



FACULTY OF INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING

**Jilin Yang**

**FEASIBILITY AND PERFORMANCE ANALYSIS OF  
MIDDLEWARE SUPPORT FOR A SITUATED VIRTUAL-  
PHYSICAL CIVIC ENGAGEMENT PLATFORM**

Master's Thesis  
Degree Programme in Computer Science and Engineering  
May 2017

**Yang J. (2017) Feasibility and performance analysis of middleware support for a situated virtual-physical civic engagement platform.** University of Oulu, Degree Programme in Computer Science and Engineering. Master's Thesis, 44 p.

## **ABSTRACT**

**With the prevalent ubiquitous computing technologies, it is possible to explore novel solutions for supporting civic engagement as a set of urban practices. One interesting urban practice is the soapbox, traditionally conceived as wooden structure, from where to hold impromptu speeches. For this thesis, a novel soapbox prototype with ubiquitous computing mediated technologies is introduced, with our focus on the feasibility and performance analysis of its middleware support, investigating how our middleware is able to meet the goals of a situated virtual-physical civic engagement platform. Based on our empirical evaluations, it is demonstrated that our prototype is effective to support civic engagement and serve purpose of continuously soapbox streaming.**

**Keywords: civic engagement, soapbox, ubiquitous computing, middleware**

## TABLE OF CONTENTS

ABSTRACT .....	2
TABLE OF CONTENTS .....	3
FOREWORD.....	4
ABBREVIATIONS .....	5
1. INTRODUCTION .....	6
1.1. Research questions and objectives .....	7
1.2. Thesis structure.....	7
2. URBAN COMPUTING AND CIVIC ENGAGEMENT .....	9
2.1. Urban computing .....	10
2.2. Urban informatics.....	12
2.3. Urban practices.....	12
3. DESIGN AND IMPLEMENTATION .....	14
3.1. Requirements.....	14
3.2. Technical solutions.....	15
3.2.1. Streaming .....	15
3.2.2. Messaging.....	16
3.2.3. Database storage.....	17
3.2.4. Reference systems .....	17
3.3. Design architecture.....	20
3.4. Metadata schemas.....	22
3.5. Middleware implementation.....	22
3.5.1. Client API.....	22
3.5.2. Middleware server.....	23
3.5.3. Middleware data storage .....	23
3.6. Middleware supported flows .....	23
3.6.1. Physical soapbox speech .....	24
3.6.2. Virtual soapbox speech .....	26
3.6.3. Hotspot feed streaming.....	27
3.6.4. Mobile devices interaction .....	28
3.6.5. Persistent storage.....	29
3.6.6. Other flows .....	29
4. EVALUATION .....	30
4.1. Lab evaluation .....	30
4.2. Field evaluation .....	31
4.3. Setup.....	31
4.4. Data .....	32
5. DISCUSSIONS AND FUTURE WORK.....	34
5.1. Original research questions .....	34
5.2. Reaching of objectives .....	35
5.3. Limitations.....	36
5.4. Future work .....	36
6. CONCLUSIONS .....	37
7. REFERENCES.....	38
8. APPENDICES.....	42

## FOREWORD

It's been a long journey.

I would like to express my sincere gratitude for the Centre of Ubiquitous Computing research group for providing me all the necessary hardware and software during the process of completing my thesis. Meanwhile, it's an honor that Professor Jukka Riekkö provided the possibility for me to have prior exchange year and further recommended me to apply for the Ubiquitous Computing theme IMP programme.

In the very beginning of introducing the joint hybrid soapbox project, I insisted on choosing the most challenging but also interesting topic – middleware support, so-called “middle man” when we joked the person who needs to fill it. It's been remarkable experience for me to advent such ride on ubiquitous computing technologies, aiming at solving the civic engagement research questions with the cutting-edge hybrid reality conceptualizations.

I'd like to thank sincerely to the beloved of mine, for her encouragement during my low time writing the thesis. Also, I'd like to send also my greatest gratitude to my parents, who are reminding me of what and how I should proceed when I was struggling to write more on thesis.

Last but not least, I'd like to express my gratitude to my supervisor Dr. Marko Jurmu who marked numerous meetings with me to clarify how we organize the thesis and guide me through a lot of hassles, while also Docent Hannu Kukka for his professional guidance especially in early development stage.

It's worth pointing out that Kai Wang, Haejong Dong, and me are in a joint team to collaborate in this joint soapbox project, and it's been a very great opportunity for me to finalize a prototype such practical which will be impossible without their sharing of thoughts.

It's been a long journey.

Oulu, 23.05.2017

Jilin Yang

## **ABBREVIATIONS**

API	Application Programming Interface
HDD	hard disk drive
HTML5	HyperText Markup Language
JSON	JavaScript Object Notation
NoSQL	Non relational structured query language
RAM	random-access memory
RESTful	REST: Representational state transfer
RTT	round-trip time
SMS	short message service
SOAP	Simple Object Access Protocol
UbiComp	ubiquitous computing
W3C	World Wide Web Consortium

# 1. INTRODUCTION

What will be your first instinct when one citizen wants to raise his opinion against a public spotlight? Answer is probably certain social media or self-media, instead of directly advocating in the streets. But once in a while, diverse means of engaging in city matters exists including the wooden soapbox originated from 19<sup>th</sup> century famously known in the Speaker's Corner, Hyde Park, London, where orators lead citizens to engage in broad public discussions and mind sharing, which a peak is shown in Figure 1 (reprinted with permissions from GeorgeLouis<sup>1</sup>) as follows.

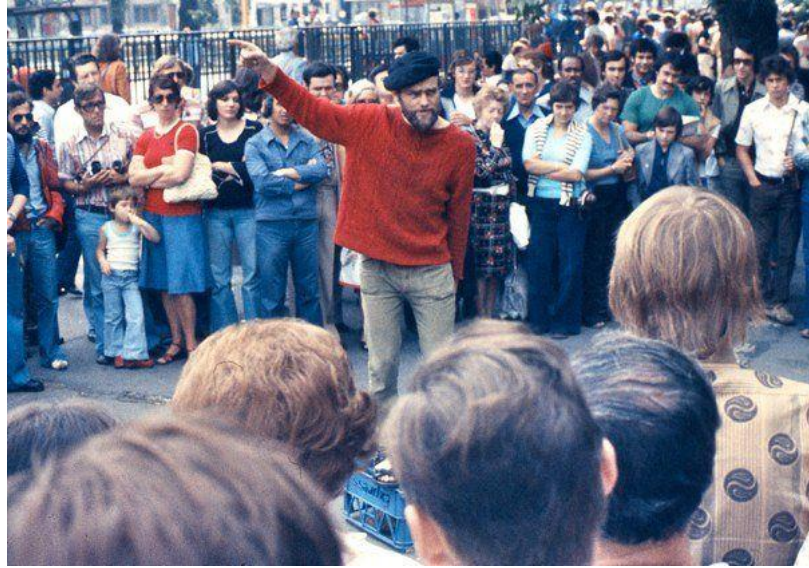


Figure 1. Orator at Speakers' Corner in Hyde Park, London, 1974.

As a typical civic engagement platform, traditional wooden soapbox is characterized such that it provides straightforward access to the in-situ crowd while optionally disclosing their identity. Even though only those who happen to be within the hearing distance at the same time of the speech can be reached, whereas some people from that crowd can choose not to pay attention. However, such advantage has been even more overwhelmingly obscured when technologies are growing so fast that citizens tend to raise their voices more often online surfing internet.

What if the traditional soapbox can be transformed into another type of medium which is more accessible to internet, more welcome for citizens with user-friendly interfaces? In terms of internet accessibility, traditional soapbox performs quite outdated, while investigating the current city circumstances, it's feasible and scalable way to reach other citizens and decision makers.

On the other hand, with increasing sensors deployed among streets of the city, various city-context data are collected and aggregated into central servers, which generates and pipelines continuous rich content back to subscribing users who are interested. Such so-called urban computing phenomena provisions a novel angle of how civic engagement can benefit from traditional soapbox, i.e. realizing soapbox with urban computing technologies. In urban computing aspects, people holding intelligent devices are given with extremely simple and easy entrance into virtual

---

<sup>1</sup> Speakers' Corner. URL: [https://en.wikipedia.org/wiki/Speakers%27\\_Corner/](https://en.wikipedia.org/wiki/Speakers%27_Corner/). Accessed: 2017-05-29.

online world. The concept of “hybrid reality” has come to denote the simultaneous presence in the physical world as well as in different online platforms through personal devices with always-on Internet connectivity. When they focus on the devices such as smartphones, they dive into virtual online world, which is fast spreading and independent of geological boundaries. But once they take their eyes off the screens, they seamlessly and effortlessly switch back to the real reality.

Therefore, we come up with a new soapbox prototype described as a situated virtual-physical civic engagement platform, which is technology-mediated combining urban computing and civic engagement. The platform prototype brings a novel angle to tackle the problem of inadequate effectiveness among current civic engagement platforms with the help of ubiquitous computing technologies, urban computing technologies and urban informatics.

### 1.1. Research questions and objectives

The objective of this thesis is to design, implement and assess middleware functionalities for a novel soapbox prototype to be more effective in civic engagement with technologies of urban computing and urban informatics, while the focus of assessment is the feasibility and performance of such middleware functionalities. The feasibility refers to two aspects: setting up speeches should not take significant time, and that streams can be multi-casted to a sufficiently large number of receivers.

With our objectives, this thesis aims for two research questions:

**Research Question 1 (RQ1):** What kind of middleware functionality is required to support technologically-mediated civic engagement within urban computing and informatics?

**Research Question 2 (RQ2):** What is the impact of the designed middleware to the technologically-mediated civic engagement prototype in terms of performance?

RQ1 will be assessed by conducting technology selection against the prototype system requirements, and by deciding where to utilize available off-the-shelf technologies and where to implement additional functionality. RQ2 will be assessed by subjecting the civic engagement prototype along with the designed middleware to both a lab evaluation as well as a small-scale field evaluation.

The contributions of this thesis mainly comprise of three aspects: First, it designs and specifies a middleware client API for different clients of the soapbox system. Second, it designs and implements a middleware server that connects and coordinates different entities of the Soapbox system. Third, it designs and implements an abstraction layer for a NoSQL database for persistent storage and management of speeches and their associated metadata. Besides, the soapbox system is evaluated and analyzed in one featured publication [35] which demonstrates its usability for civic engagement.

### 1.2. Thesis structure

The thesis is structured as follows. The first chapter introduces the background of our platform prototype and the issues it aims to solve; second chapter illustrates details of our prototype’s core concepts and each corresponding domain’s related work; third chapter focuses on design and implementation of our prototype, as well as how its

technical solutions are determined; fourth chapter describes the feasibility and performance of our prototype with evaluation data and concludes to what extent our prototype fulfills its dedicated purposes.



## 2. URBAN COMPUTING AND CIVIC ENGAGEMENT

This chapter dates to the origins and background of urban computing and urban informatics as technologies introduced in our objective and research questions. Further it presents the related work of these technologies and explains how this thesis can apply them to tackle its research questions.

Since 1988, ubiquitous computing has been brought up as a means to mediate people's lives. The term ***ubiquitous computing*** is coined by Mark Weiser stating that it seamlessly integrates computation, including information displays, into everyday physical world, which entitles ubiquitous computing the capability of accessing physical world from the very beginning [14]. From Weiser's envisioned storytale of Sal's daily life back in 1991, it's been illustrated how ubiquitous computing can allow computers to vanish into background and thus connects people to physical world seamlessly just like another spectacular invisible technology, writing, according to Weiser's metaphor. And therefore, information overload could be overcome, since he proposed ubiquitous computers can try to fit the human environments instead of vice versa. Along with Weiser's consistent idea of keeping ubiquitous computing refreshing for people and helping resolving information overload, it's no surprise that the calm technology age is suggested by Weiser in 1996, approximately 10 years after ubiquitous computing is named. Weiser states that after undertaking mainframe era, PC era, we will encounter UC (Ubiquitous Computing) era via transition of the internet and distributed computing, though they are currently still dominant components composing the giant virtual world.

However, 10 years later with constant research in context of ubiquitous computing "just around the corner", its proximate future is argued by Genevieve [15] that we should not be misled by Weiser's vision assuming ubiquitous computing era is just coming. With two distinct practical examples of Singapore (sensor ubiquity) and Korea (infrastructure ubiquity), it's argued that ubiquitous computing has already been in our reality, and the rest of it may not be embodied as Weiser's vision. Though Weiser in [17] does not claim the universality of calm technologies in fields such as an excited video game, Yvonne still reinforces that moving on from Weiser's vision, we could seek for engaging ubiquitous computing experiences as well, rather than solely pursuing calm technologies as ultimate goals of ubiquitous computing [16].

Regarding to engaging ubiquitous computing experiences located in urban spaces, the deployment of ubiquitous computing principles into urban spaces gives rise to a contemporary area of research and practice called urban computing. Meanwhile, urban informatics technologies fit in the overlapping zone with urban computing collecting all the urban information from urban spaces, which leads to various sorts of applied urban research domains. When considering the contexts of interactions happened within domain of urban informatics, there is one unusual interaction in a new form of reality: hybrid reality, resulting from physical reality and virtual reality by ubiquitous computing. The term ***hybrid reality*** means people are being always online connected to social media platform, emails, personal always-on devices, which compose a digital world where everyone is both producing and consuming data. In such hybrid reality, or so-called dual reality, often a virtual world and a physical world sensed by human-centric sensor network are mutually reflecting and merging which act as a novel medium [39] comparing with traditional multi-media such as television and radio.

On the other hand, *civic engagement* as one of the urban research domains, quoted from American Psychological Association, stands for “individual and collective actions designed to identify and address issues of public concern”. As one of the focused and applied areas in terms of cross-research of urban informatics and civic engagement platforms, information technologies undoubtedly play a crucial role in civic engagement. There are example systems blending urban computing, urban informatics with civic engagement such as the InstaBooth [20, 34] portable community engagement platform, which facilitates engagement and discussions for urban design or civic proposals. Via the InstaBooth shown as in Figure 2 (reprinted with permissions from [34]) as follows with inspiration from telephone booth, visitors are able to note down their feedback or draw pictures while interacting with media. The concepts of how urban computing impact civic engagement are illustrated in depth in the following sections.



Figure 2. The InstaBooth Photo Credit: Xavier Ho.

## 2.1. Urban computing

*Urban computing* acquires, integrates and analyzes big and heterogeneous data sets generated from urban sources consisting sensors, buildings, human beings [1]. It aims at solving major issues which cities encounter, such as traffic, logistics, security or decision-making. As quoted from Mark Weiser “Applications are of course the whole point of ubiquitous computing”, and this same thinking also applies to urban computing.

There are some reference systems developed with technologies of urban computing. For instance, an Instant Places system [5] shown in Figure 3 (reprinted with permissions from [5]) as follows in an attempt to experiment the how effective and what practices Bluetooth presence and naming is able to prompt interaction around public displays. Their system collects Bluetooth device names and instructs user-defined tags especially explicit Flickr username tags, and therefore collectively visualize labels on an in-situ large public display. With a semipublic campus bar trial in University of Minho, they found that technologies of Bluetooth naming can entice situated public interactions, and creative usage of the system is discovered by people with relatively small amounts of instruction. It indicates that urban computing associated with place and content is a good catalyst for enhancing people's public engagement with the help of public displays.

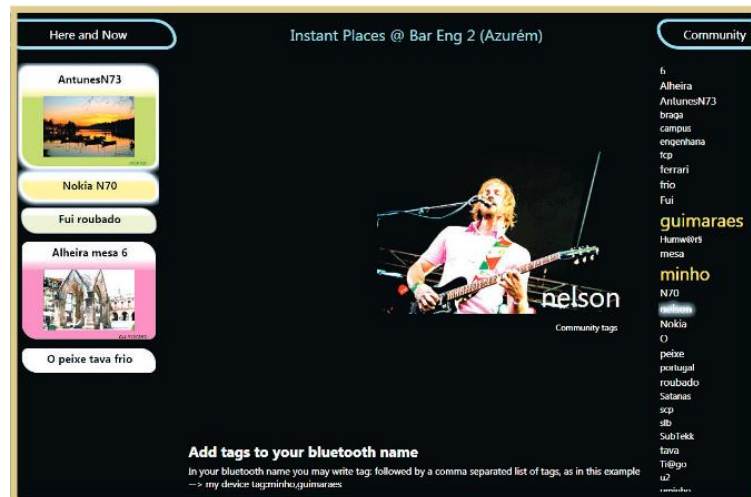


Figure 3. A type of display visualization in trial phase.

As mentioned by Vinny et al. [6], system software with ubiquitous computing has brought challenges mainly because it needs to handle heterogeneous and volatile data set surrounded by dynamic and unpredictable environments. It indicates that in data storage layer of ubiquitous computing system software, a flexible database solution will be more preferable.

What's more, Discussion in Space (DiS) system shown in Figure 4 (reprinted with permissions from [7]) as follows introduced by Ronald et al. [7] also explores the possibility of digital augmentation impacting the civic participation, with facilitation including civic discussion and opinion forum via a large public screen, deployed in Brisbane city. Utilizing their system, users can be either passerby interacting with mobile phone SMS or Bluetooth, or the remote with computers. One of its initiatives is to strengthen the decision-making basis for urban planners and engage with residences especially those who are hard to reach. With real-world deployments, they propose a way to identify the sweet spot of such civic engagement systems in [13] affecting quality of feedback and frequency of interaction. It is also concluded that the interfaces should be simple, and richer functionalities should be employed with discretions in order not to heighten the entry barriers.



Figure 4. Public screen screenshot.

## 2.2. Urban informatics

Though similar as urban computing, but what comes a step more forward towards participatory culture is *urban informatics*. Since introduced by Foth, et al. (2011), urban informatics is a multidisciplinary domain situated at the intersection of notions, trends and considerations for place, technology, and people in urban environments. Grounding in both ubiquitous computing and human computer interaction technologies, urban informatics referring to Foth's explanation, is slightly shifting the focus from the hardware aspects of the computing environment to the softer aspects, including exchanging information, communication and interaction, social networks and human knowledge mentioned [34].

When urban practices are targeted, some essential design and implementation elements need to be considered to be leveraged by urban informatics. First is mobile devices and sensing as one trending of UbiComp research, as concluded by Yong [4]. Therefore, we can visualize that mobile device as one of the mainstream UbiComp keyword, highly correlates to urban informatics since it involves people and technology, as well as place if considering the geographical context of the mobile device users. Secondly, in urban spaces, public displays are one of the rapidly increasing resources which undertake multiple sorts of civic needs, including commercial advertisement, publicize government announcement, or more general civic engagement. Therefore, it is beneficial to involve resourceful public interactive displays into implementation of civic engagement platforms. In order to investigate how public displays can stimulate civic engagement, Memarovic et al. [18] conceptualized the processes occurred around public displays (FunSquare system) into passive engagement zone, active engagement zone and discovery. From their study, stimulus of attracting people to discover the activities ongoing at public displays can come from people's topic of interest and locality reflected content, which exactly illustrate the nature prerequisites of civic engagement in public spaces.

## 2.3. Urban practices

Urban computing and urban informatics can be leveraged to re-design and influence existing urban practices, such as a use case of interactive opinion polling system shown in Figure 5 (reprinted with permissions from [41]). One example public urban displays of those urban practices is introduced by Baldauf et al. [41], on top of which are QR code scan and web applications as information technologies.

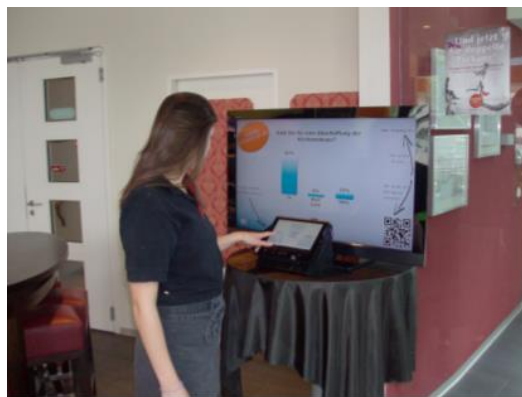


Figure 5. Interactive opinion polling system in the entrance area of a coffee house.

Public displays can be perceived as overlays to the physical world, according to the research of Müller et al. [31] about interactive public displays. Speech, body posture, and touch etc. are some of the interaction modalities supported by interactive public displays. In the Plasma Posters system by ChurchChill et al. [26] with interactive public displays, the gathered postures contribute to the success of the system to share multi-media content within their organization. However, public displays as one of the components in existing urban practices, how to design and achieve meaningful engagement with pervasive displays remains to have certain obstacles including interaction blindness and lack of participation motives as discussed by Hosio et al. [22]. In our thesis, adopting existing urban practices is our key to solving and ensuring effective attraction of such public displays usage. With sufficient hints from physical hardware deployment and web applications, the interactive features of public displays are made aware and accessible to the citizens.

Specifically, practice of giving speeches in urban environments, as one of civic engagement practices is our focus. With the technologies from urban computing and urban informatics, we want to influence such practice utilizing those frameworks from related work to design a prototype for this. We wanted to re-design the process of civic engagement through the applications consisting interactive public displays and virtual urban environments, which both are classic urban computing technologies.

One major feature of urban computing according to [1] is to integrate data from both virtual and physical worlds, which benefits our prototype in a way that brings the popularity of online social media to physical attendance for the citizens. While using urban computing to rethink how to design the system with more collective concerns rather than individual, such as a music sharing experimental study by [2], the system is designed to be situated. As internet inevitably has huge impact on civic engagement, in one way making it easier by providing accessible network entrance and bandwidth, but also in another way making it harder by isolating the focus of a crowd walking down the streets while they are simply surfing internet with their smart phones. Meanwhile, cities are equipped with more sensors, actuators, interactive dashboards with the promotion of urban computing technologies. Currently not only human beings but also cities are becoming more accustomed to feeding and reflecting with internet. Resulting from such growing comfort of interacting with virtual world from a physical world, the element of urban computing has to be considered while enhancing the traditional soapbox as a modern situated civic engagement platform.

Besides, the soapbox prototype system with speech and video streaming support is one possible answer to the question of what's next for UbiComp proposed by Abowd [23]. The prototype system is capable of facilitating 3-dimensional graphical user experience with ubiquitous computing technologies comparing with traditional 2-dimensional programming environments. This kind of platform is a novel technology-mediated way we can support civic engagement in urban environment, blending virtual and physical worlds. When it comes to the prototype, among all branches we focus on how to design and implement the middleware.

### 3. DESIGN AND IMPLEMENTATION

#### 3.1. Requirements

The soapbox platform prototype is mainly required to digitalize the traditional soapbox with technology mediated video speech streaming support. In addition to mobile devices as client-side technology for the audience, public interactive displays as one of the essential urban resources are involved for audience for implementing civic engagement platforms. It is noted regarding to the development infrastructure of this thesis, public interactive display cited and used is interchangeable to Ubi-hotspot, or hotspot for short, which will be introduced more in following setup section. Also, virtual soapbox environment is engaged to provision the possibilities to give speeches in either virtual or physical soapbox, which can act as mutually speaker or audience entities. More specifically for the middleware system among the platform prototype, there are five requirements listed as follows:

1. Real-time speech streaming, i.e. one-to-many multimedia broadcasting from soapbox to hotspot. The broadcasting process should be automatically handled by middleware server if certain pre-defined criteria are triggered.
2. Real-time speech feedback services, including like, speech and comment by text (i.e. a shoutbox). All information should be simultaneously updated in all connected application components.
3. Enable start-right-now speech and reserved speech services, with proper submitted speech meta-data from speakers. All data should be maintained in our project database.
4. A certain extent of fail-over recovery protection for soapbox sudden offline cases and also hotspot jamming cases. Soapbox should be able to reconnect online without human intervention effort, while hotspot should be resumed normal broadcasting by middleware server after short breaks.
5. Both physical soapbox web application and virtual soapbox web application should be able to use our middleware JavaScript API without conflicts. Coordination between them is realized in our middleware server.

For the joint soapbox setup requirements, there are mainly physical and virtual soapbox setups as joint efforts in addition to the middleware support as the focus of our thesis. In terms of physical soapbox setup which is discussed in [35] as shown in Figure 6 (reprinted with permissions from [35]) below, there is a key component for user detection, i.e. Wii board as weighing sensor to detect if speaker is ready. When user steps in the Wii board located in a separated soapbox broadcasting area, physical soapbox web application shown at a large screen in front of the user can be activated to serve into working mode. As for the virtual soapbox, the web application should be implemented as duplicate of physical soapbox, mimicking components such as virtual soapbox broadcasting area, virtual broadcasting screen, and virtual step-in zone for activation.

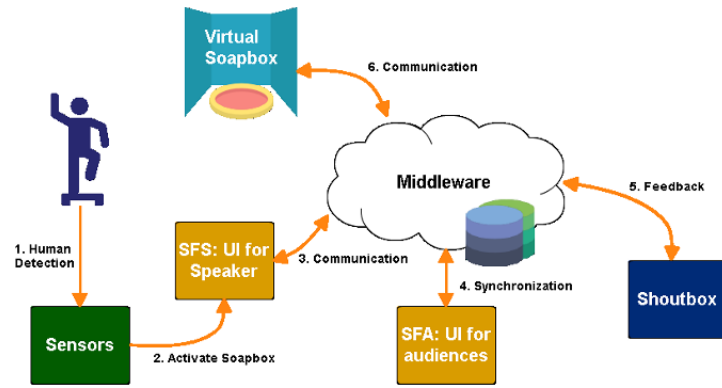


Figure 6. Example soapbox application architecture.

### 3.2. Technical solutions

We firstly identify three core components (streaming, messaging, database storage) in our middleware system and opt for the most optimal option among each one's solutions. Then we compare the complexity of tailoring the middleware part based on our requirements, and developing following design of current available ubiquitous computing frameworks. After discussions, how the middleware is implemented will be concluded regarding our system requirements.

#### 3.2.1. Streaming

For video streaming, there are a couple of options to be considered, however, the most promising standardized technology is Web Real-time Communication (WebRTC), which is initially an open source project announced by Google [8] and then further standardized by W3C. WebRTC is a specialized API definition which enables Real-Time Communication (RTC) capabilities, supporting multiple browsers and platforms such as Google, Firefox and Android. It is considered most promising and easy to use library for streaming multimedia (video and audio) in our project, because it is well documented and officially supported in mainstream browsers already wrapped in JavaScript.

In terms of WebRTC implementations (short as WebRTC in the following text), there are still choices of packaged streaming solutions available online, as well as custom implementations on top of base libraries. Integrated products such as Twilio and tokbox supporting commercial WebRTC features do not grant the flexibilities to easily integrate our messaging components. Wrapping libraries such as PeerJS and PubNub providing WebRTC functionalities have to some extent cumbersome calling procedures. OpenWebRTC from the perspective of an open-source base WebRTC implementation, has gained less popularity than the reference implementation by Google. Thus, Google WebRTC is chosen for its potential capabilities to combine our streaming and messaging contexts serving application components as part of our system's middleware. The example WebRTC architecture is shown in Figure 7 (reprinted with permissions from [32]) as follows.



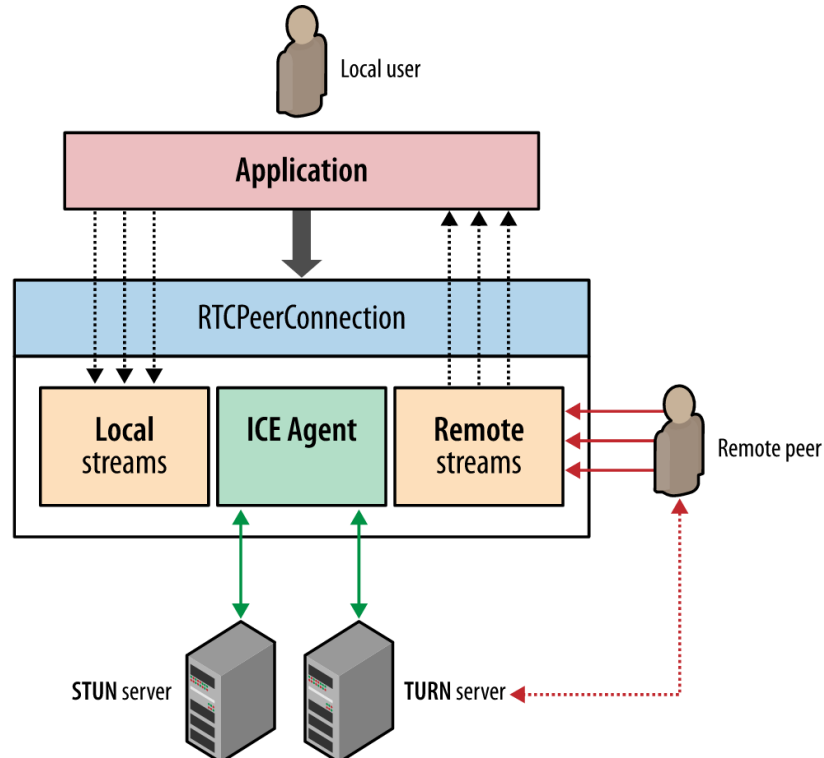


Figure 7. RTCPeerConnection API.

As an alternative, using *ffmpeg* libraries to slice the speech into multiple frames per second and use traditional image sharing technics to customize the multimedia streaming was once considered as fallback option in case WebRTC support is not mature enough in browsers. But as project proceeded, it was found that WebRTC well provides the streaming features with little latency and therefore set as the main means to implement multimedia streaming tasks.

However, the learning curve of extending and utilizing WebRTC should be also considered, as it requires some extent of understanding on the underlying protocols. These include Hypertext Transfer Protocol (HTTP), Session Description Protocol (SDP) and offer/answer mechanism. Additional terminology includes Peer, Signal channel, Session Traversal of User Datagram Protocol (UDP), Network Address Translators (NAT) server (STUN server), Traversal Using Relay NAT server (TURN server) and Interactive Connectivity Establishment (ICE). Also, the actual implementation of signaling server is left to developers for more efficiency and flexibilities, which requires more work to learn from the perspective of this work.

### 3.2.2. Messaging

For messaging, we need to mainly pick a proper application layer protocol implementation in order to develop in an agile manner. Though there are web messaging provided by HTML5 [11], WebSocket API standardized by W3C [12], and other diverse solutions with cross-platform capabilities, publish-subscribe messaging pattern is preferable because it ensures the scalability when flexible content schemas are needed.

Among all publish-subscribe implementations, RabbitMQ is one of the popular robust messaging mechanisms for applications [9]. It enables publish-subscribe



service smoothly in multiple languages including Python and JavaScript. With RabbitMQ, we can let soapbox act as a publisher, while hotspots and audience act as subscribers. For simplicity reasons, our middleware server will also act as a subscriber, observing all communication flows. We select *pika*, a pure-Python Advanced Message Queuing Protocol (AMQP) client library as the Python server side RabbitMQ implementation. For web messaging, Web-Stomp RabbitMQ plugin is utilized, exposing the Streaming Text Orientated Message Protocol (STOMP) protocol over direct or emulated HTML5 WebSockets. The main intention of Web-Stomp is to make it possible to use RabbitMQ from web browsers. It should be noticed that credentials from UbiOulu middleware are used and same exchange lying at the panOulu<sup>2</sup> network is listened.

### 3.2.3. Database storage

As discussed earlier, a NoSQL database solution will be favored for its advantage of dynamic schemas across the development and post-development process [38]. There are mainly document NoSQL including MongoDB and CouchDB, and key-value NoSQL such as Redis. Because our speech metadata and streaming signaling data will be semi-structured, it's a good practice to select document NoSQL. Moreover, comparing with CouchDB, MongoDB supports natively a JSON-like document structure, which will drastically ease the usage for application component sides, especially physical and virtual Soapbox web applications.

### 3.2.4. Reference systems

A Really Easy Displays (RED) framework [24, 25] aims to ease the development, deployment and management of multi-display applications, utilizes the pervasive nature of web DOM elements across browsers, and a Thing Broker concept originated from MAGIC Broker [28] to enable developing applications for heterogeneous display types and hardware. Four demo applications are presented with their RED framework as shown below in Figure 8 (reprinted with permissions from [25]).



Figure 8. Four RED applications: a music jukebox, a garden application, an image gallery, and a race simulation game.

The MAGIC Broker mentioned above is another middleware toolkit for developing interactive public displays applications. It is implemented by Aiman et al. [28] in the context of providing spontaneous and diverse patterns of interactions. The key aspects of such broker are about extending events abstractions beneficial from publish-subscribe mechanisms, and adopting RESTful web services protocol.

<sup>2</sup> Open Wireless Internet Access. URL: <https://www.panoulu.net/>. Accessed: 2017-02-05.

Despite the advantages that MAGIC Broker provisions such as the event flow shown below in Figure 9 (reprinted with permissions from [28]), it does not illustrate significantly more advantages than normal events broker. MAGIC Broker's support for event-based coordination remains unknown, resulting in the fact that it does not fit in our middleware focus which is broadcasting speeches while synchronizing context related interactive data.

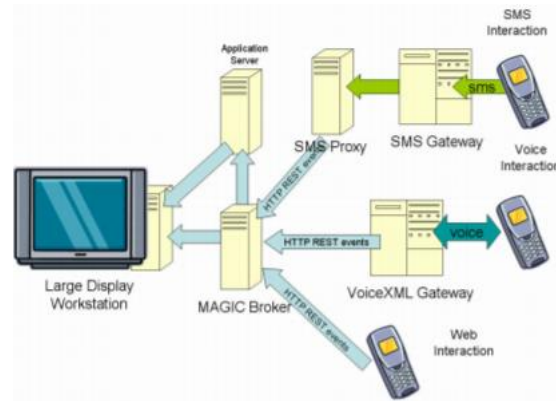


Figure 9. MAGIC Broker event flow.

Extended with MAGIC Broker II [29], this infrastructure reference system represents a such kind of focus on centrality and persistency of interaction states across multiple large public displays, while on the other hand, such focus can be management and scheduling of displayed content across multi-screens like e.g. e-Campus system by Storz et al. [33]. However, our middleware system will not focus on only either interaction state synchronization, or content scheduling, but aiming at broadcasting content in addition to synchronizing situated interactive metadata.

Another system proposed by Fabius et al. [19] is a local community polling system with tangible interactive urban screens as shown in Figure 10 and 11 (both reprinted with permissions from [19]). Polling questions are rendered in public displays and co-located people can vote with their feet by simply stepping on either side of agreeing or disagreeing the polling questions. Though in such system implementation, the control and management regarding public displays need to be integrated into civic engagement system, but there is little requirement on highly real-time data streaming or synchronization.



Figure 10. Vote With Your Feet urban screen at a bus stop.

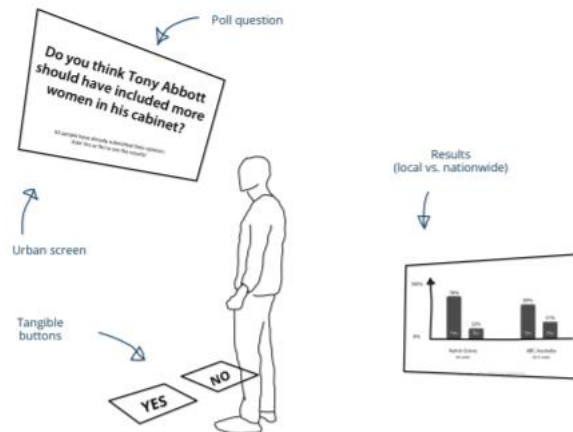


Figure 11. Vote With Your Feet concept sketch.

Similarly, the Mégaphone as Speakers' Corner proposed by Fortin et al. [27, 40] calls for bringing the art of public speaking back into cities (urban spaces) by utilizing public displays in public spaces. Mégaphone offers facilities for speakers to give speeches or perform artistic events, as shown in Figure 12 (reprinted with permissions from [40]). However, such system is limited by only digitally augmenting speakers' corner but without streaming the speeches to broader audience far from the site, let alone the lack of possibility for citizens' virtual attendance with the help of virtual reality technologies.

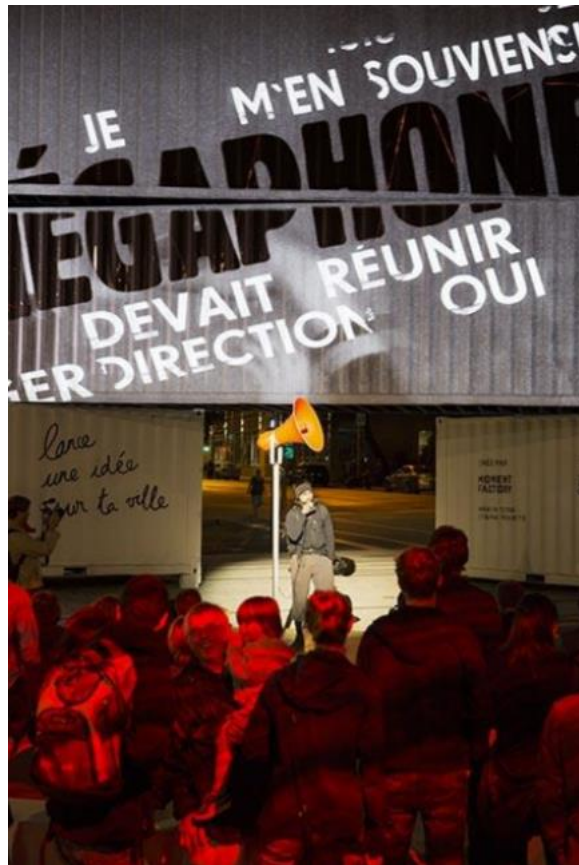


Figure 12. MÉGAPHONE's "Speaker's Corner" platform. ©2013MomentFactory / Photo: Moment Factory.

In terms of background systems or system frameworks, there are for instance middleware infrastructure Gaia proposed by Manuel et al. [10], allowing simultaneous slides presenting across multiple displays. Additionally, there are ubiquitous computing evaluation areas proposed by Scholtz and Sunny [21], including sample systems such as campus-aware system and personal interaction points system, which mainly aims at ubiquitous data and feature discrete interaction flows. These systems are restricted to the context of building or extending traditional ubiquitous computing systems, and as such are unable to fit into our focus on both streaming and messaging.

### 3.3. Design architecture

The design of our Soapbox middleware system includes defining metadata schemas, and specifying middleware requirements. Regarding the implementation, it consists of three parts: middleware client API, middleware server, and middleware data storage.

The architecture chart is shown in Figure 13 as below. During this thesis, middleware client APIs, and middleware server with embedded thin NoSQL database internal layer are developed, while message bus, WebRTC source libraries, ICE servers, NoSQL database are applied as off-the-shelf software. Besides, all relevant applications including physical soapbox, virtual soapbox, hotspot, and audience, are developed by joint researchers with assistance from this thesis.

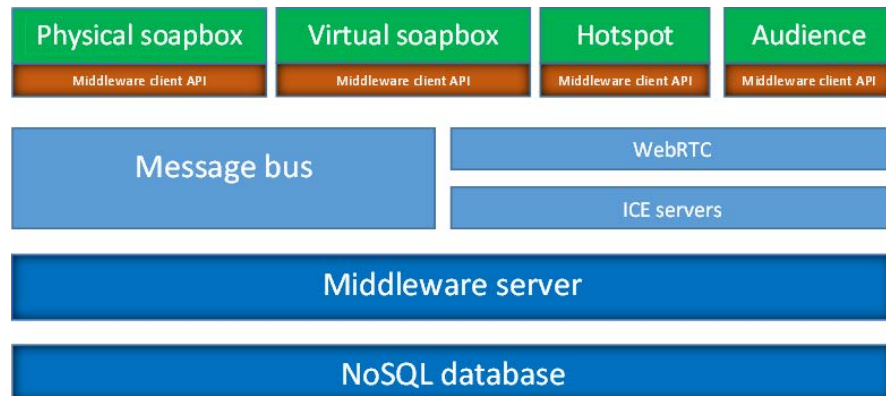


Figure 13. Architecture of the middleware system.

The typical procedure of initializing a speech streaming is shown as following Figure 14 below.

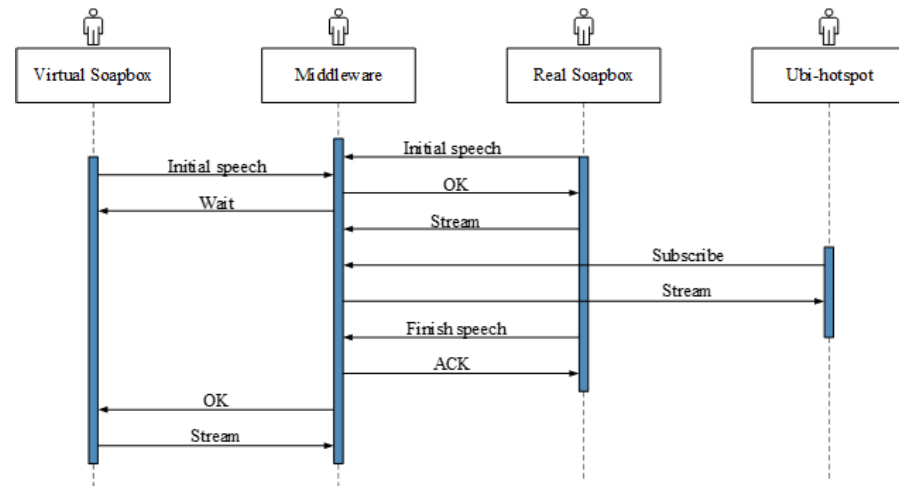


Figure 14. Speech streaming sequence diagram.

The actual work flow of the whole system is shown with following steps: The web application hosted in browser as caller and callee will both signal middleware server when they are registered to be online. Then when there is an actual speech scheduled to broadcast simultaneously, for one caller-callee speech transmission pair, the session descriptions will be exchanged via middleware server signaling interfaces for the purpose of WebRTC signaling. Moreover, there are two kinds of session descriptions: local and remote. Whenever remote session descriptions are exchanged, the premise of connecting media between caller-callee pair is established. Therefore, the pair will be able to start actual video streaming from point to point without any other proxy parties. The session descriptions employed above are the defined representative data in the JSEP offer-answer architecture which is also core principle in WebRTC technologies shown in Figure 15 as below.

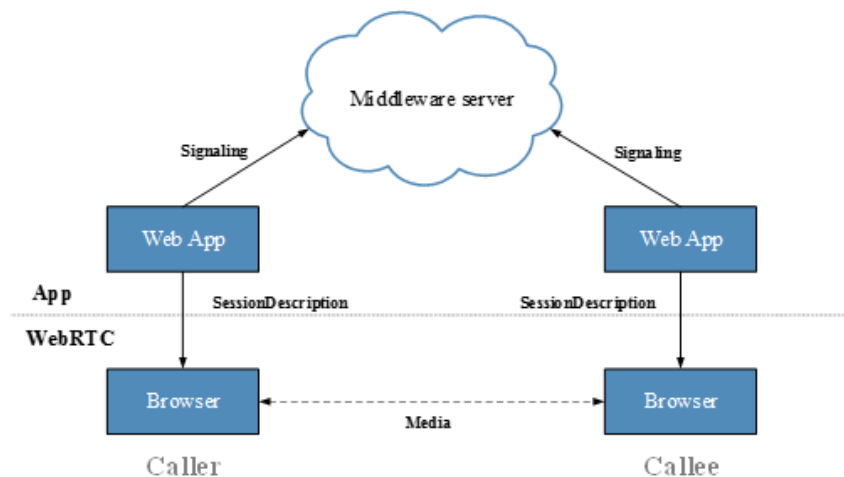


Figure 15. JavaScript Session Establishment Protocol (JSEP) offer-answer architecture.

### 3.4. Metadata schemas

There are four types of schemas defined on top of codebase. Below, we briefly introduce each one of these schemas.

In database schema, we are mainly interested in the speech registration information submitted by physical or virtual soapbox applications. Besides, timestamps and current status of each speech are also included. For the main speech document, reservation related metadata are stored providing the status of a speech and the submission information of a speech sent from actual speaker application sides.

For speech related likes, dislikes, reports, comments documents, especially *Sender* key is added for the purpose of coordinating the application logics in middleware server, which indirectly gains those messages from the message bus.

In speech registration schema, submission information provided by end users includes speaker's name, speaker's gender. Besides, reality type is added to notice middleware server about the type of speech application component (either physical or virtual). Also, length of this registered speech is inputted for expectation of speaker verification before the actual speech starts in physical Soapbox environment.

For WebRTC signaling schema, SDP data and unique index for connected party are combined together to represent the offer and answer data packages. In this way, the lower level SDP signaling for WebRTC meta-data exchange are bundled into higher level signaling managed by our middleware server, with the help of assigning unique indices to each instance of physical soapbox, hotspot or virtual soapbox. With corresponding index for the received party, all signaling messages can be routable without conflicting the sibling instances if any.

### 3.5. Middleware implementation

#### 3.5.1. Client API

The initiative of constructing middleware API is to provide a convenient means for both physical and virtual soapbox sides to communicate and enable speech broadcasting. As WebRTC strategy is chosen for supporting the speech broadcasting, and therefore, middleware API will be a JavaScript library wrapping WebRTC native API, as well as containing Web-STOMP callings for RabbitMQ.

There are four distinct roles in middleware API:

1. *Soapbox*: an object for handling physical soapbox caller. It enables video streaming from itself to all available hotspots. It receives current speech information, likes, dislikes, and comments. It only accepts input of likes and dislikes from physical soapbox speaker.
2. *Virtual*: an object for handling virtual soapbox caller. There are two sub-roles this object supports: virtual speaker and virtual audience. It enables both video streaming from itself to all available hotspots when it acts as virtual speaker, and from physical soapbox to itself when it acts as virtual audience. It receives current speech information, likes, dislikes and comments. It accepts input of likes, dislikes and comments from both roles.

3. *Hotspot*: an object for handling hotspot website caller. It enables video streaming from both physical and virtual soapboxes to itself. It receives current speech information, likes, dislikes and comments. It accepts input of likes and dislikes from actual users at hotspots.

4. *Audience*: an object for handling audience from mobile website caller. It receives current speech information, likes, dislikes and comments. It accepts input of likes, dislikes and comments from actual users who access the mobile website using mobile devices.

### **3.5.2. *Middleware server***

The reason why there will be a middleware server is mainly the lack of WebRTC signaling implementation and needs for bridging different roles of soapbox system. With the middleware server, meta-data of WebRTC can be exchanged using RabbitMQ's publish/subscribe mechanism. Moreover, metrics such as likes, dislikes, comments can be shared and synchronized across all roles as well as backend persistent database. In order to ease the controlling logics of middleware API, almost all of the decisions are made according to different conditions in middleware server.

Additionally, middleware server has control of the hotspots. Since all hotspots on production currently are hosting constantly UBI-interactive interfaces, which is deployed with a dedicated web server, the soapbox hotspot website cannot be simply pointed with full-screen mode at the hotspot when there is a speech ongoing. Therefore, the current solution is to make use of the configuration file at the hotspot web server and use SOAP wrapped API so as to make transitions between soapbox (full-screen) mode and normal UBI-interactive display mode.

### **3.5.3. *Middleware data storage***

In order to persist speech related metadata, a middleware data storage layer is implemented within the middleware server. After considering the flexibility of potential processed ubiquitous metadata mentioned in Chapter 2, a NoSQL database solution is adopted. In our case, MongoDB will be utilized. MongoDB is a NoSQL cross-platform document-oriented database solution using JSON-like documents with schemas, which is free and open-source. Middleware server will store all received data into MongoDB database instance locally or in the cloud, and retrieve all necessary data from it whenever a query is made.

## **3.6. *Middleware supported flows***

With our middleware client API and middleware server, there are several main work flows enabled for our prototype. By illustrating each of these supported work flow from the point of middleware's view, how each component of the whole soapbox prototype interacts with each other is clarified.

### **3.6.1. *Physical soapbox speech***

When a speaker walks in our physical soapbox setup (Wii board for user detection as discussed previously) and wants to start a speech, the sequence diagram is shown in Figure 16 as follows. It's noted that the role of virtual soapbox in this case is acting as a public display to receive speech streaming. It is ensured in virtual soapbox side that no new speech is attempted to start when there is already ongoing speech. Also, there is double-lock check in middleware server to ignore any new speech start request if there is any ongoing speech.

From the diagram, it can be visualized that the internal process (signaling before video streaming) of enabling WebRTC video streaming feature is quite complicated and intricate. The speech initializer and speech receiver both need to fill in their local SDP packet and configure to accept the remote counterpart's SDP packet, so as to fulfill the offer and answer of a WebRTC JSEP protocol. While normally it's the sender who acts as provider (i.e. so-called "offerer"), but with the middleware server coordinating all components with control of messaging, it's also possible for the sender to act as answerer which mediates the situation that the provider may not be aware of if the video streaming is streaming only or dual streaming. After the exchange of SDP packets, ICE candidates also need to be exchanged after the public addresses are resolved via STUN / TURN / ICE servers. The above exchanged data are transmitted via message bus messages, but also guided through the middleware server and client API so that the hassles of publishing and subscribing to message bus can be eliminated from the perspectives of the application layer.



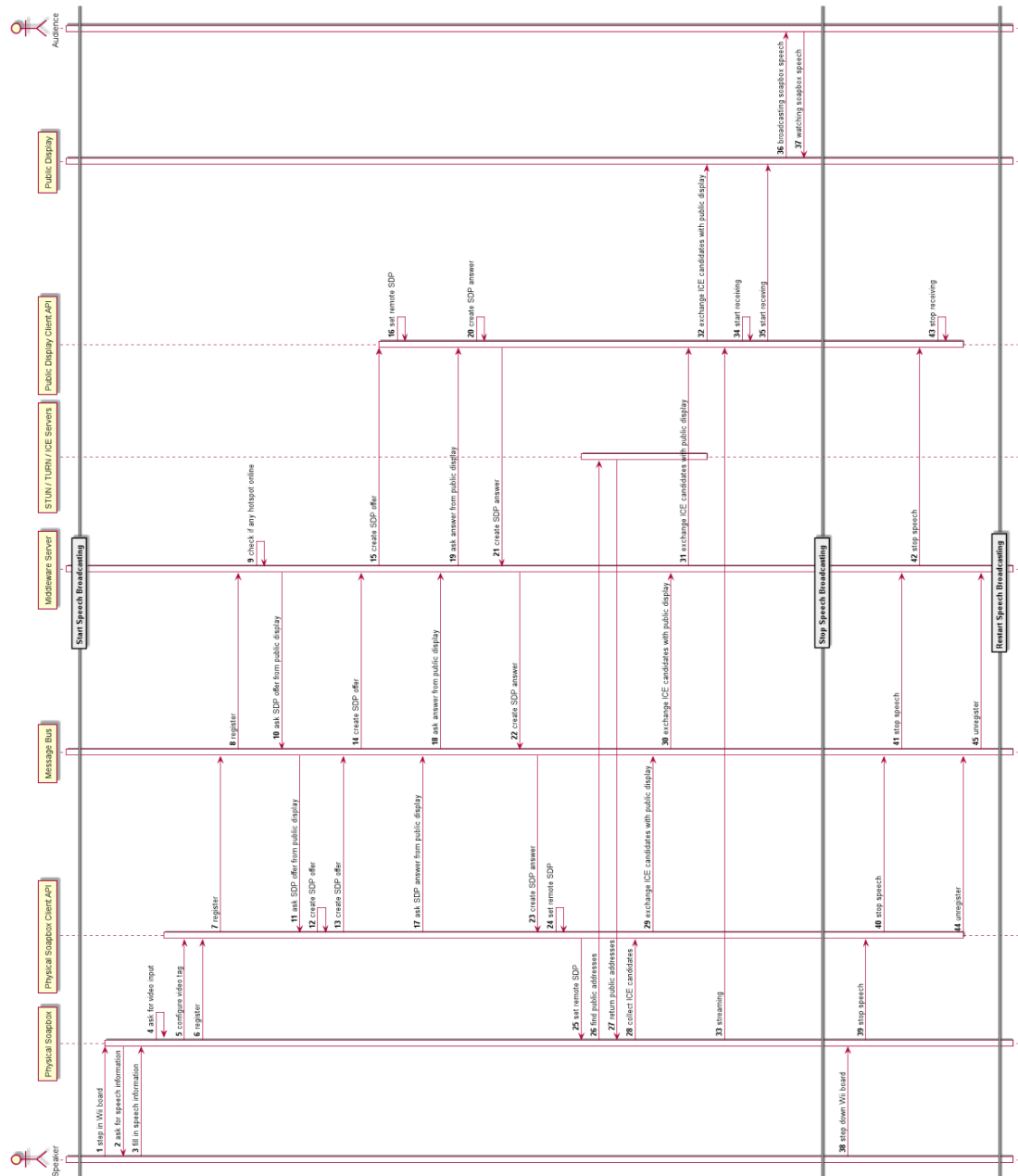


Figure 16. Physical soapbox starting a speech flow chart

Therefore, when physical soapbox represented in web application starts broadcasting a speech, there is no visible usage of messaging or WebRTC configurations. Instead the web application developer can simply configure the necessary video input and have easy to use APIs to manage the speech process.

Physical soapbox client API call looks more straightforward comparing to the inner process handled by middleware client API for physical soapbox and the middleware server signaling logics. Example code snippet is in Figure 17 as follows:

```

var soapbox = new Soapbox();
//Connect to the middleware(signaling server)
//Four params: onConnect, onError, onReceiveMessage, ConfigParams
soapbox.connect(function () {
    //onaddhotspotstream callback should be set before everything, when connected
    soapbox.onaddhotspotstream = function(event) {
        hotspotVideo.src = URL.createObjectURL(event.stream);
    };

    //Callback for knowing if the start action succeeds
    soapbox.onreceivestartfeedback = function(feedback) {
        //feedback is either "success", or "failure"
    };

    //Mandatory, register itself
    soapbox.register();

    //Start a new speech immediately
    soapbox.start(local_stream, speech_info);
});

//Try to tell middleware that it is about to close
window.onbeforeunload = function(event) {
    soapbox.stop();
};

```

Figure 17. Example code snippet of how to use client API in physical soapbox application

### 3.6.2. *Virtual soapbox speech*

When the speaker starts a speech in virtual soapbox (within virtual environment via computers), the work flow is similar as physical soapbox starting a speech as shown in the following Figure 18. The internal process of signaling video streaming has been simplified.

One major difference in this flow is that there is a unique virtual monitor role, which represents the feature that one virtual soapbox entity tries to stream the virtual environment via the virtual camera feed based on HTML5 video source configuration. In this way, the virtual monitor can provide a static bystander angle to the current virtual soapbox surrounding environment. The principles beneath the monitoring video feed of virtual monitor and virtual soapbox is the same as sender and receiver of WebRTC video streaming, or same as physical soapbox streaming to a public display screen.

### 3.6.3. Hotspot feed streaming

With the dual video streaming, the speaker can observe the facial reactions from the audience standing by the public displays which are broadcasting the speech in real time. Also, the feed shown in public display screens as well can better comfort the audience about how they will be observed and how they are interacting with the

speaker in real time. In fact, here will be multiple public displays than only one trying to connect with the soapbox when a speech is started. This is not shown in the chart for the sake of simplicity. In this case, the middleware system has the synchronization logics to make sure each public display completes the SDP and ICE data packets exchange with the soapbox before next WebRTC connection is initiated.

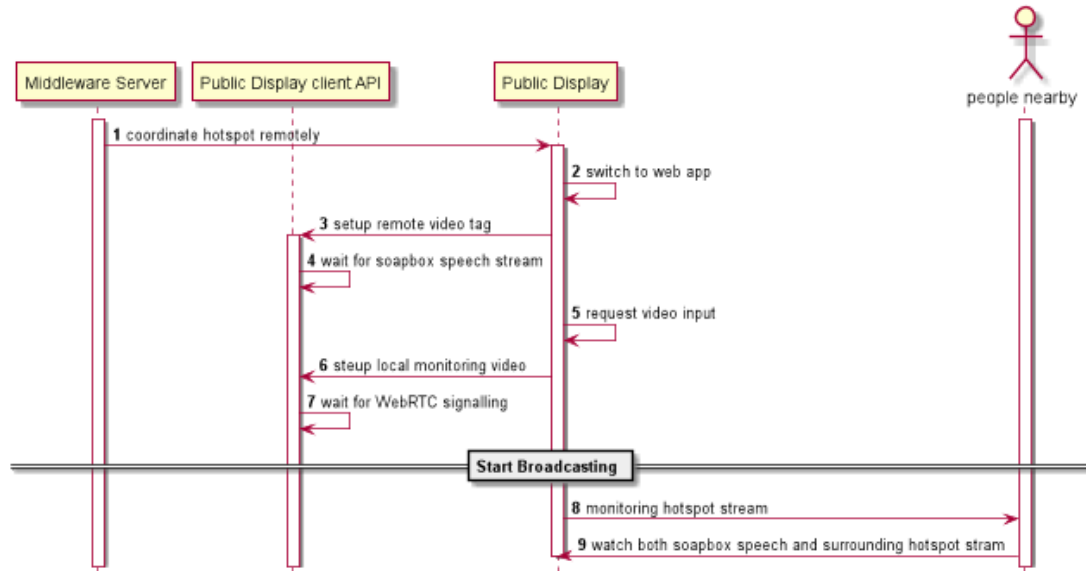


Figure 19. Hotspot feed streaming flow chart

### 3.6.4. Mobile devices interaction

When a speech is ongoing, audience can scan QR code of the soapbox or public display web applications to access the mobile web application for interacting with the speaker or audience. Users can like as shown in flow chart in Figure 20 below, or dislike, report, or comment (with one-time nickname and content) the speech.

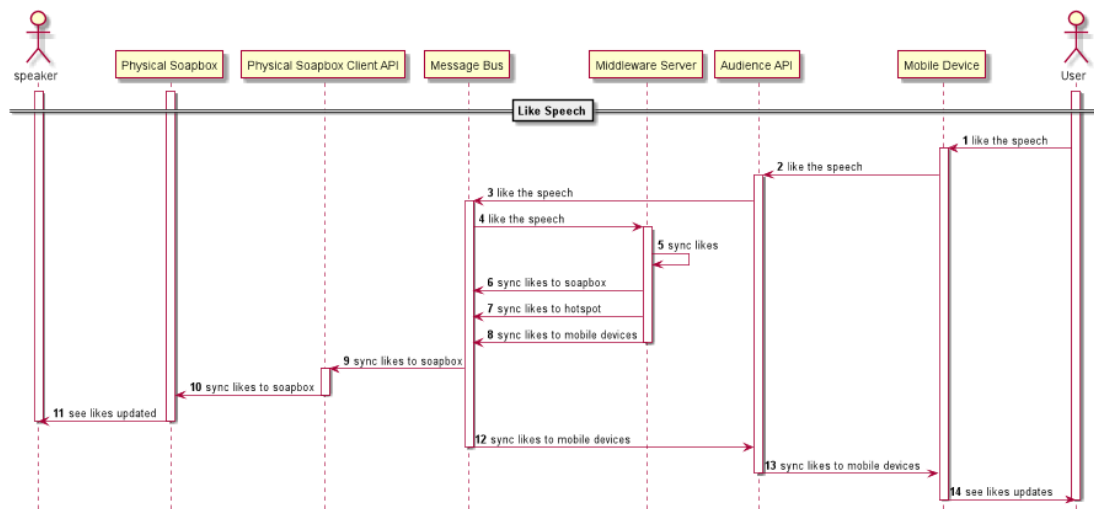


Figure 20. Mobile devices interaction flow chart

### 3.6.5. Persistent storage

In middleware server, speech related data are persisted in local database so that certain queries are enabled for continuous serving, one of those is the example of querying the history speeches in mobile devices as shown in the Figure 21 as below.

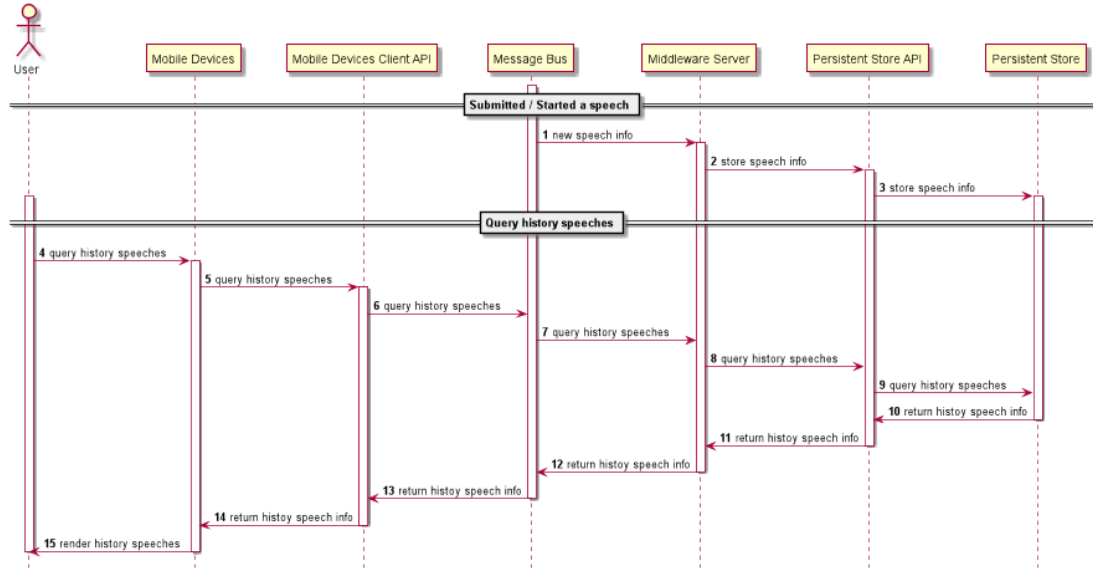


Figure 21. Persistent storage (query history speeches) flow chart

### 3.6.6. Other flows

There are additionally some minor workflows which are not the core features provided in our middleware system, but they are also valid and supported. Below, we shortly introduce each of these workflows.

First is speech reservation work flow, which allows the physical soapbox user to reserve for a speech with filled speech information and specific time period of the speech. Then later the speaker will get an email link with validation key to accessing the soapbox application index page when the speech is about to start. In this way, the speech can be provided with flexibility to be either reserved or started immediately. Such reservation feature is also enabled in mobile devices as well.

Second is local speech saving feature which allows soapbox application to save the speech locally into a video file, which can be further archived into video source repository and watched repeatedly.

## 4. EVALUATION

In order to conclude the feasibility and performance of our prototype, certain evaluations are conducted. Evaluation of our prototype comprises of two parts: laboratory evaluation and field evaluation. The former evaluation focuses on measuring quantitative performance metrics when running our prototype in different predefined use cases regarding to our system requirements. On the other hand, latter field evaluation takes our prototype into actual application usage to see if our prototype is feasible to serve solving the research questions mentioned previously.

### 4.1. Lab evaluation

For lab evaluation, there are 3 different use cases defined to measure and verify our prototype's performance: stability, streaming receiver profiling, performance.

All tests are conducted in a Windows 10 laptop (CPU: 2.5GHz, RAM: 8 GB) with wireless connection via router (bandwidth: 100 / 10 Mbps). Measured in test laptop, RTT value of RabbitMQ server (bunny.ubioulu.fi: pan0246.panoulu.net) is in average 19 ms, while RTT value of local host is in average < 1 ms. Google Chrome browser (Version 58.0.3029.96 (64-bit)) and XAMPP (Version: 5.6.3) are used together to host our lab evaluation setup.

For stability test, middleware server is running for 24 hours with constant soapbox speech streaming and one streaming receiver attending. In terms of streaming receiver, it stands for the party who is able to receive continuously speech streaming without any other interaction. In our lab evaluation, this is represented by a development sample web page of virtual soapbox receiver.

For streaming receiver profiling test, chrome developer tools plugin is adopted to collect timeline data in millisecond scale, so as to investigate whether any non-middleware aspect affects the performance of the soapbox application when initializing a stream receiver. The result is shown in Figure 22 as follows, which demonstrated that the loading of our middleware client API script takes insignificant time. Meanwhile, the largest part is from idle time which mainly comes from waiting from response of Websocket registration connection.

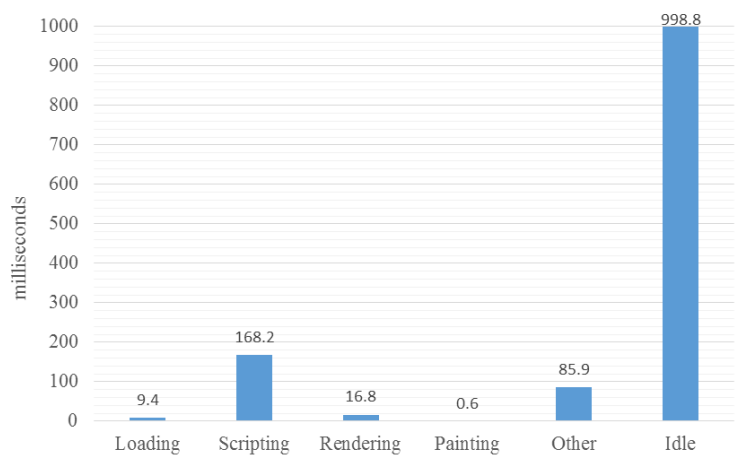


Figure 22. Summary of elapsed time for starting a stream receiver

For performance test, 9 tabs of video stream receiver test pages are sequentially opened in ~10 seconds with a soapbox speaker sample sender. Two HTML5 video tag events (onload, playing) are used to mark the latency start and end moments. The latency of playing the speech in receiver side is shown in Figure 23 as follows, which implies a fluctuating trend of insignificantly increasing latency when more stream receivers are added. However, there is no significant evidence that this latency affects only the most recently opened tab indicating that this latency can apply to all tabs, which is reasonable considering the influencing elements include end-to-end nature of provider and answerer meta-data exchange, as well as meta-data exchange coordinated in our middleware as the signaling server.

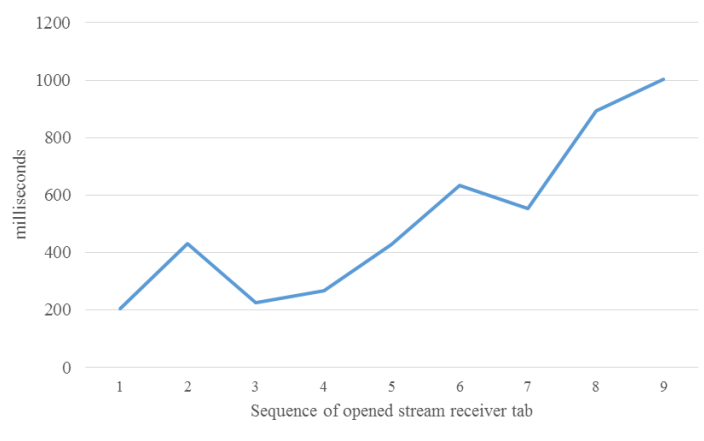


Figure 23. Latency of playing speech in stream receiver.

## 4.2. Field evaluation

Our prototype has been applied and evaluated in [35] which demonstrated the positive attitude of users during initial attempt of usage, the ease of learning, and the effectiveness of attracting more citizens for civic engagement participation. Its soapbox application architecture is shown in Figure 6.

Another evaluation with 6 speeches which is mainly discussed in this chapter is conducted with specialized focus on our prototype's supported hybrid-reality feature, while the former field evaluation highlights influence of physical soapbox interaction. Participants without any experience to soapbox system are recruited. For this evaluation, the evaluation setup and collected evaluation data are discussed in the following chapters.

## 4.3. Setup

Hardware setup for the evaluation includes the virtual soapbox environment, public displays with Ubi-hotspot, middleware server, and mobile devices (android phones) with unified wireless panOulu network.

The virtual soapbox environment is built located in CAVE shown in Figure 24 (reprinted with permissions from [37]) as follows snapshotted in one reserach, University of Oulu with a laptop as virtual soapbox application server. CAVE is an immersive virtual environment with sounds and visuals coined by Carolina et al. [36].

The virtual soapbox application server is equipped with Windows operating system. Middleware server runs on a local computer with 8GB RAM and 750GB HDD.

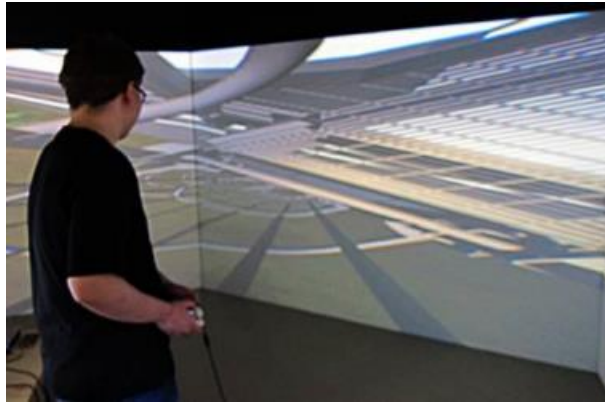


Figure 24. University of Oulu CAVE environment example scenario capture.

#### 4.4. Data

The evaluated user study consists of 6 speeches (speakers) and their corresponding speech length, amount of audience, amount of speech interactions as summarized in the following tables.

Table 1. Summary of campus use study speech meta-data

Speech No.	Speech length	Audience	Comments	Likes	Dislikes
1	11 mins	2	4	1	1
2	18 mins	2	4	1	0
3	17 mins	2	3	2	0
4	6 mins	1	1	0	0
5	17 mins	2	1	1	0
6	23 mins	11	69	6	4

Table 2. Statistical analysis of the campus use study speech meta-data

Statistics	Speech length	Audience	Comments	Likes	Dislikes
Mean	15.3 mins	3.3	13.6	1.8	0.8
Median	17 mins	2	3.5	1	0

Looking at the summary speech data in Table 1, there are some inferences to be drawn: Firstly, speeches can vary significantly in length, with shortest speech lasting six minutes whereas longest speech took 23 minutes. The longest speech in the evaluation also has most audience, comments, likes and dislikes. This illustrates that certain speeches and associated topics can draw significantly more interest in audience than others, as is expected in civic engagement. In the case of an interesting speech, it becomes relevant that the audience can take actions in committing to the conversation considering the speech. In the case of this civic engagement prototype, the options designed for committing to the conversation include submitting comments, likes and dislikes, which are familiar actions from major social media platforms. This familiarity provides a lower barrier for audiences to participate.



With calculated statistics of mean and median values from Table 2, it can be seen that the average speech length can be around 15 minutes, which is a good proposal for predefined speech reservation length, considering future development toggle. Also, the limited values of audience amount, comments, likes, and dislikes indicate that with our current use study evaluation setup, which includes only one public display as extension medium to attract audience out of speaker's vocal reach, has potentials to scale its impact by multiplying the amount of public displays. Besides, it's obvious that comments amount is larger than likes and dislikes, which implies that users may be inclined to engage in a more personalized style instead of being constrained in strict audience participatory norms, considering comments are anonymous and free to express any contents they want. Such lesson is valuable for designing the civic engagement practices regarding to user experiences.

## 5. DISCUSSIONS AND FUTURE WORK

A situated virtual-physical hybrid civic engagement platform prototype was made to address the civic engagement challenge of enhancing those civic engagement practices to be more effective, with technologies of urban computing and urban informatics. Meanwhile in this thesis, feasibility of this prototype was being empirically assessed, i.e. by subjecting the prototype to a user test and performance measurements.

### 5.1. Original research questions

As for our original research questions, we will discuss as follows individually to investigate whether and to what extent that the literature review, as well as designing, implementing and evaluating activities made in this thesis with our prototype can answer them.

**Research Question 1 (RQ1):** What kind of middleware functionality is required to support technologically-mediated civic engagement within urban computing and informatics?

First, we introduce the related work in urban computing and informatics to explore their distinct features and then align the expectation and help compare with our objective prototype system with underlying middleware support, so as to answer this research question with proper references. For instance, taking DiS system introduced in previous chapter as an example to compare, since it is one of the most similar reference systems in urban practices of giving speeches aiming at enhancing civic engagement. There is a distinct difference between their DiS system and ours, which is the underlying technology for engaging civic participation, as we use multimedia streaming instead of Twitter hashtag in their case. Since their Twitter hashtag content will be censored and displayed asynchronously, however, in our system it is allowed to observe real time civic engaging reactions to ongoing speeches in a synchronized fashion, which will be the most interesting research data to be collected. Additionally, our system design has also an initial implementation of saving speech videos locally, provisioning the possibility of later asynchronous speech content viewing in repositories.

On the other hand, this research question is carefully explored in design, implementation and evaluation phases in this thesis. When designing, we conducted a detailed technology selection for requirements for technologically-mediated civic engagement. It is defined that messaging, streaming, and persistent storing are the essential requirements for such middleware functionality as underlying support for the civic engagement practices. For each requirement of middleware functionality, related work of available solutions are discussed and compared. Furthermore, reference systems of middleware functionality is also discussed to investigate the advantages and disadvantages of implementing such middleware functionality by customizing underlying technology solutions for each proposed requirement independently. Besides, five specific requirements as design principles are proposed for our middleware functionalities supporting the prototype, i.e. soapbox platform. The necessity of real-time messaging and video streaming are carefully considered aiming for stable and fail-over functioning.

In conclusion, this research question is sufficiently answered arguing the messaging, video streaming and persistent storing as required middleware functionality within urban computing and informatics, and the validity of how these requirements are defined with the help of our literature review and detailed technology selection discussions.

**Research Question 2 (RQ2):** What is the impact of the designed middleware to the technologically-mediated civic engagement prototype in terms of performance?

In evaluation phase, it's demonstrated that the designed middleware has proper scalability of applying large amount of hardware setups for the prototype, including public displays, mobile devices, and virtual audience, based on technologies of urban computing and urban informatics. Such scalability should be able to help improve civic engagement practices to attract more attendance or audience in urban spaces, and therefore answer this research question. It's demonstrated that the lab and field evaluations conducted meet their corresponding expectations of proper middleware functioning and feasibility, resulting from no sudden crash of middleware components happened during all evaluation periods. The capability of our prototype supporting civic engagement was also emphasized, with one publication based on this thesis. In conclusion, this research question is also sufficiently answered that our designed middleware to the technologically-mediated civic engagement prototype has a positive influence on the civic engagement process, and it's performing regarding setting up multiple public displays and continuous video streaming and messaging for several speeches.

## 5.2. Reaching of objectives

As stated previously, our objective of this thesis is to design, implement and assess middleware functionalities for a novel soapbox prototype to be more effective in civic engagement with technologies of urban computing and urban informatics. It can be mainly illustrated whether the objectives are reached in implementation phase. In implementation phase, the medium selected for broadcasting in our prototype as the novel video streaming technologies are applied which enables our prototype to reach audience outside of speaker's hearing and vocal area. Comparing with traditional constraint area of speaker's corners or speaker's surroundings, our approach is significantly superior than traditional soapboxes, mediating the constraints of spaces and attendance convenience of audience. In evaluation phase, as we did not cross compare our prototype with traditional soapbox, this research question is hard to answer regarding to feasibility and performance. In overall, with all activities made in this thesis, it can be initially argued that such civic engagement platform is capable of mediating the traditional counterparts, and our civic engagement platform has proven its unique advantages in reaching out larger amount of audience over traditional counterparts.

Also, the feasibility and performance of such middleware functionality are also part of our objectives in order to answer the proposed research questions. The feasibility refers to two aspects: setting up speeches should not take significant time, and that streams can be multi-casted to a sufficiently large number of receivers. With the previous lab evaluation, it is demonstrated that the setup time for streaming is within acceptable latency range, and receivers are simulated to be capable to scale up to nine. In field evaluations, it is proven that our prototype system is feasible to handle both simulated and actual soapbox speech streaming in addition to which

meeting the design requirements. Besides, the capability of cooperating with diverse soapbox applications via built-in virtual and physical client supports demonstrates the prominence and effectiveness of adopting it as a contemporary tool in civic engagement platform. Therefore, the objectives of this thesis can be considered as sufficiently reached.

### **5.3. Limitations**

Considering the limitations of the evaluations, specifically with the field evaluation conducted and discussed in detail in this thesis, the number of participants was limited. And thus, we only have data from six participants, which limits the amount of analysis that can be done to the data. Furthermore, even though the scalability of our middleware functionality was demonstrated in lab evaluation, there are difficulties in assessing scalability of the implemented system since the numbers of participants was only six, while the hardware setup (public displays, mobile devices) are also inadequate. Therefore, more comprehensive scalability analyses should be conducted in a larger field study with significantly more participants, and preferably more hardware setups.

### **5.4. Future work**

In the future, in terms of middleware implementation, there could be some enhancement work. First, our middleware system can be generalized for more speech supported civic engagement platforms with real-time video streaming and messaging requirements. Also, currently partial signaling feature is embedded in middleware client API layer, which can be merged into middleware server for better central management for meta-data signaling and coordination of different components. From the client API perspective, different roles of speech entities based on WebRTC concepts can be divided into three categories: stream provider only, stream receiver only, dual stream provider and receiver. And therefore, the client API can be abstracted based on different roles and apply for application callers accordingly.

As for the evaluation work, more granular and quantitative evaluation can be conducted to correlate whether certain features can affect the satisfaction of user application interaction experiences, or civic engagement experiences. The further user study of in which ways citizens prefer engaging with such systems can also be conducted.

## 6. CONCLUSIONS

In this thesis, the feasibility and performance of middleware support are extensively discussed for a novel soapbox prototype system as candidate of situated virtual-physical hybrid reality civic platform. The thesis first introduces the background of contemporary civic engagement experiments, and therefore brings about the related work in the domains of ubiquitous computing, urban computing, and urban informatics. Hence, it is argued that soapbox as one of the essential urban practices can be technologically mediated via bridging urban computing and urban informatics into civic engagement system development. With the middleware system designed and implemented in this thesis for the soapbox prototype, it is demonstrated that our middleware system is capable of supporting the requirements of real-time video streaming, multi-component asynchronous and synchronous messaging, persistent storing, and easy-to-use interfaces for client applications. Specifically, it is illustrated from the development process of the soapbox prototype system that the middleware client APIs are concise and middleware server are comprehensive. Moreover, it is concluded from the lab evaluations, field evaluations and research publication that the extent of feasibility and performance can fulfill the demands of a novel civic engagement platform.

In a nutshell, our middleware support as the essential part of the technological mediated solution facilitating the civic engagement features, can be useful for diverse dimensions concerning civic engagement such as the governmental civic administrations, urban construction and planning bureaus, or civic engagement researchers. From our perspectives, it is regarded as a starting point to explore how civic engagement problems can be better resolved, and how ubiquitous computing technologies can influence practices taking place in urban spaces around the world.

## 7. REFERENCES

- [1] Zheng, Yu, et al. "Urban computing: concepts, methodologies, and applications." *ACM Transactions on Intelligent Systems and Technology (TIST)* 5.3 (2014): 38.
- [2] Bassoli, Arianna, et al. "Aesthetics: Rethinking Urban Computing." (2007).
- [3] Foth, Marcus, Jaz Hee-jeong Choi, and Christine Satchell. "Urban informatics." *Proceedings of the ACM 2011 conference on Computer supported cooperative work*. ACM, 2011.
- [4] Liu, Yong, et al. "Identity crisis of ubicomp?: mapping 15 years of the field's development and paradigm change." *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 2014.
- [5] José, Rui, et al. "Instant places: Using bluetooth for situated interaction in public displays." *IEEE Pervasive Computing* 7.4 (2008): 52-57.
- [6] Cahill, Vinny, et al. "Guest editors' introduction: Building and evaluating ubiquitous system software." *IEEE Pervasive Computing* 3.3 (2004): 20-21.
- [7] Schroeter, Ronald, and Marcus Foth. "Discussions in space." *Proceedings of the 21st Annual Conference of the Australian Computer-Human Interaction Special Interest Group: Design: Open 24/7*. ACM, 2009.
- [8] WebRTC 1.0: Real-time Communication Between Browsers. URL: <https://www.w3.org/TR/webrtc/>. Accessed: 2016-11-28.
- [9] What is RabbitMQ? URL: <https://www.rabbitmq.com/>. Accessed: 2016-11-28.
- [10] Román, Manuel, et al. "A middleware infrastructure for active spaces." *IEEE pervasive computing* 1.4 (2002): 74-83.
- [11] HTML5 Web Messaging. URL: <https://www.w3.org/TR/webmessaging/>. Accessed: 2016-11-28.
- [12] The WebSocket API. URL: <https://www.w3.org/TR/websockets/>. Accessed: 2016-11-28.
- [13] Schroeter, Ronald, Marcus Foth, and Christine Satchell. "People, content, location: sweet spotting urban screens for situated engagement." *Proceedings of the Designing Interactive Systems Conference*. ACM, 2012.
- [14] Weiser, Mark. "The computer for the 21st century." *Scientific american* 265.3 (1991): 94-104.

- [15] Bell, Genevieve, and Paul Dourish. "Yesterday's tomorrows: notes on ubiquitous computing's dominant vision." *Personal and ubiquitous computing* 11.2 (2007): 133-143.
- [16] Rogers, Yvonne. "Moving on from weiser's vision of calm computing: Engaging ubicomp experiences." *International conference on Ubiquitous computing*. Springer Berlin Heidelberg, 2006.
- [17] Weiser, Mark, and John Seely Brown. "The coming age of calm technology." *Beyond calculation*. Springer New York, 1997. 75-85.
- [18] Memarovic, Nemanja, et al. "Using public displays to stimulate passive engagement, active engagement, and discovery in public spaces." *Proceedings of the 4th Media Architecture Biennale Conference: Participation*. ACM, 2012.
- [19] Steinberger, Fabius, Marcus Foth, and Florian Alt. "Vote with your feet: Local community polling on urban screens." *Proceedings of The International Symposium on Pervasive Displays*. ACM, 2014.
- [20] Johnstone, Sarah, Glenda Amayo Caldwell, and Markus Rittenbruch. "Defining the InstaBooth: Facilitating debate and content creation from situated users." *Proceedings: MEDIACITY 5 International Conference, Workshops and Urban Interventions*. School of Architecture, Design & Environment and i-DAT (Institute of Digital Art and Technology), Plymouth University., 2015.
- [21] Scholtz, Jean, and Sunny Consolvo. "Toward a framework for evaluating ubiquitous computing applications." *IEEE Pervasive Computing* 3.2 (2004): 82-88.
- [22] Hosio, Simo, et al. "Toward Meaningful Engagement with Pervasive Displays." *IEEE Pervasive Computing* 15.3 (2016): 24-31.
- [23] Abowd, Gregory D. "What next, ubicomp?: celebrating an intellectual disappearing act." *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. ACM, 2012.
- [24] Calderon, Roberto, et al. "Developing cross-display applications using the really easy displays (red) framework." *The International Symposium on Pervasive Displays*. 2013.
- [25] Calderon, Roberto, et al. "RED: a framework for prototyping multi-display applications using web technologies." *Proceedings of The International Symposium on Pervasive Displays*. ACM, 2014.
- [26] Churchill, Elizabeth F., et al. "Sharing multimedia content with interactive public displays: a case study." *Proceedings of the 5th conference on Designing interactive systems: processes, practices, methods, and techniques*. ACM, 2004.

- [27] Fortin, Claude, Carman Neustaedter, and Kate Hennessey. "The appropriation of a digital speakers corner: lessons learned from the deployment of mégaphone." Proceedings of the 2014 conference on Designing interactive systems. ACM, 2014.
- [28] Erbad, Aiman, et al. "Magic broker: A middleware toolkit for interactive public displays." Pervasive Computing and Communications, 2008. PerCom 2008. Sixth Annual IEEE International Conference on. IEEE, 2008.
- [29] Blackstock, Michael, et al. "MAGIC Broker 2: An open and extensible platform for the Internet of Things." Internet of Things (IOT), 2010. IEEE, 2010.
- [30] N. Kaviani, M. Finke, R. Lea, and S. Fels. Investigating a design space for multidevice environments. volume 28, pages 722–729, 2012.
- [31] Müller, Jörg, et al. "Requirements and design space for interactive public displays." Proceedings of the 18th ACM international conference on Multimedia. ACM, 2010.
- [32] Browser APIs and Protocols: WebRTC - High Performance Browser Networking (O'Reilly). URL: <https://hpbn.co/webrtc/>. Accessed: 2016-12-25.
- [33] Storz, Oliver, et al. Public ubiquitous computing systems: Lessons from the e-campus display deployments. IEEE Pervasive Computing 5.3 (2006): 40-47.
- [34] Caldwell, Glenda Amayo, et al. "The InstaBooth: making common ground for media architectural design." Proceedings of the 3rd Conference on Media Architecture Biennale. ACM, 2016.
- [35] Kai Wang, Jilin Yang, Haejong Dong, and Hannu Kukka. 2016. Soapbox: a situated platform for civic engagement. In Proceedings of the 15th International Conference on Mobile and Ubiquitous Multimedia (MUM '16). ACM.
- [36] Cruz-Neira, Carolina, et al. "The CAVE: audio visual experience automatic virtual environment." Communications of the ACM 35.6 (1992): 64-73.
- [37] Jani Väyrynen, et al. Utilizing Immersive 3D for Industrial Site Design. ARea 14 workshop on Interaction and User Experience Design for Future Mixed Reality Spaces, 2014, Oulu, Finland.
- [38] Cattell, Rick. "Scalable SQL and NoSQL data stores." Acm Sigmod Record 39.4 (2011): 12-27.



- [39] Lifton, Joshua, and Joseph A. Paradiso. "Dual reality: Merging the real and virtual." International Conference on Facets of Virtual Environments. Springer Berlin Heidelberg, 2009.
- [40] Fortin, Claude, Kate Hennessy, and Hughes Sweeney. "The 'Making of' Mégaphone, an Interactive Speakers' Corner and Digitally-Augmented Agora in Public Space." Proceedings of The International Symposium on Pervasive Displays. ACM, 2014.
- [41] Baldauf, Matthias, et al. "Interactive opinion polls on public displays: studying privacy requirements in the wild." Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services. ACM, 2014.

## 8. APPENDICES

### Appendix 1. Database schema

- Speeches document

```
{
  "speech_id": <Starting DateTime String>,
  "is_speech_for_reservation": True,
  "status": "reserved" or "submitted" / "locked" / "unlocked" / "ongoing" / "over"
  "is_used": True once it becomes "ongoing" status
  "length": ,
  "unlock_timestamp":
  "start_timestamp":
  "end_timestamp":
  "password":
  "speaker_name": "Jilin Yang",
  "speech_topic": "Hello everyone",
  "submit_info": <Store info from Soapbox>
}
```

- Likes document

```
{
  "speech_id": <Same as speech_id in speeches document>,
  "timestamp":
  "sender": "audience" or hotspot name
}
```

- Dislikes document

```
{
  "speech_id": <Same as speech_id in speeches document>,
  "timestamp":
  "sender": "audience" or hotspot name
}
```

- Reports document

```
{
  "speech_id": <Same as speech_id in speeches document>,
  "timestamp":
  "sender": "audience" or "virtual-speaker" or "virtual-audience"
}
```

- Comments document

```
{
  "speech_id": <Same as speech_id in speeches document>,
  "timestamp":
  "sender": "audience" or hotspot name
  "name":
  "content":
}
```

## Appendix 2. Speech registration form schema from applications

```
{
  "name": "Jilin",
  "gender": "Male",// gender: Male, Female
  "topic": "Education",
  "reality_type": "Real",// reality_type: Real, Virtual
  "expected_length": "10"// expected_length ----- minutes
}
```

## Appendix 3. WebRTC signaling schema

- Offer

```
{
  "sdp": <sdp>
  "id": <id>
}
```

- Answer

```
{
  "sdp": <sdp>
  "id": <id>
}
```

- ICE candidate

```
{
  "ice": <ICE-candidate>,
  "id": <id>
}
```

#### Appendix 4. Middleware signaling schema

```
soapbox = { "id": <id> }

hotspots = [
  {
    "id": '16fd2706-8baf-433b-82eb-8c7fada847da',
    "name": "test_hotspot"
  }
]

virtuals = [
  {
    "id": '16fd2706-8baf-433b-82eb-8c7fada847da'
  }
]
```