

# 멀티-모바일 컴퓨팅을 위한 앱 기능 분산 프레임워크

염현석<sup>1</sup>, 박성범<sup>2</sup>, 오상은<sup>3,\*</sup>

아주대학교 <sup>1</sup>AI융합네트워크학과, <sup>2</sup>디지털미디어학과, <sup>3</sup>소프트웨어학과

{<sup>1</sup>sooscv29, <sup>2</sup>ppp3195, <sup>3,\*</sup>sangeunoh}@ajou.ac.kr

## App Feature Distribution Framework for Multi-Mobile Computing

Hyeonseok Yeom<sup>1</sup>, Seongbeom Park<sup>2</sup>, Sangeun Oh<sup>3,\*</sup>

<sup>1</sup>Department of Artificial Intelligence Convergence Network, Ajou Univ.

<sup>2</sup>Department of Digital Media, Ajou Univ.

<sup>3</sup>Department of Software and Computer Engineering, Ajou Univ.

### 요약

최근 다중기기 사용자의 증가로 멀티-모바일 컴퓨팅의 수요 또한 증가 추세에 있다. 이러한 추세에도 불구하고, 앱의 실행 범위는 아직 단일기기에 머물러 있어 폭넓은 멀티-모바일 컴퓨팅을 지원하기에는 제한적이다. 본 연구에서는 앱의 실행범위를 다중기기로 확장하는, 멀티-모바일 컴퓨팅을 위한 앱 기능 분산 프레임워크를 제안한다. 제안한 프레임워크를 통해 사용자는 즉석으로 앱 기능을 분산하여 폭넓고 새로운 멀티-모바일 컴퓨팅 사용자 경험을 제공할 수 있다. 본 연구에서는 제안한 프레임워크를 통해 멀티-모바일 컴퓨팅 앱을 기존의 모바일 프레임워크와 동일한 개발 컨벤션으로 개발 가능하여, 결과적으로 새롭고 강력한 멀티-모바일 컴퓨팅 사용자 경험을 쉽게 지원할 수 있음을 입증했다.

### I. 서론

최근 여러 모바일 기기를 소유하고 있는 사용자층이 증가하는 추세를 보이는 가운데, 모바일 기기 제조사는 해당 사용자층에 여러 기기를 혼합해서 사용할 수 있는 컴퓨팅 기술을 지원하고 있다. 삼성 갤럭시와 애플에서는 모바일 기기간 리소스 핸드오프를 지원하며, 특히 애플은 증강현실 기기를 통해 여러 모바일 기기와 함께 공간 컴퓨팅을 구사할 수 있는 시스템을 선보였다.

그러나 현존 모바일 플랫폼은 앱에 폭넓은 멀티-모바일 컴퓨팅을 지원하기에는 제공하는 시스템 리소스가 불충분하고, 이로 인해 앱 실행범위가 단일기기에 제한되어 한계를 보이고 있다. 이러한 멀티-모바일 컴퓨팅을 지원하기 위한 시도가 학계에서도 활발히 이루어지고 있다. Tap[1]은 멀티-모바일 컴퓨팅을 지원할 수 있는 시스템 리소스를 앱 수준 프레임워크로 제공한다. MobilePlus[2]는 앱이 여러 기기의 리소스를 이용할 수 있도록 기기 내 RPC 통로를 확장하여 기존 앱에서도 멀티-모바일 컴퓨팅을 지원한다. 그러나 이러한 연구에서도 아직 앱의 실행범위는 단일기기에 제한되어 있어 폭넓은 멀티-모바일 컴퓨팅 지원하기에는 부족하다.

본 논문에서는 앱 기능을 분산하여 앱의 실행 범위를 다중기기로 확장하는 멀티-모바일 컴퓨팅 프레임워크를 제안한다. 제안한 프레임워크는 사용자가 앱 기능을 즉석으로 분산할 수 있는 멀티-모바일 컴퓨팅을 지원한다. 앱 개발자는 Annotation을 이용하여 간단하게 분산가능한 앱을 개발할 수 있으며, 앱 컴파일 과정에서 주석처리가 관여하여 앱의 컴포넌트가 시스템과 긴밀하게 연결된다. 본 연구의 개념증명은 안드로이드 오픈소스 프로젝트를 시스템 수정없이 확장한 소프트웨어로 구현되었으

며, 이를 통해 구현된 앱은 분산할 기능 하나 당 약 8줄의 추가적인 LOC(Line of Code)와, 기존 앱과 큰 차이가 없는 용량을 가져 타당한 수준의 유용성을 보인다.

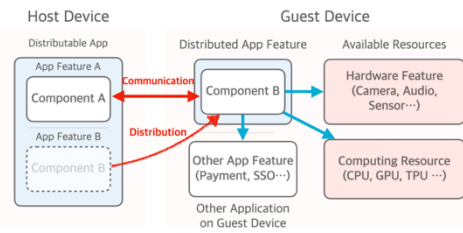


Figure 1. 분산된 앱 기능의 용례

### II. 사용사례 모델

사용자가 분산가능한 앱 기능을 사용하고자 할 때, 시스템은 주변 기기의 리스트를 보여주어 사용자가 직접 앱 기능을 분산할 기기를 선택 할 수 있도록 도와준다.

앱 기능은 앱 컴포넌트의 묶음으로 구현된다. 앱 컴포넌트는 개발자가 정의한 대로 기기의 여러 기능 및 리소스를 활용할 수 있다. 제안한 프레임워크는 이렇게 정의된 앱 기능을 분산하여 원격기기를 그대로 이용하거나, 호스트 기기의 컴포넌트와 협력하는 등 여러 형태의 멀티-모바일 컴퓨팅을 구사할 수 있다.

예를 들어, 스마트 TV에서 콘텐츠를 시청 시 추가적인 결제가 필요한 경우가 있다. 그러나 스마트 TV는 사용자 인터페이스의 한계로 결제에 불편함이 있다. 이 때 사용자는 즉석으로 자신의 스마트폰에 결제기능을 분산하여 결제를 진행할 수 있다. 이외에도 숙박시설에서는 기존의 카드키를 대체하여, 홈 IoT 앱에 등록된 홈키 기능을 고객의 모바일 기기에 즉석으로 분산하는 방식으로 새로운 사용자 경험을 제공할 수 있다.

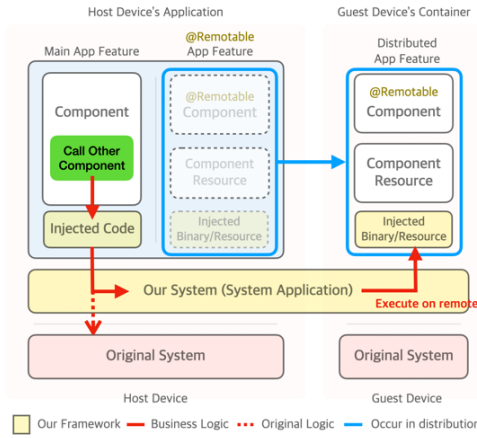


Figure 2. 앱 분산 및 상호작용 흐름

### III. 시스템 모델

앱 기능을 구성하는 앱 컴포넌트는 시스템과 깊이 연관되어 실행된다. 우리의 프레임워크는 분산된 컴포넌트 사이에서 기존의 시스템과 동일한 역할을 하는 새로운 시스템과, 이를 컴포넌트와 연결해주는 코드를 삽입할 주석처리기로 구성된다. Figure 2는 우리의 프레임워크를 통해 만들어진 앱의 분산 및 상호작용 흐름에 대한 개요를 나타낸다. 이 섹션에서는 컴포넌트 분산 시 나타나는 시스템과의 문제에 대한 해결 방법에 대해 하부항목으로 기술한다.

#### 1) 앱 기능 분산 시점 특징

우리는 앱 기능 분산 요청을 런타임에 파악할 수 있어야 한다. 컴포넌트의 실행은 항상 시스템을 거치기 때문에, 주석처리기를 이용하여 컴포넌트 실행 명령을 우리의 시스템으로 전달하기 위한 코드를 주입한다. 시스템은 명령 인자 및 개발자의 Annotation에 따라 사용자에게 연결된 기기가 나열된 선택자를 보여줌으로써 사용자의 액션에 따라 분산을 런타임에 특정할 수 있다.

#### 2) 원격 기기에서의 실행

주석처리기는 분산된 컴포넌트가 원격기기에서 실행될 수 있도록, 컴포넌트와 관련된 코드 의존성 및 리소스를 컴파일 단계에서 파악하고, 의존성 분리 과정을 거쳐 로컬 및 원격기기에서 실행 가능하도록 컴파일한다. 또한 컴포넌트가 원격기기에서 실행되기 위해서는 컴포넌트의 정보가 해당 시스템에 미리 등록이 되어야 하는데, 이를 위해 분산될 컴포넌트를 실행할 컨테이너 앱을 해당 컴포넌트의 메타데이터와 같이 실시간으로 컴파일하여 앱 컴포넌트와 같이 전송한다. 원격 기기의 시스템은 컨테이너 앱을 통해 앱 컴포넌트를 동적으로 로드하고 기존 앱과 서명을 비교하여 안전하게 실행할 수 있다.

#### 3) 컴포넌트간 인터랙션 처리

우리는 물리적으로 다른 공간에서 실행중인 컴포넌트간 모든 인터랙션을 하나의 앱에서 실행되는 컴포넌트간의 인터랙션처럼 보이도록 처리해 주어야 한다. 우리의 시스템은 컴포넌트간 호출 인터페이스를 동적으로 확장하고, 호출을 RPC로 치환하여 인터랙션을 유지한다. 이러한 호출 간에는 기기 중속적인 리소스 교환 및 접근이 일어날 수 있는데, 이 때는 프록시 패턴을 통해 인터랙션이 제한없이 일어날 수 있도록 지원한다.

앱	분산할 기능 수	LOC <sup>1)</sup>	추가 LOC <sup>1)</sup>	기존 용량 <sup>2)</sup> (KB)	수정 후 용량 <sup>2)</sup> (KB)
스마트TV	1	1044	8	145.7	156.8
키오스크	2	824	15	510.6	548.5

1) Line of Code, 2) 해당 앱 기능의 총 용량을 나타냄

Table 1. 기존 앱의 수정 후 차이

### III. 평가

본 연구에서는 평가를 위해, 제안한 프레임워크를 이용하여 개발한 앱의 유용성 평가를 진행했다. Table 1의 앱은 평가를 위해 구현된 앱이며, 개념증명을 탑재한 안드로이드 레퍼런스 기기에서 정상적으로 멀티-모바일 컴퓨팅을 구사했다. Table 1은 각 앱의 분산을 위한 코드 추가 전/후 비교하는 내용을 나타낸다.

추가 LOC는 개발자가 분산 가능한 앱을 개발하고자 할 때 추가적으로 발생하는 엔지니어링 부담을 나타내는 척도로 볼 수 있다. 개발자는 분산할 기능에 Annotation을 적용하게 되므로, 추가 LOC는 전체 LOC와 상관관계가 없으며, 분산할 기능 수에 비례해 증가한다. 평가의 결과로 추가 LOC는 일반적으로 앱을 작성하는데 소요되는 전체 LOC에 비해 1~2%의 미만의 비중을 차지함을 볼 수 있으며, 결론적으로 이는 분산의 적용이 비교적 쉽다는 것을 의미한다. 또한, 컴파일 과정에서 추가되는 앱 용량 또한 큰 변화를 나타내지 않아 충분히 납득가능한 수준으로 보인다.

### IV. 결론

본 연구에서는 앱 컴포넌트를 분산하여 멀티-모바일 컴퓨팅을 지원하는 모바일 플랫폼을 제안했다. 제안한 프레임워크로 보다 폭넓은 앱 기능을 멀티-모바일 컴퓨팅을 지원할 수 있고, 기존 모바일 플랫폼의 확장으로 만들어져 기존 모바일 애플리케이션과 완벽하게 호환이 가능했다. 또한, 분산된 앱 컴포넌트는 하나의 앱처럼 작동하여 기존 플랫폼과 동일한 보안수준을 가진다. 제안한 프레임워크의 평가로, 멀티-모바일 컴퓨팅을 위한 라이브러리를 개발자가 비교적 쉽게 이용할 수 있을 것으로 보여 유용성을 입증했다. 향후 연구로, 본 연구로 인해 새롭게 나타나는 컴포넌트간 실행 관계 유형에 대한 정의 및 해당하는 대표적인 사용사례에 대한 연구와, 컴포넌트간 호출로 인해 발생하는 전역변수의 동시성 문제를 해결하고자 한다.

### 참고 문헌

[1] 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.

[2] Oh, Sangeun, et al. "Mobile plus: Multi-device mobile platform for cross-device functionality sharing." Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services. 2017.