



## 231015 10. 알림 시스템 설계

최근 많은 프로그램이 채택한 인기 있는 기능

각종 정보를 고객에게 비동기적으로 제공

단순히 모바일 푸시 알림에 한정되지 않음

### 알림 시스템의 종류

- 모바일 푸시
- SMS 알림
- 이메일

### ▼ 문제 이해 및 설계 범위 확정

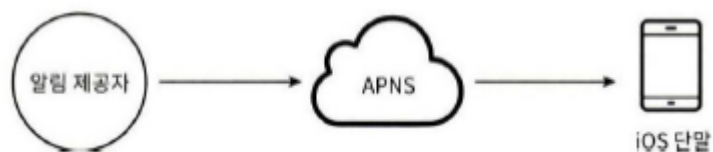
알림 시스템이 어떻게 구현되는지에 대한 깊은 이해가 있어야 확장성 높은 알림 시스템 구축 가능

### ▼ 2. 개략적 설계안 제시 및 동의 구하기

#### ▼ 알림 유형별 지원 방안

알림 메커니즘이 어떻게 동작하는지

#### iOS 푸시 알림



- 알림 제공자
  - 알림 요청을 만들어 APNS로 보내는 주체

- 단말 토큰
  - 알림 요청을 보내는 데 필요한 고유 식별자
- 페이로드
  - 알림 내용을 담은 JSON 딕셔너리

```
{
  "aps": {
    "alert": {
      "title": "Game Request",
      "body": "minu wants to",
      "action_loc_key": "PLA"
    },
    "badge": 5
  }
}
```

- APNS
  - 애플이 제공하는 원격 서비스
  - 푸시 알림을 iOS 장치로 보내는 역할
- iOS 단말
  - 푸시 알림을 수신하는 사용자 단말

## Android 푸시 알림



iOS 푸시 알림과 비슷한 절차

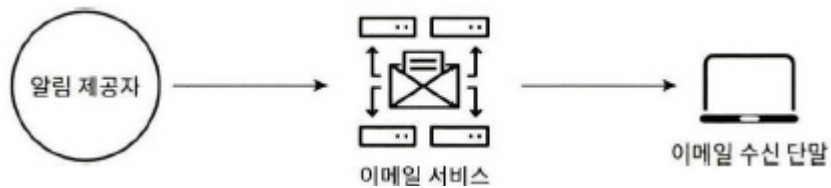
APNS 대신 FCM(Firebase Cloud Messageing)을 사용함

## SMS 메세지



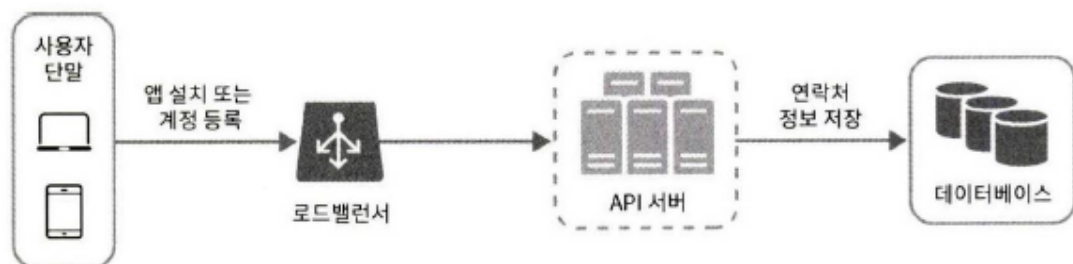
보통 제3사업자의 서비스(트윌리오, 넥스모, ...)를 많이 이용  
사용 서비스라 이용요금을 내야 함

## 이메일



고유 이메일 서버를 구축하거나 사용 이메일 서비스 이용  
사용 이메일 서비스(센드그리드, 메일chimp)를 더 많이 이용함

### ▼ 연락처 정보 수집 절차

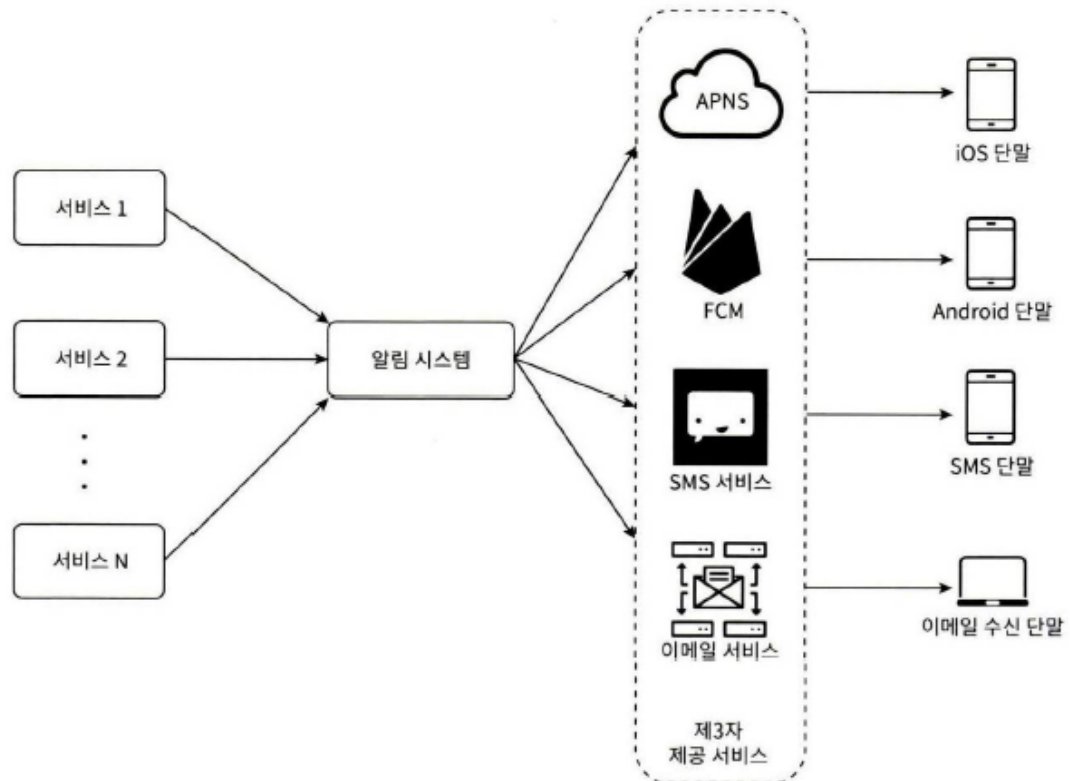


알림을 보내려면 모바일 단말 토큰, 전화번호, 이메일 주소 등의 정보가 필요

앱을 설치하거나 처음으로 계정을 등록할 때 API 서버가 사용자의 정보를 수집하여  
DB에 저장

### ▼ 알림 전송 및 수신 절차

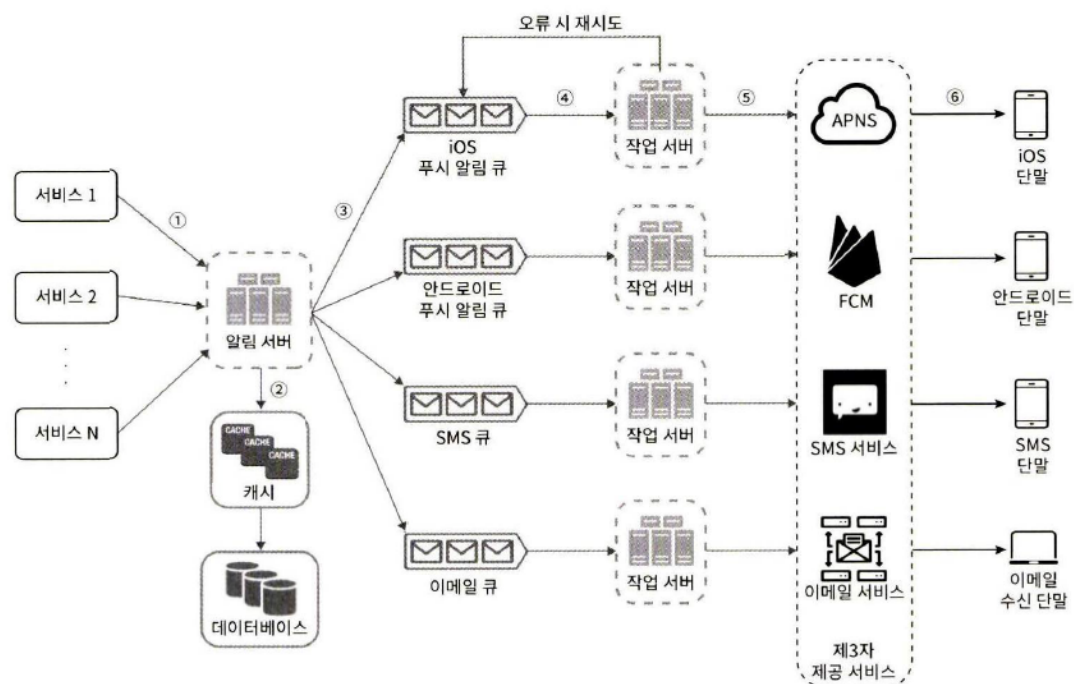
개략적 설계안(초안)



- 1...N 서비스
  - 해당 서비스들은 각각 마이크로서비스, 크론잡, 분산 시스템 컴포넌트 등으로 구성되어 있음
  - ex) 사용자에게 납기일을 알리고자 하는 과금 서비스, 배송 알림을 보내는 쇼핑몰 웹 사이트, ...
- 알림 시스템
  - 알림 전송, 수신 처리의 핵심
  - 서버 1개만 사용한다고 가정
  - 서비스에 알림 전송을 위한 API 제공
  - 제3자 서비스에 전달할 알림 페이로드 생성
- 제3자 서비스
  - 사용자에게 알림을 실제로 전달하는 역할
  - 확장성을 필수로 고려해야 함
  - 특정 서비스는 특정 시장에서 사용 불가능할 수도 있음
    - FCM은 중국에서 사용 불가

- 단말
  - 사용자가 알림을 수신 받는 곳
- 문제점
  - SPOF
    - 알림 서비스 서버가 하나이기 때문에 발생
  - 규모 확장성
    - 한 대 서비스로 푸시 알림에 관한 모든 것을 처리하기 때문에 DB나 캐시 등 중요 컴포넌트의 규모를 개별적으로 늘릴 방법이 없음
  - 성능 병목
    - 알림을 처리하고 보내는 것은 자원을 많이 필요로 하는 작업일 수 있음
    - 모든 것을 한 서버로 처리하면 사용자의 트래픽이 많이 몰리는 시간에는 시스템이 과부하 상태에 빠질 수 있음

### 개략적 설계안(개선)



- 개선한 점

- DB와 캐시를 알림 시스템의 주 서버에서 분리
- 알림 서버 증설
- 알림 서버가 자동으로 수평적 규모 확장이 이루어질 수 있도록 함
- 메시지 큐를 이용해 시스템 컴포넌트 사이의 강한 결합을 끊음
- 1...N 서비스
  - 알림 시스템 서버의 API를 통해 알림을 보낼 서비스들
- 알림 서버
  - 알림 전송 API
    - 스팸 방지를 위해 사내 서비스 또는 인증된 클라이언트만 사용 가능
  - 알림 검증
    - 이메일 주소, 전화번호 등에 대한 기본적 검증을 수행
  - DB or 캐시 질의
    - 알림에 포함시킬 데이터를 가져오는 기능
  - 알림 전송
    - 알림 데이터를 큐에 넣음
    - 본 설계안의 경우 하나 이상의 메시지 큐를 사용하므로 알림을 병렬적으로 처리 가능
- 캐시
  - 사용자 정보, 단말 정보, 알림 템플릿 등을 캐싱
- DB
  - 사용자, 알림, 설정 등 다양한 정보를 저장
- 메세지 큐
  - 시스템 컴포넌트 간 의존성을 제거하기 위해 사용
  - 다량의 알림이 전송되어야 하는 경우를 대비한 버퍼 역할도 함
  - 본 설계에서는 알림의 종류별로 별도 메시지 큐를 사용
  - 제3자 서비스 가운데 하나에 장애가 발생해도 다른 종류의 알림은 정상 동작
- 작업 서버
  - 메시지 큐에서 전송할 알림을 꺼내서 제3자 서비스로 전달하는 역할

- 제3자 서비스
- 단말

### **알림을 사용자에게 전송하는 과정**

1. API를 호출하여 알림 서버로 알림을 전송
2. 알림 서버가 사용자 정보, 단말 토큰, 알림 설정 같은 메타데이터를 캐시나 DB에서 추출
3. 알림 서버가 전송할 알림에 맞는 이벤트를 만들어서 해당 이벤트를 위한 큐에 넣음
4. 작업 서버가 메시지 큐에서 알림 이벤트를 꺼내서 알림을 제3자 서비스로 전송
5. 제3자 서비스가 사용자 단말로 알림 전송

## **▼ 3. 상세 설계**

### **▼ 안정성**

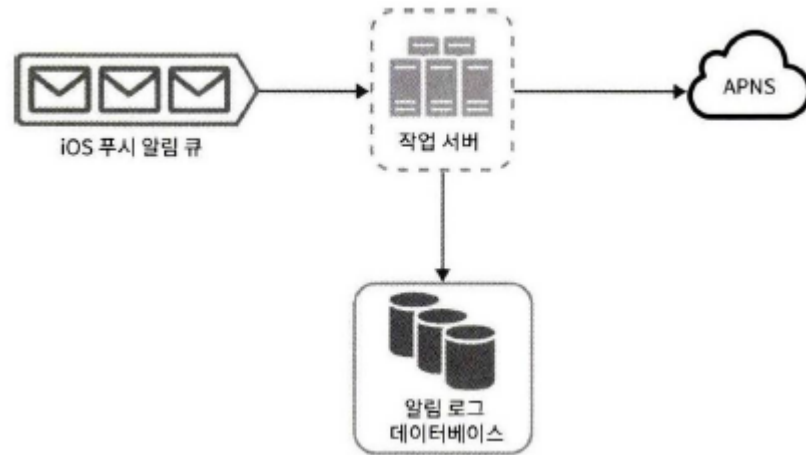
분산 환경에서 운영될 알림 시스템을 설계할 때는 안정성을 확보하기 위한 사항을 반드시 고려해야 함

### **데이터 손실 방지**

알림 전송 시스템의 가장 중요한 요구사항 가운데 하나는 어떤 상황에서도 알림이 소실되면 안 된다는 것

알림이 지연되거나 순서가 틀려도 괜찮지만 사라지면 곤란

알림 시스템은 알림 데이터를 데이터베이스 보관하고 재시도 메커니즘을 구현해야 함



알림 로그 데이터베이스를 유지하는 것도 한 가지 방법

### 알림 중복 전송 방지

같은 알림이 여러 번 반복되는 것을 완전히 막는 것은 불가능

분산 시스템의 특성상 가끔은 같은 알림이 중복되어 전송되기도 함

중복 전송 빈도를 줄이려면 중복을 탐지하는 메커니즘을 도입

오류를 신중하게 처리해야 함

보내야 할 알림이 도착하면 그 이벤트 ID를 검사하여 이전에 본 적이 있는 이벤트인지 살피고 중복된 이벤트라면 버리고 그렇지 않으면 알림을 발송하는 것도 한 가지 방법

### ▼ 추가로 필요한 컴포넌트 및 고려사항

#### 알림 템플릿

파라미터나 스타일, 추적 링크를 조정하기만 하면 사전에 지정한 형식에 맞춰 알람을 만들어 내는 틀

대형 알림 시스템은 하루에도 수백만 건 이상의 알림을 처리

메시지 대부분은 형식이 비슷

유사성을 고려하여 알림 메시지의 모든 부분을 처음부터 다시 만들 필요 없도록 해 줌

### 알림 설정

사용자는 이미 너무 많은 알림을 받고 있어서 쉽게 피곤함을 느낌



많은 웹사이트와 앱에서는 사용자가 알림 설정을 상세히 조정할 수 있도록 하고 있음  
해당 정보는 알림 설정 테이블(user\_id, channel, opt\_in 컬럼으로 구성)에 보관  
설정을 도입한 뒤에는 알림을 보내기 전에 반드시 해당 사용자가 해당 알림을 켜 두었는지 확인 필요

### **전송률 제한**

사용자에게 너무 많은 알림을 보내지 않도록 한 사용자가 받을 수 있는 알림의 빈도를 제한

알림을 너무 많이 보내면 사용자가 알림 기능을 아예 꺼 버릴 수도 있기 때문에 필수로 고려해야 함

### **재시도 방법**

제3자 서비스가 알림 전송에 실패할 경우 해당 알림을 재시도 전용 큐에 넣음  
같은 문제가 계속해서 발생하면 개발자에게 통지

### **푸시 알림과 보안**

iOS와 안드로이드 앱의 경우 알림 전송 API는 appKey와 appSecret을 사용하여 보안을 유지

인증 혹은 승인된 클라이언트만 해당 API를 사용하여 알림을 보낼 수 있음

### **큐 모니터링**

알림 시스템을 모니터링 할 때 큐에 쌓인 알림의 개수가 중요

너무 크면 작업 서버들이 이벤트를 빠르게 처리하고 있지 못하다는 뜻

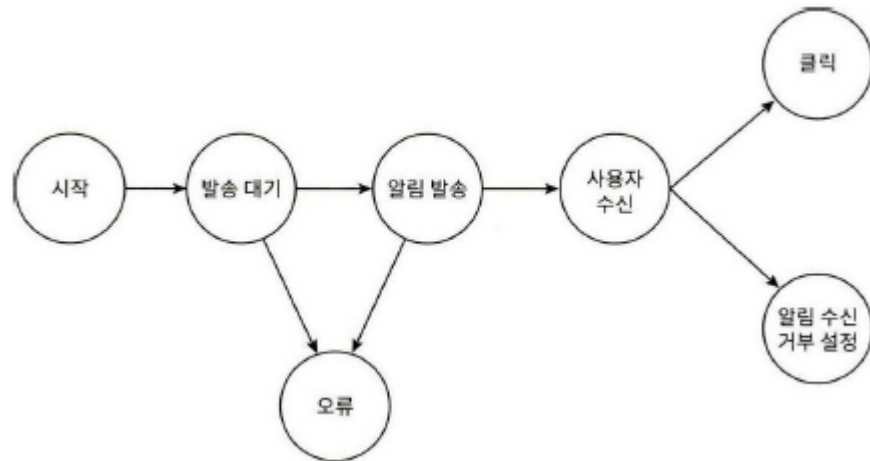
작업 서버를 증설하는 게 바람직

## 이벤트 추적

알림 확인율, 클릭율, 실제 앱 사용으로 이어지는 비율 같은 메트릭은 사용자를 이해 하는데 중요

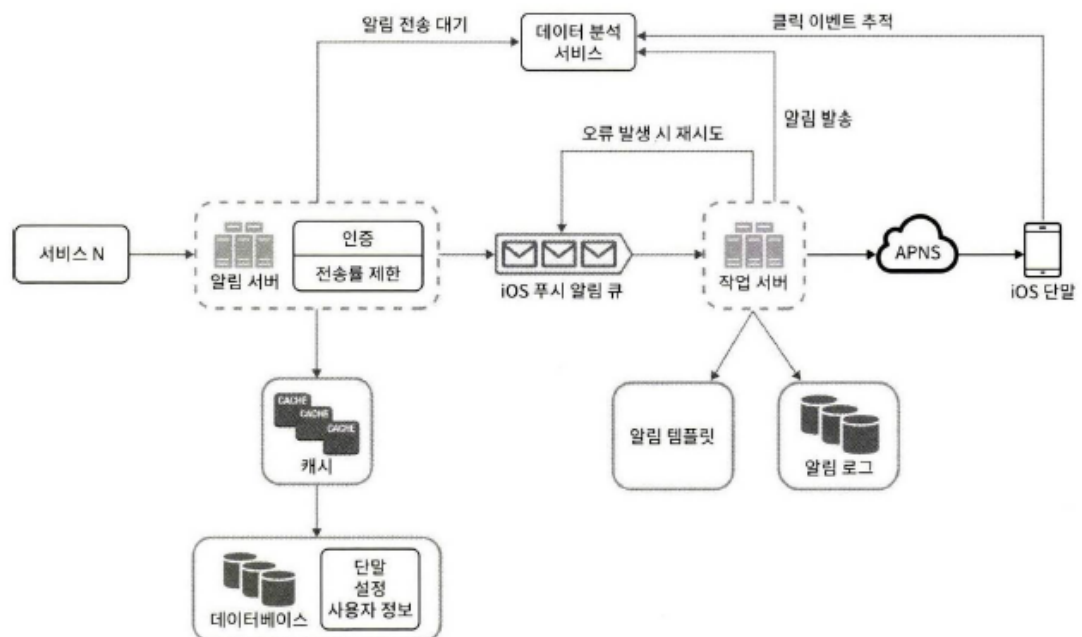
데이터 분석 서비스는 보통 이벤트 추적 기능도 제공

알림 시스템을 만들면 데이터 분석 서비스와도 통합해야만 함



데이터 분석 서비스를 통해 추적하게 될 알람 시스템 이벤트의 사례

## ▼ 수정된 설계안



- 알림 서버에 인증과 전송률 제한 기능이 추가
- 전송 실패에 대응하기 위한 재시도 기능이 추가

- 전송에 실패한 알림은 다시 큐에 넣고 지정된 횟수만큼 재시도
- 전송 템플릿을 사용하여 알림 생성 과정을 단순화하고 알림 내용의 일관성을 유지
- 모니터링과 추적 시스템을 추가하여 시스템 상태를 확인하고 추후 시스템을 개선하기 쉽도록 설계

## ▼ 4. 마무리

알림은 중요한 정보를 계속 알려준다는 점에서 필요불가결한 기능

- 안정성
  - 메시지 전송 실패율을 낮추기 위해 안정적인 재시도 메커니즘 도입
- 보안
  - 인증된 클라이언트만이 알림을 보낼수 있도록 appKey, appSecret 등의 메커니즘을 이용
- 이벤트 추적 및 모니터링
  - 알림이 만들어진 후 성공적으로 전송되기까지의 과정을 추적하고 시스템 상태를 모니터링하기 위해 알림 전송의 각 단계마다 이벤트를 추적하고 모니터링할 수 있는 시스템을 통합
- 사용자 설정
  - 사용자가 알림 수신 설정 조정 가능
  - 알림을 보내기 전 반드시 해당 설정을 확인하도록 설계
- 전송률 제한
  - 사용자에게 알림을 보내는 빈도 제한 가능

## ▼ 토론

알림 우선순위를 설정하고 싶을 때는 어떻게 해야 할까