10장

강의, 교재명	가상 면접 사례로 배우는 대규모 시스템 설계 기초1편
생성 일시	@2024년 2월 20일 오후 8:43

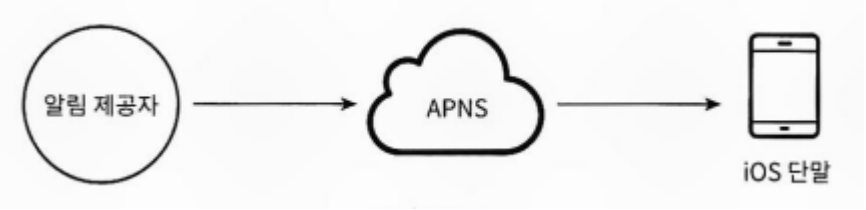
알림 시스템 설계

문제 이해 및 설계 범위 확정

- 푸시 알림, SMS 메시지, 이메일 중 어떤 종류의 알림을 지원하는지
- real-time인지 soft-real-time인지
  - soft-real-time은 약간의 지연을 허용한다.
- IOS, 안드로이드, 랩톱/데스크탑 중 어떤 단말을 지원하는지
- 클라이언트 애플리케이션, 서버 스케쥴링 중 어느 것이 알림을 생성할지
- 알림 받지 않기 같은 설정 지원할지
- 하루에 몇 건의 알림을 보낼지

알림 유형별 지원 방안

IOS 푸시 알림



1. 알림 제공자 : 알림 요청을 만들어 애플 푸시 알림 서비스(APNS)로 보내는 주체
   요청을 만들려면 단말 토큰과 페이로드 데이터가 필요
   a. 단말 토큰 : 알림 요청을 보내는 데 필요한 고유 식별자
   b. 페이로드 : 알림 내용을 담은 JSON 딕셔너리

c. 예시

```
{
  "aps": {
              "alert": {
                 "title": "Game Request",
                 "body": "Bob wants to play chess",
                 "action-loc-key":"PLAY"
              },
              "badge":5
          }
}
```

2. 알림 서비스(APNS) : 애플이 제공하는 원격 서비스이며 IOS 장치로 보내는 역할 담당

안드로이드 푸시 알림

그림 10-3

1. IOS와 비슷하지만 APNS이 아니라 FCM을 사용

SMS 메시지

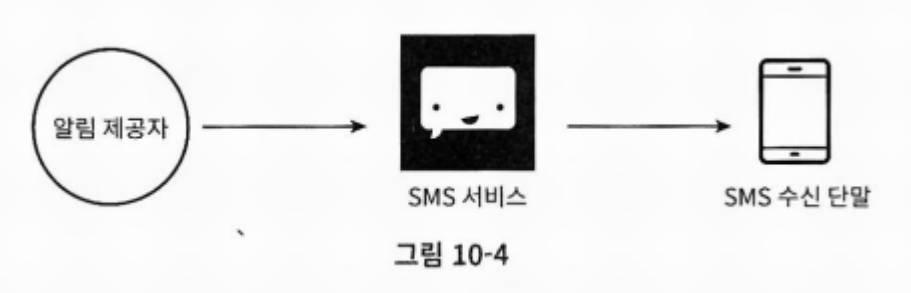


그림 10-4

1. 트윌리오, 넥스모 같은 제 3 사업자의 상용 서비스를 이용함으로 비용이 발생

이메일

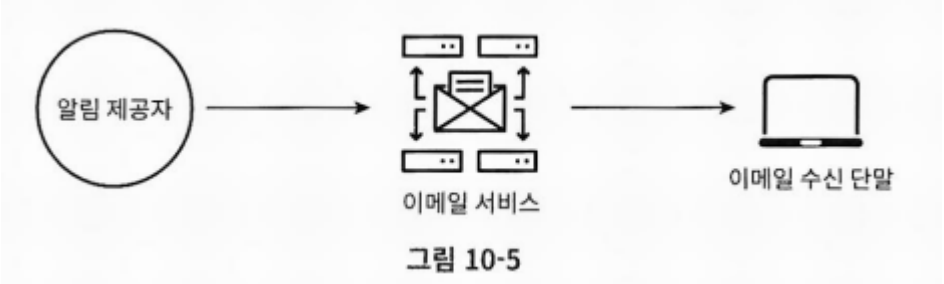
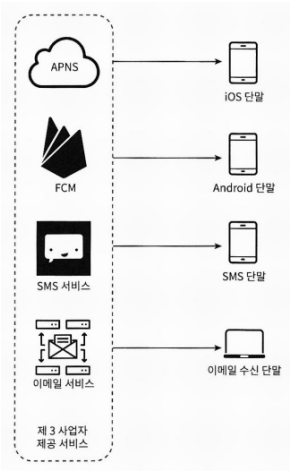


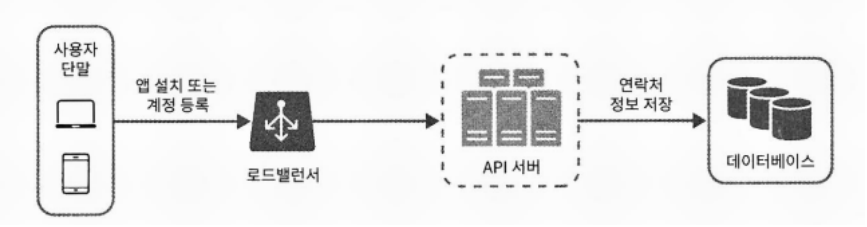
그림 10-5

회사마다 자체 서버 구축하기도 하지만 센드그리드, 메일치프와 같은 상용 이메일 서비스를 주로 이용

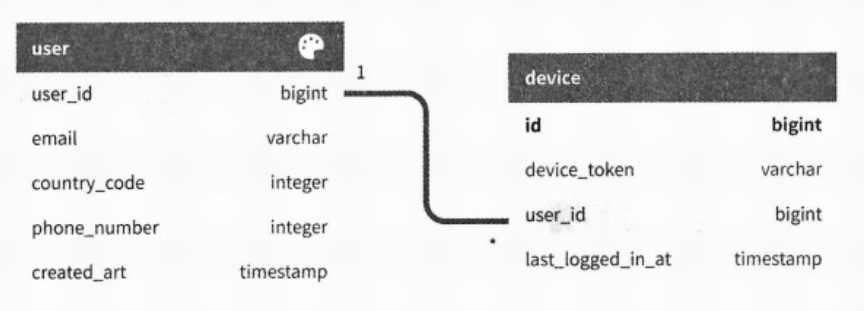


알림 유형별 시스템

연락처 정보 수집 절차

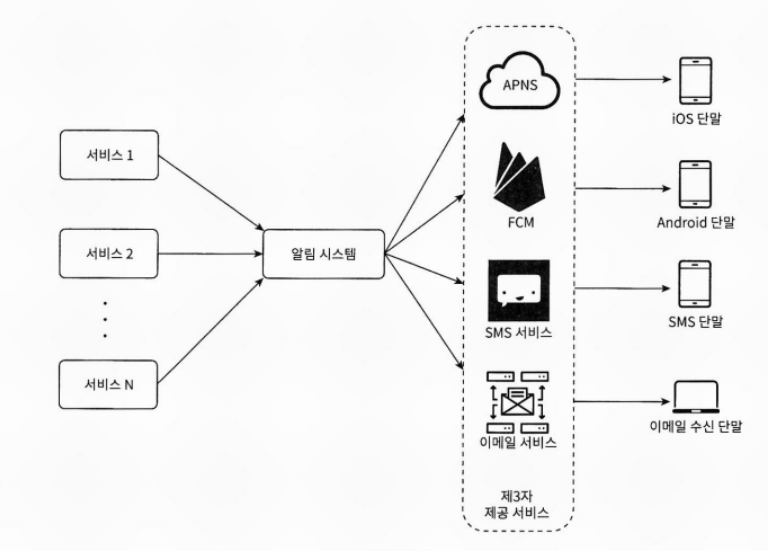


- 알림을 보내려면 단말 토큰, 전화번호, 이메일 주소 등의 정보가 필요
- 우리 앱을 설치하거나 처음으로 계정 등록하면 API 서버는 해당 사용자의 정보를 수집하고 데이터베이스에 저장



- 이메일과 전화번호는 user테이블에
- 한 사용자는 여러 단말을 가질 수 있고 모든 단말에 알림을 전송해야 하기에 단말 토큰은 device 테이블에

개략적 설계안



1부터 N까지의 서비스

- 크론잡
- 마이크로서비스
- 분산 시스템 컴포넌트

알림 시스템

- 서비스 1~N에 알림 전송을 위한 API 제공해야함
- 1개의 서버에서 구축
- 제3자 서비스에 전달할 알림 페이로드를 만들 수 있어야 함

제 3자 서비스

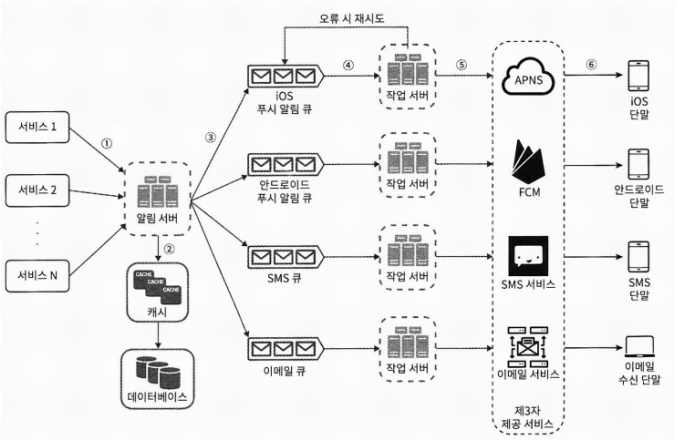
- 알림을 실제로 전달하는 역할을 함
- 확장성을 고려해 서비스를 통합하거나 제거할 수 있어야 함
- 중국에서는 FCM을 사용할 수 없고 Jpush나 PushY를 사용

문제점

1. SPOF : 1개밖에 없는 서버에 장애가 발생하면 서비스 전체의 장애로 이어짐
2. 규모 확장성 : 데이터베이스나 캐시 등 중요 컴포넌트의 규모 확장 불가능
3. 성능 병목 : 트래픽 과부하

개선

1. 데이터베이스와 캐시를 알림 시스템의 주 서버에서 분리
2. 알림 서버 증설
3. 메시지 큐 이용하여 시스템 컴포넌트 사이의 결합을 끊음



- 1~N개의 서비스 : 알림 시스템 서버의 API를 통해 알림을 보낼 서비스
- 알림 서버
  - 알림 전송 API : 스팸 방지를 위해 사내 서비스나 인증된 클라이언트만 이용 가능
  - 알림 검증 : 이메일 주소, 전화번호 등에 대한 기본적 검증 수행
  - 데이터베이스 또는 캐시 질의 : 알림에 포함시킬 데이터(알림 템플릿 등등)를 가져오는 기능

    - 알림 전송 : 알림 데이터를 메시지 큐에 넣어서 병렬 처리 가능
    - 알림의 종류별로 별도의 메세지 큐를 구성하여 제3자 서비스 장애 대응
- API 호출 시 전송할 데이터의 예시

```
{
  "to": [
    {
      "user_id": 123456
    }
  ],
  "from": {
    "email": "from_address@example.com"
  },
  "subject": "Hello, World!",
  "content": [
    {
      "type": "text/plain",
      "value": "Hello, World!"
    }
  ]
}
```

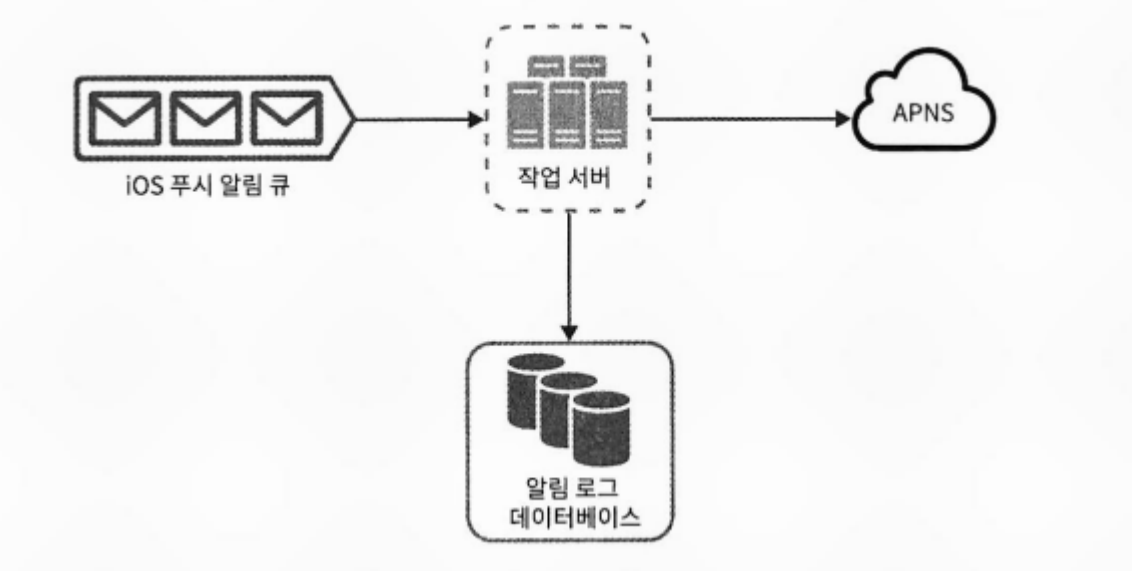
과정

1. API 호출하여 알림 서버로 알림을 보냄
2. 알림 서버는 사용자 정보, 단말 토큰, 알림 설정 같은 메타데이터를 캐시나 데이터베이스에서 가져옴
3. 알림을 알림 유형에 따른 메세지 큐에 넣는다.
4. 작업 서버는 메세지 큐에서 알림 이벤트를 꺼내고 제3자 서비스로 보냄
5. 제3자 서비스는 사용자 단말로 알림 전송

상세 설계

안정성

- 데이터 손실 방지
    - 알림이 지연되거나 순서가 틀리는 것 정도는 허용해도 데이터 자체가 사라지는건 안됨
    - 알림 데이터를 데이터베이스에 보관하고 재시도 매커니즘 구현



- 알림 중복 전송 방지
  - 알림이 도착하면 이벤트 ID를 검사하여 이전에 본 적이 있는 이벤트인지 살핀다.
  - 중복된 이벤트라면 버림
  - 중복 전송을 100% 방지하는건 불가능(Why?)

추가로 필요한 컴포넌트 및 고려사항

알림 템플릿

여러분이 꿈꿔온 그 상품을 우리가 준비했습니다. [item_name]이 다시 입고 되었습니다! [date]까지만 주문 가능합니다!

- 인자나 스타일, 추적 링크만 조정하여 다시 만들지 않아도 됨
- 일관성 유지

알림 설정

user_id	bigInt	
channel	varchar	# 알림이 전송될 채널. 푸시 알림, 이메일, SMS 등
opt_in	boolean	# 해당 채널로 알림을 받을 것인지의 여부

- 사용자에게 설정을 제공하기 위해 설정 테이블을 따로 만듬

- 알림을 보내기 전에 설정 테이블을 거치도록 함

전송률 제한

- 사용자에게 너무 많은 알림을 보내지 않도록 함

재시도 방법

- 제3자 서비스가 알림 전송에 실패하면, 해당 알림을 재시도 전용 큐에 넣는다.
- 같은 문제가 계속해서 발생하면 개발자에게 통지

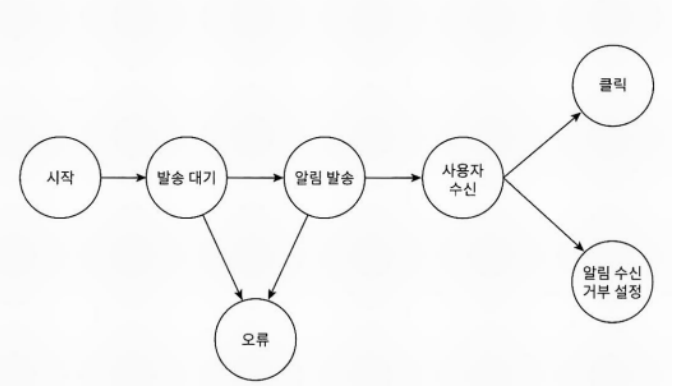
푸시 알림과 보안

- appKey와 appSecret을 사용해 인증되거나 승인된 클라이언트만 알림을 보낼 수 있게 함

큐 모니터링

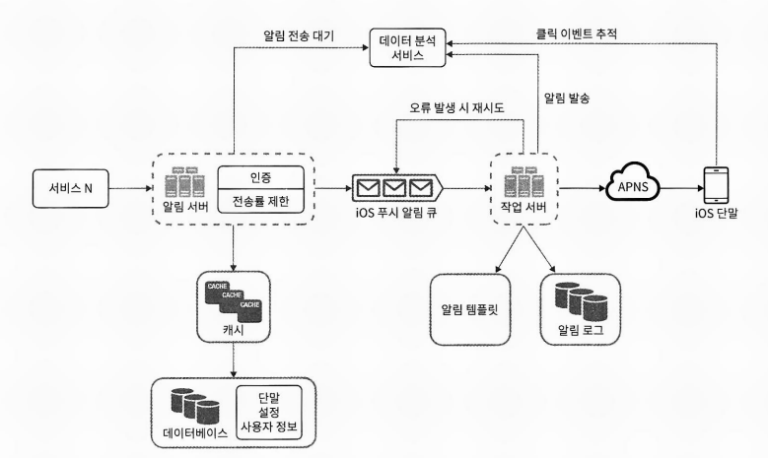
- 알림의 개수를 모니터링하여 작업 서버를 증설할지 말지 결정

이벤트 추적



- 알림 확인율, 클릭율, 실제 앱 사용으로 이어지는 비율 등등의 메트릭은 사용자를 이해하는데 중요한 요소임
- 데이터 분석 서비스와 통합할 필요가 있다.

수정된 설계안


알림 전송 대기
데이터 분석 서비스
클릭 이벤트 추적
알림 발송
오류 발생 시 재시도
서비스 N
알림 서버
인증
전송률 제한
iOS 푸시 알림 큐
작업 서버
APNS
iOS 단말
캐시
알림 템플릿
알림 로그
데이터베이스
단말
설정
사용자 정보