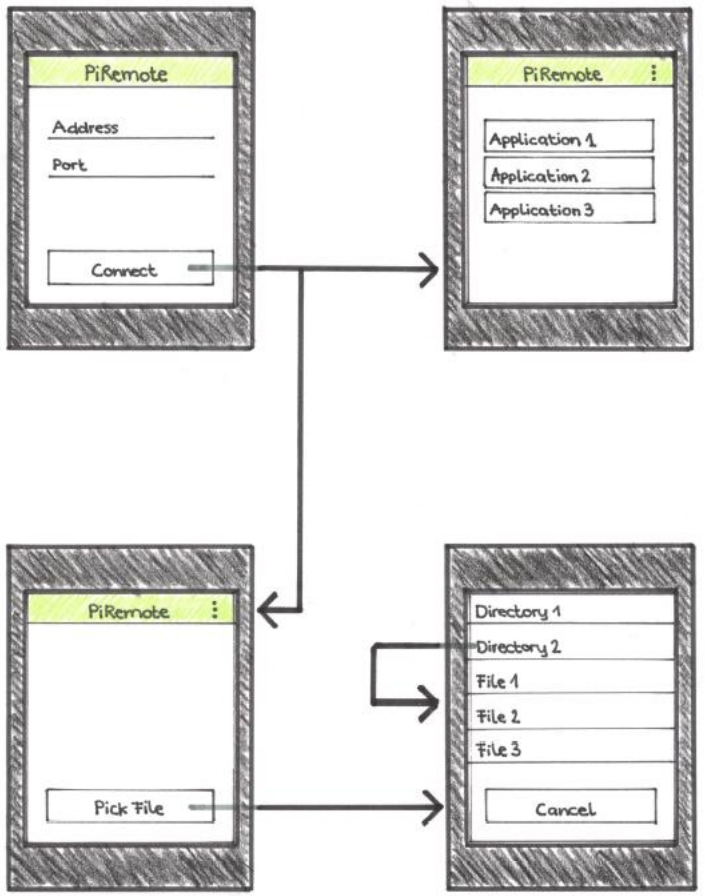
- 2-4 multiplayer game
- Local puck and mallet config
- Global points, state config
- Communication
- Physics
- Graphics



PiRemote (Remote control framework)

HS2015: Sandro Kalbermatter, Andrina Denzler, Mickey Vänskä, Fabian Murer, Francois Wirz, Julia Badertscher

- Raspberry Pi (server)
- Android devices (clients)
- APIs for the server and client
- App can use the APIs to design their own custom logic
- Clients and servers states synchronized automatically



Jass card game

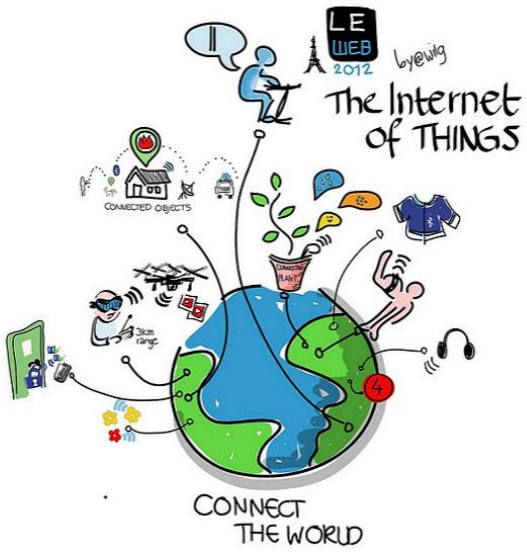
HS13: Fabian Stutz, Jannick Griner, Priska Pietra, Dejan Mircic, Michael Franz, Nicolas Forster

- Client-Server architecture
- Server = tablet
- Clients = mobile phones

- 3 main components to consider:
  - Networking
  - Game logic
  - GUI

Internet of Things Development Platforms

- We will provide IoT hardware for 4 projects
  - Estimote Beacons
  - TI SensorTag
- Make a dummy object smart
  - Change how people interact with the world around them



- "The Internet of Things (IoT) is the network of physical objects embedded with electronics, software, sensors, and network connectivity."
- Create opportunities for more direct interaction between the physical world and virtual world

Estimote Beacons

- ARM processor, BLE radio, sensors (accelerometer & temperature)
- iBeacon and Eddystone compatible
  - BLE device periodically broadcasting a unique ID
  - Location awareness applications

- Useful Links:
  - http://developer.estimote.com/
  - https://github.com/Estimote
  - Google Beacons: https://developers.google.com/beacons/?hl=en
  - https://github.com/google/eddystone
  - https://www.youtube.com/watch?v=SrsHBjzt2E8

TI SensorTag

- ARM processor, BLE radio, 10 sensors
  - Light, digital microphone, magnetic sensor, humidity, pressure, accelerometer, gyroscope, magnetometer, object temperature, and ambient temperature

- Links:
  - http://www.ti.com/tool/cc2650stk
  - https://store.ti.com/cc2650stk.aspx
  - http://processors.wiki.ti.com/index.php/CC2650_SensorTag_User%27s_Guide

In summary

- Use of Version Control Systems recommended (e.g., Git, Mercurial, or SVN)

- Deliverables
  - Project proposal (3 – 4 pages)
  - Code
  - 1-minute-madness presentation

- Important Dates
  - Project starts now
  - Project proposal due on November 18, 2016
  - Code, 1-minute-madness due on December 18, 2016
  - Final presentation on December 19, 2016
  - Your exact presentation slots will be specified later

Happy programming!