einz

Distributed Systems – Project Proposal

Clemens Bachmann
13-932-488
baclemen@student.ethz.ch

Josua Cantieni
15-919-038
josuac@student.ethz.ch

Fabian Gessler
15-939-341
fgessler@student.ethz.ch

Christian Knieling
14-923-809
knielinc@student.ethz.ch

Eric Mink
15-917-057
minker@student.ethz.ch

Silvia Siegrist
15-935-893
sisilvia@student.ethz.ch

ABSTRACT

This proposal should conclude the initial planning phase. This is where you choose a project, set your goals, clarify your ideas, and find the materials you will need. Concisely state *(a)* System overview *(b)* software and hardware you intend to use in this project, *(c)* expected deliveries of this project.

OUR PART: We chose to create an android application which allows to play the game "einz" which is very similar to the popluar UNO cardgame. The goal is to be able to play this game with friends wherever you are, as long as you have an android smartphone and access to the same local area network. For this purpose we will create an android application which is able to take the role of server and client at the same time. The device of one of the players is used as the server for the game which saves the state of the game and is responsible for synchronization. In this way, there is no extra server needed, except for the lookup of the players in case that they are not in the same LAN.

1. INTRODUCTION

Introduction to the problem you are working on, why is it important and what makes it challenging to solve, spell out the distributed systems challenges and how you plan to tackle them. State the technical problem you intend to solve. Indicate how it might be useful. This can be brief; it is just an introduction to the next section.

Use this section as well for background information, particularly if your project is building on previous work. If it is doing that, you need to refer to the work, describe it, and say how you are extending it. What are the new ideas?

Use references such as books [7], papers [9], or specifications [8] whenever available. Web-sites for documentation [5], tutorials, etc., are a special case. In a thesis, you would put them as footnotes. At this stage, however, you will only have a few "real references", so we put the Web-sites into the bibliography. Cite every source you used throughout the project.

OUR PART: We build a distributed game similar to the known card game "UNO" by Mattel [6]. Because you might often find yourself wanting to play a game with friends - e.g. while you are waiting for the next train - but without a set of cards to play it, it would be useful to always carry the cards on you. We make this easy by implementing a similar game on the phone as a native application.

Obvious difficulties awaiting us include the coordination of a team consisting of six people, each with different skillsets and time available. Also, we intend to create an easily extensible codebase so that we can first build the base game and in a second phase add further rules without much effort. This poses difficulties on its own as we have to learn coding patterns such as using factories to make the program code more modular.
Technical problems will probably be the selection of one device as the server, smooth and clean communication between the clients and the server, and implementing concurrency within the server. Selecting which device will be the server could be done by communicating P2P and choosing the device with the smallest IMEI, or just by letting the users choose.

For the usual difficulties in networking like message ordering and making sure the peers actually get the messages, we will rely on TCP.

2. SYSTEM OVERVIEW

This is the core of the proposal. It is where you spell out your technical plan and explain the project design. Expected evaluation/demonstration issues would also be addressed in this section. Use helpful figures such as Figure 2 and Figure 3, explain the figures in the text where you reference them.

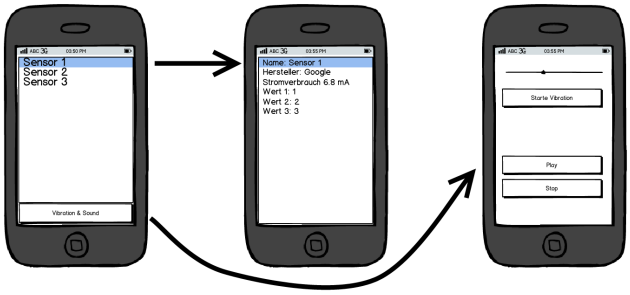

Figure 1: Only include useful figures. Do not simply copy something from a Web.

OUR PART: We propose a modular approach, building first the baseline functionality of the game, followed by further improvements like NAT-punchthrough to allow players from behind different routers to play via the internet with each other or adding more variations and rules. The additional rules will be available to the user of the device the server runs on before game start, such that we can dynamically change the way the game works.
Specifically, we would set up a Lookup-server that is reachable for all clients and thus allow the routers to setup forwarding on ports that we know. The LUS can then inform every client about the other client's IP address and ports, through which they can communicate as if they were within the same subnet, as already implemented in the first phase (See Figure 2). The gameserver on one phone will have the option to choose whether to use the LUS or only accept players from within the same LAN. Because of this, we will need a local WiFi in order to demonstrate our app.

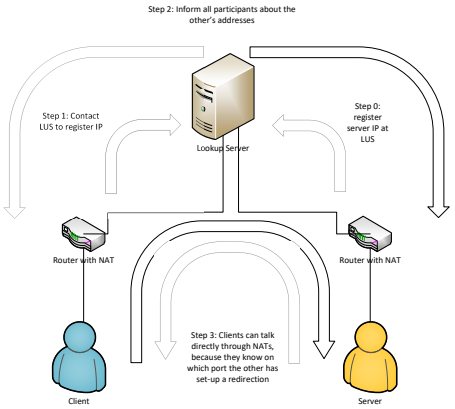

Figure 2: NAT traversal

First of all, we implement a simple server-client setup and define the format and the messages that should be provided as an interface between the client and the server. [1] At the same time, we can start implementing the user interface and writing this proposal.
In a second step, once the basic communication between server and client works and the exact behaviour of the two parts has been defined, we implement the serverside game logic and the previously defined messaging functionality. We will make it so that the server does most of the computations and only sends the client its state, containing the players hand, the number of cards other players are holding and the most recently played cards. The client receives also a list of possible actions, from which it can choose one without having any clientside support for them, except for the user interface. This allows us to later add gamemodes and rules with ease, because we will only need to change a few classes. We are also trying to keep the classes themselves very modular to make updating them - e.g. by adding further messages - just as simple.
Once the game is in a working state with a few rules such as an additional card like the one where the player can choose a color or the well-known "+2" card from UNO, we will decide whether we need to focus on bugfixing, cleaning up code, implementing a LUS or adding more rule options. Adding a LUS might impose additional difficulties because some mobile carriers might use symmetric NATs. If these difficulties arise, we will probably resort to only implementing NAT-traversal for the other (easier) types of NATs.

So how exactly will we enable dynamic rules? The UI will feature settings on the device which acts as the server. These will mostly be checkboxes or multiple-choice dropdown menus. This means that we will not feature user-designed rules. However, we intend to make use of the efforts we put into the extensibility of our codebase.

Regarding the difficulty of working in a larger team than most of us are used to, we intend to simplify the cooperation by using git [2] and trello [4]. We have also appointed a Team Lead whose task it is to distribute the work so that everybody can contribute meaningfully to the Project yet also learn something while doing so.

To demonstrate our application in the end, we will need at least a working WiFi network and maybe internet as well, depending on how much of the NAT traversal we will have implemented.

TODO: wer macht eine Grafik als Ãijberblich zum aufbau und vielleicht noch eine zum UI?

3. REQUIREMENTS

Describe system setup, components, external libraries, hardware etc.

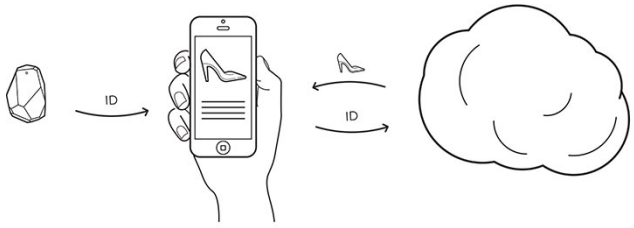

Figure 3: System Overview [3]

OUR PART:
TODO:Josua beschreibst du das "system setup"? TODO:jeder der "external libraries, hardware etc." benutz bitte diejenigen beschreiben
Our App runs on Android devices with at least Android 5.0 installed. We use the local wireless connection to communicate between the devices. One device will act as a server and waits for the clients to connect. Once every player is connected the user that started the server can start the game.

4. WORK PACKAGES

Breakdown the work to subtasks to meet the project requirements. Define and describe these tasks.

- **WP1**: XYZ ...
- **WP2**: Set and Configuring Backend Serve ...
- **WP3**: Integration ...
- **WPx**: ...

Stick to a concise, scientific writing style.
OUR PART: There work will be broken down into the following subtasks: TODO: mÃijsste man den WPs noch andere namen als nummern geben? TODO: jeder beschreibt das WP fÃijr das er verantworlich ist. dÃijrft meine beschreibungen gerne lÃűschen ;)

- **WP1**: Define Client-Server Communication
  We define the protocol that the client and the server uses to communicate. Once we agreed on a message exchange protocol the developement of the server and the client can be done separately.
- **WP2**: Server Game Logic
- **WP3**: Client UI
- **WP4**: Dynamic Message Parser
- **WP5**: Implement Client Server Protocol

5. MILESTONES

The milestones section provides a work plan for carrying out the project. This is your schedule for getting the project done. Clearly state how the work packages will be distributed among the team members.
OUR PART:

5.1 Milestone 1

- Define Client Server Protocol

5.2 Milestone 2

- Getting client server to communicate with the Protocol

5.3 Milestone 3

- Implement Game logic and Client UI

We think it is important to have someone who has an overview of the whole project, therefore we assigned a project manager. For the implementation of the application we made two groups, one for the server side and one for the client side. There is one person responsible for the organisation and delegation of the work to other people for both server and client side. These two people also have to set an API to make the two parts work together in the application.

As project manager and organizing the structure of the project: Josua

Server: Responsible for organisation and also helps implementing: Eric, helps with implementation: Fabian

Client: Responsible for organisation and also helps implementing: Chris, helps with implementation: Clemens

UI and Logo: Chris

First responsable for writing the proposal, then helps implementing where there is need: Silvia

6. REFERENCES

[1] documentation_Messages.md, JSON Interface Documentation. `https://github.com/lucidBrot/einz/blob/master/protocols/documentation_Messages.md`. Accessed on 13 Nov 2017.

[2] Einz git repository. `https://github.com/lucidBrot/einz`. Accessed on 13 Nov 2017.

[3] Estimote. `http://estimote.com/`. Accessed on 26 Oct 2015.

[4] Our trello dashboard. `https://trello.com/b/4F0OdNPI/distributed-systems`. Accessed on 13 Nov 2017.

[5] Services: Sending Notifications to the User. `http://developer.android.com/guide/components/services.html#Notifications`. Accessed on 29 Aug 2013.

[6] "UNO" seller. `http://shop.mattel.com/shop/en-us/ms/uno-game`. Accessed on 13 Nov 2017.

[7] E. Burnette. *Hello, Android: introducing Google's mobile development platform*. Pragmatic Bookshelf, 3 edition, 2010.

[8] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1. RFC 2616, 1999.

[9] R. T. Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. Phd thesis, UC Irvine, 2000.