

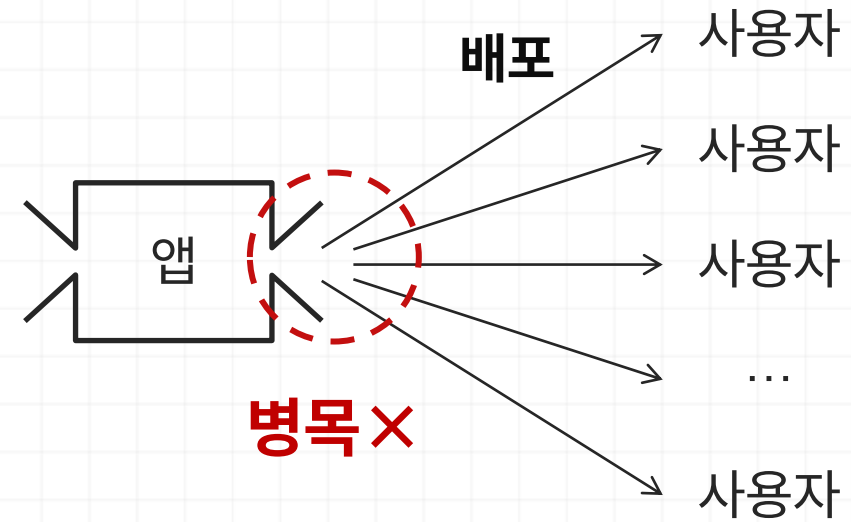
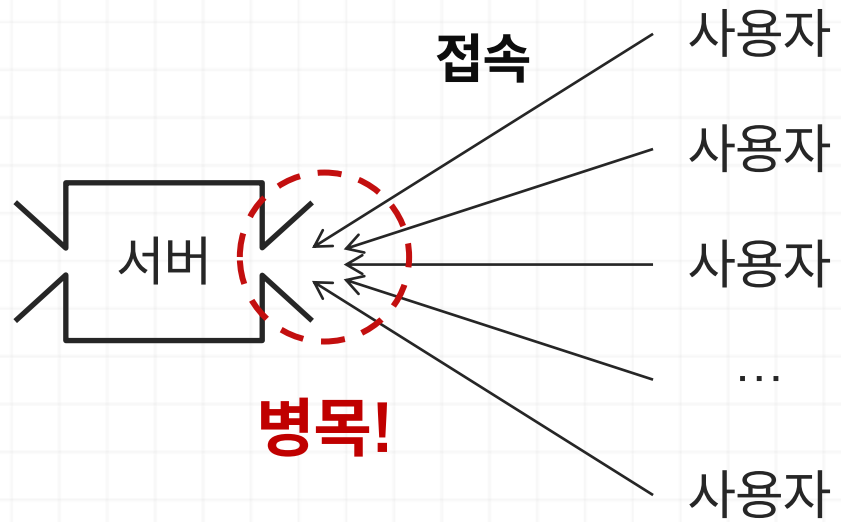
# 앱 Scalability와 아키텍처

---

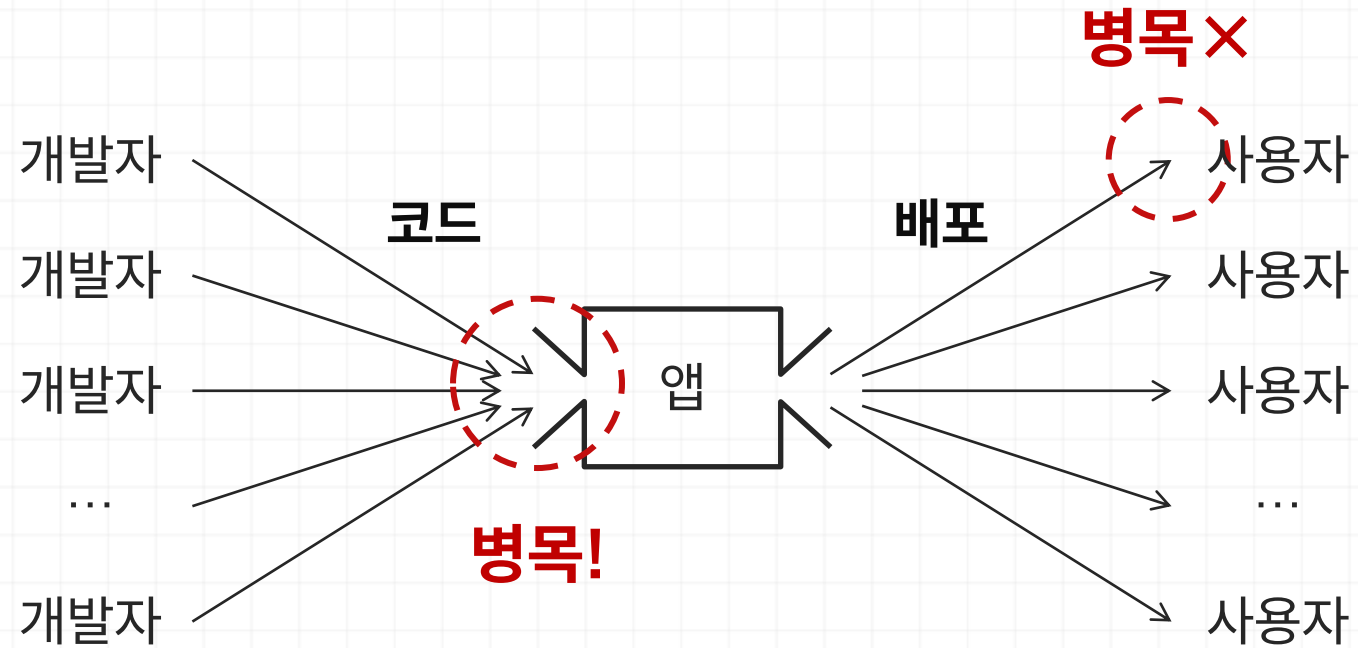
모바일 개발자에게 Scalability란 어떤 의미인지 생각해봅시다.  
100명의 앱 개발자가 하나의 슈퍼앱을 개발하면 어떤 문제들이 발생할까요?  
앱의 빠른 성장을 지탱해주는 아키텍처의 특징이 뭔지 알아봅시다.



# 백엔드와 모바일 개발의 차이



# 앱 개발의 병목



# Merge Request 20시간 후...



Request to merge [feature/117-117-117](#) into [main](#)

The source branch is **117 commits behind** the target branch



# Scale과 연관된 다양한 문제들

## 개발자가 겪는 문제

- 빌드 시간 증가
- 코드 충돌 증가
- QA, 디버깅 시간 증가
- 끊이지 않는 회귀 버그
- 개발 소요 시간 증가

...

## 사용자가 겪는 문제

- 앱 Startup 시간 증가
- 과도한 자원 사용 (배터리, 네트워크)
- 앱 용량 증가
- 버그 증가
- 앱 안정성 하락

...

# 확장 가능한 앱 아키텍처의 특징

## 명확한 역할 구분

객체/모듈은  
하나의 역할만 수행하며,  
데이터의 흐름을 쉽게  
따라갈 수 있다.

## 변화에 유연

의존성은 decoupling  
되어있고, 구조가 단순하기  
때문에 유연하다.

## 테스트가 용이

유닛 테스트가 어렵다면  
구조 개선이 필요하다는 신호이며,  
테스트는 품질 관리에 유용하다.

# 확장 가능한 앱 아키텍처의 특징

## 1부 코드 레벨 아키텍처

로직을 분산시키고 객체를 조립해  
재사용 가능한 코드를 만드는 방법

## 변화에 유연

의존성은 decoupling  
되어있고, 구조가 단순하기  
때문에 유연하다.

## 테스트가 용이

유닛 테스트가 어렵다면  
구조 개선이 필요하다는 신호이며,  
테스트는 품질 관리에 유용하다.

# 확장 가능한 앱 아키텍처의 특징

## 1부

### 코드 레벨 아키텍처

로직을 분산시키고 객체를 조립해  
재사용 가능한 코드를 만드는 방법

## 2부

### 모듈 아키텍처

유지 보수를 고려한 코드 구조,  
여러 개발자가 동시에  
효율적으로 일할 수 있는 방법

### 테스트가 용이

유닛 테스트가 어렵다면  
구조 개선이 필요하다는 신호이며,  
테스트는 품질 관리에 유용하다.



# 확장 가능한 앱 아키텍처의 특징

## 1부

### 코드 레벨 아키텍처

로직을 분산시키고 객체를 조립해  
재사용 가능한 코드를 만드는 방법

## 2부

### 모듈 아키텍처

유지 보수를 고려한 코드 구조,  
여러 개발자가 동시에  
효율적으로 일할 수 있는 방법

## 3부

### 자동화된 테스트

유닛 테스트, 스냅샷 테스트,  
통합 테스트, E2E 테스트