

이기종 플랫폼 간 UI 분산 지원을 위한 모바일 프레임워크

이름 이지은

지도교수 오상은

연구배경

최근 다양한 폼·팩터의 디스플레이들이 등장하면서 UI들을 여러 기기 디스플레이로 분산하고 활용하는 다중 서피스 상호작용에 관한 연구가 많이 이루어지고 있다.

하지만 현재까지 제안된 연구들은 <표1>에서 확인할 수 있듯이 동종 플랫폼 간의 UI 분산만을 지원하여 다른 플랫폼들로 이루어진 IoT 환경에서 적용하기 어렵거나, 이기종 플랫폼 간의 UI 분산이 가능하더라도 OS의 수정이 불가피하다.

	FLUID	FLUID-XP	Colladroid
OS 수정 여부	O	O	X
이기종 간 UI 분산 여부	X	O	X

<표1> 선행 연구

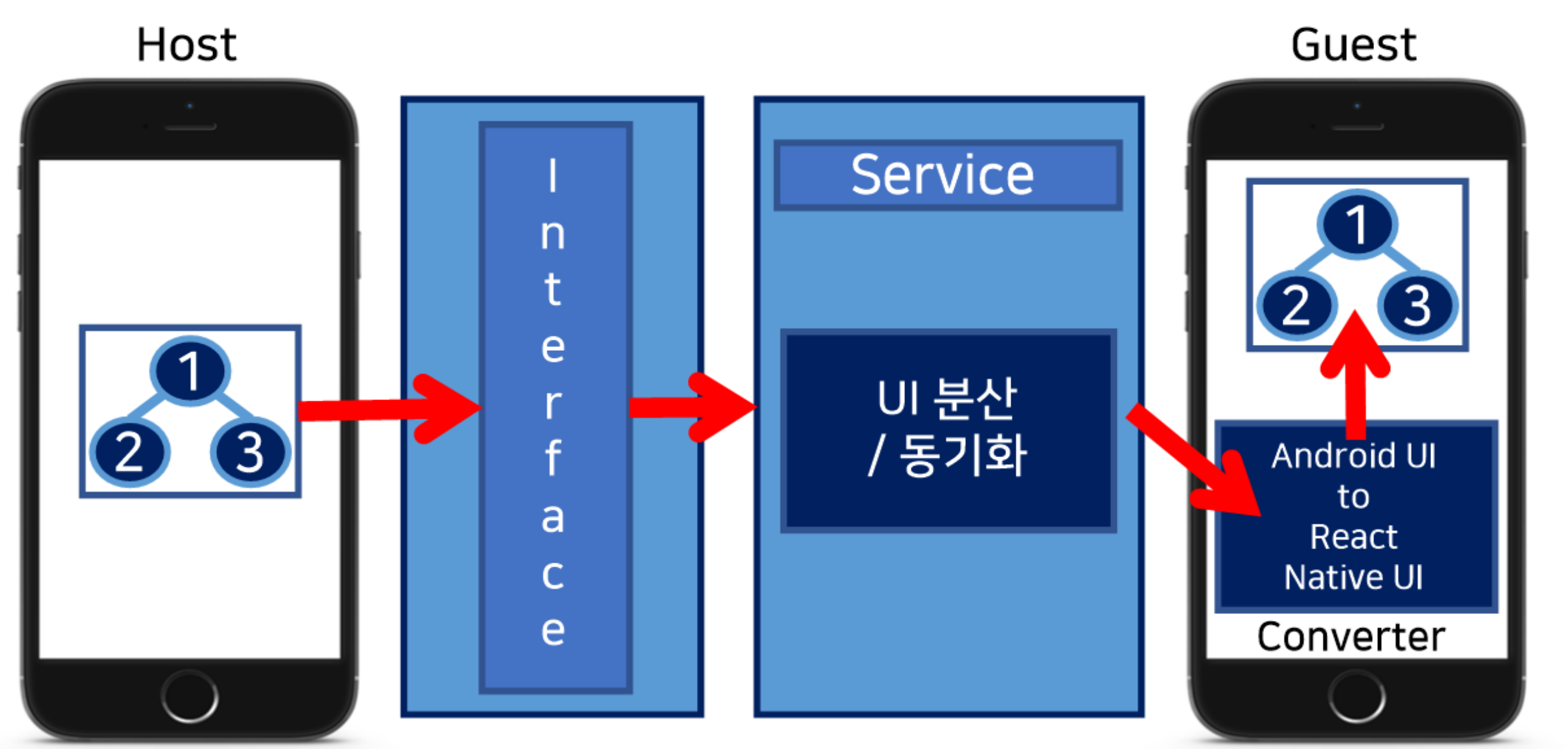
연구내용

본 연구는 네이티브 UI 요소에 대응하는 웹 UI 요소를 새로 만들어 다른 기기로 분산할 수 있도록 지원한다.

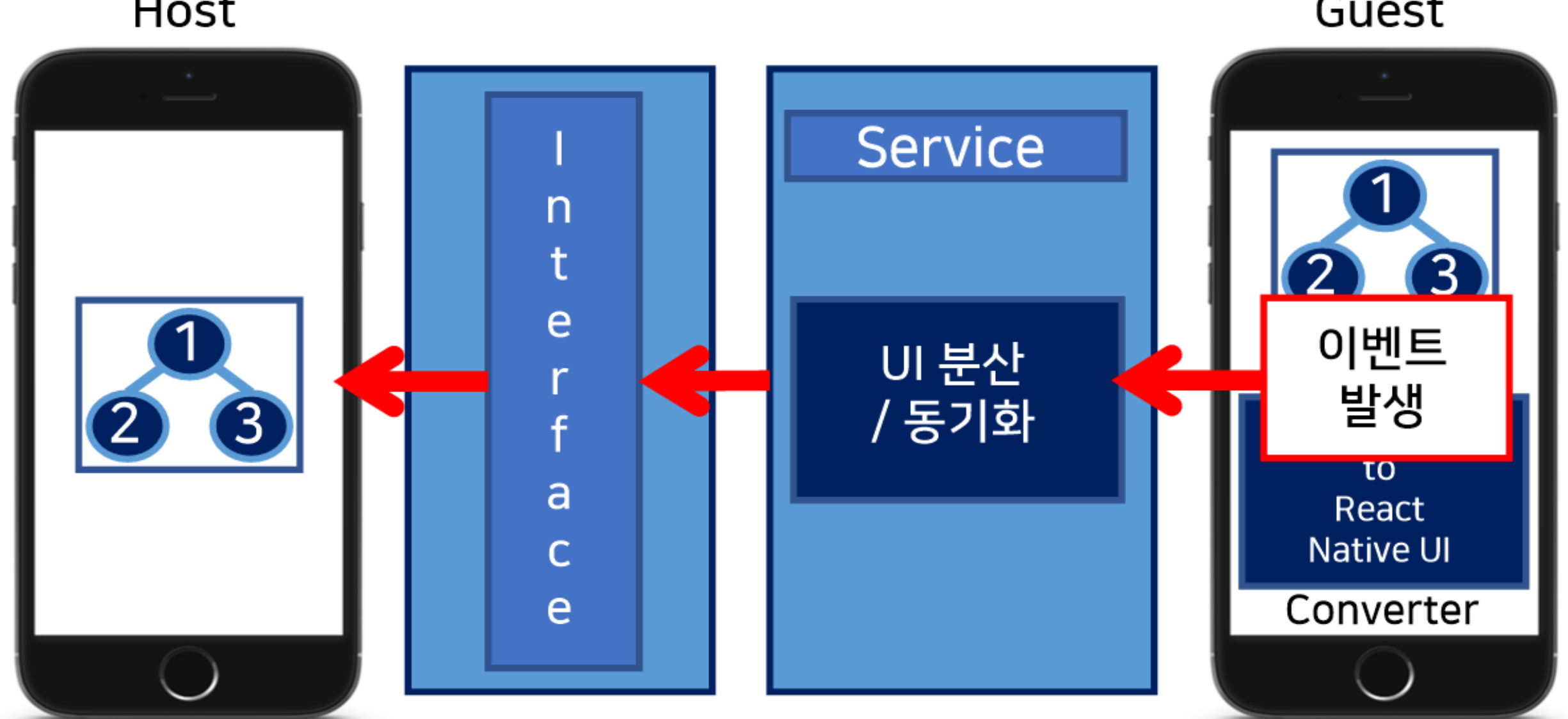
제안기법

본 연구는 Layout을 유지하면서 Android native UI를 React Native UI로 변환하는 메커니즘을 제안한다.

작동하는 방식은 아래의 <그림1>, <그림2>와 같다.



<그림1> UI 분산 방식



<그림2> 분산 UI 이벤트 작동 방식

➡ "OS 수정 없이 이기종 플랫폼 간 UI 분산 가능"

결과 및 분석

환경

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

테스트

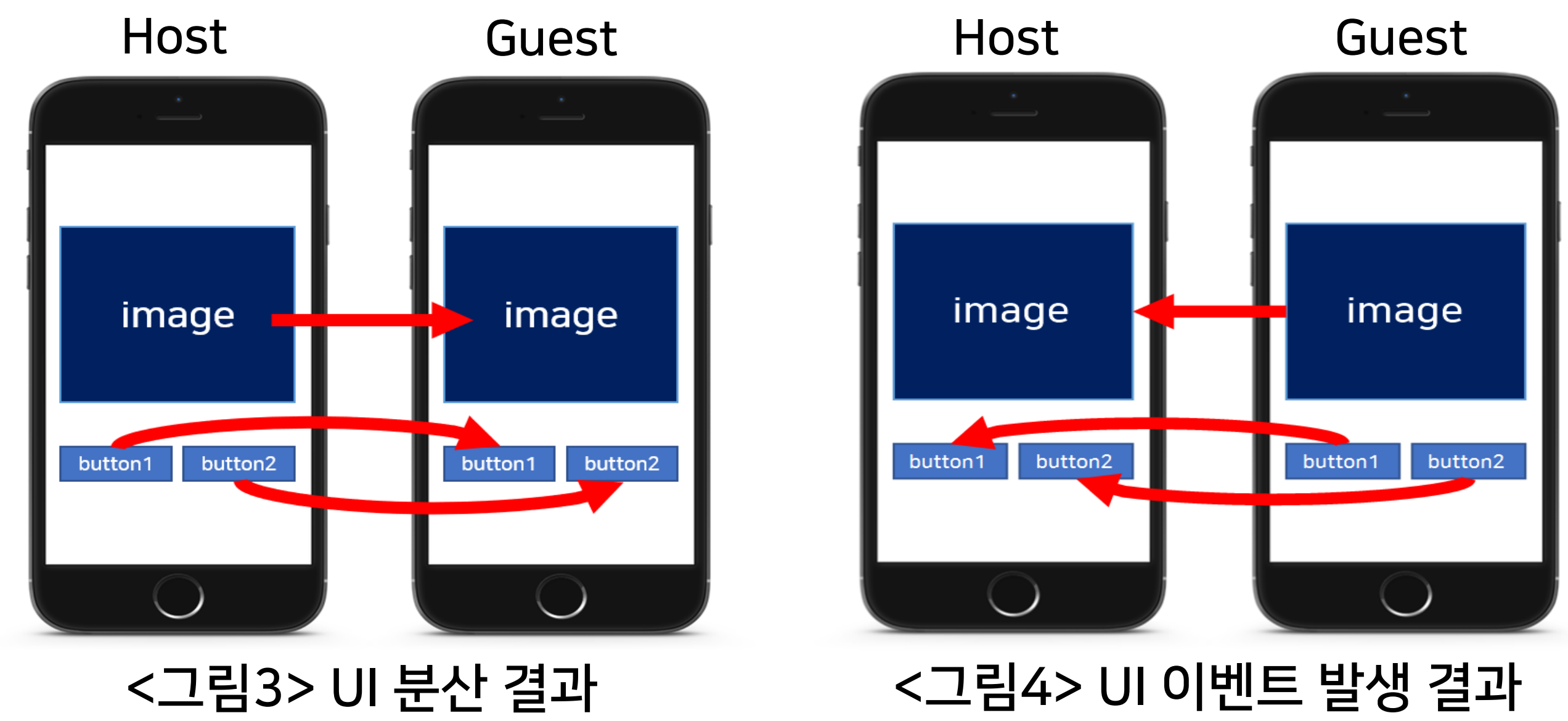
- Host(Google Pixel 5)에서 Guest(Google Pixel 4a)로 5가지의 UI 분산 시 소요 시간 측정

결과

- Host_process(Host Device 처리 시간) → 매우 작음
- Render_time → UI마다 큰 차이가 없음
- Transfer → UI마다 다름
- <그림4>와 같이 Guest의 분산된 UI에 Press Event를 주었을 때 Host UI가 정상 작동함

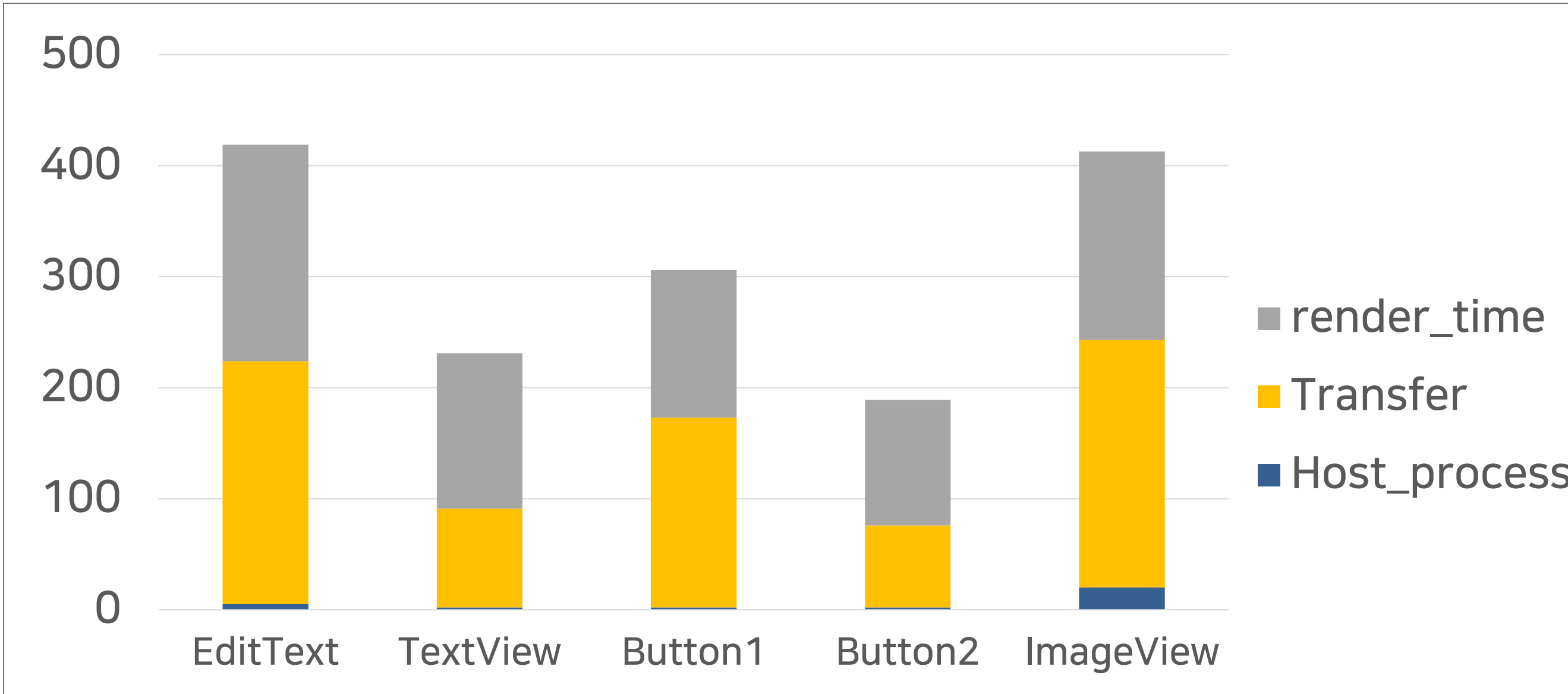
분석

- <그림5>과 <표2>를 보면 Transfer Time은 Text 기반 UI들의 경우에 보내는 정보의 크기가 유사하지만 차이가 큰 것으로 보아 대부분의 변동의 원인으로 추정됨



<그림3> UI 분산 결과

<그림4> UI 이벤트 발생 결과



<그림5> UI 분산 소요 시간

	Host_process	Transfer	render_time	total
EditText	5	219	195	419
TextView	2	89	140	231
Button1	2	171	133	306
Button2	2	74	113	189
ImageView	20	223	170	413

<표2> UI 분산 소요 시간