Developing Bleeding-edge microservice solutions for complex problems

Non-intrusive technology in Walking Meetings

VIGNESH MEENAKSHI SUNDARAM
Developing Bleeding-edge microservice solutions for complex problems: Non-intrusive technology in Walking Meetings

Cloudlösning baserad på mikrotjänster med bleeding-edge technology vid utveckling av interaktivt tekniskt support för gå-möten

Vignesh Meenakshi Sundaram
vms@kth.se

Computer Science and Communication, Master of Science in Security and Mobile Computing

Supervisor: Helena Tobiasson
Examiner: Jan Gulliksen
KTH – Royal Institute of Technology
June 12, 2017
ABSTRACT

The last decade has seen an emergence of various types of cloud services and development frameworks offered by leading companies in the software industry. While each of these services has been used to solve specific tasks, their specifications have changed over time as they have matured. Therefore, integrating these components to solve a whole new task tends to get tricky due to their incompatible and experimental nature. While some technology components might continue to be developed, others might deprecate. In this thesis, using a user-centered design and agile development approach, we have attempted to develop a cloud solution using microservice software architecture by integrating state-of-the-art technology components to solve a totally new task of providing a non-intrusive technology experience during walking meetings. We present our results based on interaction with the research group, user studies as a part of the research study “Movement of the mind”, and expectations of the working prototype within the context of walking meetings. We also present the features of the prototype and our motivation for choosing the tools to develop them. Finally, we discuss the development challenges faced during our attempt and conclude whether it is plausible to integrate various components of bleeding-edge technology to solve complex real-life problems or rather wait for these technologies to mature.

SAMMANFATTNING

# Table of Contents

ABSTRACT .............................................................................................................................................. 1

Keywords ............................................................................................................................................... 1

INTRODUCTION ..................................................................................................................................... 1

Monolithic Software Architecture .......................................................................................................... 1

Microservice Software Architecture ..................................................................................................... 1

Bleeding-edge technology ....................................................................................................................... 1

  Cloud service -The Google Cloud ...................................................................................................... 2
  React Native ......................................................................................................................................... 2
  Firebase ............................................................................................................................................... 2
  Flic Bluetooth button ........................................................................................................................ 2

Integrating bleeding-edge technology components ............................................................................... 2

THEORY / RELATED WORK .................................................................................................................... 2

  Choosing non-intrusive technology .................................................................................................. 2
  Walking Metro .................................................................................................................................... 2
  Meeting to GO ................................................................................................................................. 3

  Monolithic architecture in existing solutions .................................................................................. 3

METHODS ................................................................................................................................................ 3

  User-Centered Design ....................................................................................................................... 3
    User Experience and Experience Design ...................................................................................... 3
    Interactive Devices ........................................................................................................................ 3

  Agile Development ........................................................................................................................... 3
    Adapting the system based on user needs .................................................................................... 3
    Extreme Programming .................................................................................................................. 3

RESULTS .................................................................................................................................................. 4

  Expectation and features of the Cloud solution for walking meetings (P1) ...................................... 4
    Web portal ...................................................................................................................................... 4
    Mobile application ......................................................................................................................... 4
    Flic button .................................................................................................................................... 4
    Meeting analysis .......................................................................................................................... 4

  Designing the architecture of the Cloud solution (P2) ..................................................................... 5
    Front-end component of the Web portal ....................................................................................... 5
    Back-end component of the Web portal ....................................................................................... 5
    Mobile application and Flic button ............................................................................................... 5
    Focus on usability .......................................................................................................................... 5

  Evaluation of user experiences based on the first version of the prototype (P3) ................................. 5
    Non-intrusiveness of the technology ............................................................................................ 6
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sources of controlling the Flic button</td>
<td>6</td>
</tr>
<tr>
<td>Trusting the technology</td>
<td>6</td>
</tr>
<tr>
<td>Recording while walking</td>
<td>6</td>
</tr>
<tr>
<td>Feedback mechanism while using the Flic button</td>
<td>6</td>
</tr>
<tr>
<td>The Flic button as a wearable</td>
<td>7</td>
</tr>
<tr>
<td>Preparing for the meeting</td>
<td>7</td>
</tr>
<tr>
<td>Access to meeting analytics</td>
<td>7</td>
</tr>
<tr>
<td>Potential benefits of the application</td>
<td>7</td>
</tr>
<tr>
<td>Finalizing the architecture and second version of the prototype (P4)</td>
<td>8</td>
</tr>
<tr>
<td>The back-end service</td>
<td>8</td>
</tr>
<tr>
<td>The Android mobile application</td>
<td>8</td>
</tr>
<tr>
<td>DISCUSSION</td>
<td>8</td>
</tr>
<tr>
<td>User-centered Agile Approach</td>
<td>8</td>
</tr>
<tr>
<td>Understanding the culture</td>
<td>8</td>
</tr>
<tr>
<td>Observing and learning</td>
<td>8</td>
</tr>
<tr>
<td>Customized Features and Iterative Development</td>
<td>9</td>
</tr>
<tr>
<td>Conception</td>
<td>9</td>
</tr>
<tr>
<td>Architecture design</td>
<td>9</td>
</tr>
<tr>
<td>Feature implementation considerations</td>
<td>9</td>
</tr>
<tr>
<td>Challenges faced during development</td>
<td>9</td>
</tr>
<tr>
<td>Updating meeting specifics</td>
<td>9</td>
</tr>
<tr>
<td>Speech-to-Text</td>
<td>9</td>
</tr>
<tr>
<td>Limitation of Flic recording on the iOS platform</td>
<td>10</td>
</tr>
<tr>
<td>Audio format conversion from 3GP to FLAC</td>
<td>10</td>
</tr>
<tr>
<td>Benefits of choosing the FLIC button</td>
<td>10</td>
</tr>
<tr>
<td>Technical benefits</td>
<td>10</td>
</tr>
<tr>
<td>Usability benefits</td>
<td>10</td>
</tr>
<tr>
<td>Ethics</td>
<td>10</td>
</tr>
<tr>
<td>Sustainability</td>
<td>10</td>
</tr>
<tr>
<td>Economic Sustainability</td>
<td>11</td>
</tr>
<tr>
<td>Ecological Sustainability</td>
<td>11</td>
</tr>
<tr>
<td>Social Sustainability</td>
<td>11</td>
</tr>
<tr>
<td>Future work</td>
<td>11</td>
</tr>
<tr>
<td>Longer User Tests</td>
<td>11</td>
</tr>
<tr>
<td>Exploring the market for more potential users</td>
<td>11</td>
</tr>
<tr>
<td>Extending the development</td>
<td>11</td>
</tr>
<tr>
<td>CONCLUSION</td>
<td>12</td>
</tr>
</tbody>
</table>
Developing Bleeding-edge microservice solutions for complex problems: Non-intrusive technology in Walking Meetings

Vignesh Meenakshi Sundaram
vms@kth.se

ABSTRACT
The last decade has seen an emergence of various types of cloud services and development frameworks offered by leading companies in the software industry. While each of these services has been used to solve specific tasks, their specifications have changed over time as they have matured. Therefore, integrating these components to solve a whole new task tends to get tricky due to their incompatible and experimental nature. While some technology components might continue to be developed, others might deprecate. In this thesis, using a user-centered design and agile development approach, we have attempted to develop a cloud solution using microservice software architecture by integrating state-of-the-art technology components to solve a totally new task of providing a non-intrusive technology experience during walking meetings. We present our results based on interaction with the research group, user studies as a part of the research study “Movement of the mind”, and expectations of the working prototype within the context of walking meetings. We also present the features of the prototype and our motivation for choosing the tools to develop them. Finally, we discuss the development challenges faced during our attempt and conclude whether it is plausible to integrate various components of bleeding-edge technology to solve complex real-life problems or rather wait for these technologies to mature.

Keywords
Bleeding-edge, microservice architecture, walking meeting; non-intrusive technology; cloud solution; user-centered design; agile development

INTRODUCTION
In this section, we briefly describe software design architecture from both monolithic and microservice-based perspectives. After presenting relevant state-of-the-art technology components for a cloud solution, we then discuss the possibility of integrating these components to create a non-intrusive technology experience.

Monolithic Software Architecture
From Cloves et al. (2016), we understand that since the early days of the internet until 2010, cloud solutions were mostly monolithic in nature. This meant that solutions typically consisted of a traditional client-server setup and computation was carried out in a centralized manner. In its essence, the monolithic software architecture relied on database and computation logic programmed on the server, and solutions delivered to client-originated requests. The idea of a monolithic architecture can be attributed to any solution that is not decoupled, modularized, and decentralized. This could include a standalone web server serving requests to a browser client, or even a mobile application that provides a service without interacting with an external component.

Microservice Software Architecture
From Dragoni et al. (2017) and Killalea et al. (2016), we can deduce that the microservice software architecture, in its essence, is a distributed software architecture which consists of various smaller components or modules that cohesively work together to solve a bigger problem. The attributes of such a system are:

1. Each module solves a single problem and solves it very well. This means that it is responsible for its own behaviour and functionality. Therefore, is becomes easier to test each module independently.
2. Modules are loosely coupled, meaning their functions are not related to each other and, therefore, do not impact each other.
3. The sum of the modules is greater than their individual contributions. This means that integration of various modules has potential to solve a bigger and more complicated problem.

Bleeding-edge technology
Bleeding-edge technology is usually experimental in nature, and can take time to mature. Over the course of the last two years, few development frameworks representing cutting-edge technology from a cloud solution perspective have started to mature. Facebook’s React Native (Paul et al, 2016) framework has made cross-platform mobile application development much easier, Google Cloud (Krishnan et al, 2015) has offered a plethora of Big Data cloud-based services to run machine learning algorithms and backend automation. Firebase, acquired by Google, has provided a real-time database service that maintains data consistency across multiple devices. Furthermore, the Low-Energy Bluetooth button Flic by Flic.io has made home automation via smartphone much more effortless. (Sourced from documentation on Firebase and Flic websites, 2017)
Cloud service -The Google Cloud
The Google Cloud is a system of cloud solutions powered by Google technology that offer a wide range of services. As a developer, one can get to leverage Google technology to process API requests and focus only on the business logic of the application without worrying about server downtimes, database configuration, and other DevOps issues. (Krishnan et al, 2015, p. 121)

React Native
Summarizing the description by Paul et al. (2016) on page 4, powered by Facebook technology and their patented Virtual DOM technique, React Native comprises an underlying rendering engine that uses Javascript code to render native elements of the platform on which the application is being used. Thus, mobile applications that interact with external API services and provide mobile solutions can work seamlessly on both iOS and Android devices when built with the React Native framework.

Firebase
Powered by Google technology, Firebase provides the leverage to focus only on the logic of data flow. It takes care of features such as data handling, database processing, real-time data delivery, and user authentication. The firebase code can be embedded on both server side logic and mobile applications, thus avoiding duplicated database copies while still operating in a mutually exclusive manner. (Source: Firebase website, 2017)

Flic Bluetooth button
Released by Shortcut labs, the Flic.io button is a smart button that operates on Bluetooth Low Energy that can connect to a user’s smartphone and automate tasks, such as making a call, recording a speech, and carry out a series of tasks. Furthermore, Flic also supports a plethora of third-party module integrations. (Source: Flic website, 2017)

Integrating bleeding-edge technology components
To minimize the intrusive nature of using a mobile phone during the walking meeting, we intend to use a Flic button to record the proceedings of the meeting as audio files into a mobile application running on React Native. These audio files can then be sent via an API interface to a backend service running on Google Cloud, which in turn can produce transcripts of the meeting. The Firebase database can act as a data-bridge between backend and front-end components, thus maintaining data-integrity. This combination, a distributed architecture, could thus result in a seamless user experience and be implemented to reduce user-distraction while letting them focus on the goals of the meeting and still enjoy the benefits of a wholesome walking experience. Therefore, this is an exciting time to push the boundaries of distributed software architecture and try to find solutions to problems that monolithic architectures could not solve.

THEORY / RELATED WORK
In this section, we discuss current work being done in the field of non-intrusive technology following which, we present how microservice-based architecture can be more beneficial in providing non-intrusive technology.

Choosing non-intrusive technology
Schraefel (2015) discusses how knowledge work – an activity that primarily involves applying mental abilities into work – and physical activity such as sports can have a profound effect on the cognitive ability of an individual. He introduces us to the following notion on how design factors can influence our cognitive activity: ‘Evidence from physiology and neurology shows unequivocally that when we connect the brain and the body, our cognitive performance improves.’ (p. 34)

According to the book “Human Walking in Virtual Environments” by Steinicke et al. (2013), the authors mention walking as a physical activity that has a complex set of biomechanical processes associated with it. These processes are carried out by the brain based on cognitive perceptions. In Chapter 6, the authors suggest that to fully experience the benefits of walking, one must allow the sensory perceptions to operate in cohesion. Suchman L. (1987), one of the first scholars to address the relationship and interactions between humans and machines with a focus on the idea of human-like machines mentioned in her research: ‘Because of the asymmetry of user and machine, interface design is less a project of simulating human communication than of engineering alternatives to interaction’s situated properties.’ (Suchman, 1987, p. 185)

In a research conducted by Fernaeus et al (2008), one of the themes addressed is a subjective interpretation of technology components by users. The authors suggest a design pattern to provide the flexibility to interpret and manipulate technological components based on the context of the situation. This in turn would complement views of Suchman (1987) and help bridge the cognitive parts of the human brain and the technology component, thus enabling a human-like machine centric approach instead of vice-versa. Furthermore, this phenomenon could in turn lead to a higher retention effect and lesser hindrance to the brain’s reception of the concerned technology component.

Walking Metro
To solve the problem of non-intrusive technology in walking meetings, Ahtinen et al. (2016) conducted an explorative study based on user prospects towards walking meetings and designed a mobile application called Walking Metro. The results of their study showed that designing a ubiquitous solution involved three primary components:

1. Design for acceptability – This was critical to making sure the technology used was visually appealing, intuitive, and fostered a higher retention rate.

2
2. Non-interruptive guidance – Users should not be overwhelmed using the component and thus get distracted away from the goals of the meeting. This in turn meant that user interaction was supposed to be both seamless and still guide the user towards fulfilling the meeting objectives.

3. Discreet persuasion and stimulation – This meant that users expected some sort of incentive to motivate them to continue using the application. The incentives could be based on data collected from the meeting – steps covered, ability to share the walking path, etc.

**Meeting to GO**
As a part of the “Med Rörelse i Tankarna” project, according to the design-research conducted by Tobiasson et al. (2014) in participatory walking meetings, the subjects felt excited by the inclusion of technology in aiding walking meetings. The technology solution was an Android mobile application called Meeting to GO. Following user feedback, the next iteration needed lesser dependence of the mobile phone and a speech-to-text feature to substitute note-taking during the meeting.

**Monolithic architecture in existing solutions**
The mobile solution developed by Ahtinen et al. (2016) was a mobile application with possibly a web service as a backend, while Tobiasson et al. (2014) developed a native mobile application. To truly create a non-intrusive technology experience, the technology solution should comprise of many components besides a mobile application. Even if the mobile application interacts with an external server, the arrangement would still fit within a “semi-monolithic” architecture implementation. Therefore, in this paper, we aim to develop the next iteration of the Meeting to GO application as a microservice cloud solution using bleeding-edge technology components.

**METHODS**
In this section, we present our methods that led to the design and development of the prototype.

**User-Centered Design**
“System performance is a result of particular users doing particular tasks (using a particular technology) in a particular context. When you are thinking about designing interactive systems, you will need to consider all of these aspects together.” (Ritter et al., 2014, Chapter 3, p. 74). Therefore, it was important to design the architecture keeping the end users in mind.

**User Experience and Experience Design**
“User experience and experience design focus on and foreground the users’ feelings, emotions, values, and their immediate and delayed responses.” (Ritter et al., 2014, Chapter 2, p. 44). Therefore, to understand the mindset of the end user, techniques such as Role-play, Method acting, and Wizard of Oz were followed. Furthermore, active discussions with the researcher in the project and user input gained from work done on the project prior to this thesis were instrumental in the iterative design and development of the prototype.

**Interactive Devices**
“Touch is usually regarded as the third most important sense after vision and hearing” (Ritter et al., Chapter 3, p. 62). As a result, we explored various use cases of the Flic button to identify whether it was convenient to use as an interactive device and also was an apt input mechanism from the user into the system.

**Agile Development**
“A fundamental principle of agile methodologies is to welcome late change as one cannot know in advance what the system under development should be like and how it should be implemented.” (Cockton et al., 2014, chapter 9, p. 208). The agile development process was adopted as, initially, it was not clear how non-intrusive the technology was expected to be. It is important to mention that prior to work this project, technical expertise existed only in the Google Cloud services Google App Engine and Google Compute Engine. Therefore, additional software components and expertise were added to the prototype as the agile process evolved.

**Adapting the system based on user needs**
De Vito Dabbs et al (2009) developed an interactive health technology by involving patients in the development process to understand their needs better. Inspired by this approach, we developed features within the prototype so user needs and expectations could be met.

**Extreme Programming**
Chamberlain et al (2006) used Extreme Programming technique to develop a “web-based system where the novelty with the project was that consideration of security requirements were deferred until functionality was complete” (Chamberlain et al, 2006, p. 154). Taking inspiration from this approach, we decided to implement minimalistic versions of individual components in the prototype and perform user tests on the non-functional version. Following user feedback, we added functionality and developed the working prototype by extending the minimal versions.

Therefore, by following a User-Centred Agile approach, we developed a working prototype while adhering to The Brundtland definition of sustainability as “development that meets the needs of the present without compromising the ability of future generations to meet their own needs” (Missimer et al., 2017, p. 43)
RESULTS
As the processes of user-centered design and iterative agile development were intertwined in this study, we present the combined results of both approaches.

Table 1 summarizes the steps taken where we explored user expectations, iteratively designed the prototype, and evaluated user experiences based on the prototype design, which helped finalize the development using the user-centered agile approach. The aim of this approach was to design and develop a working prototype based on interaction with the user study group that already existed within this project. The first phase (P1) involved gathering information to understand the context. Discussions during this phase included understanding who the potential users were, how they would interact with prototype, and in what environment the prototype would be used. Following the discussion, features and requirements of the prototype were decided. The second phase (P2) included an iterative design of the initial version of the prototype. In this phase, based on chapters 5 and 6 from Ritter et al. (2014), care was taken to ensure that the user focused on the goals of the meeting instead of getting overwhelmed with the technology. The architectural design evolved with continuous interaction and discussion with the researcher in the project, and a user-test performed at the backyard of the KTH campus. In the third phase (P3), reviews from six users of the study group in Umeå were gathered and analyzed. The users included an expert in Interaction Design, engineering and PhD students, and potential enterprise customers of the prototype from medical and sports fields who were already aware of this research. The fourth phase (P4) included the final design and development of the working prototype. During this phase, focus was on meeting the initial requirements of the prototype without diluting the result.

<table>
<thead>
<tr>
<th>Phase</th>
<th>Method</th>
<th>Focus</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1. Expectations and features of a cloud solution for walking meetings</td>
<td>User-centric discursive approach</td>
<td>Explore the technical requirements and expectations of potential users during walking meetings.</td>
</tr>
<tr>
<td>P2. Designing the architecture of the cloud solution prototype</td>
<td>Iterative Design</td>
<td>Design the workflow based on user needs from P1, without implementing its functional components.</td>
</tr>
<tr>
<td>P3. Experiences based on the first version of the prototype</td>
<td>Semi-structured interviews</td>
<td>Evaluate user meeting experiences based on the design from P2.</td>
</tr>
<tr>
<td>P4. Finalizing the architecture and second version of the prototype</td>
<td>Iterative design and development</td>
<td>Based on user feedback from P3, finalize the missing components to develop a working prototype.</td>
</tr>
</tbody>
</table>

Table 1. User-Centered Agile Development

The following sections summarize the results of our user-centered design and agile development of the Meeting to GO prototype based on data collected after interaction with the research group and potential users of the cloud solution.

Expectation and features of the Cloud solution for walking meetings (P1)
In this section, we present the prototype requirements following discussions on earlier versions of Meeting to GO, the vision of the research group, functional features, and technical considerations needed to architect the cloud solution. This is the first phase of the design and development process. The working prototype needed to include a web postal and a mobile application. It was expected to comprise the following features:

Web portal
1. Here, users could create and manage meetings. They could specify various properties of the meeting such as its time, duration and location.
2. Users could add other users of the system to be participants of the meeting.
3. The homepage had to consist of a list of meetings that users could subscribe to / unsubscribe from.

Mobile application
1. Here, users could login and view meetings which they had created or were participating in.
2. The mobile application could record media files including audio, video and images.

Flic button
1. The Flic button could be used as an interactive tool when recording media. Single-click, double-click and long-press features of the button could also be explored for additional functionality.
2. During the meeting, users could toggle on and off voice recording using the single-click feature of the Flic button.

Meeting analysis
1. A background processing system was needed to analyze the meeting based on data recorded using the flic button and mobile application.
2. The minutes of the meeting had to be generated via a speech-to-text mechanism and displayed on a dashboard on the web portal so the creator of the meeting could view them.
3. Additional features of the processing system also included location tracking and biometric activity such as steps walked. This was inspired from Ahtinen et al. (2016) observation on motivating users to continue using the application.
**Designing the architecture of the Cloud solution (P2)**

This is the second phase of the design and development process. An overview of the system architecture designed is shown in figure 1. The architecture evolved following continuous interaction and discussion with the researcher in this project, simulating walking meetings to understand the extent of detachment from the mobile device and convenience of the Flic button while keeping in mind the meeting goals.

**Front-end component of the Web portal**

We developed the front-end component of the web portal using the React web framework (Horton et al, 2016). Firebase was integrated to implement user authentication namely user-signup, login, and password recovery. Data flow on the web interface was implemented using the Redux (Paul et al, 2016, p. 160) software design pattern.

**Back-end component of the Web portal**

We developed the back end of the web portal using Google App Engine, which is a Platform as a Service (PaaS) solution from Google Cloud (Krishnan et al, 2015, Chapter 5). The web framework used was Webapp2, while the rest of the backend components included a Memcached service and a queue service (Task Queue) to process tasks on the background.

**Mobile application and Flic button**

The mobile application was built with React Native and had a very basic layout that showed the meetings created on the web portal. A flic button was configured to record voice and save it into the mobile device as an audio file.

**Focus on usability**

The first version of the prototype was simplistic in nature and did not cover a wide variety of features. However, there were certain placeholders that were meant to guide the user through various stages of the non-intrusive technology experience. At this stage, our goal was to see if our initial design provided a technology experience which was simple, non-intrusive, and did not require a huge learning curve for the users while walking during the meeting. We experimented our design with a short user test at the backyard of the KTH campus.

**Evaluation of user experiences based on the first version of the prototype (P3)**

This is the third phase of the design and development process. In this section, we document the experiences of a study group in Umeå consisting of six users from different backgrounds as they used the first version of the design. The users were already familiar with the research as they had been involved in earlier test scenarios with the research team.

An indoor walk between 5 to 10 minutes was conducted with each user along a predetermined path, with the subject of the meeting being their agenda for the Easter holidays. During the meeting, they were asked to use the Flic button as an interaction tool instead of a smartphone, that could stimulate the start and stop of a voice-recording action. Following the walk, the users were presented with a semi-structured interview containing open-ended questions highlighting their experience, suggestions, and what they expected from the outcome of the walk.
Non-intrusiveness of the technology
Amongst three males specialized in interaction design, two of them above 40 years of age and another 20-25 years old, the general thought on the technology was that using a Flic button was a welcome move as an alternative to using the smartphone during the meeting.

‘Yes, this is exactly what I had imagined’ – P3-U5 (Research and School Coordinator, Sports Education, female, 40+)

‘We all know that technology at standard meetings is still crappy and unstable – we joke about the projector, adapters that are missing. You change in a way the behavior of meetings’ – P3-U2 (PhD student, Interaction Design, male, 40+)

Sources of controlling the Flic button
A recurring theme of discussion was whether there was a single or multiple sources for recording the conversation.

‘Should a microphone be provided to all? Maybe there are room for misunderstandings if everybody has just one flic and smartphone’ – P3-U5 (Research and School Coordinator, Sports Education, female, 40+)

‘The difference if all attendees has the system or if only the boss, or the manager of the meeting’ – P3-U4 (PhD student, Public Health and Clinical Medicine, female, 40+)

Trusting the technology
Acceptance of the technology depends on the familiarity and of the participants with this kind of technology and how secure they feel in having their speeches recorded.

‘Trusting the technology may depend on the background experience – tech or not tech’ – P3-U1 (Designer, Interaction Design, male, 50+)

‘Having the record option only in the hand of one individual is in a way providing a power imbalance’ – P3-U2 (PhD student, Interaction Design, male, 40+)

Recording while walking
There were some interesting views on the way recording was performed while walking.

‘Do not want to stop and talk – it gave me an impression that it is hampering the conversation.’ – P3-U2 (PhD student, Interaction Design, male, 40+)

‘Maybe one should stop when recording something of importance’ – P3-U5 (Research and School Coordinator, Sports Education, female, 40+)

Feedback mechanism while using the Flic button
When using the Flic button to toggle the on/off recording feature, users did not know if the recording had started or not.

‘Need some feedback maybe through vibration’ – P3-U2 (PhD, Interaction Design, male, 40+)
The Flic button as a wearable
There were some perspectives on how the Flic button could be used as a wearable.

'The button is small enough to get lost. It might be designed as a ring like Lord of Rings’ – P3-U2 (PhD, Interaction Design, male, 40+)

‘Design as a wristband or pen, or a place where it can be stored, like a MeetingtoGO-box’ – P3-U4 (PhD student, Public Health and Clinical Medicine, female, 40+)

Preparing for the meeting
Users mentioned it was important to prepare beforehand for the meeting and that some of them would be keen on knowing its specifics

‘What do I need to do before the meeting? Create a profile, invite attendees to the meeting, name the meeting, a plan and a discussion where to go/walk are needed’ – (PhD student, Public Health and Clinical Medicine, female, 40+)

Access to meeting analytics
There were questions on how the minutes of the meetings could be recorded and accessed.

‘I use paper and pen to take notes. Minutes are taking after it has been said’ – P3-U3 (Student, Computer Science, male, 20+)

‘The text that is generated might benefit from being colored as to mark-up sections of higher importance’ – P3-U2 (PhD student, Interaction Design, male, 40+)

‘How do I get access to the audio-files?’ – P3-U5 (Research and School Coordinator, Sports Education, female, 40+)

Potential benefits of the application
There were varied views on the potential benefits of the solution, beyond just as a tool for walking meetings.

‘See potential user domain such as rehabilitation, business, personal development coaching’ – P3-U4 (PhD student, Public Health and Clinical Medicine, female, 40+)

‘Is this made only for one task at the time one task at one walking meeting?’ – P3-U3 (Student, Computer Science, male, 20+)

@Figure 4: Back-end service workflow
Finalizing the architecture and second version of the prototype (P4)
This is the final phase of the design and development process.

The back-end service
Figure 4 shows the process in which the Backend service worked on Google Cloud. A combination of services such as PubSub, Compute Engine, and Speech API were used to run the internal engine that converted the speech recorded during the meeting to text. PubSub was used as a notification mechanism between the Google services whenever data processing had to be automated. The audio files recorded on the mobile device using flic were stored into Google Cloud Storage via Firebase. Following this, an AJAX request was sent to the api service via HTTP with the user-id, meeting-id, and file-id. Once the api service received this information, it triggers PubSub (A Google cloud service for message-queuing and notifications) which had a watcher script configured on Google Compute Engine (Krishnan et al., 2015, Chapter 4). This watcher script fetched the file from Google Cloud Storage based on the AJAX parameters and produced a transcript of the audio file using Google Speech to Text service. The transcript was then saved in the user database. Another HTTP AJAX request was sent to the API service informing the transcript had been saved. This finally triggered an email sent to the moderator of the meeting (for which the transcript was produced).

The Android mobile application
The mobile application was built using React Native for the Android platform. It shared the same Firebase real-time data as that on the website. This arrangement ensured data-integrity across all platforms, since there was just a single copy of the database. The mobile application comprised the following features:

1. User profile, Signup and Login - Authentication was implemented using Firebase npm package.
2. Creating a meeting – Users could create a meeting and specify the meeting title, duration, and description
3. Meeting chat rooms – Users could now discuss meeting specific requirements and decide locations of the meeting. In case of any changes, these changes could be communicated in the chat thread.
4. Timeline – The home screen contained a “timeline” of meetings along with the option to join them if the user had not added themselves to those meetings yet.
5. Record meeting – Conversations during a meeting could be recorded via the Flic button. Furthermore, additional features were implemented such as location tracking and biometric logging via steps walked during the meeting.
6. List of meetings created and joined
7. Sync recorded files – Audio file recorded and saved via the Flic button could be synced to the cloud so the transcripts of the meeting could be generated.
8. Notify users when specifics of a meeting have changed.
This was implemented with OneSignal.

DISCUSSION
The design and development of the working prototype followed a user-centered agile approach. The prototype evolved systematically during the four phases as a product conception, design iteration, user feedback, and iterative development of the working version. In this section, we first discuss our user-centered agile approach, following which, we break down our encounters, design choices, and development challenges faced as the prototype evolved. We conclude the section by discussing the ethics and sustainability in our work and mentioning future work on how the working prototype could be extended further.

User-centered Agile Approach
Using principles learned from Cockton et al. (2016), to guide the design process, information from users and researcher in the project was gathered through continuous interactions with the researcher, short user tests, and an extended user test with a selected user group.

Understanding the culture
To conceptualize the prototype design, it was important to understand the mindset of the end user. To achieve this, techniques such as role-play and method acting were followed with the researcher to understand the culture of walking meetings (Cockton et al., 2016, Chapter 2) in the form of walking, discussing, sketching functionalities and receiving information of user-involvement that had taken place prior to this thesis work. Additional information was gained by simulating walking meetings, a short user test at the backyard of the KTH campus, and experimenting with microphone usage on single and multiple participants (Ritter et al., 2014, Chapter 1, pp 23-24).

Observing and learning
To further develop the prototype with the focus of meeting the user expectations, the initial version of the design was then tested out on selected users via a talk-and-walk-trough activity. During this activity, the focus was to explore functionalities and gather detailed feed-back in relation to the design (Cockton et al., 2016, Chapter 3). After each walk, a seated meeting took place where the user was encouraged and given an opportunity to share their experience of the system and what they thought should be further developed or changed so they could use the system within their everyday practice. This was done by acting it out on the prototype – for example, one of the users showed how she imagined the interactive button to be integrated in a bracelet. Positive reviews by the users while using the Flic button provided the confidence that our choice of Flic
was apt according to Chapter 3 by Ritter et al. (2014) in recording user input, thereby moving towards the correct direction in providing a non-intrusive technology experience.

**Customized Features and Iterative Development**

In our quest to customize the users’ needs within the context of the walking meeting (De Vito Dabbs et al, 2009), we received valuable insight on further development of the prototype based on ideas gained during discussion with the user group. This further motivated us to focus on developing features on the mobile application using Extreme Programming technique (Chamberlain et al., 2006) that would also help keep the users engaged (Ahtinen et al, 2016). Prior to the user tests, minimal functionality was implemented on the web front-end, backend service, and mobile application along with a simple input mechanism from Flic button to the mobile device before the user tests. Following user feedback, we iteratively added features to the minimalistic versions and connected them via Google PubSub, thus completing development of the entire working prototype.

**Conception**

Based on the discussions with the research group during the early stages, it was evident that the prototype had to include various components to bring a truly non-intrusive technology experience. This called for consideration of various technology components:

1. Wearables or similar technology components – In our quest to find an alternative to using a mobile device, we thought the Flic button was ideal, as it was not only intuitive to use but also easy to integrate during application development. From their documentation, it was straightforward to retrieve recorded flic audio files from a third-party application source code.
2. Mobile application framework – It was a logical to choose React Native as the framework to develop the mobile application as we could use the same source code (with minor tweaks) for both iOS and Android.
3. Back-end framework – We decided to go with Google Cloud over Amazon Web Services (AWS) as we could leverage many of Google’s cloud features such as Compute Engine and App Engine.
4. Speech-to-text transcription engine – We has initially decided to implement the transcription engine using IBM Watson instead of Google Speech as we felt sending the audio files to the Watson API service directly from the mobile device was a straightforward process.
5. Web front-end framework – After carefully considering Angular 2, we decided to use the React framework as we wanted to implement the data flow using the Redux (Paul et al, 2016, p. 160) software design pattern and felt it would lead to optimized performance when coupled with React’s Virtual DOM rendering technique.

**Architecture design**

Based on user experience and user input, the architecture for the cloud solution was designed with the following idea. To minimize the cumbersome use of a smartphone, we could use a Flic button to provide user input to a mobile application running on React Native and record voice, take pictures, and even film videos. These media files could then be sent to Google Cloud via an API interface, which in turn talked to IBM Watson and produced transcripts of the meeting. The Firebase database could act as a data-bridge between server and client components, thus maintaining data-integrity. This could further drive the meeting goals by providing real-time analytics of the meeting while the mobile application and the server exchanged data via the API interface. This combination, a distributed architecture, could thus result in a seamless user experience and be implemented to reduce user-distraction while letting them focus on the goals of the meeting and still enjoy the benefits of a wholesome walking experience.

**Feature implementation considerations**

For the user studies as part of P3, we had a few basic features ready. We got valuable insight on further development of the prototype based on ideas gained during discussion with the users. Positive reviews by the users while using the Flic button provided the confidence that our choice of Flic was apt according to Chapter 3 by Ritter et al. (2014) in recording user input, thereby moving towards the correct direction in providing a non-intrusive technology experience. This further motivated us to focus on developing features that would not only keep the users engaged, but also provide valuable content within the context of the walking meeting.

**Challenges faced during development**

In this section, we discuss the critical challenges encountered while developing the cloud solution.

**Updating meeting specifics**

While tracking location, there were ideas of drawing the path in real time on a map. This would have made it possible to view the walking path directly on the mobile device as the meeting progressed. However, React-Native’s implementation of the Map view was different for iOS and Android. Whereas this was a direct implementation on iOS, it was a complex process in Android and had a longer implementation estimate. Therefore, we decided to not show the walking path in real time on both devices.

**Speech-to-Text**

Initially, IBM Watson API service was considered for Speech-to-Text transcription and planned to be linked to Google App Engine via a REST API. However, the
prediction model from IBM needed training, which further hampered the testing process, and called for an already trained speech transcription service. In March 2017, Google opened a lot of their cloud services to developers, including the Google Speech API. Towards late April, the network architecture for Meeting to GO was restructured so that internal components such as Compute Engine and App Engine could be integrated via PubSub.

Limitation of Flic recording on the iOS platform
Due to security considerations for iOS development, Flic recording on iPhone has significantly restricted the testing on iOS using the microphone. Therefore, the Meeting to GO application mainly focused on Android. It is important to point out that majority of employees at corporations use iPhones instead of Android phones. This implies that making the Flic button usable on the iOS platform would have meant a wider reach of the user base and greater room for future feedback. Keeping this in mind, to achieve maximum spread of users across both platforms, a provision has been made within the mobile application to trigger the microphone internally on the iPhone. This provision is implemented alongside the functions that track both location and steps taken during the meeting. Technical implementation of this feature can be reserved for future work.

Audio format conversion from 3GP to FLAC
The audio files saved from Flic were in 3GP multimedia format which was not supported by Google Speech API. Therefore, we decided to convert the audio file to FLAC format which was not only recognized but also the recommended format by Google. We leveraged the cloud conversion service “CloudConvert” to convert the saved file from Google Storage and inform our API service on App Engine when the conversion was complete. After conversion, the new file location was sent to the API service, which triggered a PubSub notification that in turn prompted Compute Engine to fetch the converted file, produce the transcription with the Speech API, and then save the text transcript into Firebase under the corresponding meeting which the audio file was linked to.

Benefits of choosing the FLIC button
While considering providing a truly non-intrusive experience, it was important to have not only an autonomous system, but also a simple mechanism to register user input.

Technical benefits
Initial consideration included an Internet of Things (IoT) based solution such as a Raspberry Pi implementation to track motion-sensing or gesture-recognition. After concluding such a solution was going to take longer to implement, we decided to use the Flic button. As discussed in the section under Conception, this decision proved to be effective as it not only quickened the pace of mobile development, but also proved to be an almost equally cost-effective solution when compared to the Raspberry Pi implementation.

Figure 5. Usability of the Flic button

Usability benefits
Based on user feedback, we observed that the Flic button was ideal to be used as a device for user-interaction, due to its soft texture. It felt as a more convenient alternative to a smartphone device. The fact that it could also be packaged as a wearable made the Flic button more beneficial from a usability standpoint, thus adhering to Chapter 3 by Ritter et al. (2014).

Ethics
From a software development perspective, the working prototype is an integration of various tools and services that are openly available to the developer community. Therefore, it makes sense to share the lessons learned during the development phases and open-source the code, so interested research groups could use similar technology to explore other interesting integrations. From a software security perspective, care has been taken to develop the mobile application to preserve integrity and confidentiality of users. Seeing that the prototype has potential to evolve as a standalone application for personal motivation, care has been taken to ensure data protection following Google Security and Compliance (2017). The source code has been written keeping in mind the best software development practices. While open source boilerplate code was used as a starting framework for both web and mobile versions, the underlying logic is an authentic attempt at using Flux architecture for the web front-end and Singleton architecture for the mobile application. Although the prototype is limited in its functionalities for iOS, the choice of React Native has resulted in providing an equal experience opportunity on both iOS and Android platforms.

Sustainability
The microservice architecture of the working prototype enables sustainability (Missimer et al, 2017) in three important areas.
**Economic Sustainability**

1. Efficiency of resources – By using React Native to develop the mobile application, the cost to hire and manage native mobile developers individually for Android and iOS platforms is greatly reduced (Paul et al, 2016, page 6, paragraph 1; Why Companies Should Consider React Native, 2017). This in turn would mean opportunity for deeper user-reachability, additional user feedback, and more qualitative bug-fixes at the cost of a single React Native developer. However, it is important to mention that for testing the application, the appropriate devices on the platforms are necessary. Furthermore, for text-to-speech, communication between Google Cloud and CloudConvert during audio file conversion takes place only when necessary (CloudConvert and Google Cloud, 2017). Techniques such as long polling and web sockets which can consume time and resources have been replaced by the publish-subscribe technique (Rivera et al, 2014; Pub/Sub Benefits, 2017).

2. Optimal maintenance cost – A Flic button costs $35 (Flic, 2017), React Native development is free (open source), while Google Cloud and Firebase costs are charged on a Pay-Per-Use basis (Google Cloud Pricing, 2017), which roughly sums up to $20 per month if used extensively. Therefore, with the help of open source documentation, free Software Development Kits (SDK) such as Google App Engine SDK, React Native SDK, and free Integrated Development Environments (IDE) such as Android Studio, the entire system can be maintained with a one-time cost of $35 and a monthly budget of $20.

3. Optimal replacement cost – Various components of the prototype have been chosen such that replacing one of them would not influence the rest of the system. For example, the Flic button could be replaced by a VR headset, yet the rest of the cloud components could function intact, thus saving cost to modify the entire system.

**Ecological Sustainability**

1. Environment-friendly – Since data is processed and saved on Google’s data centers which will become 100% environment friendly in 2017 (Environmental Report – Google, 2016), the technology that powers our working prototype would not contribute to pollution, wastage, or degradation of the environment.

2. Minimal consumption of battery, storage, and power – As the prototype has been engineered to automate most of the processing between the vicinity and the cloud, it prevents battery draining and memory consumption on the mobile device.

**Social Sustainability**

1. Free flow of body movement and better health – The non-intrusiveness of the technology enables users to enjoy the benefits of walking movements (Swedish National Institute of Public Health, 2010; Tobiasson et al., 2012; World Health Organization, 2014).

2. Active participation – Using the prototype while walking contributes to both creative and social aspects of the meeting (Vallance et al, 2011; Marily et al, 2014)

3. Physical reliance on the Flic button – the prototype relies on physically manipulating the flic button with actions such as single-click, double-click and hold. This would prevent it from being usable by users with disabilities, which further diminishes its reachability. An alternative could be to use cognitive computing – the technology that drives Google Home and Amazon Echo to provide user input to the system. (Ebling, 2016, page 5)

**Future work**

With the current version of the working prototype functional, we believe the following steps would help take this project to subsequently better levels.

**Longer User Tests**

Outside the scope of this thesis, a longer user testing has been planned to take place. Initially at the time of the thesis proposal a second thesis student was expected to co-work on the topic. However, that student selected another topic. During the thesis proposal, the technology had not been explored and as user expectations for the functionalities was high, the work in this thesis focused quite strongly on iterative development due to the experimental nature of the bleeding-edge technology components.

**Exploring the market for more potential users**

As there has been an interest amongst potential enterprise customers based on the prototype it has potential to spin off as a full-fledged product for use within companies. Considering this, it would also make sense to release the mobile application on the Appstore and PlayStore to a larger audience. This would be essential to understand the mass appeal of the prototype and whether it would make sense to develop additional features.

**Extending the development**

Due to time constraints, it was not possible to implement the recording feature for the iPhone via the FLIC button. Therefore, from a development perspective, it would make sense to include this feature so we could enable equal use cases for both iOS and Android users. Since tests are essential to validate any software solution, we plan on
writing additional unit and functional tests to improve the robustness of the mobile application. We also intend to explore addons to support user inputs with alternatives to the Flic button, such as custom IoT solutions built on Raspberry Pi and even machine-learning or cognitive computing technology that powers Amazon Echo and Google Home. It would also make sense to explore integration with Google solutions such as Cloud DataFlow and Javascript technologies such as WebAssembly (WASM, 2017).

CONCLUSION
It is possible although risky to integrate bleeding-edge technology components to solve complex problems. There is a chance that they might not be compatible. To make them compatible, we might have to go through many hurdles, and even after crossing the hurdles, the product might have been diluted or complicated that it doesn't meet our initial requirements. While most of our components initially chosen remained the same, the Speech to text engine nearly was a failure. We were fortunate to have found CloudConvert that made it possible to have our audio files converted, so the transcription process could be integrated into our design without affecting the rest of the architecture. The working prototype continuously evolved throughout the process of the user-centered design and agile development approach in which lessons were learned after overcoming development challenges and interacting with potential users of the cloud solution. As with any agile development approach, the prototype needs to be evaluated by potential users. This final step has not been thoroughly accomplished in this thesis due to time constraints. However, we have been able to integrate bleeding-edge technology components to create a non-intrusive technology experience within a walking meeting scenario. Thus, we show that now is a great time to push the boundaries of cloud computing and try to create truly non-intrusive technology to solve complex problems. Therefore, from an application perspective, the solution has the potential to be used at both enterprise-level scenarios and self-help apps (motivational, goal-setting, etc). On that note, we are happy to share our source code with anybody interested. We encourage further exploration in this direction and interaction with each other when developing such solutions, hoping that this would be a small step towards creating a more wide and open design and development community.

ACKNOWLEDGEMENT
I would like to take this opportunity to thank my supervisor Helena Tobiasson for having provided the vision for this project during the initial phases. I would also like to thank AFA Försäkring for providing the research grant for this project.

REFERENCES


De Vito Dabbs, A; Myers, B; McCurry, K; Dunbar-Jacob, J; Hawkins, R; Begey, A; Dew, M; (2009) User-Centered Design and Interactive Health Technologies for Patients. Computers Informatics Nursing. 2009, 27: 1-16. 10.1097/NCN.0b013e3181920769


Pub/Sub Benefits, Retrieved on April 21, 2017 from https://cloud.google.com/pubsub/docs/overview#benefits


