

## 11. 뉴스 피드 시스템 설계

# 1단계. 문제 이해 및 설계 범위 확정

- 모바일 앱, 웹 둘다 지원해야
- 주요 기능
  - 스토리를 올리고, 친구들이 올리는 스토리를 볼 수 있어야
- 스토리 표시는 시간흐름 역순
- 사용자별 최대로 가질 수 있는 친구수는 5000명
- 1000만 DAU
- 스토리에는 이미지, 비디오 등의 미디어 파일 포함 가능

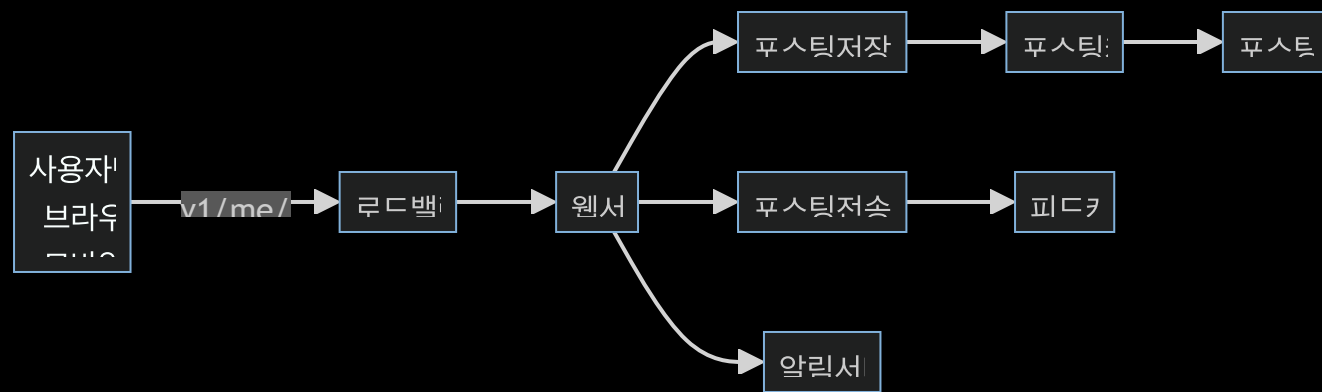
## 2단계: 개략적인 설계안 제시 및 동의 구하기

- 피드 발행
  - 스토리 포스팅 시, 캐시와 DB에 기록. 새 포스팅은 친구의 뉴스 피드에도 전송
- 뉴스 피드 생성
  - 모든 친구의 포스팅을 시간 역순으로 모음

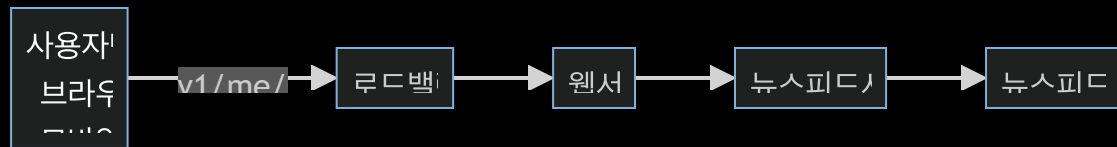
# 뉴스 피드 API

- 피드 발행 API
  - 스토리 포스팅을 위한 API. POST /v1/me/feed
  - 인자
    - body: 포스팅 내용
    - Authorization 헤더: API 호출을 인증하기 위함
- 피드 읽기 API
  - 뉴스 피드 조회. GET /v1/me/feed
  - 인자
    - Authorization 헤더: API 호출을 인증하기 위함

# 피드 발행



## 뉴스 피드 생성



## 3단계: 상세 설계

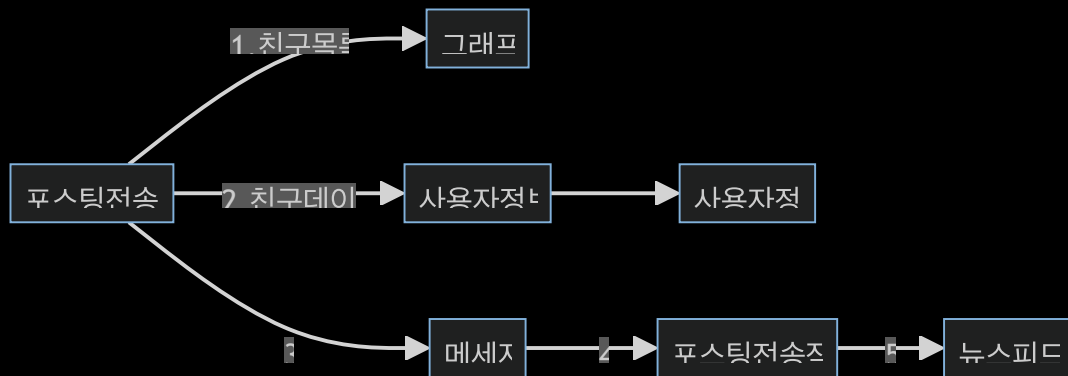
## 웹 서버

- 클라이언트와 통신
- 인증, 처리율 제한 등의 기능도 수행
- 스팸 차단. 유해한 콘텐츠 업로드 방지



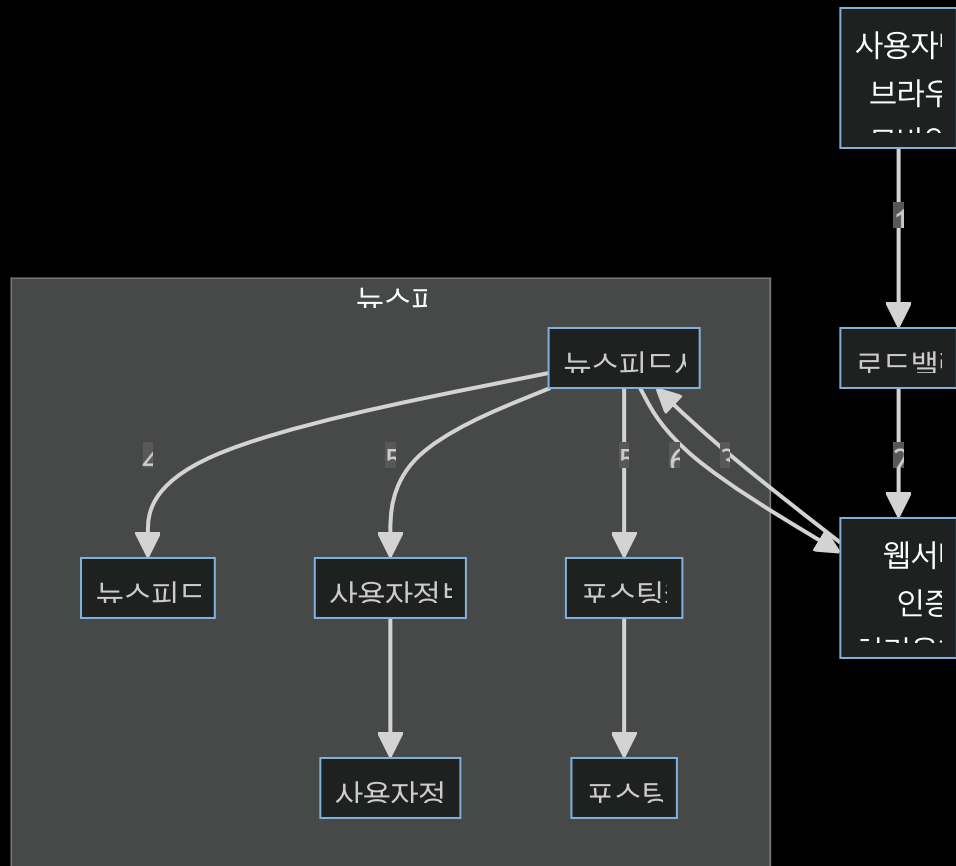
# 포스팅 전송(팬아웃) 서비스

- 팬아웃
  - 어떤 사용자의 새 포스팅을, 그 사용자와 친구 관계에 있는 모든 사용자에게 전달
- 2가지 모델이 있음
  - 쓰기 시점에 팬아웃(push)
    - 장점: 실시간 갱신. 뉴스 피드를 읽는데 드는 시간이 짧아짐
    - 단점: 핫키 문제. 친구가 많은 사용자라면, 모두의 뉴스 피드를 갱신하는데 오래 걸림
  - 읽기 시점에 팬아웃(pull)
    - 장점: 비활성화 사용자는 이 모델이 유리. 핫키 문제 없음.
    - 단점: 뉴스 피드 읽는 데 많은 시간 소요.
- 채택할 방법
  - 대부분의 사용자에게 대해서 푸시모델 사용
  - 친구/팔로워가 아주 많은 사용자의 팔로워에게는 풀모델 사용
  - 안정해시를 이용해 요청/데이터를 고르게 분산



1. 그래피 DB는 친구관계/추천을 관리하기 적합
2. 사용자 정보 캐시에서 친구정보 조회 후 일부를 필터링. 예: 누군가의 피드 업데이트 무시 설정
4. 뉴스 피드 캐시에 <포스팅ID, 사용자ID> 순서쌍을 추가. 캐시 크기에 제한.

# 피드 읽기 흐름 상세 설계



## 캐시 구조

- 5가지 계층으로 구분
  - 뉴스피드
  - 콘텐츠: 인기콘텐츠, 일반콘텐츠
  - 소셜그래프: 팔로워, 팔로잉
  - 행동: 좋아요, 답글, 기타
  - 횃수: 좋아요횃수, 답글횃수, 기타

# 4단계: 마무리

- 추가 논의
  - 데이터베이스 규모확장
    - 수직적 vs 수평적 규모 확장
    - 주-부 다중화
    - 복제본에 대한 읽기 연산
    - 일관성 모델
    - 데이터베이스 샤딩
  - 웹 서버 무상태 운영
  - 데이터를 많이 캐시할 방법
  - 여러 데이터센터 지원 방법
  - 메시지 큐를 사용해 컴포넌트 간 결합도 낮추기
  - 핵심 메트릭에 대한 모니터링