# Deliverable 3: Sprint #1

for

Danske Bank: Peer-to-peer

Group Delta:
Jesper Borgstrup, Thomas Kjeldsen and Mads Ohm Larsen

March 11, 2011

## 1 Requirements for this deliverable

1. Doing a demo in class (on 2011-03-09)
2. Giving us access to your source code
3. Handing in a collection of your sprint material
4. Describing a sprint retrospective (e.g., as a set of bullets outlining what when well, what went wrong, and how you will improve for the next sprint)

## 2 Source code access

Our source code is available on Github from: https://github.com/omegahm/DBP2P

Please note that we're working on multiple branches (use the button switch branch to view another branch).

The Sprint 1 development branch is named "dustytuba".

If you wish to checkout our code (read-only) using Git, then use git clone with this URL: `git://github.com/omegahm/DBP2P.git`

For this deliverable, your group must have completed the first sprint and docu- mented it by: 1. Doing a demo in class (on 2011-03-09) 2. Giving us access to your source code 3. Handing in a collection of your sprint material 4. Describing a sprint retrospective (e.g., as a set of bullets outlining what when well, what went wrong, and how you will improve for the next sprint) Please hand in 2-3 as one PDF file.

Sprint Demo The Scrum Master should present the results of the first sprint to the class in the form of a running system. Consider the following: • Show you sprint backlog: what was planned, what was finished, and what was the process? • Present finished stories only in a short, fast-paced demo (5-10 minutes). Focus on what has been done (in contrast to how it was done) • Run on emulator or device (for Android code), make sure that everyone can see and understand the interaction Sprint Material To assess your progress, we need access to the following1 : 1 Make sure that you keep track of your sprint material corresponding to the end of sprints. You might need them for the exam 1 • source code. A version number and access method is sufficient • product backlog (before and after the sprint) • sprint backlog • any other material (e.g., burndown chart) that illustrates your progress

# 3 Purpose/Vision

The aim of this project is to provide a library, which Android App developers can utilize to facilitate easy secure communication between two phones over Bluetooth.

Providing secure communication can be reduced to two discting problems: 1) securing communication against eavesdropping and 2) validating the identity of the other party to protect against man-in-the-middle style attacks.

We will use Bluetooth as the transport medium over which the transfers take place, which in addition to cryptographic measures adds additional resilience by forcing a potential attacker to be in close proximity to the user.

Bluetooth offers advantages compared to transmitting data over the mobile network, since it is is quite efficient in terms of both power drain and transfer speed and it is not subject to usage fees. *** SOMETHING ABOUT EASE (or rather lack thereof) of setting up Bluetooth connections (from a user perspective) ***

The library will be able to support multiple identity providers, which provide identification of the client and transport the neccessary information (out of band) for setting up the secure bluetooth transport. Support for using Bump as an identity provider will be included as a part of the project, but the library could be extended with further identity providers such as NFC (Near Field Communication) or 3rd party solutions e.g. a custom authentification server set up by a Bank or by NemID.

Information exchanged through Bump is subject to the Bump privacy policy, which specifically states that Bump is allowed to retain all exchanged information. For co-op games and other non-sensitive applications this may be good enough, but for exhange of financial information it is not neccesarily good enough. *** REWRITE/REMOVE: not true ***

*** REWRITE / REFOCUS below ***

We have chosen to focus our sample application and prototype to target financial institutions and bank where high security and transactional confidentiality is a prerequisite, but the library will be of value to anyone who wants to utilize the Bump API while safeguarding transactions.

Banks who use our library will be able to facilitate easy transfer of money from one customer's account to another customer's account, by merely bumping their phones together.

Danske Bank is one potential customer, since its marketing strategy is throughly based on a strong presence in both social media and on the mobile platform. Their mobile banking application already has a number of technology features and it is one of the key differentiating features compared to their competitors among Danish banks.

In addition to easing transactions, adding Bump would add yet another "cool" feature, which users will hopefully show off to friends and family similar to the recent/current voice recognition "craze" (Voice Search, Voice and Translate from Google, Tellme voice search for Windows Phone 7 etc.), thus adding further value through viral marketing.

Finally it's worth noting that PayPal, which is a major international player in micro-payments, currently has Bump functionality in their mobile app. While PayPal does not pose a significant threat to established financial institutions at present time it's worth staying on the forefront in case of a sudden surge in either micro-payments or Bump-technology "craze".

### 3.1 Roadmap

Our roadmap is divided into our five sprints.

These five sprints, and their goals, are as follows:

- Sprint #1 (March 11$^{th}$, 2011)
  - Read up on relevant documentation, APIs and product and process brief refined. Implementation of a fully working prototype that allows two Android smartphones to be bumped together and transfer information between them.

- Sprint #2 (April 1st, 2011)
  - Sample App that can perform secure transfers between two phones. Functionality packaged as a distributable library.
- Sprint #3 (April 29th, 2011)
  - Different approaches as how to secure information in transit have been evaluated and benefits and drawbacks have been thoroughly examined. Prototype app can perform secure communication using the preferred approach.
- Sprint #4 (May 20th, 2011)
  - UI polished and user experience evaluated.
- Sprint #5 (June 10th, 2011)
  - Final product delivered to customer
  - Paper finished and submitted

# 4 User/Customer

Our customers are Android app developers, banks and financial institutions in particular, who are interested in utilizing the Bump to facilitate communication or transactions between two phones, but are not interested in leaving entirely in the hands of the creators of Bump and having clear text data be subject to the Bump privacy policy.

Sune Lomholt has agreed to act as a representative of our product owner(s).

# 5 Team resources, roles and obligations

The team consists of three Master's students at the Department of Computer Science of the University of Copenhagen (DIKU):

| **Name** | E-mail | Telephone |
|---|---|---|
| Jesper Borgstrup | jesper@borgstrup.dk | (+45) 61 30 30 81 |
| Thomas Kjeldsen | thomas@thomaskjeldsen.dk | (+45) 61 30 80 01 |
| Mads Ohm Larsen | mads.ohm@gmail.com | (+45) 60 16 39 53 |

## 5.1 Skills

In our team we will need some different skills. This is the distribution on how we find ourselves fitted for these skills:

| Skills | High | Medium | Low |
|---|---|---|---|
| Android development | Jesper | Thomas | Mads |
| Configuration Management | Mads | Jesper and Thomas | |
| Scrum Master | | | Jesper, Thomas and Mads |
| User involvement | | Mads and Jesper | Thomas |
| Test | Mads and Thomas | Jesper | |

### 5.2 Resources

Correspondence between team members and roles:

| Team Member | Roles and responsibilities | Strengths and focus area |
|---|---|---|
| Jesper | Coder | Development |
| Thomas | Scrum Master, tester | Test |
| Mads | SCM maintainer, tester | Test, user involvement |

These roles are very loose, and will be subject to change throughout the course. E.g. we will all be Scrum Master at some point during this course.

## 6 Team empowerment

- The team breaks down and estimates prioritised work items in the sprint backlog
- The team jointly determines how to perform the work, including possible ad hoc planning meetings
- Each team member plans their own daily work in respect for planned activities/meeting

## 7 Team values

The following are agreements made between the project members

- We're respectfully nice to each other.
- If someone's feeling overwhelmed they're entitled to a hug (subject to a daily maximum of 3 and a weekly maximum of 5)
- We show up on time, and if we're prevented or running late we let each other know.
- We speak openly about problems
- We share information, be that information from or to product owner, or between us, via e-mail, the configuration management system or other ways
- We participate equally, meaning that everybody have an equal work load

# 8 Team processes

- Our code will be open for everybody to see
- We have agreed on using git, and our code will be stationed at GitHub
- We have agreed on working together in person, but also being available on both a messenger platform and on Skype - or eMeeting
  - eMeeting is a platform suggested by Sune Lomholt

## 8.1 Configuration management

We use Git as our SCM system. Our repository is hosted publicly on GitHub.

We're using a Test Driven Development approach, which imposes the following structure on the `master` branch:

- Code committed to the central repository must be thoroughly covered by JUnit tests.
- Tests must not fail when comitted, even if this means methods are stubbed.
- Code must compile.

## 8.2 Office rules

When working together, e.g. sitting together and working on stories, the following should be respected:

- No loud music or noises
- Procrastination should not affect others

## 8.3 Calendar planning

We have agreed to do a daily virtual scrum logging, where each participant writes an entry firstly about what he did today, what he will do tomorrow, and if any problems have occurred.

Sprint demos will occur at, or around, the ending of a sprint, depending on when the product owner have got time. At the time of sprint demo, we will properly want to do a sprint planning, for the following sprint, together with the product owner.

At the end of a sprint, we will have a so-called sprint retrospective, discussing what went well and what could be approved for next sprint.

# 9 Team performance & progress monitoring

To support our process we are interested in a scrum tool that, especially because we have no offices or static whiteboards, facilitates a virtual collaboration using the scrum model.

We are using Google Docs as our scrum tool and have set up a logbook of (virtual) daily scrum meetings, burndown chart, sprint- and product backlogs.

Throughout Sprint 1 we will will assess how the tools are working to ensure that they are being used to their maximum potential and they are providing the neccesary product support.

Pending this assessment we will consider evaluating additional scrum tools, such as IBM's Rational Team Concert (based on the Jazz collaboration platform).