Problem Statement Document

Table of Contents

1. The Problem

Different use cases exists that leverage from peer to peer architectures like e.g. sharing data in applications when no Internet connection is available using Bluetooth or WiFi.
In other use cases with Internet connection, no dedicated server is required, that might be a bottleneck and that might also lead to high costs.
Nevertheless applications need to implement a lot of logic in order to allow peer-to-peer communication like finding other services, connecting to them and reliably sending and receiving data in a special representation.
The main goal of the project will be the development of a framework that simplifies the development of peer-to-peer applications by providing different services and simplifying complex details.

2. Use Cases

The framework shall mainly support the following use cases

1) **Find other users:** It should be possible to find other users with the same application over Bonjour or with the help of a web server.
2) **Connect to other users:** It should be possible to connect to other devices. It might be required to develop a master-slave mechanism for this connection to increase reliability of network transfer. This needs to be evaluated
3) **Send data to other users:** It should be possible to send data to other users in form of objects. The application does not need to care about the representation of the objects.

4) **Receive data from other users:** It should be possible to receive data from other users in form of objects. The application does not need to care about the representation of the objects.
5) **Configure connection:** It shall be possible to configure some connection settings like the maximum of possible connections, the desired representation or whether connections can be added dynamically at runtime

Part of the framework development is to identify and prioritize additional use cases.
Implemented use cases need to be demonstrated with a sample application.

3. Functional Requirements

The developed framework shall mainly consist of two services:

1) **Connection service:** This service finds other users the application can connect to. It offers to connect via Bonjour (WiFi or Bluetooth) and if configured over a server in the Internet which single purpose is to provide a similar service to Bonjour, which is limited to local area networks. This service also allows the user to connect to available other users by providing a nice user interface.

2) **Communication service:** After the user connected his application to other users, he wants to send/receive data over TCP to other users. The communication service provides object-oriented interfaces and simplifies the process of sending and receiving data. It automatically cares about the representation of the data and provides an interface and two example representations (XML, JSON) for data transfer. It provides interface to send objects over the network (including the automatic serialization) and to retrieve these objects on the other side (including the automatic deserialization). With the provided interface for representations, it allows to add new representation possibilities in an application.

Part of the framework development is to identify and prioritize additional requirements.
Implemented requirements need to be demonstrated with a sample application.

4. Nonfunctional Requirements

The framework shall fulfill the following nonfunctional requirements:

- **Extensibility:** The framework provides extension mechanisms like e.g. the creation of own Representations

- **Flexibility:** With the help of delegation and blocks the application is able to customize the behavior of the framework when it is useful
- **Performance:** Even large data can be sent over the network in a short time. The framework shall support compression and authentication
- **Reliability:** Network transfer shall be as reliable as possible
- **Usability:** It should be easy to develop an application with the framework. This includes well design interfaces that can easily be understood and hiding complex details from the application's developer.

5. Target Environment

The framework shall be developed for iOS 6.x and should be compatible with iOS 5.x. It shall support both, iPads and iPhone applications. Additionally the framework shall also run on Mac OS X 10.7.x and 10.8.x. If parts of the framework (like e.g. UI elements) cannot be developed to automatically support both platforms, two different implementations shall be provided.

6. Deliverables

1) RAD, Requirements analysis document (with the given template)
2) SDD, System design document (with the given template)
3) Source code of the framework and a sample application (commented in Doxygen style)
4) Developer Documentation incl. a tutorial how to use the framework

7. Schedule

• Thu, October 11th 2012:	Kickoff Meeting
• Mid of December	Design Review
• End of semester:	Client Acceptance Presentation

8. Client Acceptance Criteria

All high and medium prioritized functional requirements and use cases are implemented. Nonfunctional requirements are fulfilled or, if not possible, a good argumentation is documented why they cannot be fulfilled. All deliverables of section 6 are handed out in a good quality.