실시간 다국어 채팅 애플리케이션 아키텍처

1. 기술 스택

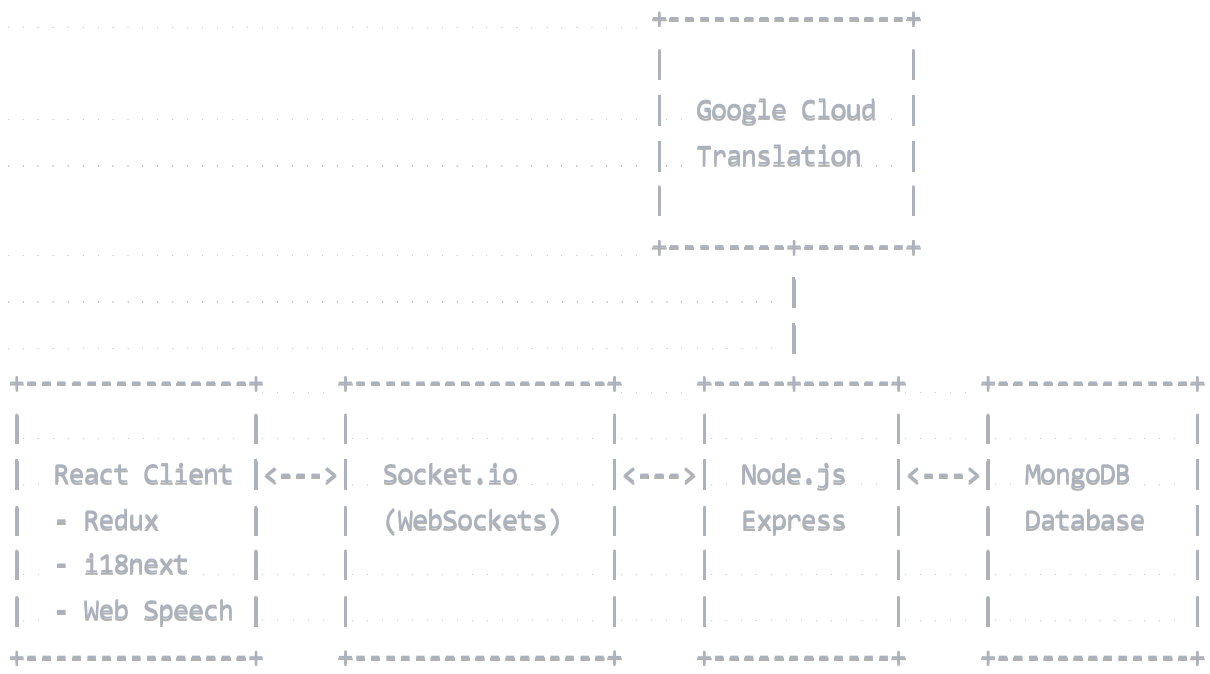
프론트엔드

- **React**: 사용자 인터페이스 구축
- **Redux**: 상태 관리
- **Socket.io-client**: 실시간 양방향 통신
- **i18next**: 다국어 지원 (한국어, 영어, 말레이어)
- **Web Speech API**: STT(Speech-to-Text) 및 TTS(Text-to-Speech) 기능

백엔드

- **Node.js & Express**: 서버 구축
- **Socket.io**: 실시간 양방향 통신
- **MongoDB**: 데이터베이스 (메시지, 사용자, 채팅방 정보 저장)
- **JWT**: 인증 관리
- **Google Cloud Translation API**: 다국어 번역 (무료 할당량 내에서 사용)
- **bcrypt**: 비밀번호 암호화

2. 시스템 구성도



3. 핵심 기능

사용자 관리

- 회원가입 및 로그인
- JWT 기반 인증
- 사용자 프로필 관리

채팅 기능

- 개인 채팅방 생성
- 그룹 채팅방 생성 및 관리
- 메시지 전송 및 수신
- 메시지 읽음 상태 표시 및 관리

관리자 기능

- 채팅방 관리 (생성, 삭제, 사용자 초대/추방)
- 사용자 관리
- 로그 및 통계 확인

다국어 지원

- 실시간 메시지 번역 (한국어, 영어, 말레이어)
- 인터페이스 언어 전환

음성 기능

- STT(Speech-to-Text): 음성 입력을 텍스트로 변환
- TTS(Text-to-Speech): 수신된 메시지를 음성으로 변환

읽음 상태 관리

- 메시지 읽음 상태 추적
- 읽음 표시 관리 기능

4. 데이터베이스 스키마

사용자 (Users)

- _id: ObjectId
- username: String

- email: String
- password: String (해시됨)
- profilePicture: String (URL)
- language: String (기본 언어 설정)
- createdAt: Date
- updatedAt: Date

채팅방 (Rooms)

- _id: ObjectId
- name: String
- type: String (private/group)
- creator: ObjectId (ref: Users)
- participants: [ObjectId] (ref: Users)
- admins: [ObjectId] (ref: Users)
- createdAt: Date
- updatedAt: Date

메시지 (Messages)

- _id: ObjectId
- roomId: ObjectId (ref: Rooms)
- sender: ObjectId (ref: Users)
- content: String
- originalLanguage: String
- translations: Object (언어코드: 번역된 텍스트)
- readBy: [ObjectId] (ref: Users)
- createdAt: Date

5. API 엔드포인트

인증 API

- POST /api/auth/register
- POST /api/auth/login
- GET /api/auth/user

사용자 API

- GET /api/users
- GET /api/users/
- PUT /api/users/
- DELETE /api/users/

채팅방 API

- GET /api/rooms
- POST /api/rooms
- GET /api/rooms/
- PUT /api/rooms/
- DELETE /api/rooms/
- POST /api/rooms/
  /participants
- DELETE /api/rooms/
  /participants/

메시지 API

- GET /api/rooms/
  /messages
- POST /api/rooms/
  /messages
- PUT /api/messages/
  /read

6. 웹소켓 이벤트

클라이언트 → 서버

- 'join_room': 채팅방 입장
- 'leave_room': 채팅방 퇴장
- 'send_message': 메시지 전송
- 'mark_read': 메시지 읽음 표시

서버 → 클라이언트

- 'receive_message': 새 메시지 수신
- 'user_joined': 사용자가 채팅방에 입장
- 'user_left': 사용자가 채팅방에서 퇴장
- 'message_read': 메시지 읽음 상태 업데이트

7. 확장성 고려사항

- **수평적 확장**: Node.js 서버 복제 및 로드 밸런싱
- **캐싱**: Redis를 사용한 메시지 및 사용자 정보 캐싱
- **메시지 큐**: 번역 요청 처리를 위한 메시지 큐 도입
- **WebRTC**: 향후 화상 채팅 기능 추가 가능성