

Collab Note

시작발표

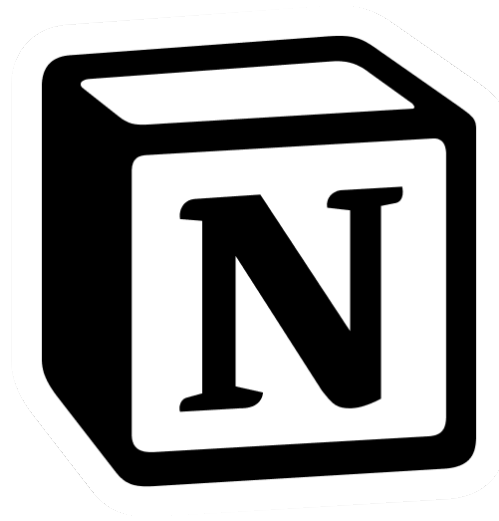
2023년 1학기 WAP WEB TEAM 1

201911430 김민수



프로젝트 소개

실시간 화상 회의 및 동시 편집 협업 툴입니다.



주제 선정 동기

Notion이나 Figma 등 협업 툴을
사용하면서 동시 편집 기술에 대해
흥미를 느끼고 직접 구현해 보고
싶었음.

Zoom을 이용한 비대면 화상 회의
수업 중 Zoom에서 WebRTC를
통한 서버 없이 브라우저 간 직접
통신하는 것을 알게 되었고,
적용해보고 싶었음.

프로젝트 설명

❖ WebRTC: 실시간 화상 회의

- ❖ WebRTC(Web Real-Time Communication)란 웹 애플리케이션과 사이트가 중간자 없이 브라우저 간에 오디오나 영상 미디어를 포착하고 마음대로 스트림을 할 뿐 아니라, 임의의 데이터도 교환할 수 있도록 하는 기술.

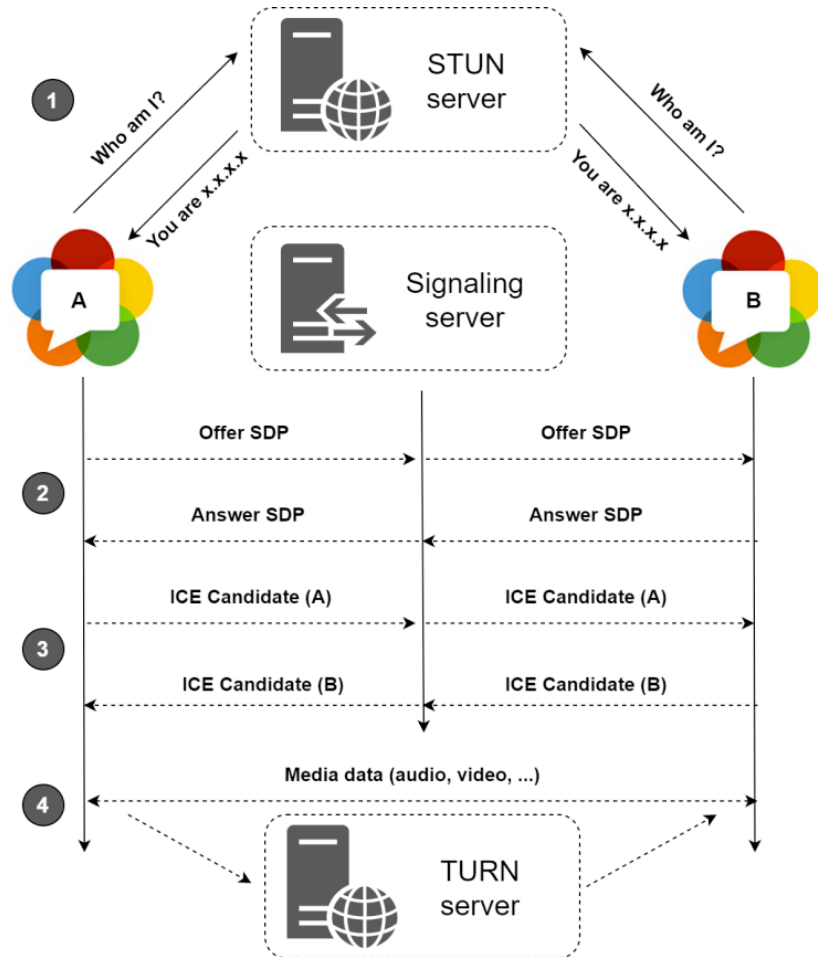
ex) Zoom, Google Meet, Facebook

❖ CRDT: 동시 편집 기술

- ❖ CRDT(Conflict-Free-Replicated Data Types)란 동시 수정 작업이 발생하는 분산 시스템에서 데이터 일관성을 보장하는 알고리즘.

ex) Figma, Google Docs, Visual Studio Code Share

프로젝트 설명-WebRTC



WebRTC는 "Web Real-Time Communication"의 약자로 음성 통화, 영상 통화, 파일 공유 등의 실시간 통신 기능을 제공하는 오픈 소스 프로젝트입니다.

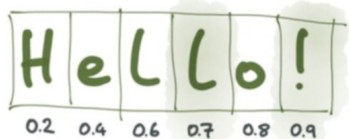
WebRTC는 다양한 장점이 있습니다. 웹 브라우저에서 바로 사용할 수 있기 때문에 별도의 소프트웨어나 플러그인을 설치할 필요가 없습니다. 또한 WebRTC는 P2P(Peer-to-Peer) 기반의 네트워크를 사용하기 때문에 서버에서 중개하는 시스템을 사용하지 않아도 되어 서버 부하를 줄일 수 있습니다.

참고: <https://webrtc.org/?hl=ko>

프로젝트 설명-CRDT



↓ edit


$$\{(0.2, A, "H"), (0.4, A, "e"),$$
$$(0.6, A, "L"), (0.8, A, "o")\}$$

∪

$$\{(0.7, A, "L"), (0.9, B, "!")\}$$

=

$$\{(0.2, A, "H"), (0.4, A, "e"),$$
$$(0.6, A, "L"), (0.7, A, "L"),$$
$$(0.8, A, "o"), (0.9, B, "!")\}$$

CRDT는 "Conflict-Free-Replicated Data Types"의 약자로 정의는 "어떤 변경사항을 받으면, 순서와 상관없이 변경사항만 같은면 같은 상태"입니다. CRDT에서 각 개체를 유니크한 값으로 봅니다. 따라서 텍스트가 추가될 때 새로운 유니크한 값을 생성해서 넣어주게 된다. 예를 들어 인덱스 0.6과 0.8사이에 "l"을 넣을 경우 0.7이 된다. 하지만 두 글자를 동시에 merge하면 섞이게 된다.

참고: https://channel.io/ko/blog/crdt_vs_ot

프로젝트 설명-CRDT

CRDT의 문제를 보완하는 알고리즘 기법들

Text editing CRDTs		
Logoot	(Weiss, Urso & Holli 2009/2010)	interleaving 😞
LSEQ	(Nédelec, Holli, Mostefaoui & Desmontils, 2013)	interleaving 😞
RGA	(Roh, Jeon, Kim & Lee, 2011)	lesser anomaly 😞
Treedoc	(Proença, Marquès, Shapiro & Leticia, 2009)	no interleaving (conjectured) 😊
WOOT	(Oster, Urso, Holli & Imine, 2006)	no interleaving (conjectured) 😊
Astrong	(Attiya, Burckhardt, Gotsman, Morrison, Yang & Zawirski, 2016)	interleaving 😞

두 글자가 섞이는 문제를 해결하기 위한 RGA, Treedoc, WOOT 등의 알고리즘이 있다. 알고리즘들 중 우선 구현 가능성이 있는 RGA(Linked List)를 선택했고, 3개의 정수형 튜플(우선순위, Upstream 피어 ID, Upstream 논리 시계)로 이루어진 리스트의 Identifier를 추가한 논문을 발견하고 적용해 볼 예정입니다.

논문 참고: <https://pages.lip6.fr/Marc.Shapiro/papers/rgasplit-group2016-11.pdf>

프로젝트 설명-스택



- ❖ Server: **Nest**, Postgresql, Prisma, Swagger
- ❖ Client: **React**(+Vite)
 - ❖ Data Fetching: Axios, Tanstack Query
 - ❖ Global State: Zustand
 - ❖ UI: Emotion, Framer-motion, Storybook
- ❖ ETC:
 - ❖ **Typescript**
 - ❖ Pnpm의 workspace를 이용한 monorepo 방식
 - ❖ Socket.io, Socket.io-client 를 통한 Socket 통신

프로젝트 설명-개발

WebRTC와 CRDT의 class, hook, manager 구현에
집중할 예정이므로 구조도, 순서도는 존재하지 않습니다. 허허...



개발 일정

	1주차 03/20~03/26	2주차 03/27~04/02	3주차 04/03~04/09	4주차 04/10~04/16	5주차 04/17~04/23	6주차 04/24~04/30
WebRTC	WebRTC 공부		Class 구현		중간 고사	
CRDT	CRDT 공부					
UI	Common Component 제작					

	7주차 05/01~05/07	8주차 05/08~05/14	9주차 05/15~05/21	10주차 05/22~05/28	11주차 05/29~06/04	12주차 06/05~06/11
WebRTC	Hook 구현	Manager 구현	Refactoring			정리
CRDT				Refactoring		
UI	Page. Component Detail					

QnA

