

온라인 토론 플랫폼



AGORA



01 기획 배경 및 프로젝트 소개

02 프로젝트 기능 소개

03 프로젝트 시연

04 기술 스택 소개

05 추후 계획

기획 배경



올바른 토론 문화를 형성해 의사소통 능력을 함양할 수 있는 서비스를 만들자!

기획 배경

학생들의 의사소통 능력 향상을 위해 토론 교육이 필요하다고 생각하시나요?



전남 OO고등학교 교사

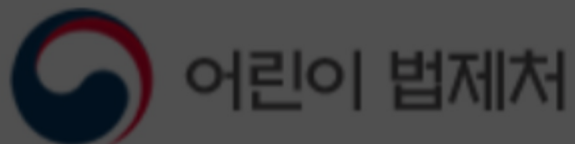
필수적이라고 생각합니다! 자신의 의견을 합리적 근거를 들어
표현할 수 있는 능력은 향후 입시 면접에 큰 도움이 됩니다.

만약 온라인 화상 토론 플랫폼이 있다면, 해당 플랫폼 활동이 입시 준비에 도움이
될까요?



전남 OO고등학교 교사

네, 도움될 것 같습니다. 생기부에 특색있는 활동으로 기재할 수 있을 것 같습니다.



- 텍스트 바탕
- 법 관련한 주제 한정
- 초등학교 고학년 대상
- 일정 회원만 참여 가능

실시간 소통 채널 부재
한정된 주제
제한된 타겟층



- 텍스트 바탕
- 근거 자료로 인터넷 기사만 가능

실시간 소통 채널 부재
현장감 x



- 온라인 바탕 토론 교육
- 영어 회화를 위한 온라인 토론 교육
- 대회를 위한 교육에 초점

자율성 x
제한된 콘텐츠

텍스트 기반 → 실시간성X
토론 '대회' 중심
주제 제한적



자유로운 토론 학습을 위한 **온라인 화상 토론** 플랫폼

- ✓ 비대면 실시간 소통 채널 제공하는 서비스
- ✓ 시공간적 제약을 받지 않는 플랫폼 서비스
- ✓ 토론을 매개로 의사소통 능력을 함양할 수 있는 온라인 서비스

시연 영상

시연

기술 스택



Vuejs



Webpack



NGINX



Kurento



WebSocket



Kafka



**Spring
Boot**



**Spring
Security**



**Spring
Data JPA**



**Query
DSL**



MySQL



Redis



**AWS
EC2**



GitLab



Jenkins



**Docker
Compose**

기술 스택



Vuejs



Webpack



NGINX



Kurento



WebSocket



Kafka



Spring
Boot



Spring
Security



Spring
Data JPA



Query
DSL



MySQL



Redis



AWS
EC2



GitLab



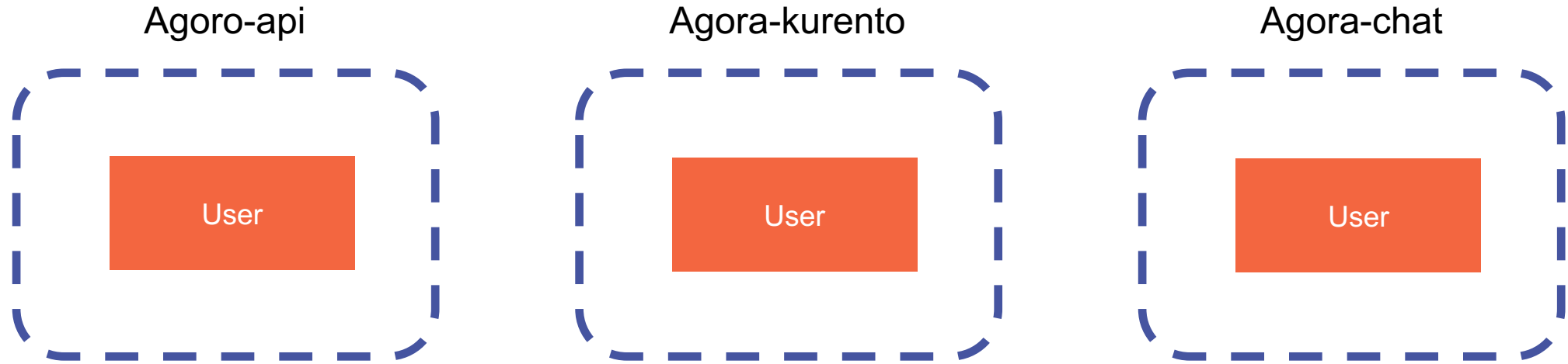
Jenkins



Docker
Compose

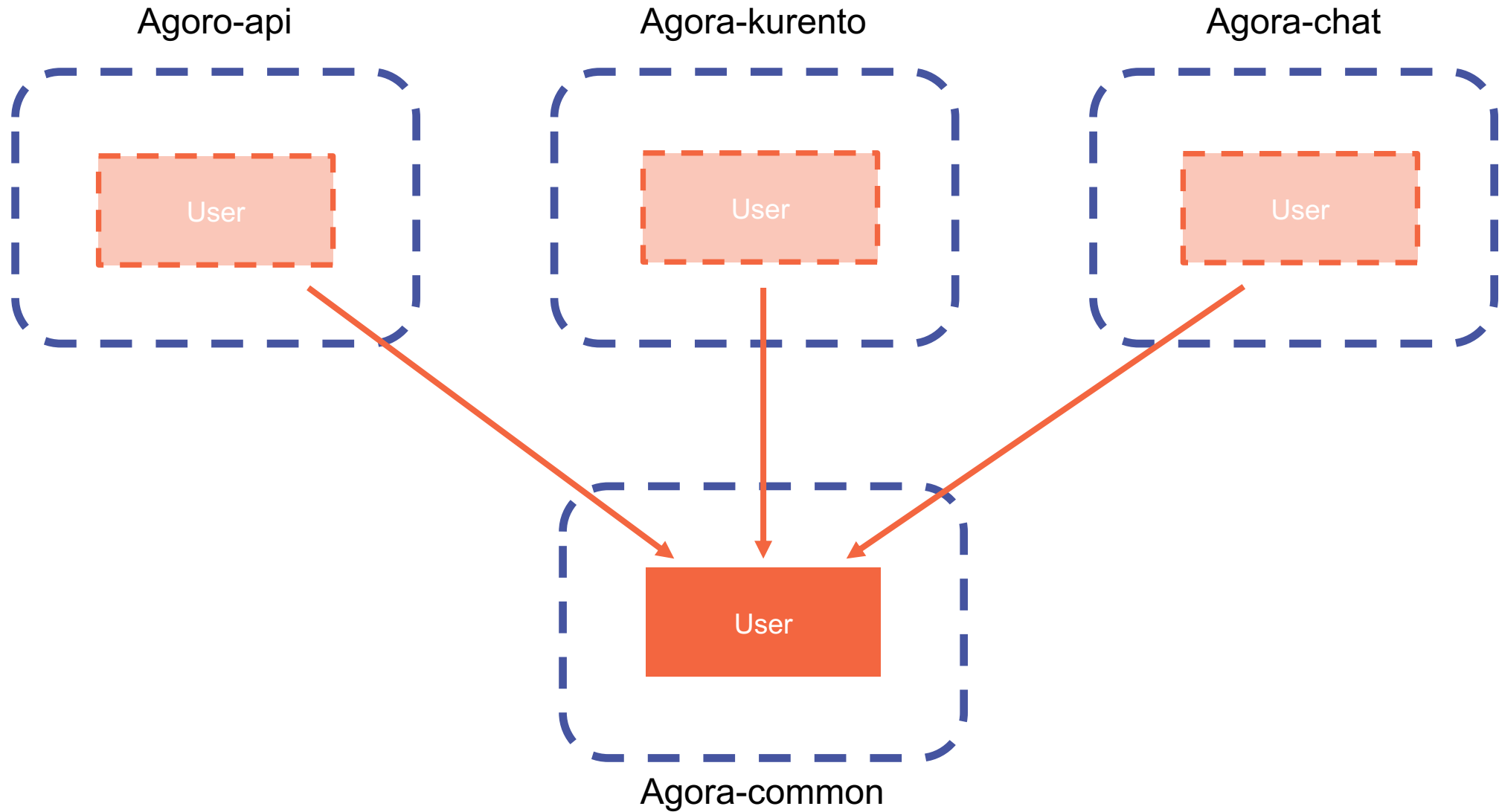
+멀티 모듈..

멀티 모듈 프로젝트

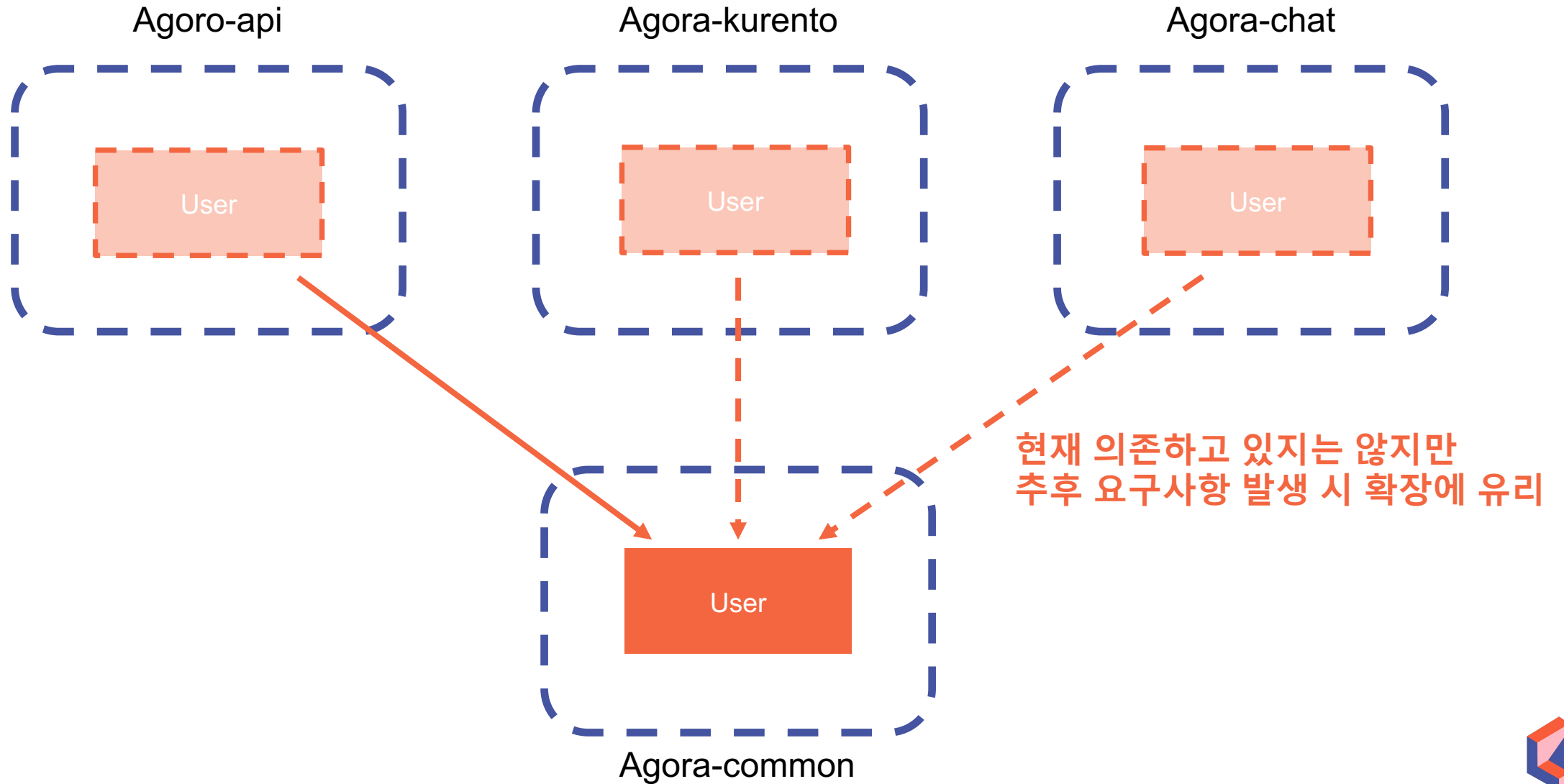


각 단일 프로젝트에서 공통적으로 사용되는 User 도메인을 복붙해야하는 문제 발생!

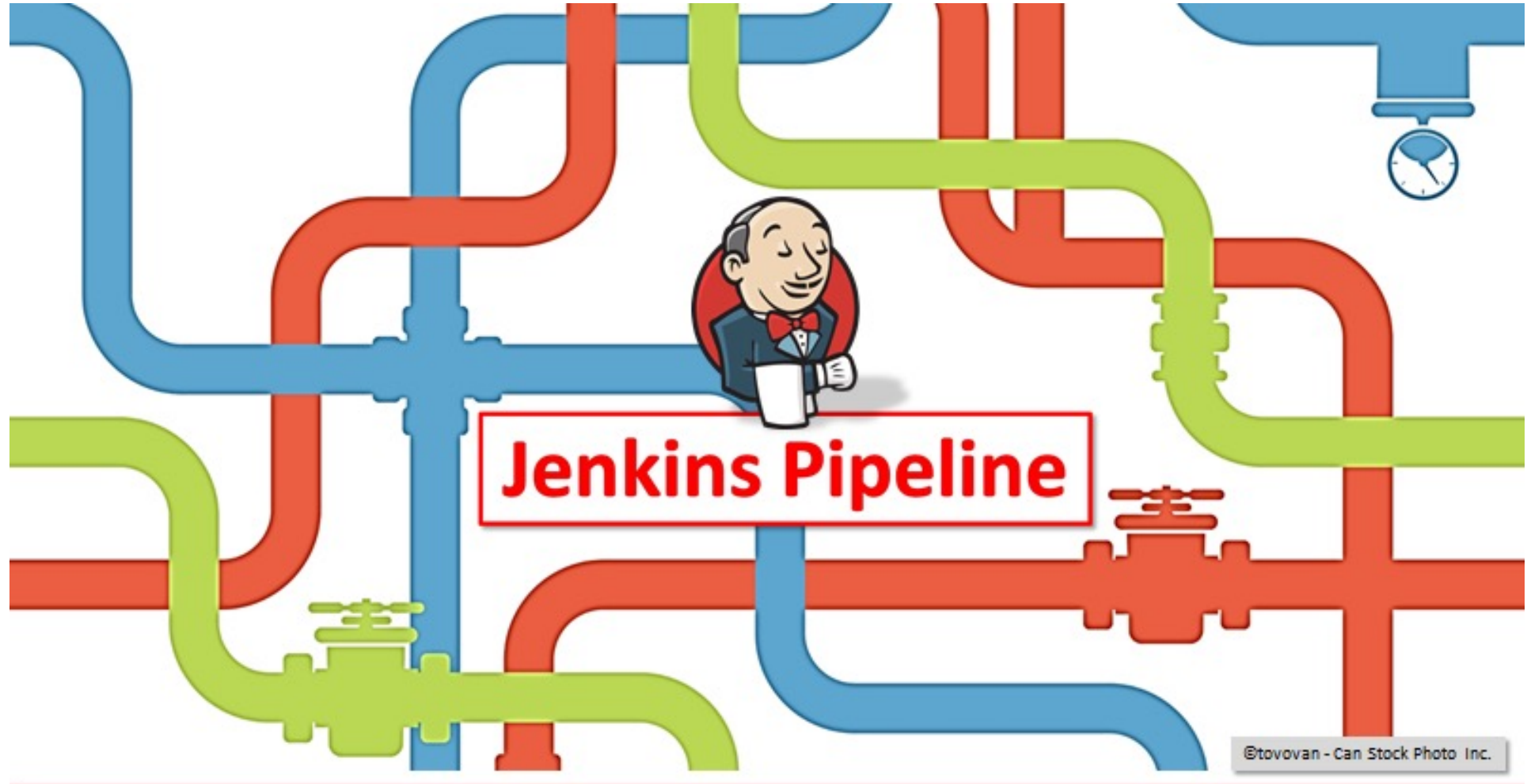
멀티 모듈 프로젝트



멀티 모듈 프로젝트



Jenkins Pipeline을 통한 배포 자동화



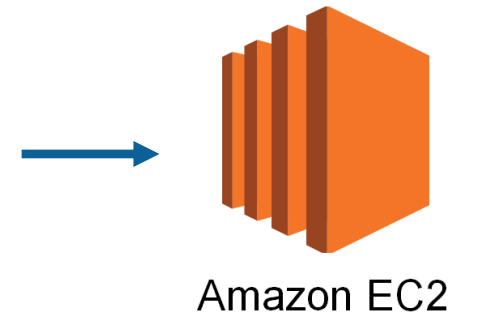
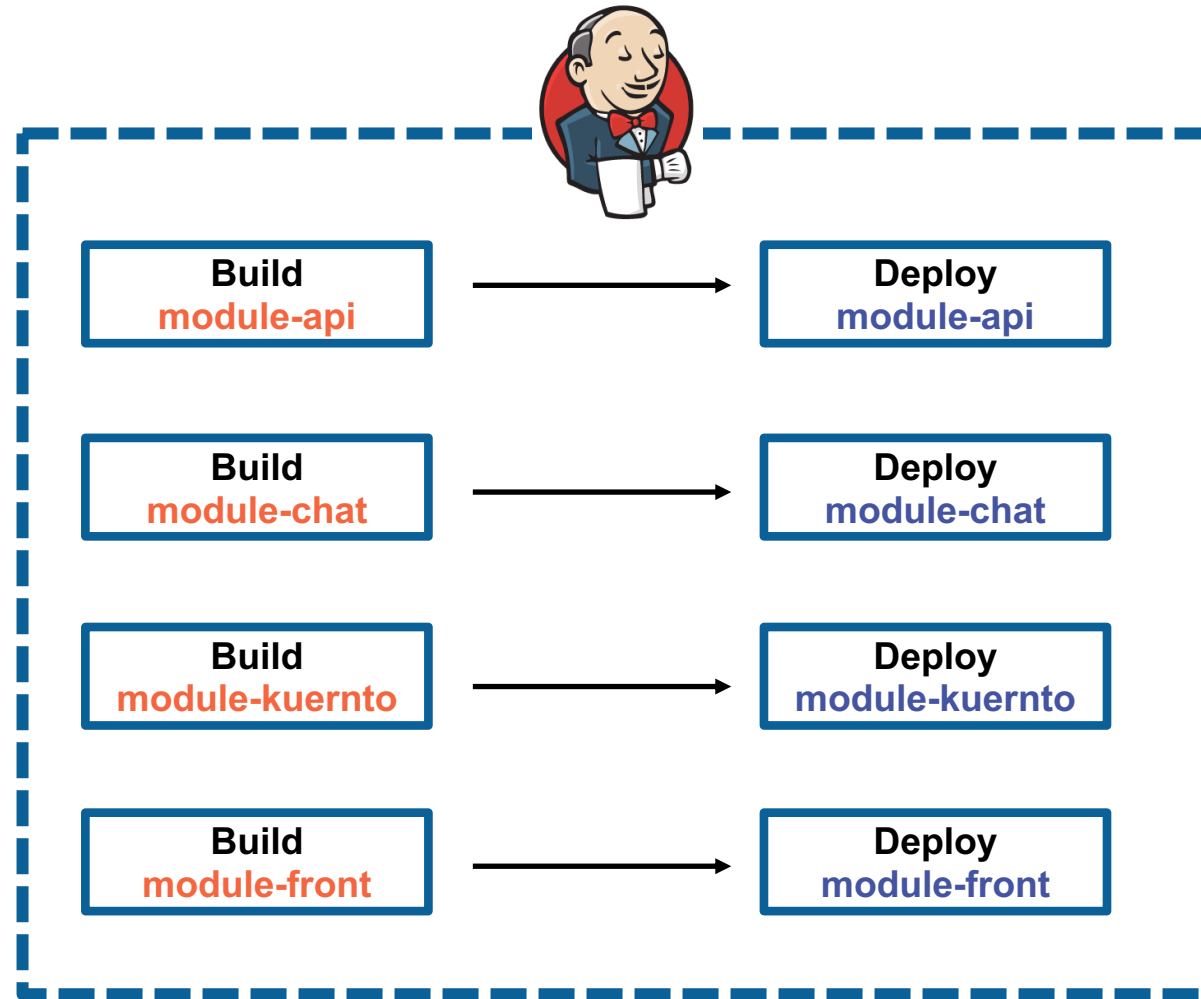
Jenkins Pipeline을 통한 배포 자동화

```
pipeline
{
  ...
  stages {
    ...
    stage('build-module-api') {
      when {
        anyOf {
          changeset "backend/module-core/**"
          changeset "backend/module-api/**"
        }
      }
      steps {
        echo 'Build Start "${APP_API}"'
        sh 'chmod +x backend/gradlew'
        sh 'backend/gradlew -p backend/${APP_API} build -x test'
        echo 'Build End "${APP_API}"'
      }
    }
    stage('build-module-chat') {
      ...
    }
    stage('build-module-kurento') {
      ...
    }
    stage('build-front') {
      ...
    }
  }
}
```

when 구문에 지정한 디렉토리에 변경사항이 생기면 해당 stage가 실행되도록 지정

== 변경된 특정 모듈만 배포 자동화

Jenkins Pipeline을 통한 배포 자동화

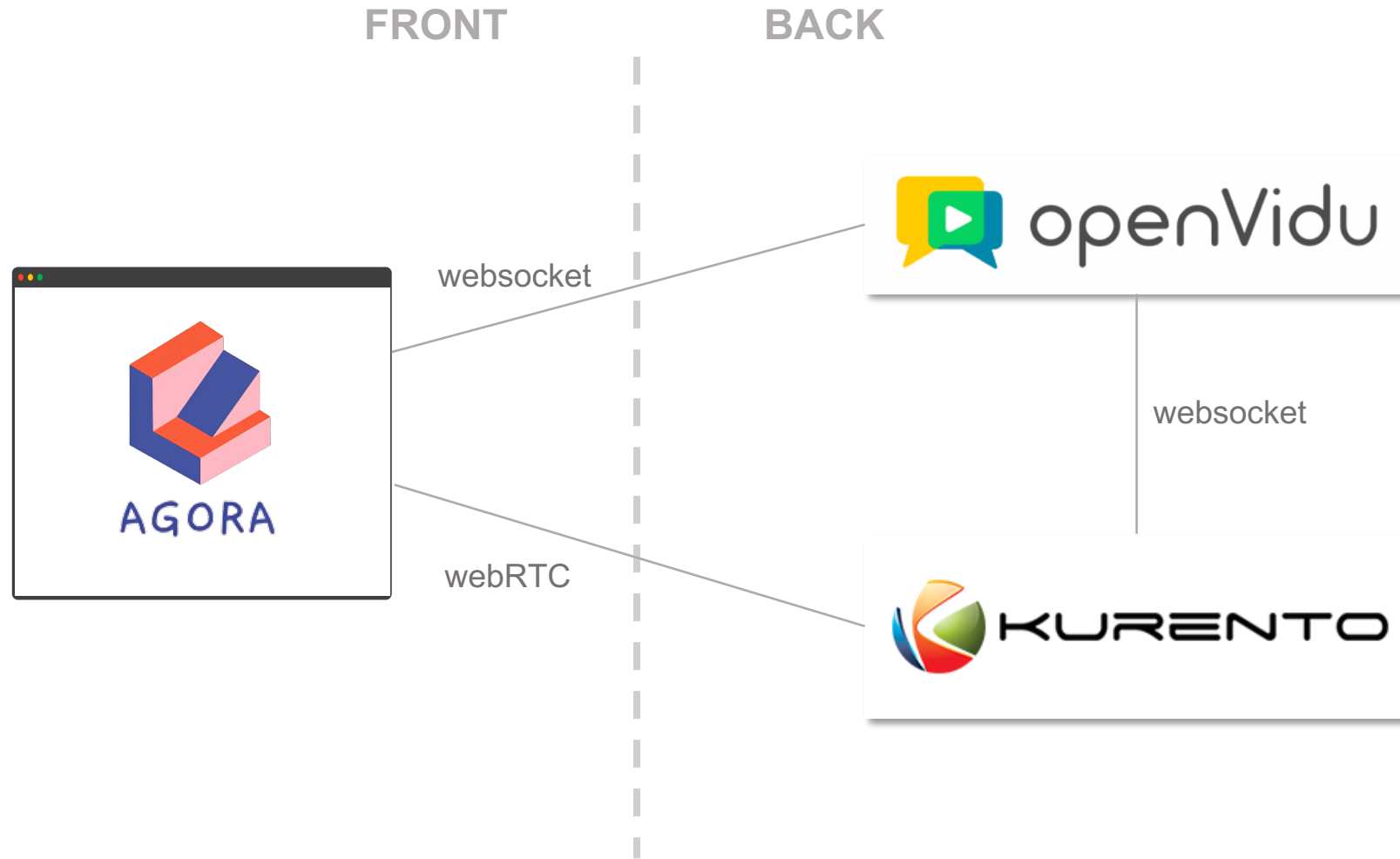


젠킨스 파이프라인을 통한 배포 자동화

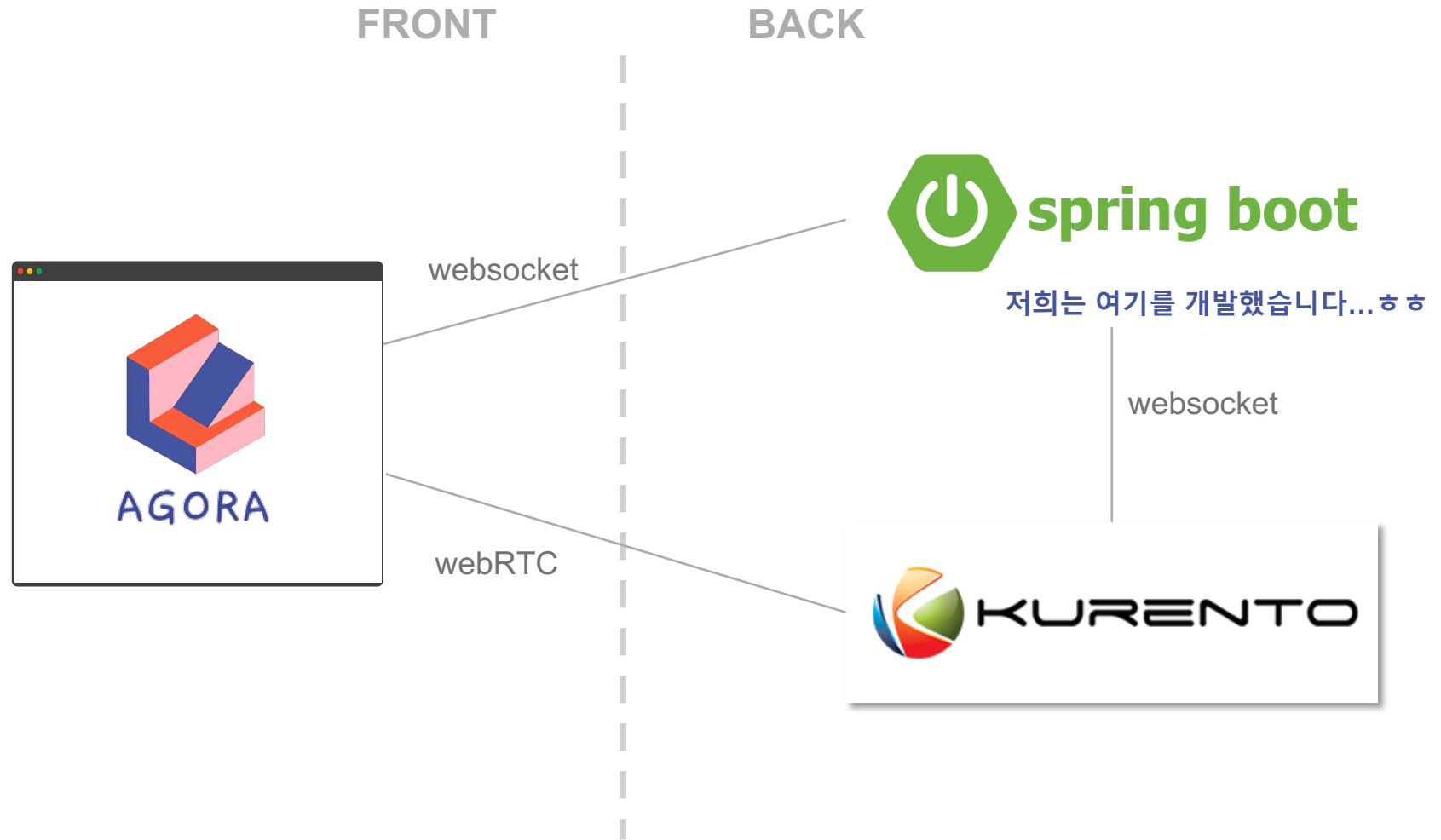
Stage View

		Declarative: Checkout SCM	Declarative: Tool Install	environment	build- module- api	build- module- chat	build- module- kurento	build- front	deploy- module- api	deploy- module- chat	deploy- module- kurento	build- deploy- front
Average stage times: (Average <u>full</u> run time: ~38s)		757ms	95ms	0ms	18s	0ms	0ms	2min 10s	49ms	50ms	50ms	790ms
#335	2월 16일 21:55 1 commit	890ms	108ms					1min 1s				1s
#334	2월 16일 21:52 1 commit	880ms	88ms					3min 0s				1s
#333	2월 16일 20:49 No Changes	531ms	91ms									
#332	2월 16일 20:41 1 commit	601ms	89ms					2min 58s				473ms failed
#331	2월 16일 20:31 1 commit	587ms	94ms					2min 15s failed	50ms failed	51ms failed	52ms failed	51ms failed
#330	2월 16일 17:48 1 commit	598ms	93ms									

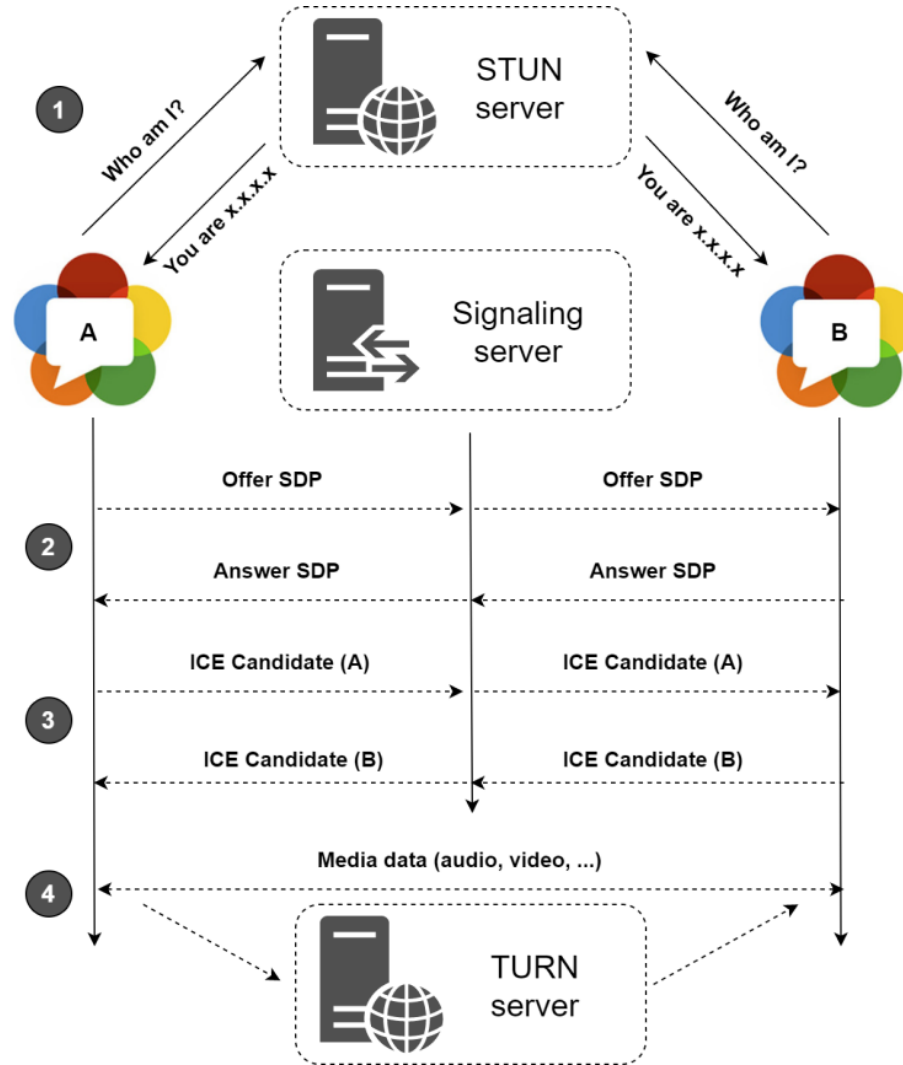
Kurento 기반 화상 애플리케이션



Kurento 기반 화상 애플리케이션



Kurento 기반 화상 애플리케이션



Kurento 기반 화상 애플리케이션

JSON 형태의 웹소켓 메시지

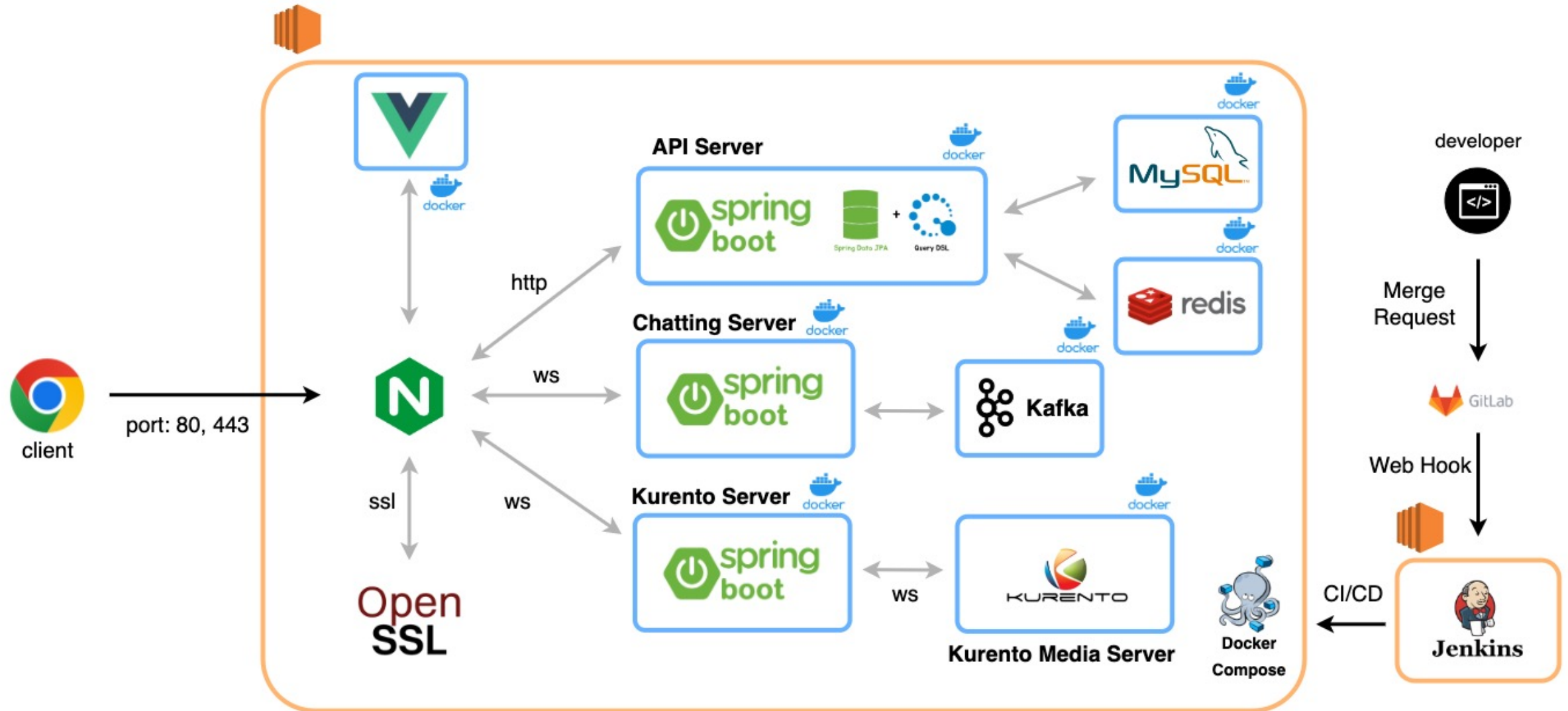
```
let message = {  
  id: 'joinRoom',  
  userName: name,  
  debateId: debateId,  
  title: title,  
  position: position,  
}  
sendMessage(message);
```

웹소켓으로 들어오는 메시지를 처리하는 Server 단 코드

```
switch (jsonMessage.get("id").getAsString()) {  
  case "createRoom":  
    createRoom(jsonMessage);  
    break;  
  case "joinRoom":  
    joinRoom(jsonMessage, session);  
    break;  
  case "joinRoomAsAudience":  
    joinRoomAsAudience(jsonMessage, session);  
    break;  
  case "receiveVideoFrom":  
    final String senderName = jsonMessage.get("sender").getAsString();  
    final Participant sender = registry.getByName(senderName);  
    final String sdpOffer = jsonMessage.get("sdpOffer").getAsString();  
    user.receiveVideoFrom(sender, sdpOffer);  
    break;  
  case "leaveRoom":  
    leaveRoom(user);  
    break;  
}
```

WebSocket을 통해 JSON 메시지를 주고 받고, 메시지의 id를 기준으로 기능을 분기

시스템 아키텍처



추후 계획



- ✓ 코드 리팩토링 및 아키텍처 재설계
- ✓ 토론 시스템 진행 기능 구현
- ✓ Kurento 서버 최적화

진짜 마지막으로...



“바퀴를 재발명하지 마라”

Thank you



AGORA