

Zap

A Cross-platform Library for Multi-device Applications

12조 미디어프로젝트 아이디어 발표

목차

1. 개요
2. 배경: 문제의식 및 선행연구
3. 제안: 시스템 요구사항 및 설계
4. 계획: 목표 및 실현 범위

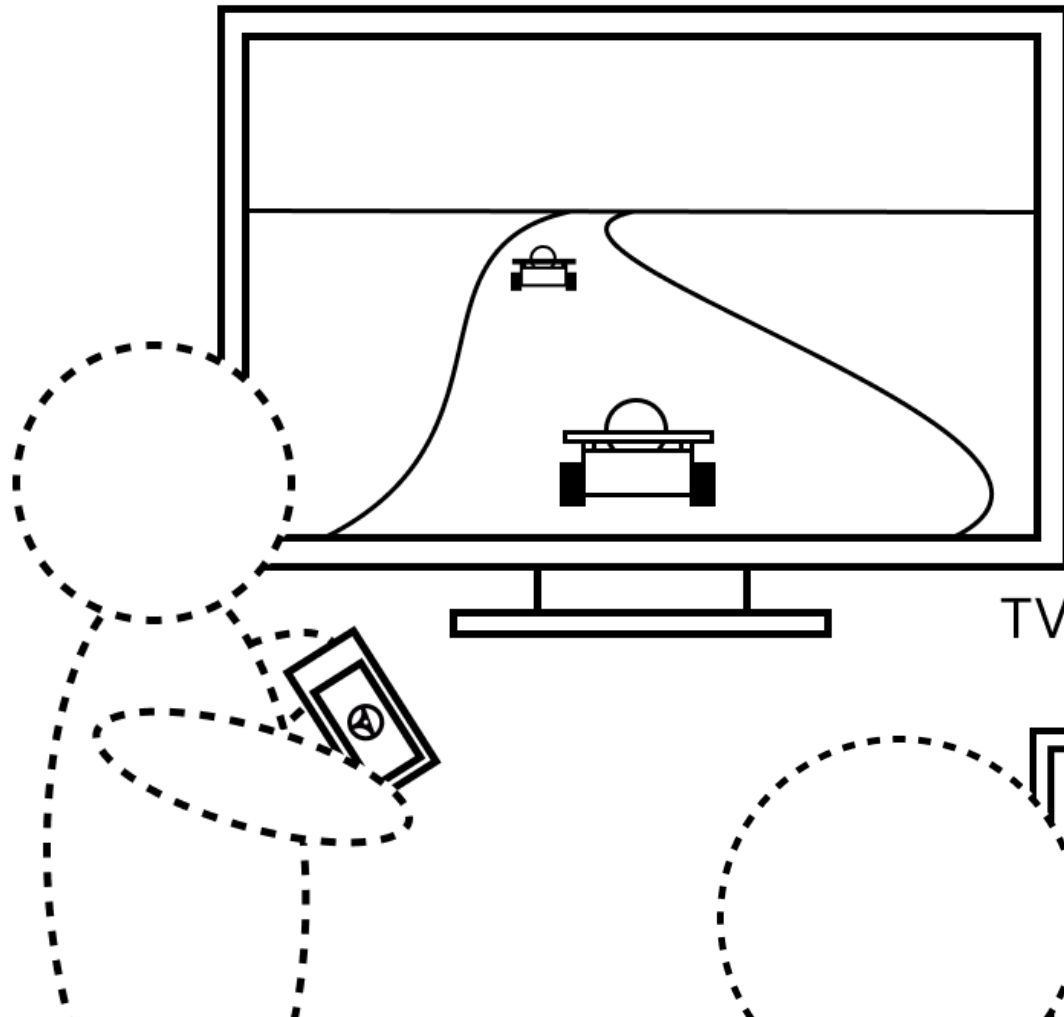
1. 개요

“멀티 디바이스 앱”을 위한 “크로스 플랫폼” 라이브러리

- 모바일 기기의 각종 인터페이스, 센서를 다양한 기기에 연동해 활용할 수 있도록 돕는 라이브러리.
- 프로그래머가 라이브러리를 이용해 쉽게 멀티 디바이스 애플리케이션을 구현할 수 있도록 돕는다.

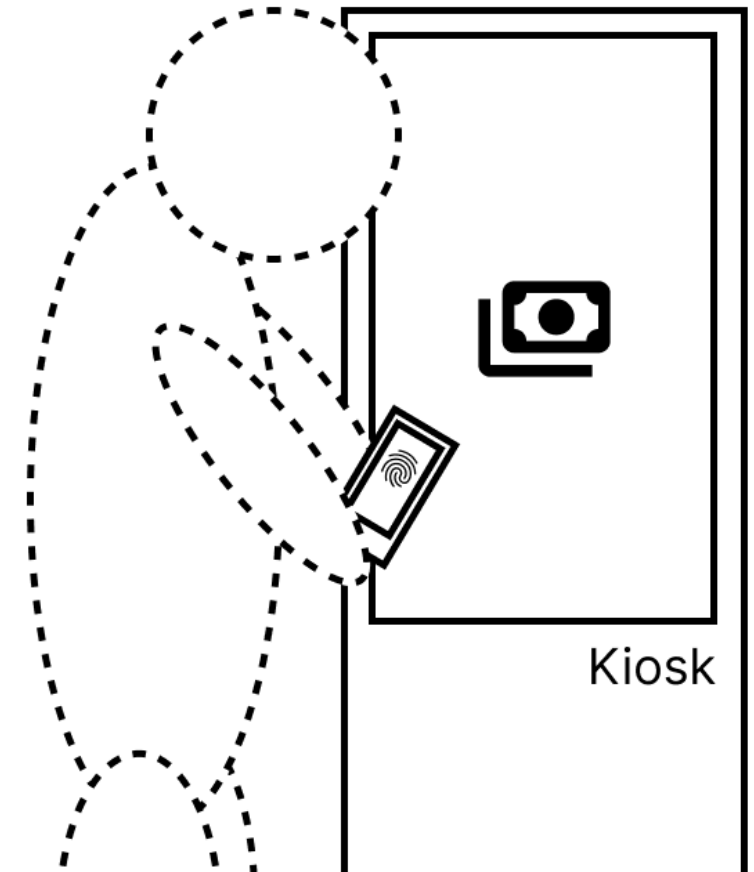
1. 개요

기울기 센서를 연동해
모션 인식 컨트롤러 구현



Kiosk

지문 인식 센서를 연동해
보안 결제 시스템 구현



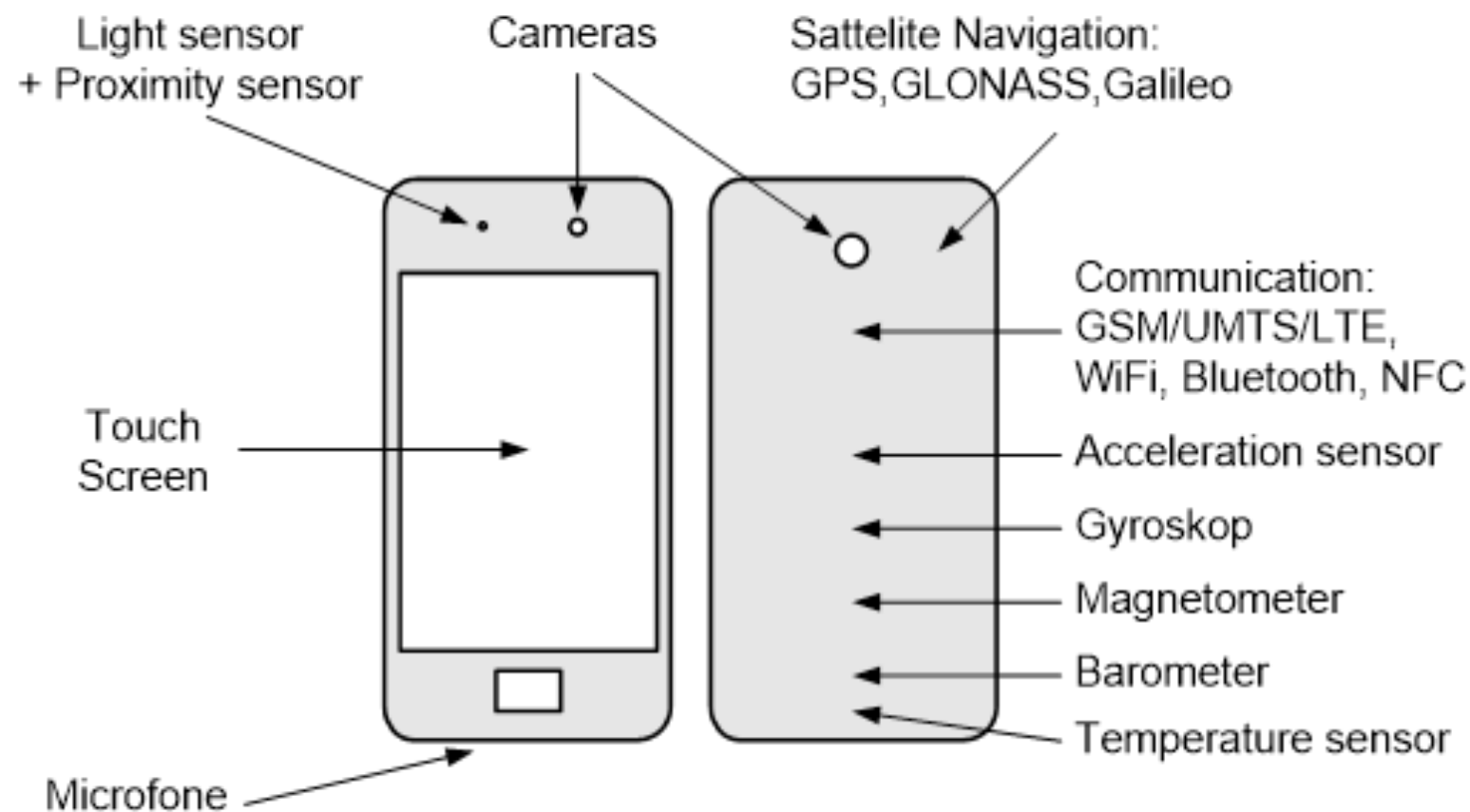
PC

클립보드를 연동해
심리스 작업 환경 구현



2. 배경

우리는 모바일 기기의 기능을 충분히 활용하지 못한다.



- 모바일 기기에 각종 센서가 탑재되어 있다.
- 하나의 기기에만 종속되어 있어 확장이 불가능.

2. 배경

선행 연구가 실용적인 라이브러리를 제공하지 않는다.



Tap: An App Framework for Dynamically Composable Mobile Systems

Naser AlDuaij
Department of Computer Science
Columbia University
New York, New York, USA
alduaij@cs.columbia.edu

Jason Nieh
Department of Computer Science
Columbia University
New York, New York, USA
nieh@cs.columbia.edu

ABSTRACT

As smartphones and tablets have become ubiquitous, there is a growing demand for apps that can enable users to collaboratively use multiple mobile systems. We present Tap, a framework that makes it easy for users to dynamically compose collections of mobile systems and developers to write apps that make use of those impromptu collections. Tap users control the composition by simply tapping systems together for discovery and authentication. The physical interaction mimics and supports ephemeral user interactions without the need for tediously exchanging user contact information such as phone numbers or email addresses. Tapping triggers a simple NFC-based mechanism to exchange connectivity information and security credentials that works across heterogeneous networks and requires no user accounts or cloud infrastructure support. Tap makes it possible for apps to use existing mobile platform APIs across multiple mobile systems by virtualizing data sources so that local and remote data sources can be combined together upon tapping. Virtualized data sources can be hardware or software features, including media, clipboard, calendar events, and devices such as cameras and microphones. Leveraging existing mobile platform APIs makes it easy for developers to write apps that use hardware and software features across dynamically composed collections of mobile systems. We have implemented a Tap prototype that allows apps to make use of both unmodified Android and iOS systems. We have modified and implemented various apps using Tap to demonstrate that it is easy to use and can enable apps to provide powerful new functionality by leveraging multiple mobile systems. Our results show that Tap has good performance, even for high-bandwidth features, and is user and developer friendly.

CCS CONCEPTS

• **Human-centered computing** → Ubiquitous and mobile devices; Ubiquitous and mobile computing systems and tools; Ubiquitous and mobile computing design and evaluation methods; • **Software and its engineering** → Software libraries and repositories; Development frameworks and environments; Middleware; Operating systems; Peer-to-peer architectures; API

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
MobiSys '21, June 21–July 2, 2021, Virtual, WI, USA.
© 2021 Copyright held by the owner(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-4443-8/21/06...\$15.00
<https://doi.org/10.1145/3458864.3467678>

languages; • **Computer systems organization** → Client-server architectures; Peer-to-peer architectures; • **Networks** → Mobile ad hoc networks.

KEYWORDS

mobile computing; distributed computing; operating systems; mobile devices; remote display; Android; iOS

ACM Reference Format:

Naser AlDuaij and Jason Nieh. 2021. Tap: An App Framework for Dynamically Composable Mobile Systems. In *The 19th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '21)*, June 24–July 2, 2021, Virtual, WI, USA. ACM, New York, NY, USA, 14 pages. <https://doi.org/10.1145/3458864.3467678>

1 INTRODUCTION

With mobile systems evermore ubiquitous, users rely on them as essential accessories of modern day life to socialize, share, and interact with other people. Individual users often own multiple mobile systems [84] and groups of users have many mobile systems at their disposal. There is a growing demand to provide users with a seamless experience across multiple mobile systems, not just use them as separate, individual systems. Our work on M2 [1] demonstrates various examples of useful functionality that can be achieved with multiple mobile systems working together, including combining multiple mobile systems into a mobile multi-headed display surface to provide a big screen experience for all users and turning a pair of mobile systems into a portable motion-based gaming console. We refer to the ability to combine the functionality of multiple mobile systems as *multi-mobile computing*.

Although multi-mobile computing has the potential to provide a wide range of powerful new app functionality, there are two key challenges that stand in the way of further adoption. First, there is no general, easy-to-use mechanism to connect impromptu collections of mobile systems that works across heterogeneous systems and networks. Even basic information sharing across mobile systems often requires users to tediously exchange phone numbers or email addresses, a burdensome process especially for the types of ephemeral interactions that occur among mobile users, not to mention that users may not want to reveal such personal information. Some approaches require users to login to cloud infrastructure and connect those cloud accounts together, an inconvenient process at best that requires network infrastructure and fails otherwise [41, 45]. Other approaches such as Apple's AirDrop [8] remove some of this burden to enable media sharing across mobile systems, but do not generalize to allow apps to go beyond this limited functionality to perform a wider range of tasks.

Session 9: Nuts and Bolts

MobiSys '19, June 17–21, 2019, Seoul, Korea



Heterogeneous Multi-Mobile Computing

Naser AlDuaij
Department of Computer Science
Columbia University
New York, New York, USA
alduaij@cs.columbia.edu

Alexander Van't Hof
Department of Computer Science
Columbia University
New York, New York, USA
alexvh@cs.columbia.edu

Jason Nieh
Department of Computer Science
Columbia University
New York, New York, USA
nieh@cs.columbia.edu

ABSTRACT

As smartphones and tablets proliferate, there is a growing demand for multi-mobile computing, the ability to combine multiple mobile systems into more capable ones. We present M2, a system for multi-mobile computing that enables existing unmodified mobile apps to share and combine multiple devices, including cameras, displays, speakers, microphones, sensors, GPS, and input. M2 introduces a new data-centric approach that leverages higher-level device abstractions and hardware acceleration to efficiently share device data, not API calls. To support heterogeneous devices, M2 introduces device transformation, a new technique to mix and match different types of devices. Example transformations include combining multiple displays into a single larger display for better viewing, or substituting accelerometer for touchscreen input to provide a Nintendo Wii-like experience with existing mobile gaming apps. We have implemented M2 and show that it (1) operates across heterogeneous systems, including multiple versions of Android and iOS, (2) can enable unmodified Android apps to use multiple mobile devices in new and powerful ways, including supporting users with disabilities and better audio conferencing, and (3) can run apps across mobile systems with modest overhead and qualitative performance indistinguishable from using local device hardware.

CCS CONCEPTS

• **Human-centered computing** → Mobile computing; Ubiquitous and mobile computing systems and tools; Mobile devices; • **Software and its engineering** → Client-server architectures; Operating systems; Peer-to-peer architectures.

KEYWORDS

mobile computing; distributed computing; operating systems; mobile devices; remote display; Android; iOS

ACM Reference Format:

Naser AlDuaij, Alexander Van't Hof, and Jason Nieh. 2019. Heterogeneous Multi-Mobile Computing. In *The 17th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '19)*, June 17–21, 2019, Seoul, Republic of Korea. ACM, New York, NY, USA, 14 pages. <https://doi.org/10.1145/3307334.3326096>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
MobiSys '19, June 17–21, 2019, Seoul, Republic of Korea.
© 2019 Copyright held by the owner(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-4461-8/19/06...\$15.00
<https://doi.org/10.1145/3307334.3326096>



Figure 1: Multi-mobile computing using fused devices

1 INTRODUCTION

Users rely on tablets and smartphones for their everyday computing needs. Individual users often own multiple mobile systems of various shapes and sizes [79], and groups of users often have many mobile systems at their disposal. Since the vision of dynamic composable computing [72], there has been a growing demand to provide users with a seamless experience across multiple mobile systems, not just use them as separate, individual systems. We refer to the ability to combine the functionality of multiple mobile systems into a more capable one as *multi-mobile computing* [1, 2].

Three examples help illustrate some of the possibilities. First, multi-mobile computing makes it straightforward to remote control other devices such as cameras, and provide richer input modalities beyond touchscreen input such as motion-based game controllers or more accessible interfaces for users with disabilities. Second, multi-mobile computing makes it easy for users to combine their tablets together in a self-organizing multi-headed display and input surface for a big screen experience for all users anywhere, even when a big bulky screen is not available, as shown in Figure 1. Similarly, multiple smartphones can combine cameras together to provide panoramic video recording without specialized hardware. Third, multi-mobile computing makes it easy to use users' smartphones distributed across a room to leverage their microphones from multiple vantage points. Together, they can provide superior speaker-identifiable sound quality and noise cancellation for audio conferences, without costly specialized equipment. Unlike simple one-to-one I/O sharing approaches [4, 15, 30, 52] such as Apple AirPlay [10] which can display content from a smartphone to an Apple TV [11], multi-mobile computing envisions a broader, richer experience with the ability to combine multiple devices from multiple systems together in new ways.

Although multi-mobile computing has the potential to provide a wide range of powerful new app functionality, three key challenges must be met to turn this potential into reality. First, mobile systems are highly heterogeneous; on Android alone, more than 24,000 different systems are available [55]. They are tightly integrated hardware platforms that incorporate a plethora of different hardware devices using non-standard interfaces. Many different versions of software



A-Mash: Providing Single-App Illusion for Multi-App Use through User-centric UI Mashup

Sunjae Lee^{1*}, Hoyoung Kim¹, Sijung Kim¹, Sangwook Lee¹, Hyosu Kim², Jean Young Song³
Steven Y. Ko⁴, Sangeun Oh⁵, Insik Shin^{1,6*}

¹KAIST, S. Korea ²Chung-Ang University, S. Korea ³DGIST, S. Korea
⁴Simon Fraser University, Canada ⁵Ajou University, S. Korea ⁶Fluiz Corp., S. Korea
^{*}{sunjae1294, insik.shin}@kaist.ac.kr

ABSTRACT

Mobile apps offer a variety of features that greatly enhance user experience. However, users still often find it difficult to use mobile apps in the way they want. For example, it is not easy to use multiple apps simultaneously on a small screen of a smartphone. In this paper, we present A-Mash, a mobile platform that aims to simplify the way of interacting with multiple apps concurrently to the level of using a single app only. A key feature of A-Mash is that users can *mask* up the UIs of different existing mobile apps on a single screen according to their preferences. To this end, A-Mash 1) extracts UIs from unmodified existing apps (*dynamic UI extraction*) and 2) embeds extracted UIs from different apps into a single wrapper app (*cross-process UI embedding*), while 3) making all these processes hidden from the users (*transparent execution environment*). To the best of our knowledge, A-Mash is the first work to enable UIs of different unmodified legacy apps to seamlessly integrate and synchronize on a single screen, providing an illusion as if they were developed as a single app. A-Mash offers great potential for a number of useful usage scenarios. For instance, a user can mashup UIs of different IoT administration apps to create an *all-in-one* IoT device controller or one can mashup today's headlines from different news and magazine apps to craft one's own news headline collection. In addition, A-Mash can be extended to an AR space, in which users can map UI elements of different mobile apps to physical objects inside their AR scenes. Our evaluation of the A-Mash prototype implemented in Android OS demonstrates that A-Mash successfully supports the mashup of various existing mobile apps with little or no performance bottleneck. We also conducted in-depth user studies to assess the effectiveness of the A-Mash in real-world use cases.

CCS CONCEPTS

• **Human-centered computing** → Graphical user interfaces; User interface management systems; User centered design.

KEYWORDS

Multi-tasking; User Interface mashup; Multi app execution; Mobile platform; Multi-app single-screen

ACM Reference Format:

Sunjae Lee^{1*}, Hoyoung Kim¹, Sijung Kim¹, Sangwook Lee¹, Hyosu Kim², Jean Young Song³, Steven Y. Ko⁴, Sangeun Oh⁵, Insik Shin^{1,6*}. 2022. A-Mash: Providing Single-App Illusion for Multi-App Use through User-centric UI Mashup. In *The 20th Annual International Conference on Mobile Computing And Networking (ACM MobiCom '22)*, October 17–21, 2022, Sydney, NSW, Australia. ACM, New York, NY, USA, 13 pages. <https://doi.org/10.1145/3495243.3560522>

1 INTRODUCTION

Mobile apps have made our lives significantly easier and convenient. Various apps provide a wide variety of features that greatly simplify and enrich the end-user experience in a variety of areas, including social media, entertainment, and shopping. Furthermore, as the mobile apps become increasingly sophisticated and diversified, smartphone users often want to use their smartphones across app-boundaries [21, 37]. A typical example is where users perform multi-tasking. For instance, when running outside for an exercise, a user might want to constantly check her location using Google maps, keep track of her records through Run tracker app, and even listen to music through a music player app. However, since modern-day mobile platforms do not support multiple apps to use the smartphone screen simultaneously, she needs to constantly switch between apps to perform each task independently.

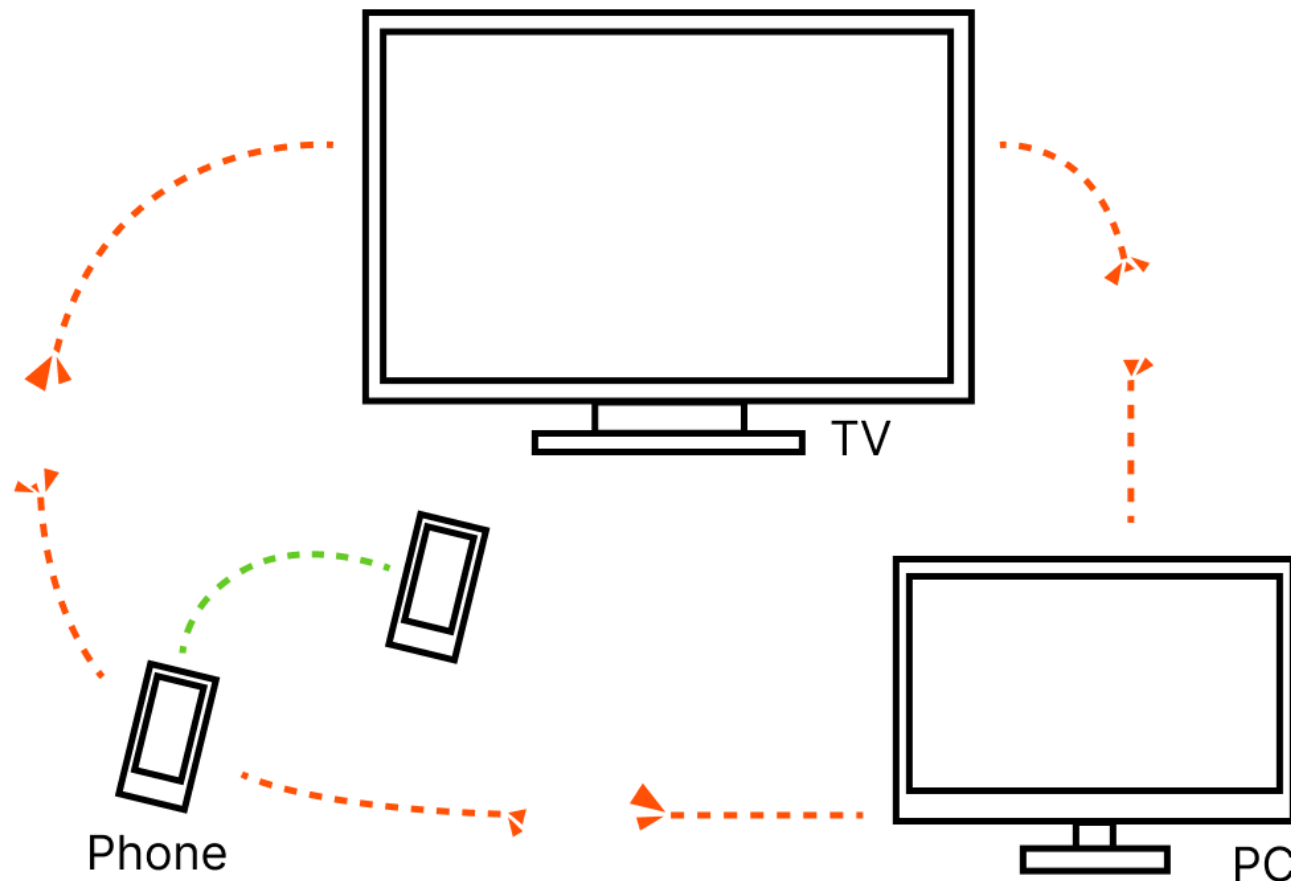
Recently, a few mobile platforms have introduced new techniques that allow multiple apps to run concurrently in a single screen. Split-screen mode [2, 15] allows two apps to run side by side by splitting the screen into two mini windows. Free-form mode [29] allows individual apps to be drawn on a separate movable windows. However, not to mention that it only supports maximum of two apps at a time, it is inconvenient to interact with the apps when they are drawn in such small windows. On the other hand, a couple of studies [18, 38] have considered a similar problem in the web environment and introduced techniques to mash up UI elements of different webpages to create a single multi-purpose web interface. Such an approach can be an effective solution in mobile environments as well since it can not only remove the need to switch between apps when performing tasks that involve multiple apps, but also display only the necessary UIs in the given small size of screen. However, there is a large gap (i.e., code availability) between the web and mobile environments to apply such techniques directly to mobile apps (see Section 10).

• 멀티 디바이스 컴퓨팅에 대한 많은 선행 연구가 있다.

• “즉시 사용 가능한” 솔루션을 제공하지 않는다.

2. 배경

크로스 플랫폼을 지원하는 멀티 디바이스 솔루션이 없다.



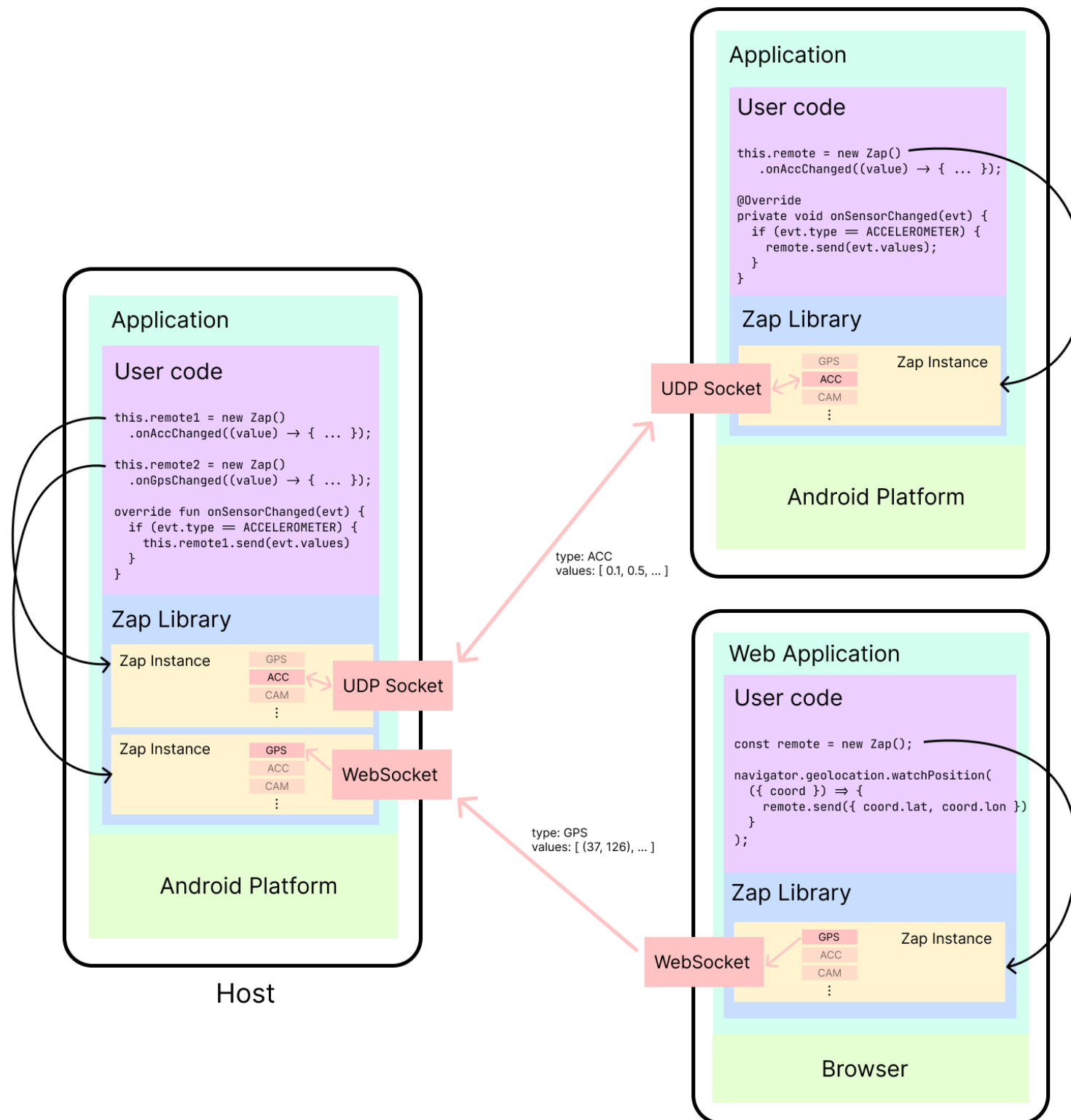
- 모바일 기기와 달리 PC, TV 등의 기기에는 센서가 없다.
- 넓은 유비쿼터스 환경을 위해 크로스 플랫폼을 지원해야.

3. 제안

Zap: 멀티 디바이스 앱을 위한 크로스 플랫폼 라이브러리

- 라이브러리를 통해 시스템을 수정하지 않고 동일한 인터페이스로 다양한 종류의 기기를 연동할 수 있다.
- A 기기가 B 기기의 데이터소스를 이용할 수 있다.
e.g., 로컬 기기가 원격 기기의 가속도 센서를 이용.
- 같은 통신 프로토콜로 다른 플랫폼과 통신할 수 있다.
e.g., PC에서 모바일 기기의 센서를 이용.

3. 제안



4. 계획

- 애플리케이션 레벨 라이브러리를 오픈소스로 배포.
- Android, Java/Kotlin, (+ JavaScript) 구현.
- 동작센서, GPS, 클립보드, 위젯, (+ 미디어) 지원.
- 라이브러리를 이용해 구현할 수 있는 예시 앱 개발.

참고자료

- Takeda, Genyo, and Howard Cheng. “Spatially-correlated human-machine interface.” U.S. Patent No. 11,173,392. 16 Nov. 2021.
- AlDuaij, Naser, and Jason Nieh. “Tap: an app framework for dynamically composable mobile systems.” *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services*. 2021.
- AlDuaij, Naser, Alexander Van’t Hof, and Jason Nieh. “Heterogeneous multi-mobile computing.” *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*. 2019.
- Oh, Sangeun, et al. “FLUID: Flexible user interface distribution for ubiquitous multi-device interaction.” *The 25th Annual International Conference on Mobile Computing and Networking*. 2019.