

앱 변환 프레임워크: Runtime Configuration

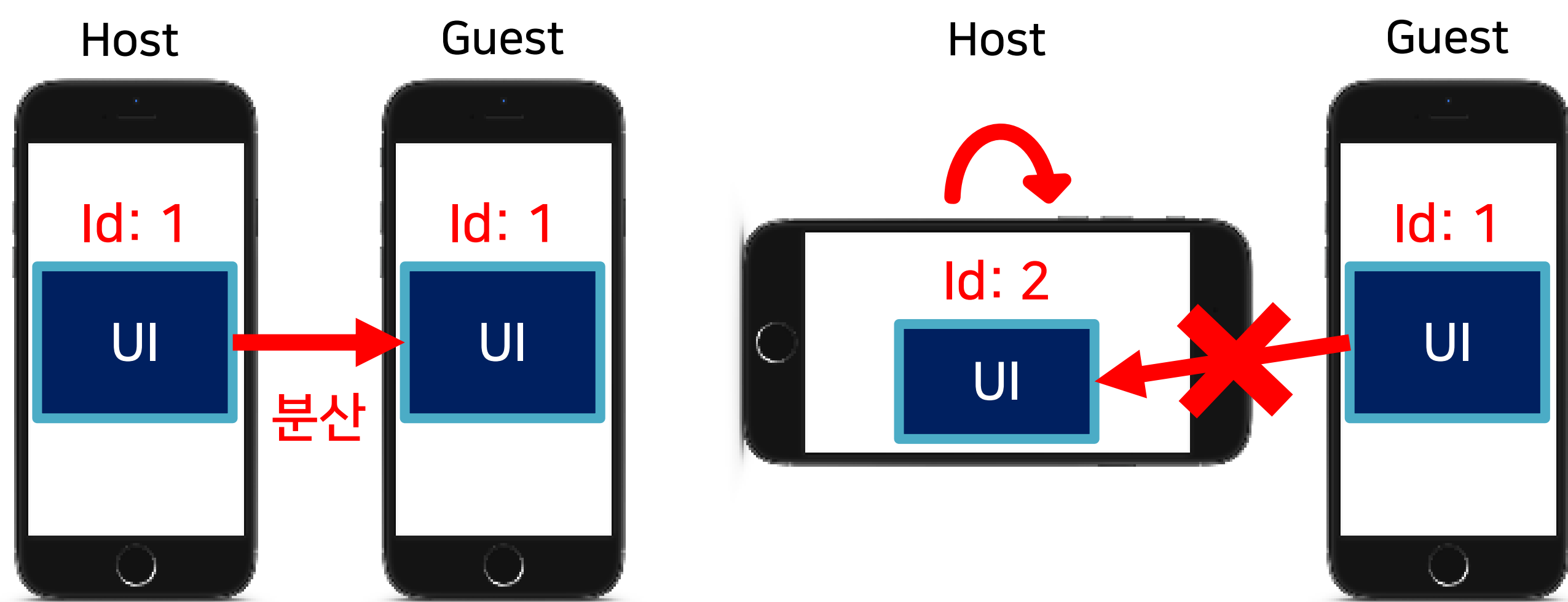
이름 이지은

지도교수 오상은

연구배경

Host App과 Guest App이 존재하고, <그림1>처럼 Host에서 Guest로 UI를 분산 했다고 가정했을 때 Host의 UI id와 Guest의 분산된 UI id는 같다. 이 상황에서 <그림2>처럼 Host가 회전하면 이전과 다른 layout으로 화면을 구성하게 되는데, 이때 같은 역할을 수행하는 Host의 UI id가 달라진다. 이러한 경우에 Guest의 터치, 키보드 입력 이벤트는 정상적으로 Host App에 전달될 수 없다.

→ “회전 시 Host와 Guest App의 상호작용 상실”



<그림1> UI 분산

<그림2> 상호작용 상실

선행 연구 조사

선행 연구인 RuntimeDroid의 경우, Activity가 재시작을 하지 않게 함으로써 모든 Activity Data가 live 상태를 유지하도록 한다. Activity가 재시작하지 않으면 새 resource가 자동으로 load되지 않는 문제가 있는데, HotR을 통해 이를 해결한다.

연구내용

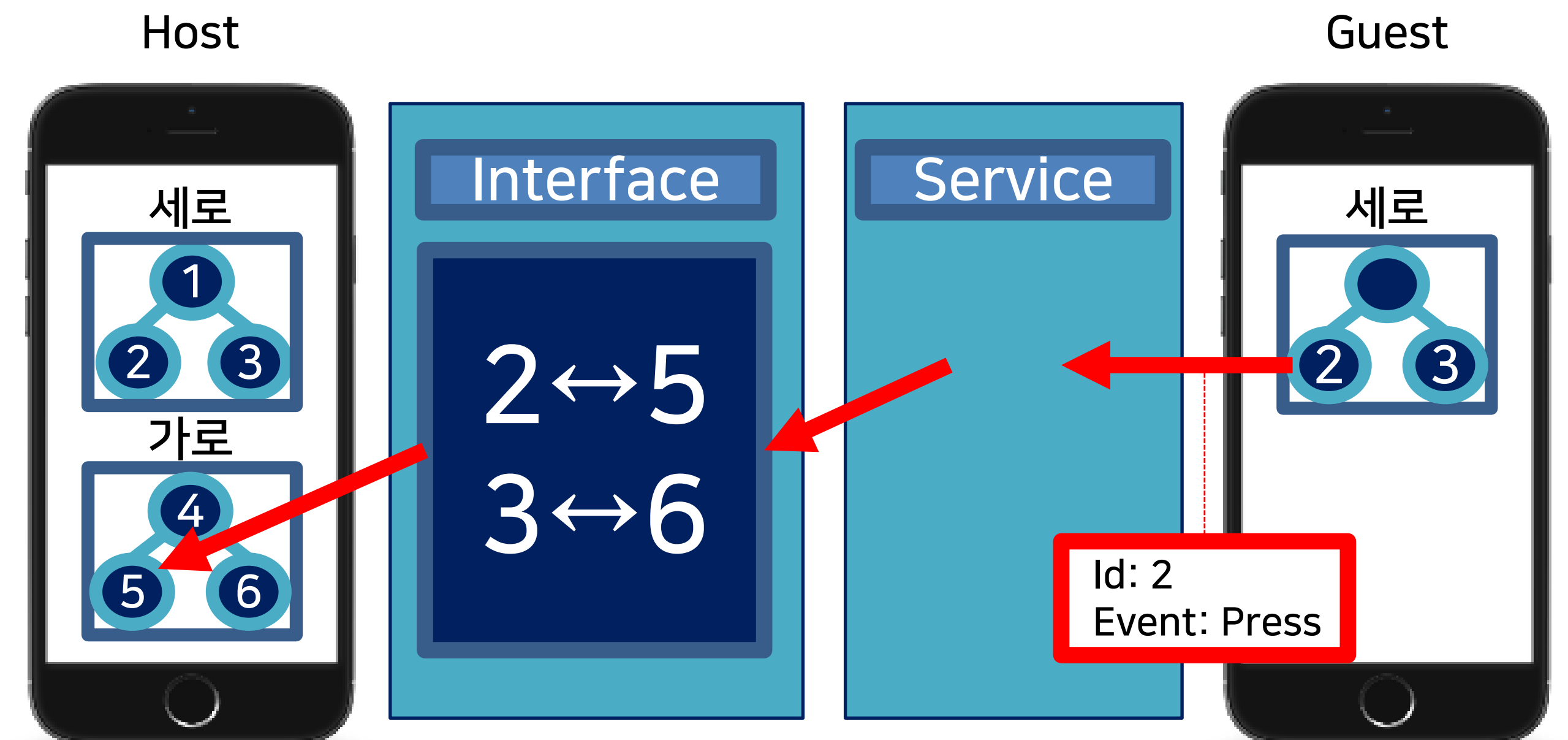
본 연구는 Activity의 재시작 여부와 무관하게 Activity의 구성 요소가 framework와 동일하게 작동하도록 지원한다.

→ “Activity 재시작 시 Activity 구성요소의 기능 유지”

제안기법

본 연구의 핵심 아이디어는 UI mapping이다. Host App에는 세로 모드용 UI tree와 가로 모드용 UI tree가 각각 존재한다.

Host App이 세로 모드에서 Guest로 UI를 분산하면 Guest App에는 <그림3>과 같이 세로 UI tree가 분산된다. 이때 Host App을 회전하여 가로 모드로 전환하면 Guest App에는 그대로 세로 UI tree가 남아있지만, Host App에는 가로 UI tree가 적용된다.



<그림3> UI mapping 기법

위 상황에서 Interface에 id 2를 가진 UI는 id 5를 가진 UI와 같고, id 3을 가진 UI는 id 6을 가진 UI와 같다는 정보가 있다면, Guest에서 id 2인 UI에 대해 Press event가 발생했을 때, Host에서 id 5를 가진 UI가 정상적으로 작동한다.

결과 및 분석

환경

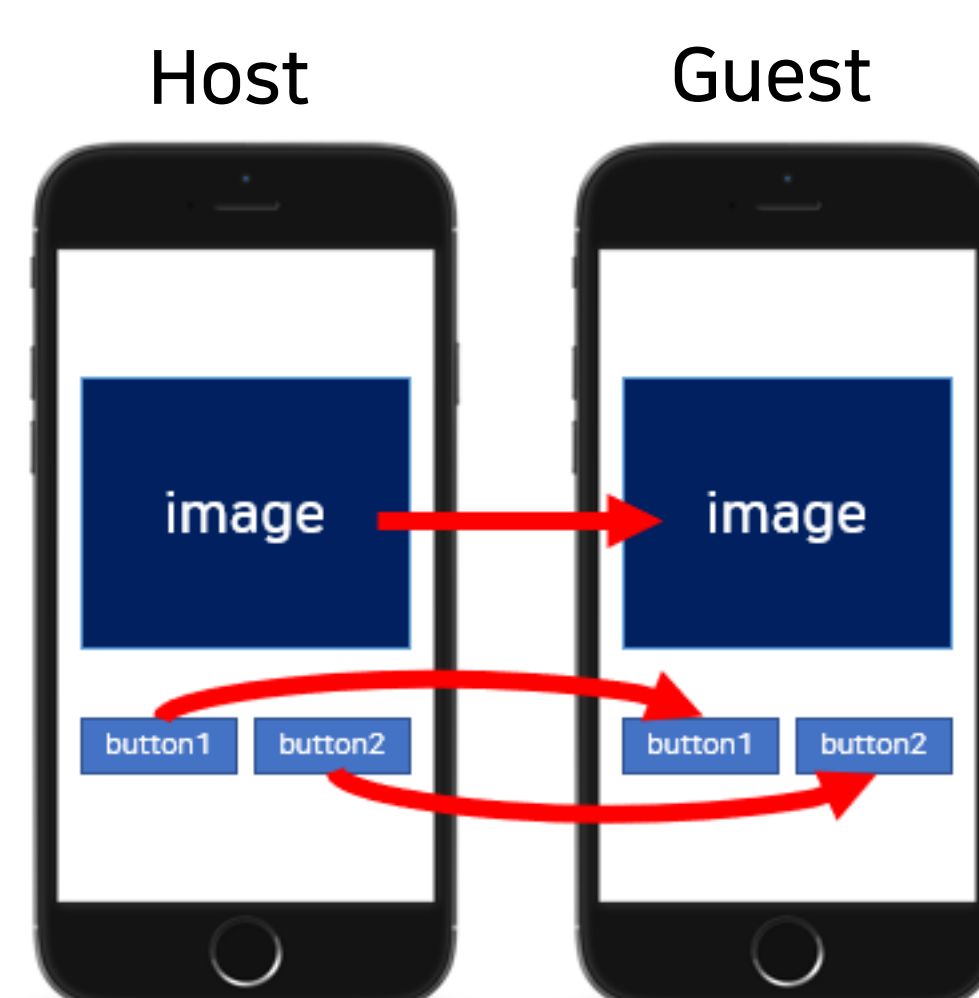
- Google Pixel 5 (Android 11 탑재)
- Google Pixel 4a (Android 11 탑재)
- 2.4GHz 대역폭의 wifi 네트워크

테스트

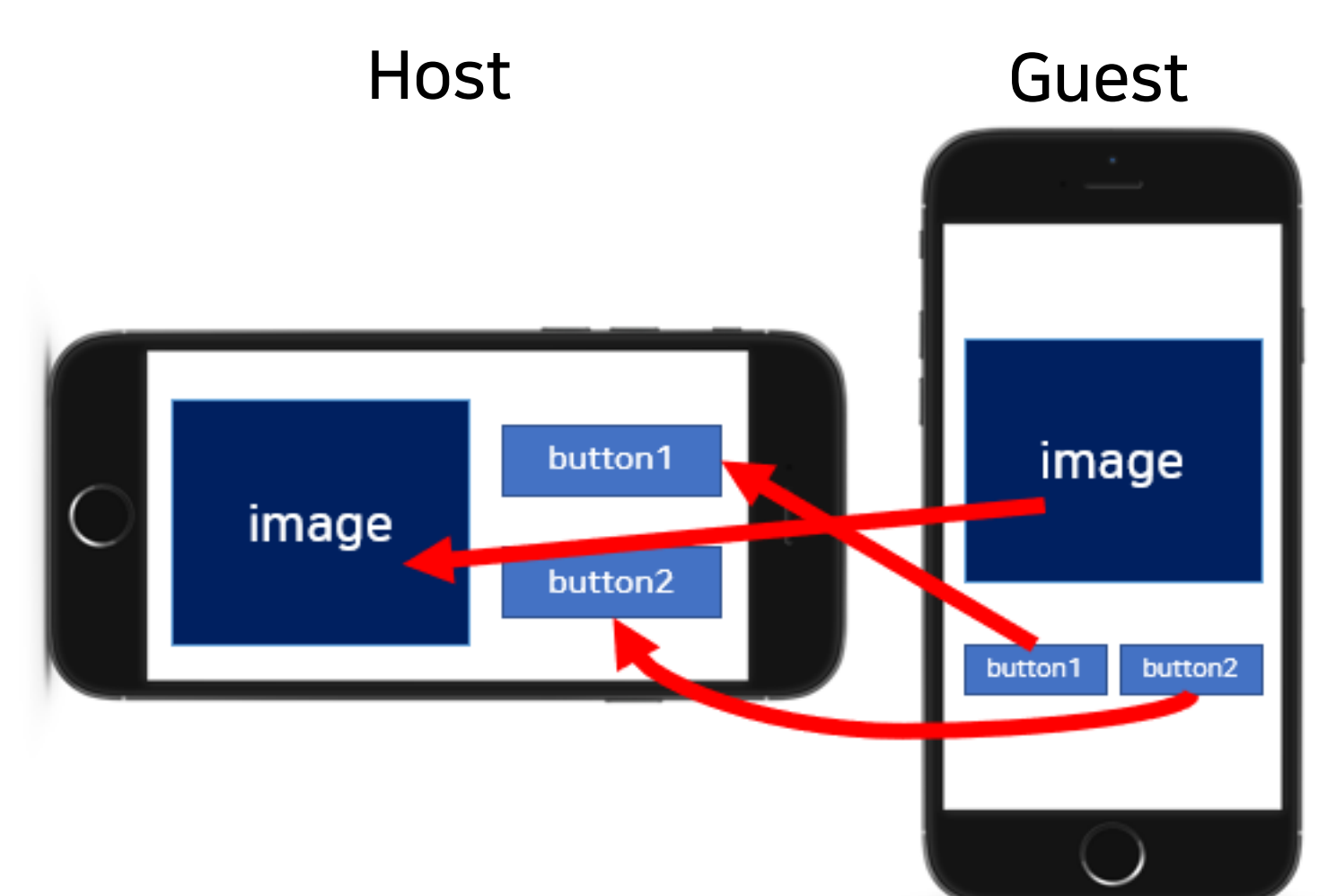
1. Host(Google Pixel 5)에서 Guest(Google Pixel 4a)로 5가지의 UI 분산
2. Host 가로로 회전
3. Guest의 분산된 UI에 Press Event 발생 시 결과 확인

결과

- <그림4>와 같이 Host의 세로 UI가 Guest에 분산됨
- Host 회전 시, Host UI는 가로 UI로 전환됨
- Guest UI에 Press Event 발생 시, Host의 mapping된 UI에 Event가 정상적으로 전달됨



<그림4> 분산 테스트



<그림5> 상호작용 테스트

➔ “회전 시 Host와 Guest App의 상호작용 유지”