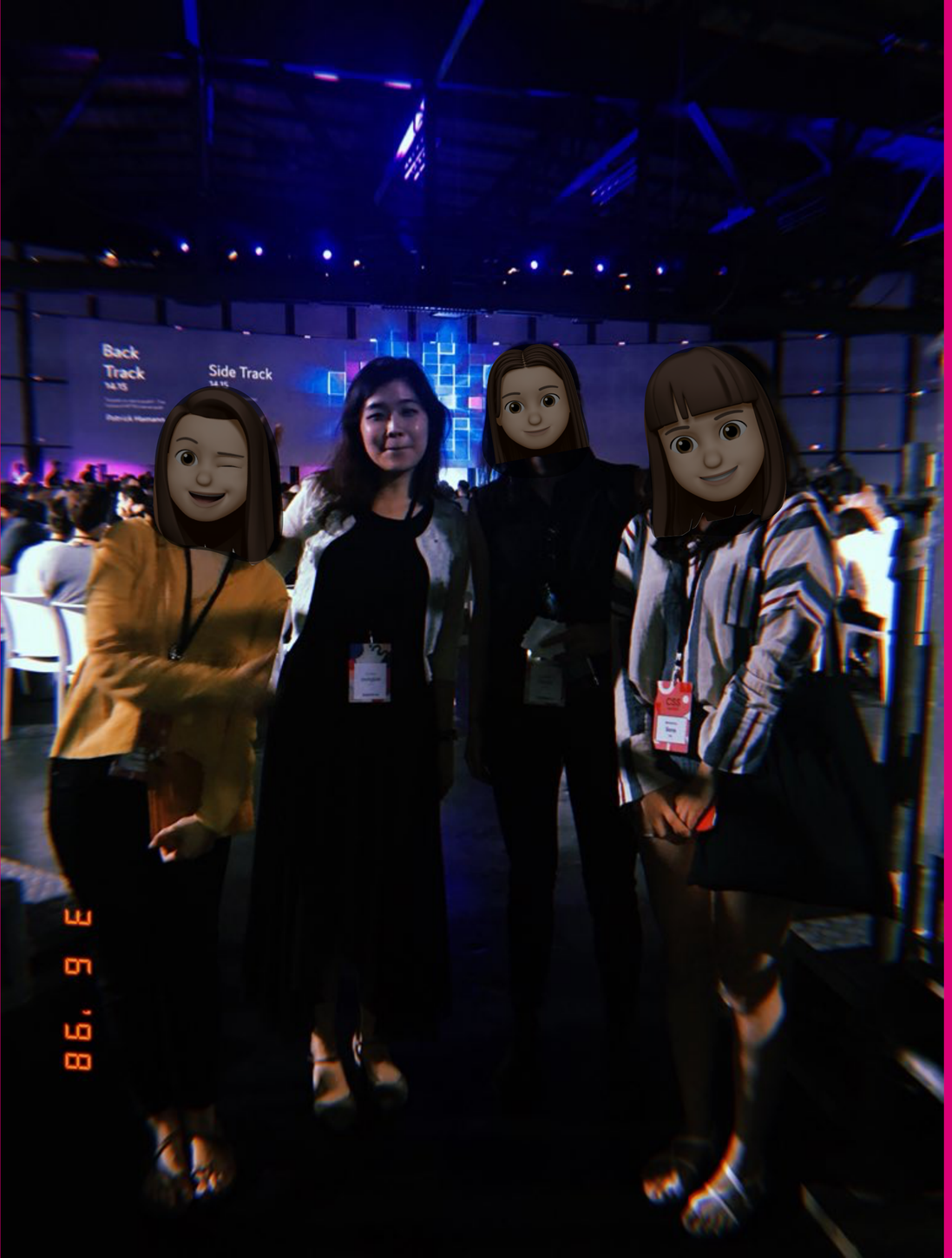
JS JSConf.korea

Users > 이정은 JeongEun Lee > JS JSConf.korea

if (extensible design)
return "work life balance"

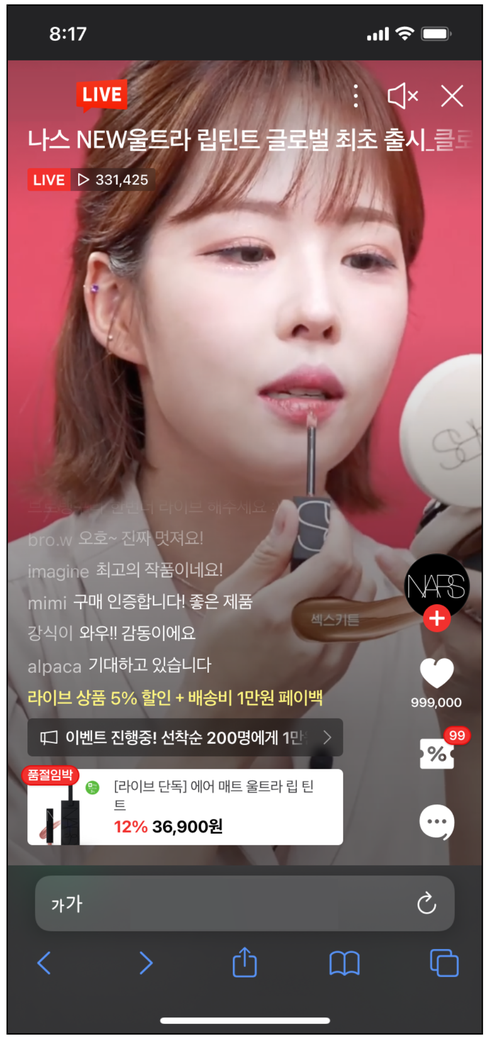


DEVELOPER

JeongEun Lee.


nexmo
zalando
NODESOURCE
TensorFlow
Google Cloud
NEIGHBOURHOODIE SOFTWARE
SINNERSCHRADER
Part of Accenture Interactive
slack
Glitch
IBM
Bloomberg
sum.cumo
twilio
AMP
cobot
adesso
npm
algolia
shopify
mozilla
Microsoft Edge
stripe
CLOUDFLARE
ebay TECH
JSConfEU

Today is ☀️



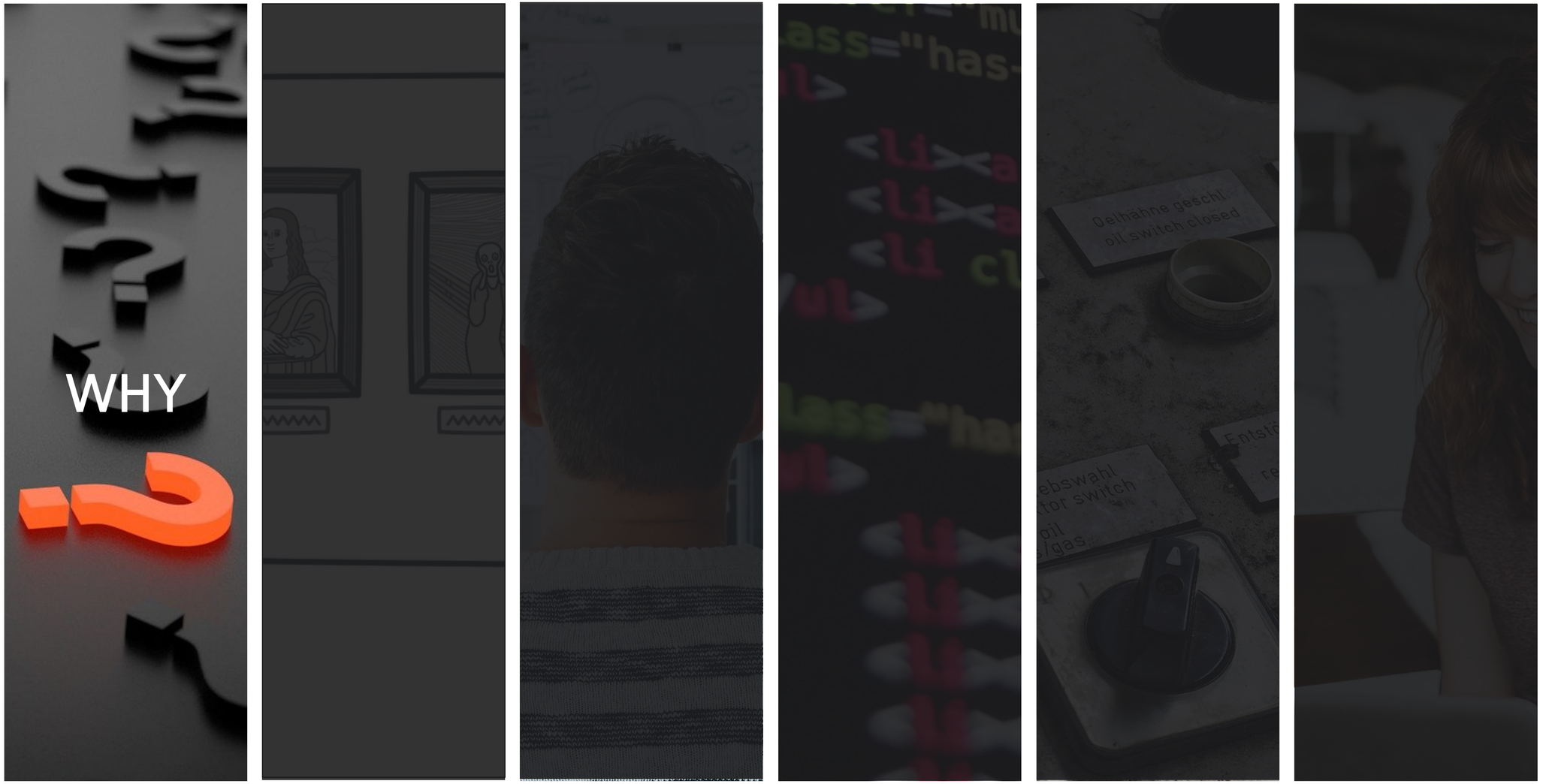
WHY

PROBLEM

SOLUTION

BENEFIT

1.



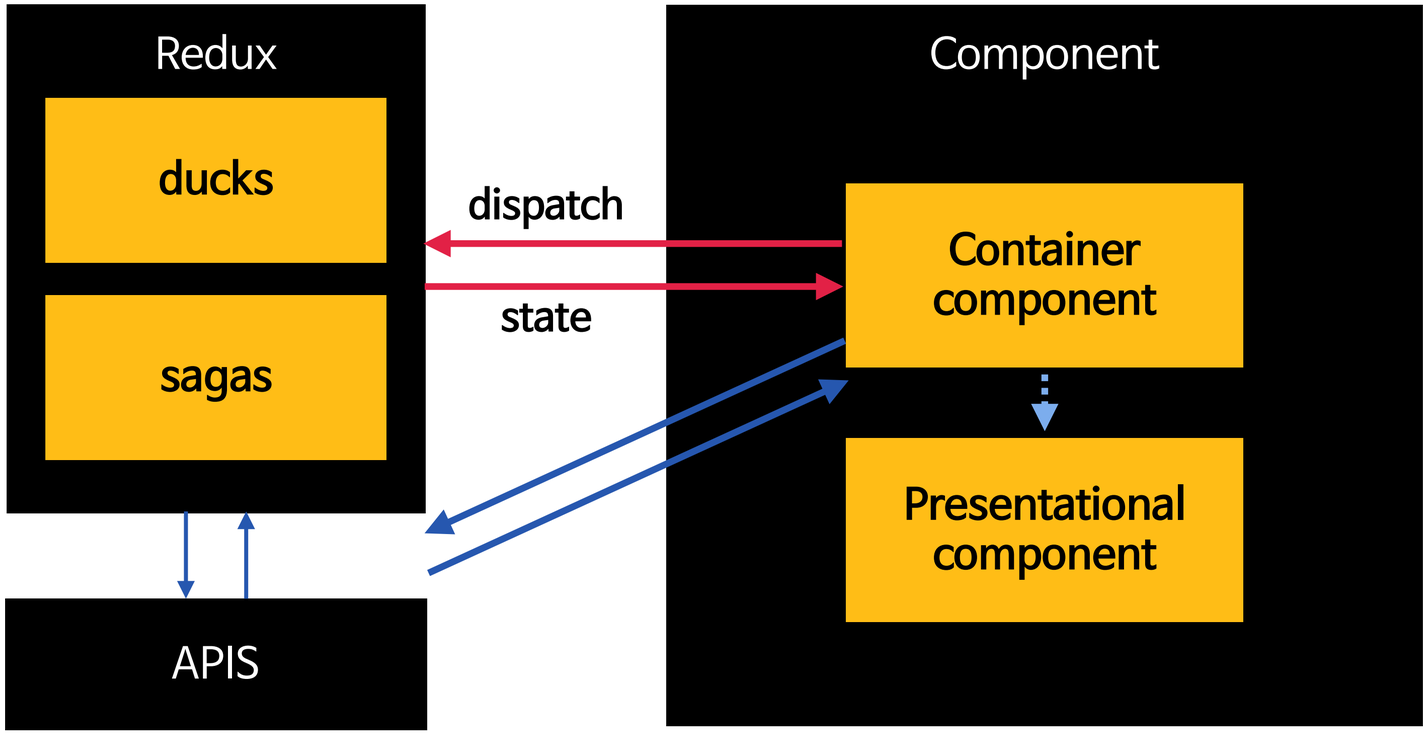
TOO MANY REQUESTS

2.

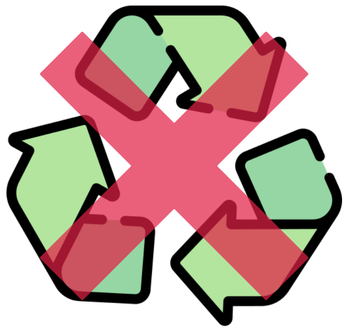
PROBLEM 1


WHY
Structure without considering scalability
Oelhähne geschl.
oil switch closed

PREVIOUS STRUCTURE



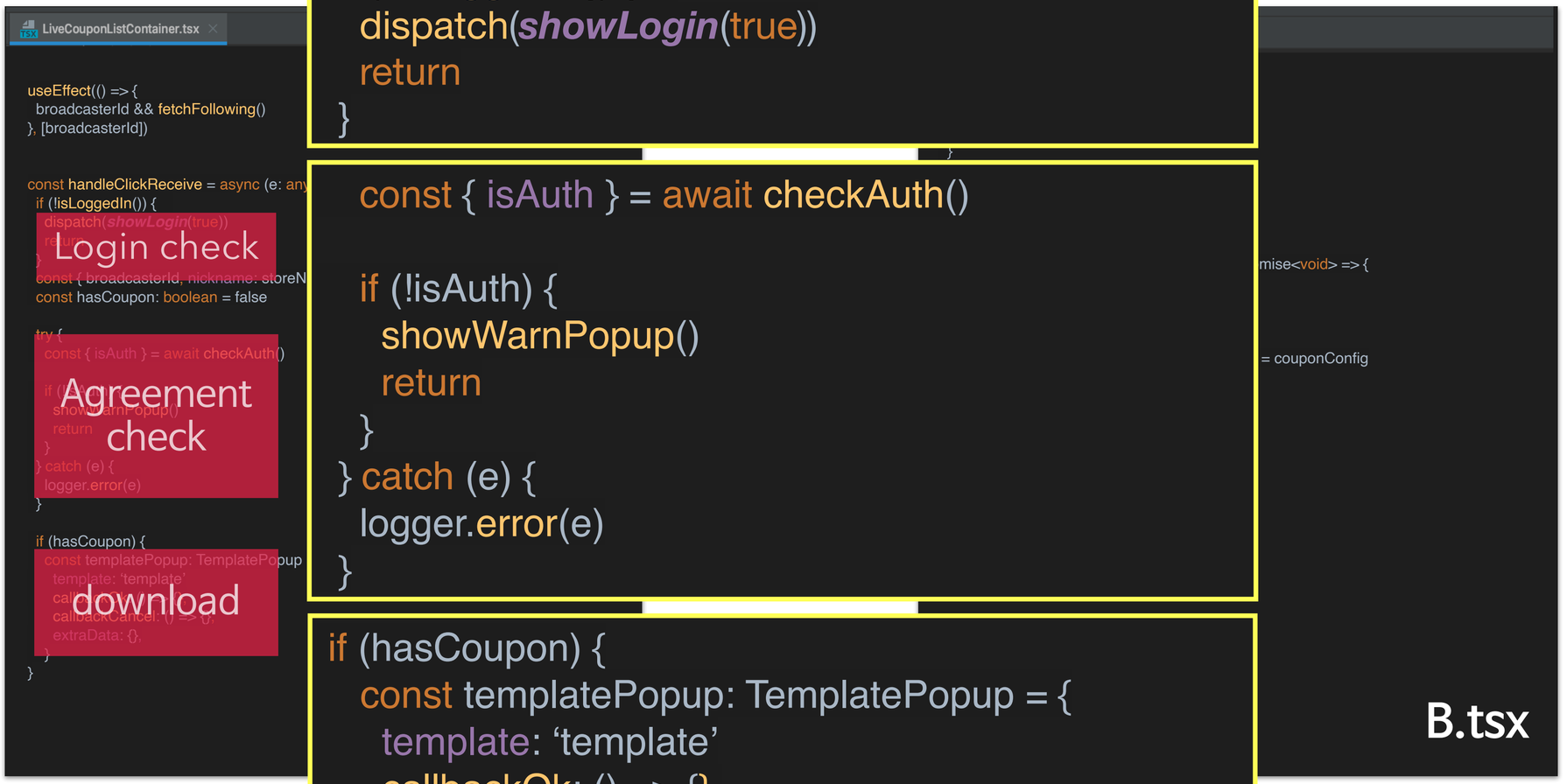
PROBLEM 1



Not reusable

Duplicate code

PROBLEM 1

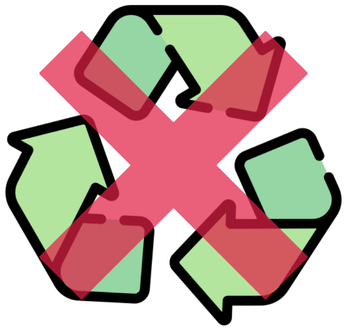



```
if (!isLoggedIn()) {
  dispatch(showLogin(true))
  return
}
```

```
  const { isAuth } = await checkAuth()

  if (!isAuth) {
    showWarnPopup()
    return
  }
} catch (e) {
  logger.error(e)
}
```

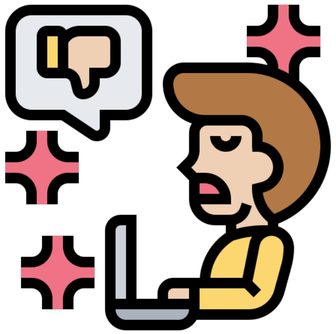
```
if (hasCoupon) {
  const templatePopup: TemplatePopup = {
    template: 'template'
    callbackOk: () => {},
    callbackCancel: () => {},
    extraData: {},
  }
}
```

PROBLEM 1



Not reusable

Duplicate code



Mistake

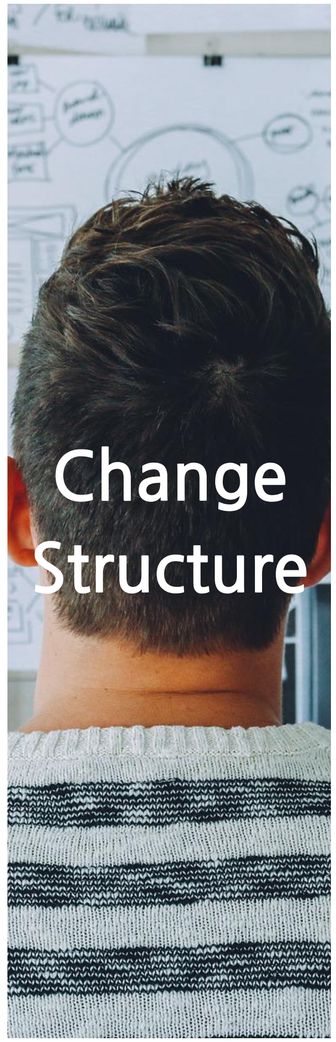
Large Component

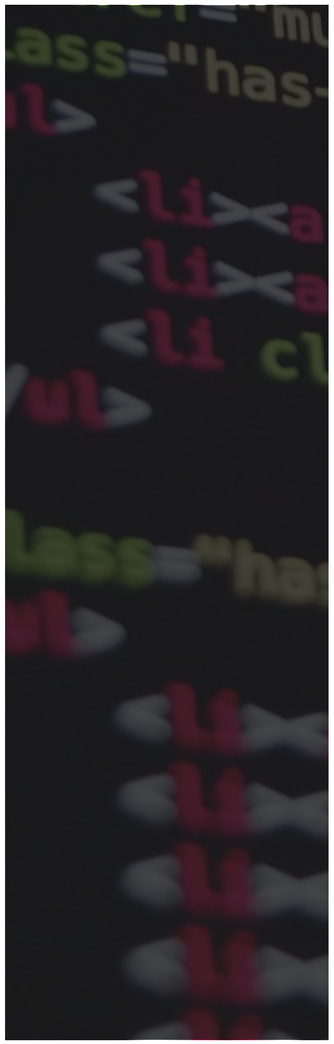
3.

SOLUTION 1

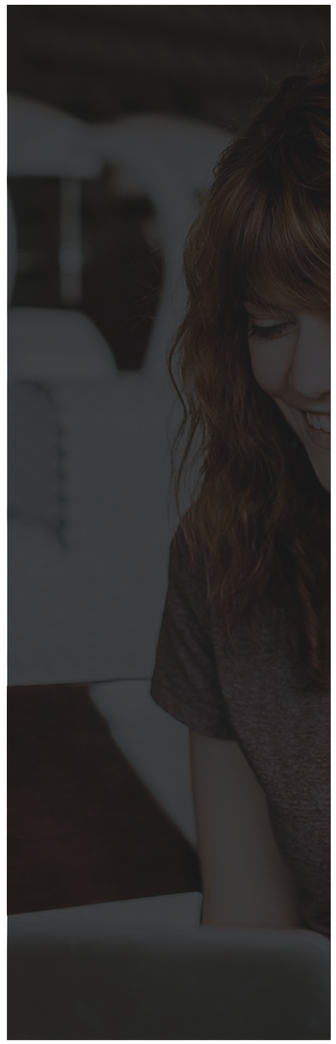

WHY


Structure without
considering scalability

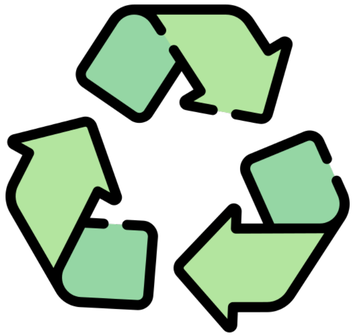

Change
Structure



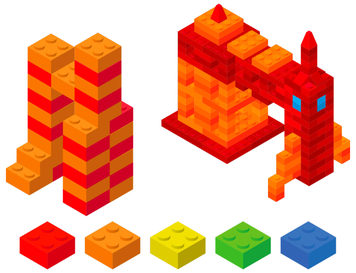

Oelhahne geschl.
oil switch closed



POINT



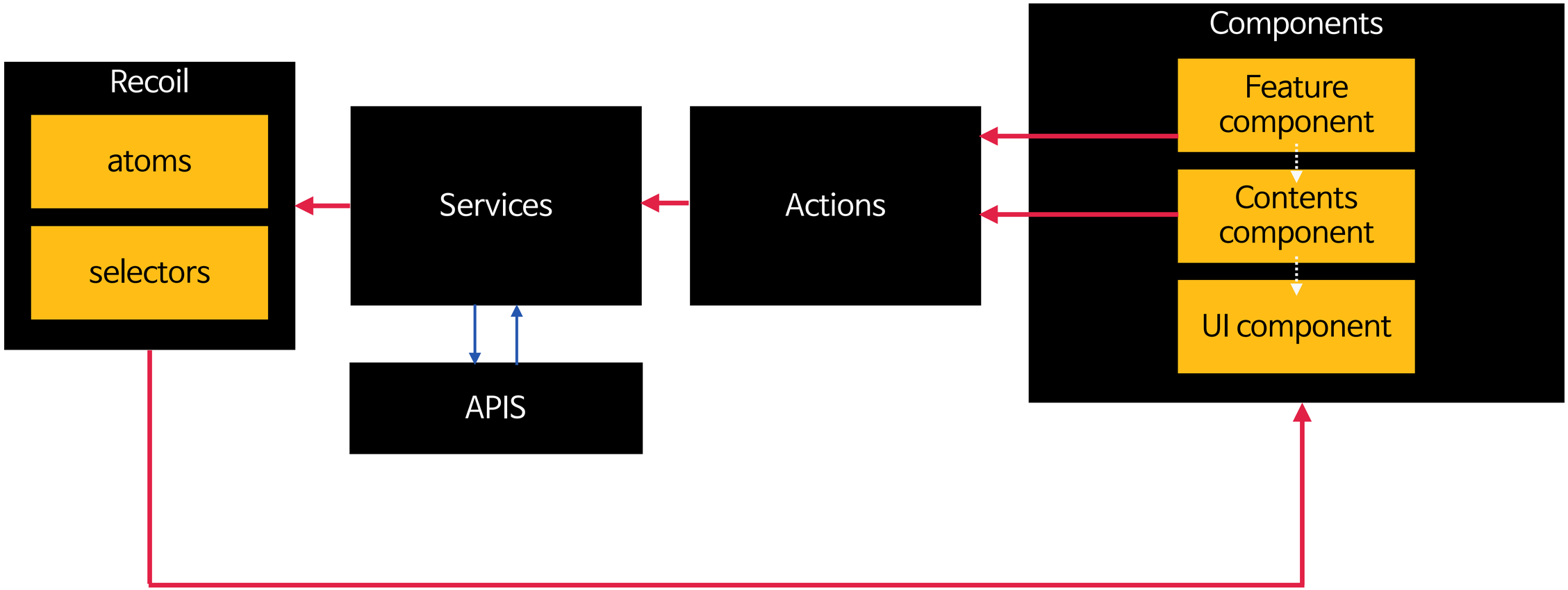
Reusable



Composable

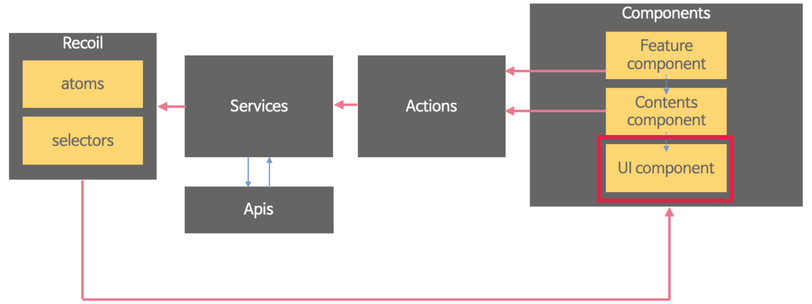
DRY

FLOW


Components
Feature component
Contents component
UI component
Actions
Services
Recoil
atoms
selectors
APIS


JSConf
Korea2020
구독
JSCONF
KOREA
2020
coming soon
개발자A Good!
개발자B 유용하네요
개발자C 기대됩니다
개발자D 온라인은 어디서 보나요?

UI Component (1/4)



JSConf Korea2020

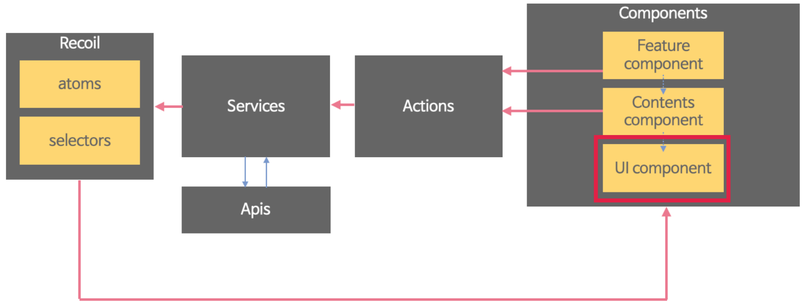
구독

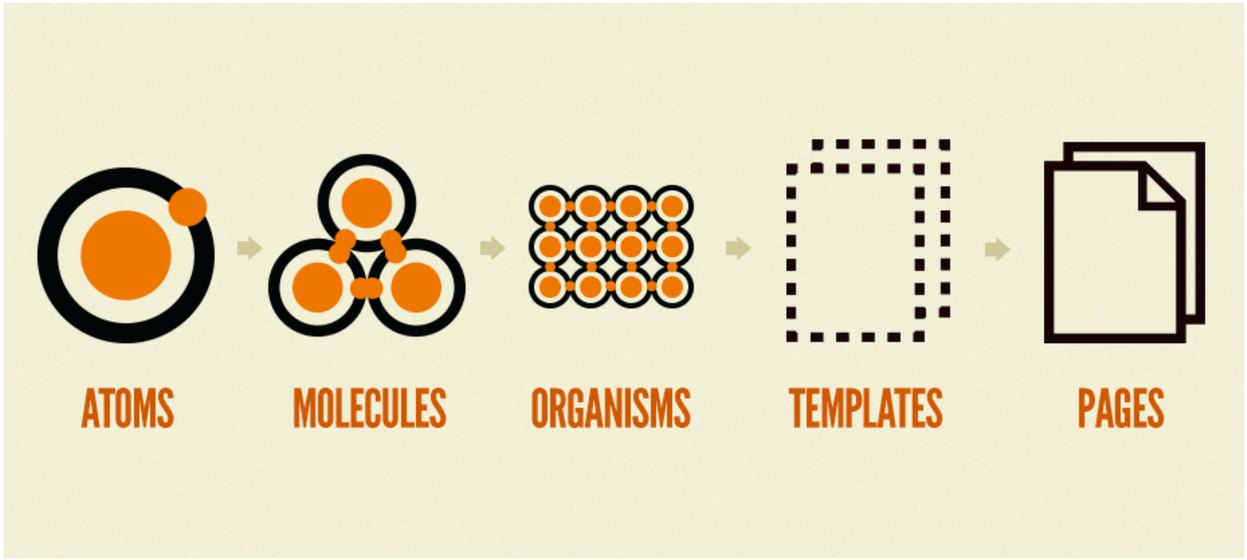
JSCONF KOREA 2020

coming soon

개발자A 좋아요
개발자B 유용하네요
개발자C 기대됩니다
개발자D 온라인은 어디서 보나요?

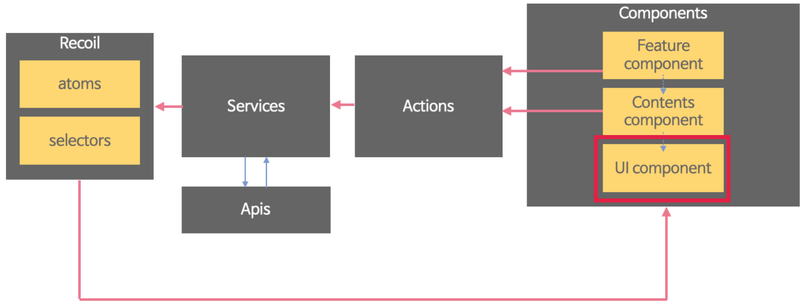
UI Component (2/4)





Atomic design

UI Component (3/4)



Step 1

JSConf Korea2020

Text

개발자A Good!

구독

Step 2

JSConf Korea2020

Logo

개발자A Good!

Comment

Step 3

개발자A Good!
개발자B 유용하네요
개발자C 기대됩니다
개발자D 온라인은 어디서 보나요?

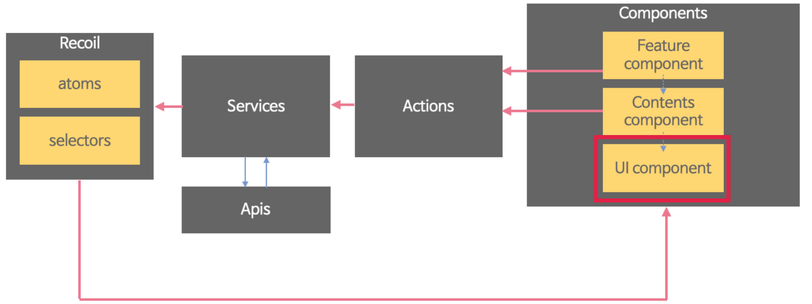
Comment list

JSCONF KOREA 2020

coming soon

Video

UI Component (4/4)



Storybook

- Button
- Dialog
- Form
- Layer
- Layout
- Tab
- Text
- UI

Reusable common UI Components

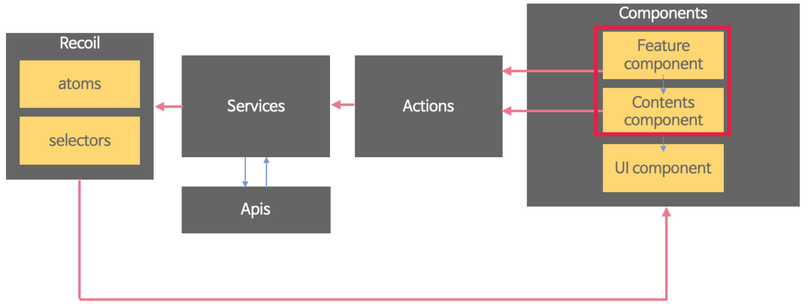
- Badge
- Button
- Dialog
- Layer
- Layout
- Modal
- UI
- Viewer

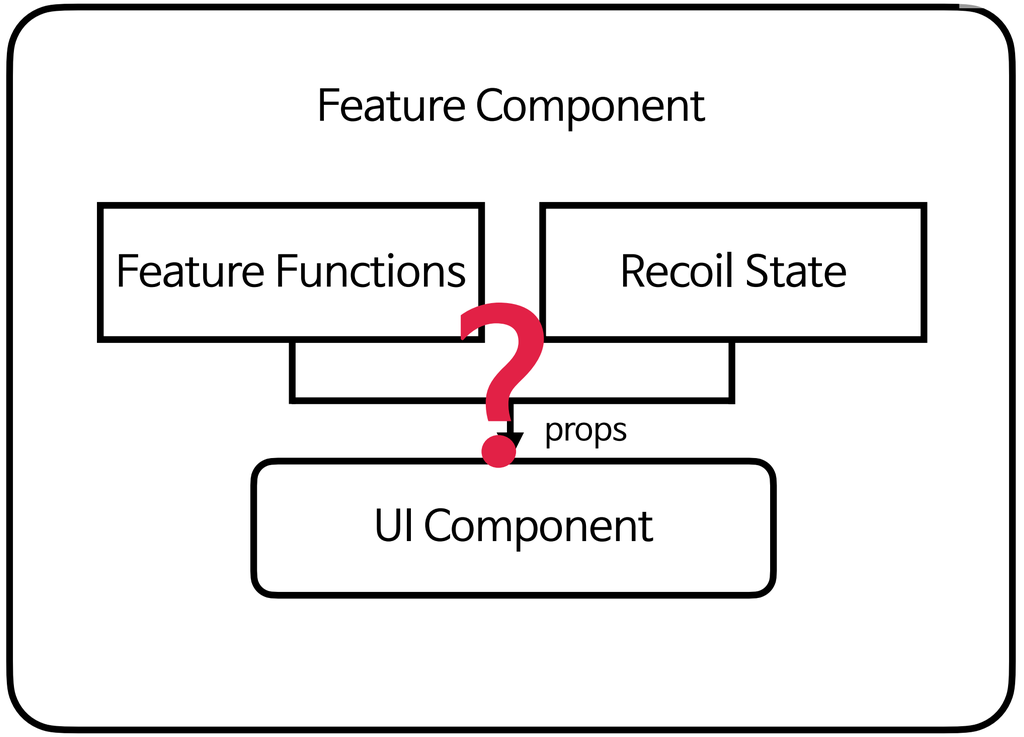
Composed UI Components

- PC Version type A
- PC Version type B
- Play Speed
- Volume Control

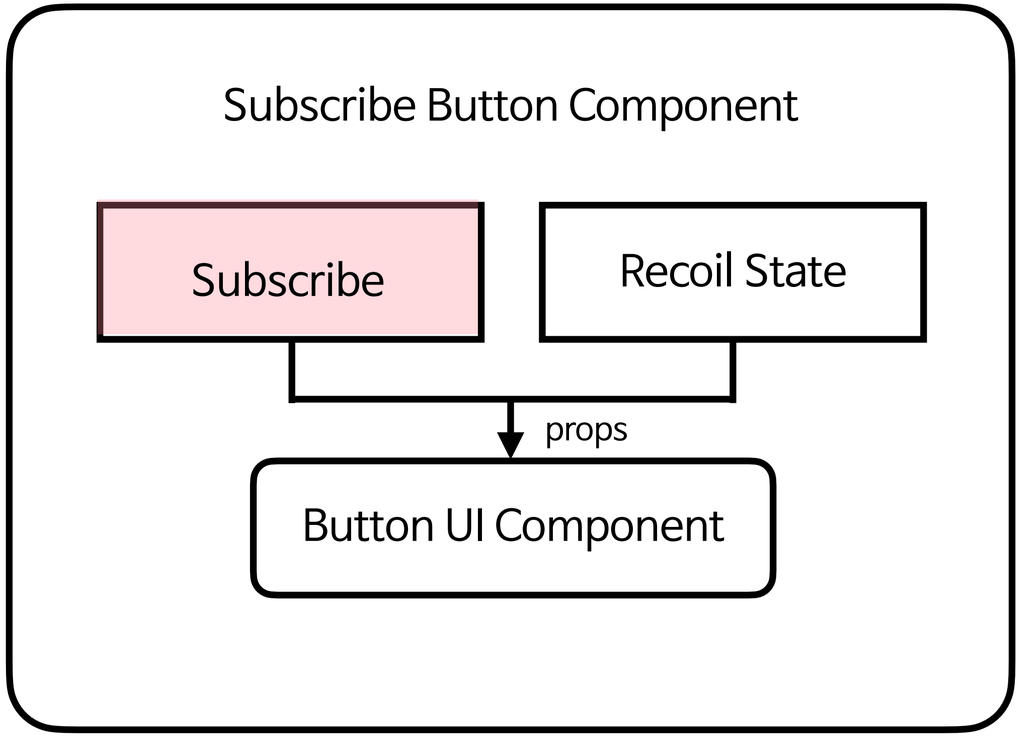
Custom Layout

Feature Component

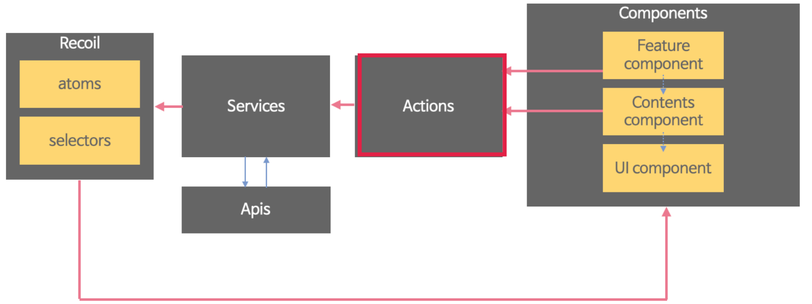





JSConf
Korea2020
구독
구독
JSCONF
KOREA
2020
coming soon
개발자A 좋아요
개발자B 유용하네요
개발자C 기대됩니다
개발자D 온라인은 어디서 보나요?


Subscribe Button Component
Subscribe
Recoil State
props
Button UI Component

Action Layer (1/3)

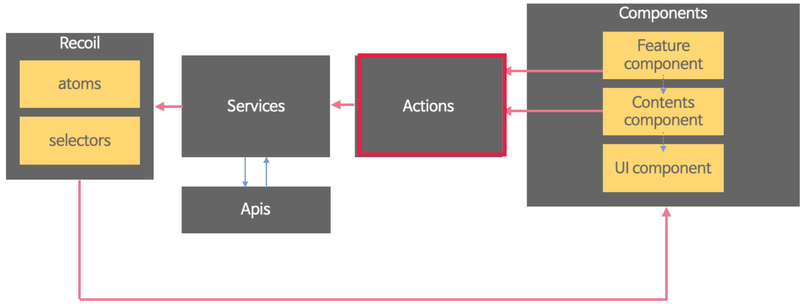


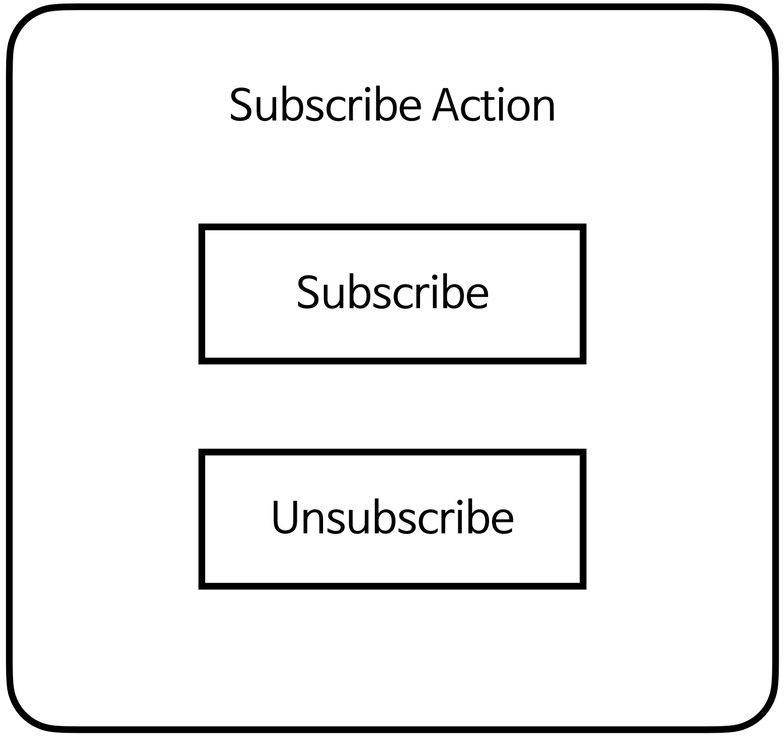
Subscribe Action

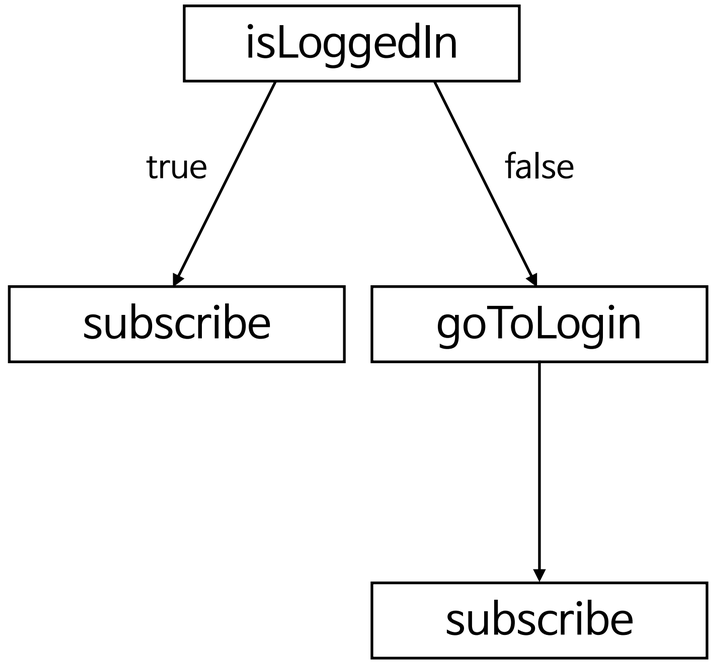
Subscribe

Unsubscribe

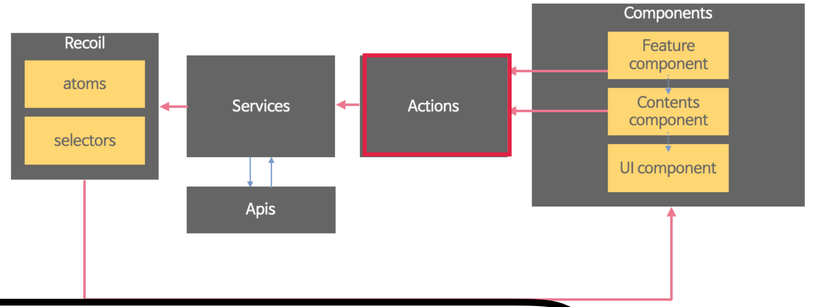
Action Layer (2/3)

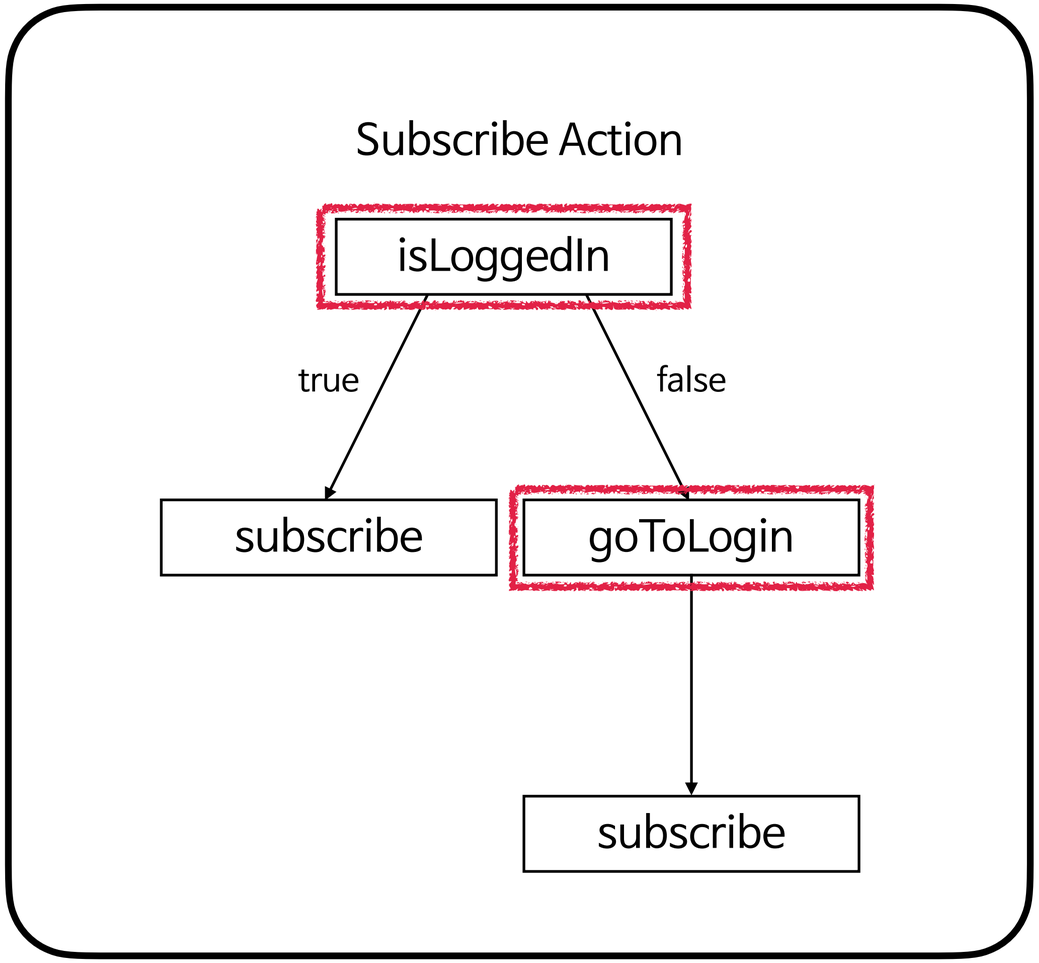


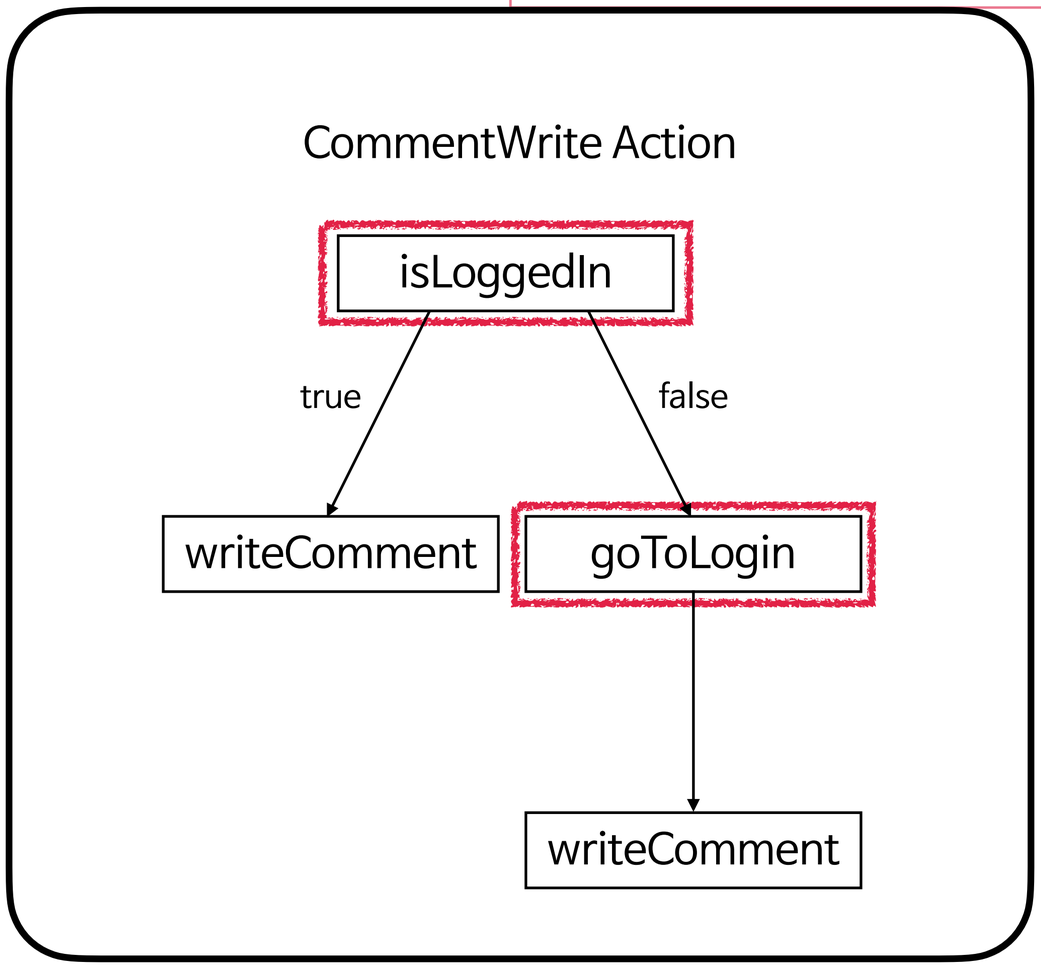




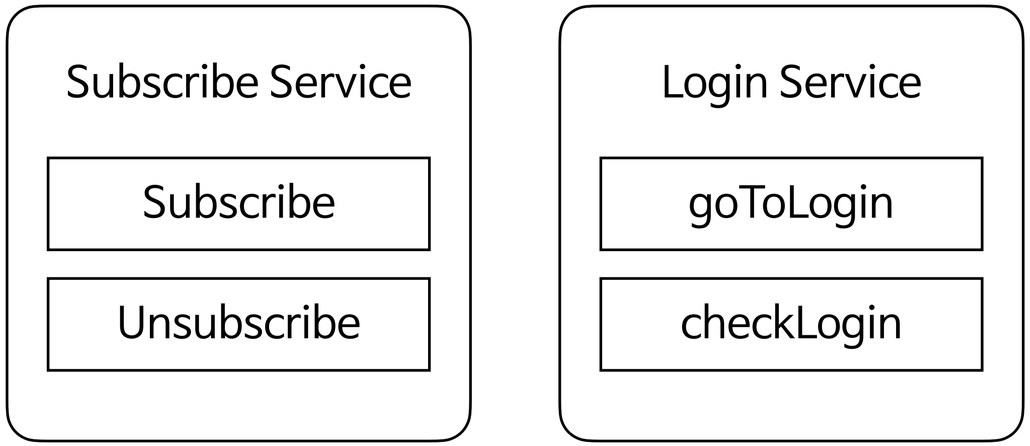
Action Layer (3/3)





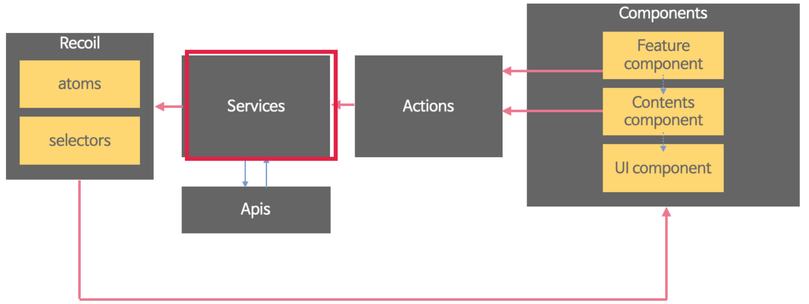


Service Layer (1/2)




```
// loginService
const checkLogin = (func) => {
  return new Promise((resolve, reject) => {
    if (isAuthenticated) {
      resolve(func())
      return
    }

    goToLogin()
  })
}

// SubscribeService
const openNotificationPopup  = () => {
  const result = window.confirm('알림 설정 하시겠습니까?')
  if (result) {
    // api 콜
    console.log('알림설정 완료')
    // store 업데이트
    return result
  }
  return false
}
```

Service Layer (2/2)



Role of Service Layer

1. SRP (Single Responsibility Principle)
2. Connect to an API
3. Update value of a Recoil State

Action Layer - Service Layer Execute

Role of Action Layer

```
openNotificationPopup: () => {
  loginService.checkLogin(() => {
    const result = notificationService.openNotificationPopup()
    result && couponService.download()
  })
}
```

Action Layer – Service Layer Execute

Role of Action Layer

1. Compose Service Layer
2. Connect business logic with Feature Component

```
openNotificationPopup: () => {
  loginService.checkLogin(() => {
    const result = notificationService.openNotificationPopup()
    result && couponService.download()
  })
}
```

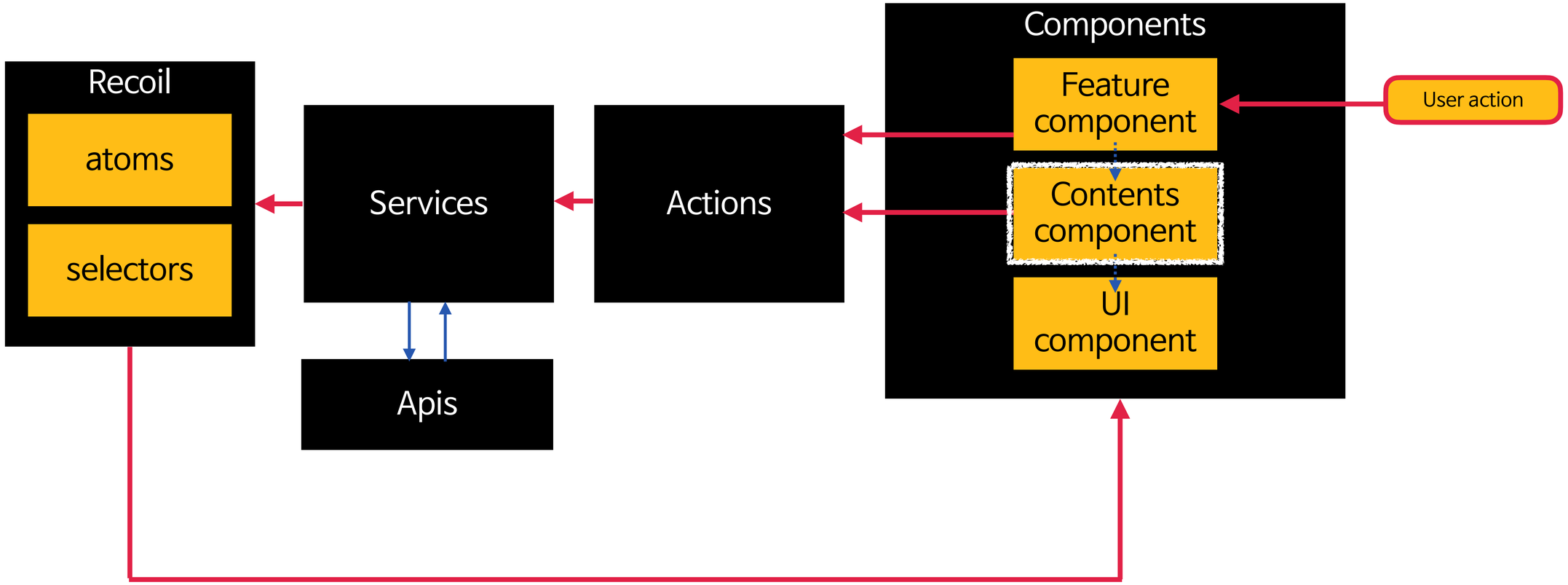
Feature Component - Action Layer Execute

```
const ProfilButtonComponent = () => {
  ...
  const handleClickProfile = useCallback(
    (e) => {
      if (isNotificationOn) {
        notificationAction.openCancellationPopup()
      } else {
        notificationAction.openNotificationPopup()
      }
    },
    [isNotificationOn],
  )

  return (
    <ProfileButton
      profileInfo={broadcastProfile}
      isSetAlarm={isNotificationOn}
      onClick={handleClickProfile}
    />
  )
}
```

```
const NoticeButtonComponent = () => {
  const handleClickNotice = useCallback(
    (e) => {
      notificationAction.openNotificationPopup()
    },
    [isNotificationOn],
  )

  return (
    <NoticeButton
      isSetAlarm={isNotificationOn}
      onClick={handleClickNotice}
    />
  )
}
```

CURRENT STRUCTURE

CURRENT STRUCTURE

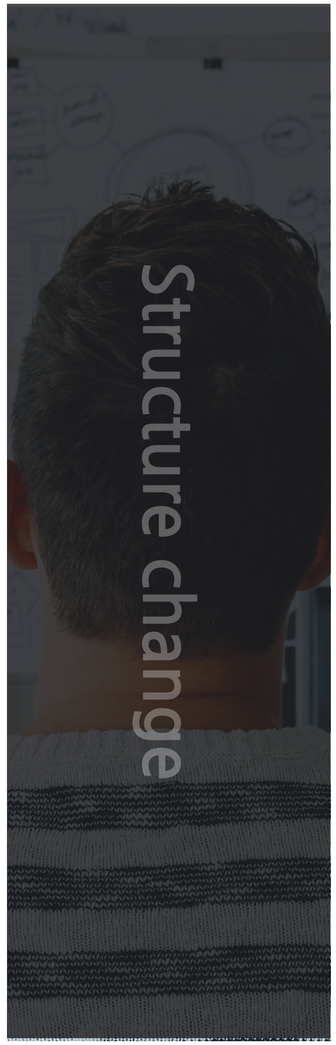


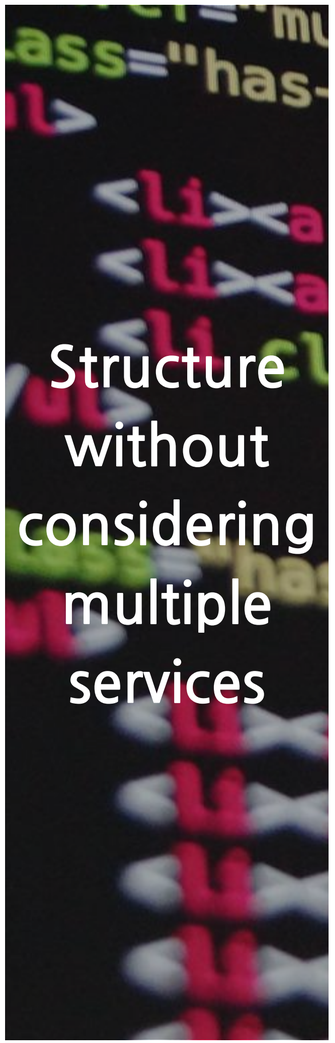
4.

PROBLEM 2

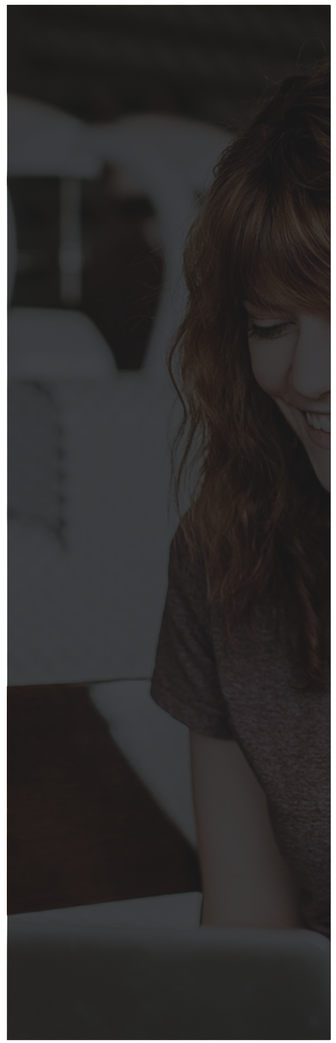

WHY


Structure without
considering scalability

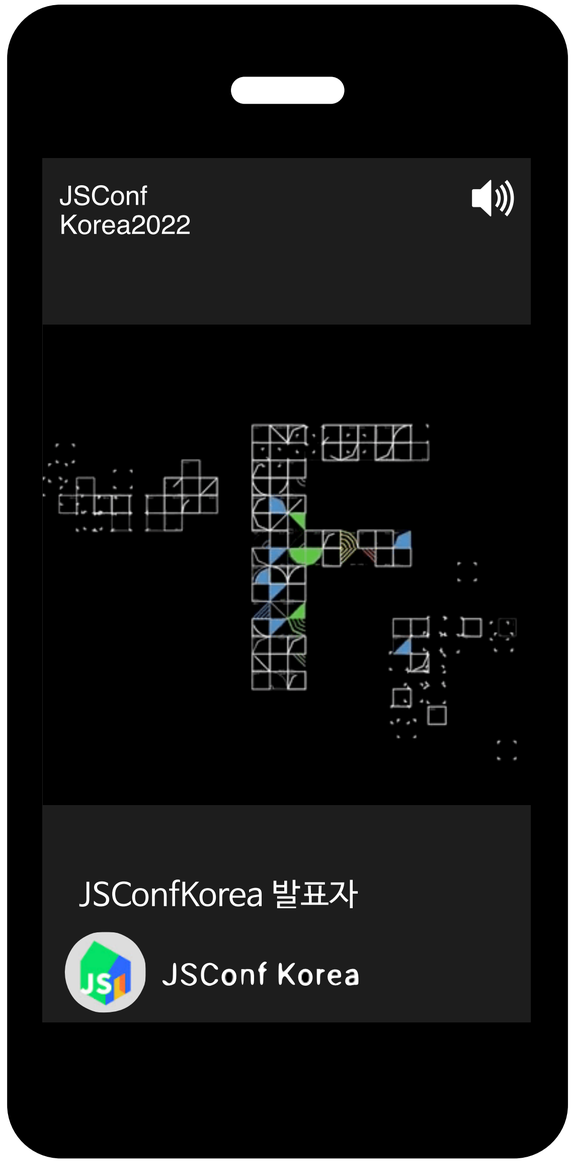

Structure change


Structure
without
considering
multiple
services


Oelhahne geschl.
oil switch closed



Service for 2020



Service for 2022

PROBLEM 2

PROBLEM 2

```
const isABCService = (serviceId: string): boolean => serviceId ===
SERVICE_ID.ABC
const isXYZService = (externalId: string): boolean => serviceId &&
externalId === SERVICE_ID.XYZ
.
.
.
const getUseFeature = () => {.
  if (isABCService(serviceId)) {
    return true
  } else if (isXYZService(serviceId)) {
    return false
  } else {
    return hasCondition
  }
}
```

```
const renderLogo = useMemo(() => {
  // 1. 커스텀 로고
  if (logoSrc) {
    const logoWidth: number = isABCService(serviceId) ? 91 : 79
    const logoHeight: number = isABCService(serviceId) ? 24 : 28
    const logoSrc: string = isABCService(serviceId) ? '/abcLogo.jpg' : logoSrc

    return <img src={logoSrc} width={logoWidth} height={logoHeight} alt={logoName} />
  }
}, [serviceId])
```

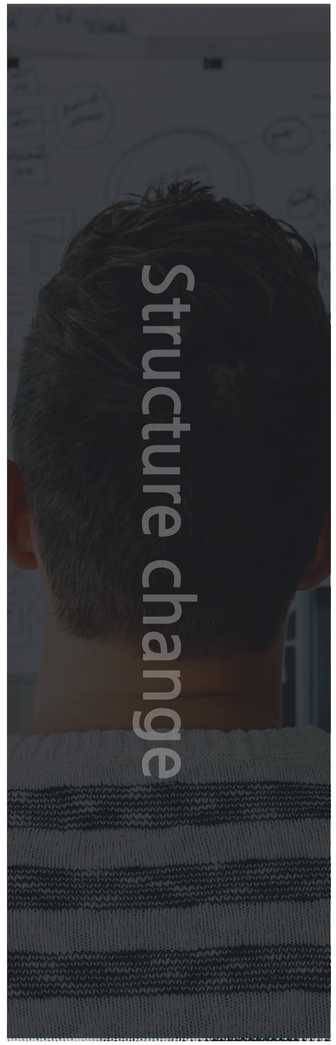
if-else Hell

5.

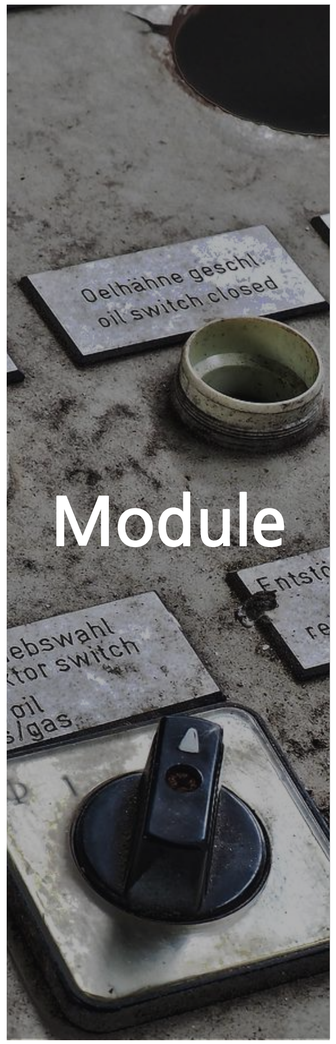
SOLUTION 2

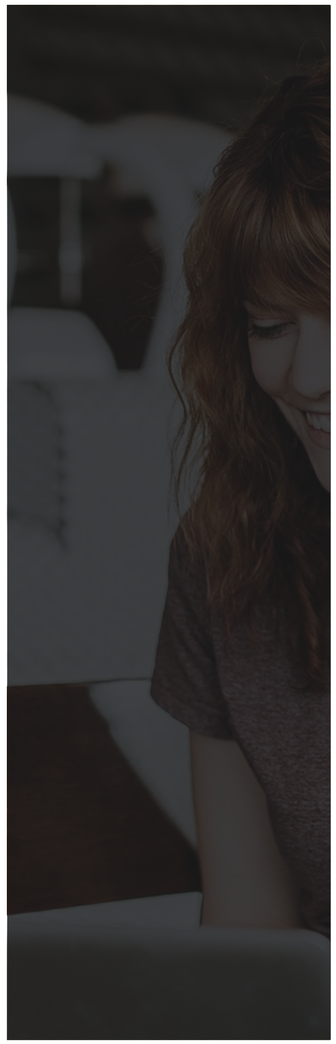

WHY


Structure without
considering scalability

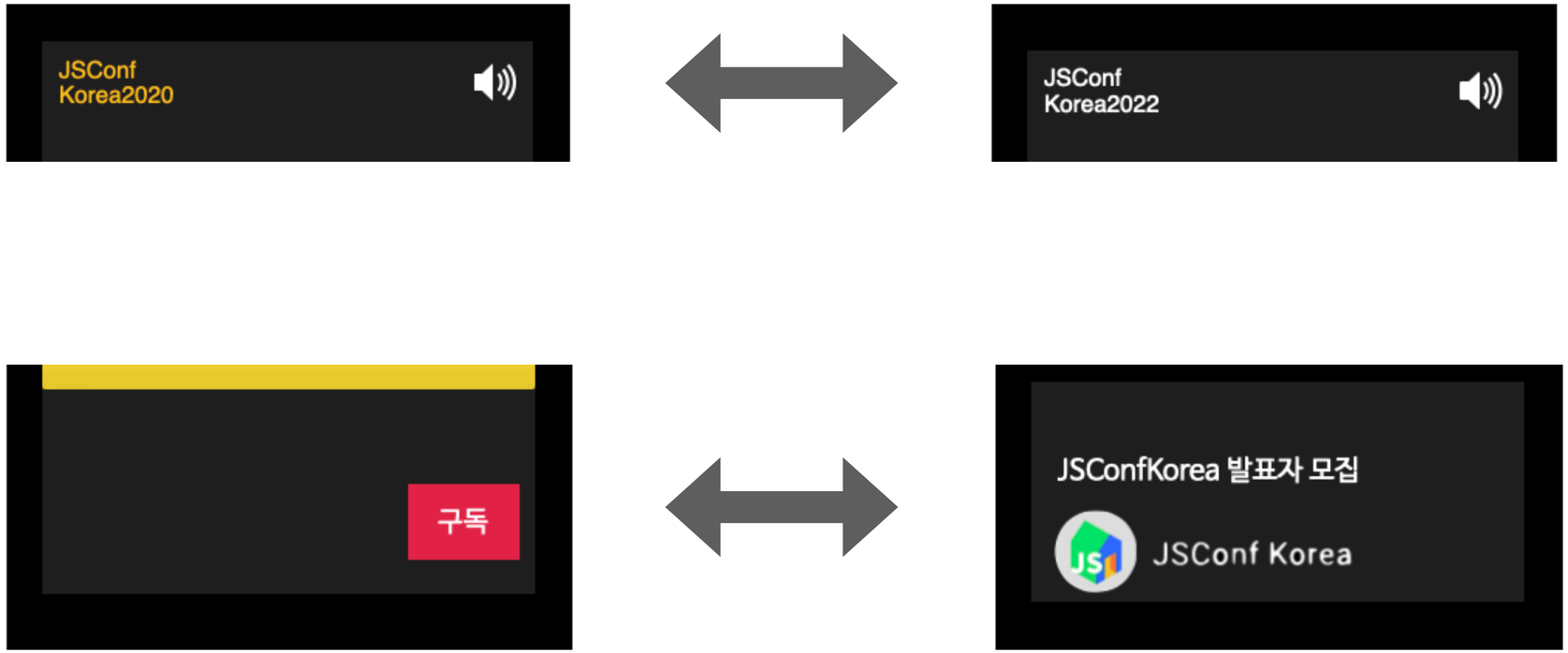

Structure change


Structure without
considering multiple services


Oelhähne geschl.
oil switch closed
Module



POINT

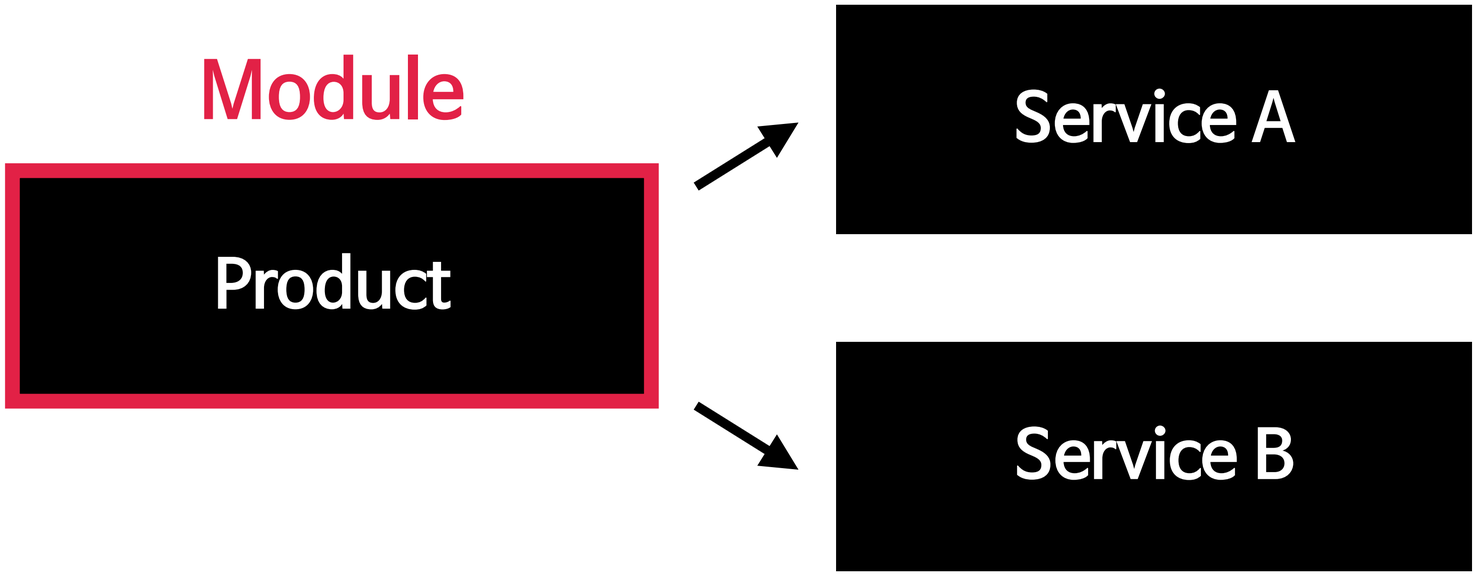

JSConf
Korea2020
JSConf
Korea2022
구독
JSConfKorea 발표자 모집
JSConf Korea

POINT

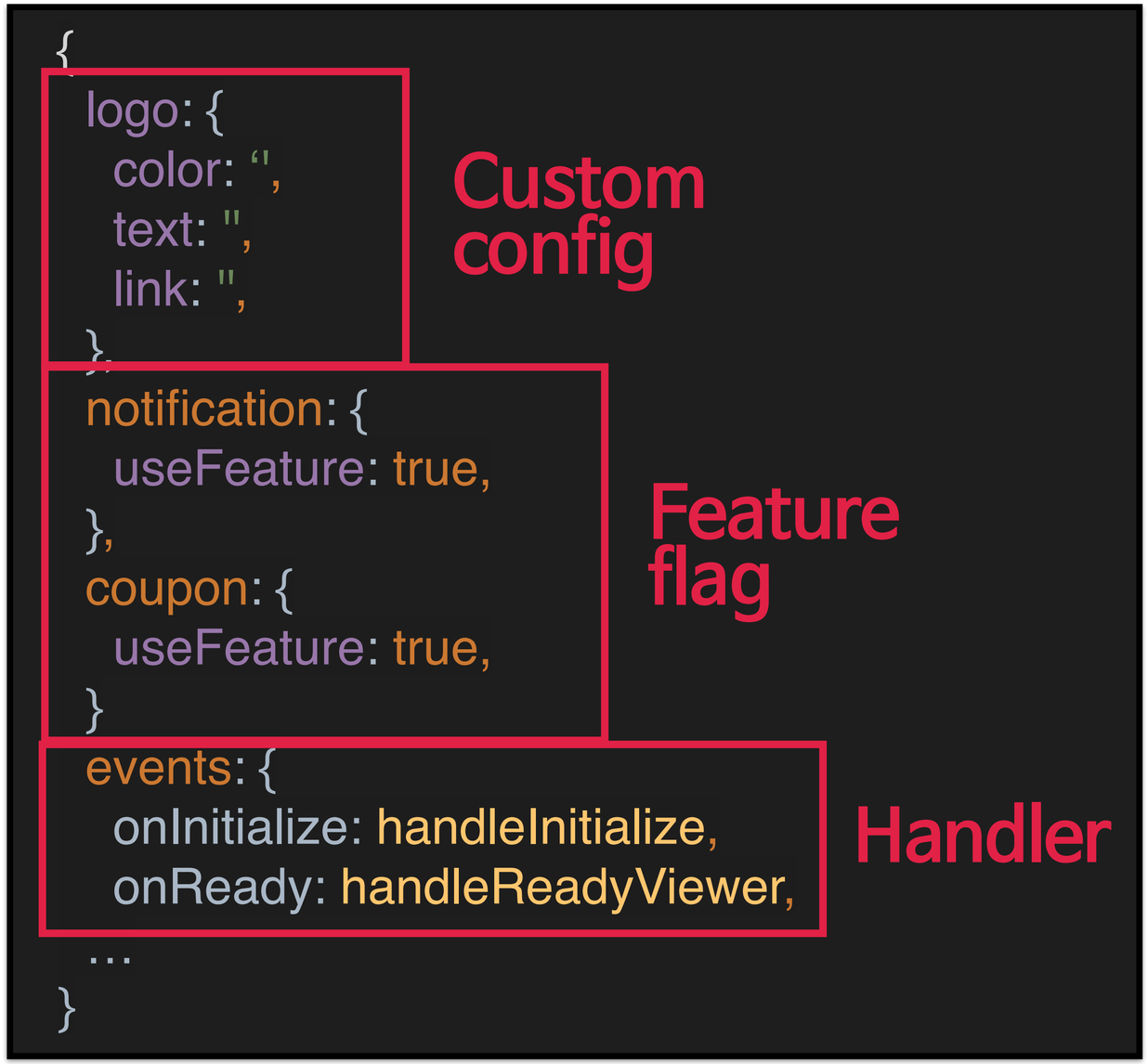
Product

!=

POINT



Config


{
logo: {
color: '',
text: '',
link: '',
},
notification: {
useFeature: true,
},
coupon: {
useFeature: true,
}
events: {
onInitialize: handleInitialize,
onReady: handleReadyViewer,
...
}
Custom config
Feature flag
Handler

Custom Config

```
{
 logo: {
  text: ‘JSConf₩nKorea2020}
  style: { fontColor: ‘yellow’ }
  onClick: () => {
   location.href = ‘https://2020.jsconf.kr’
  }
 }
 …
}
```

JSConf
Korea2020

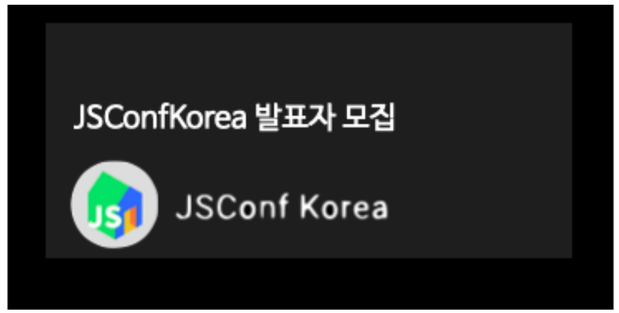
```
{
 logo: {
  text: ‘JSConf₩nKorea2022}
  style: { fontColor: ‘white’ },
  onClick: () => {
   location.href = ‘https://2022.jsconf.kr’
  }
 }
 …
}
```

JSConf
Korea2022

Feature flag (1/2)

```
{
 ...
 title: {
  useFeature: false
 },
 profile:  {
  useFeature: false
 },
 subscribe: {
  useFeature: true
  type: 'textButton'
 }
}
```


```
{
 ...
 title: {
  useFeature: true
 },
 profile:  {
  useFeature: true
 },
 subscribe: {
  useFeature: false
 }
}
```



Feature flag (2/2)

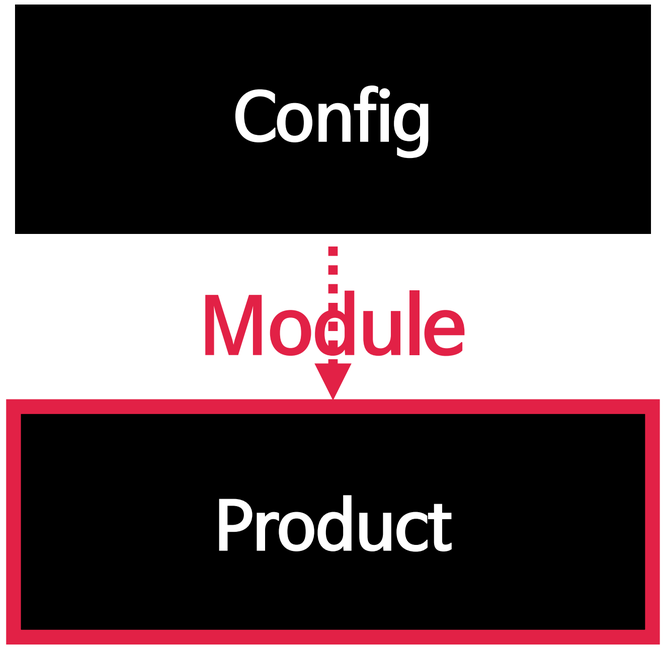
Feature Flag - How to Control

```
const featureFlagConfig = useRecoilValue(featureFlagSelector)
const canVisible = useMemo( factory: () => {
  // visibility config 와 상관 없이
  return ignoreVisibilityConfig || isVisibleByVisibilityConfig
}, deps: [isVisibleByVisibilityConfig, ignoreVisibilityConfig])

// 컴포넌트 마운트 여부 체크(설정 없을 경우 기본값 true)
const shouldMount = shouldComponentMount( payload: { featureFlagConfig })
```

Feature Flag - How to use in Components

```
export default connectVisibilityController(CommentsComponent, config: {
  displayType: ComponentDisplayType.VISIBILITY,
  shouldComponentMount: ({ featureFlagConfig : FeatureFlagConfig }) => {
    return featureFlagConfig.useChat
  },
})
```

Module (1/2)



JSConf
Korea2020

JSCONF KOREA 2020

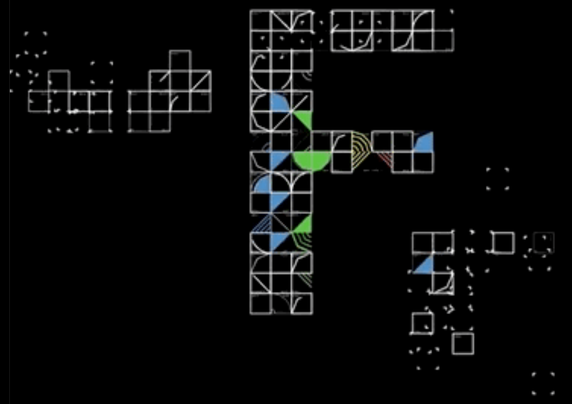
coming soon

구독

```
{
 logo: {
  text: 'JSConf₩nKorea2020}
  style: { fontColor: 'yellow' }
  onClick: () => {
   location.href = 'https://2020.jsconf.kr'
  }
 }
 title: {
  useFeature: false
 },
 profile:  {
  useFeature: false
 },
 subscribe: {
  useFeature: true
  type: 'textButton'
 }
}
```

Module

JSConf
Korea2022



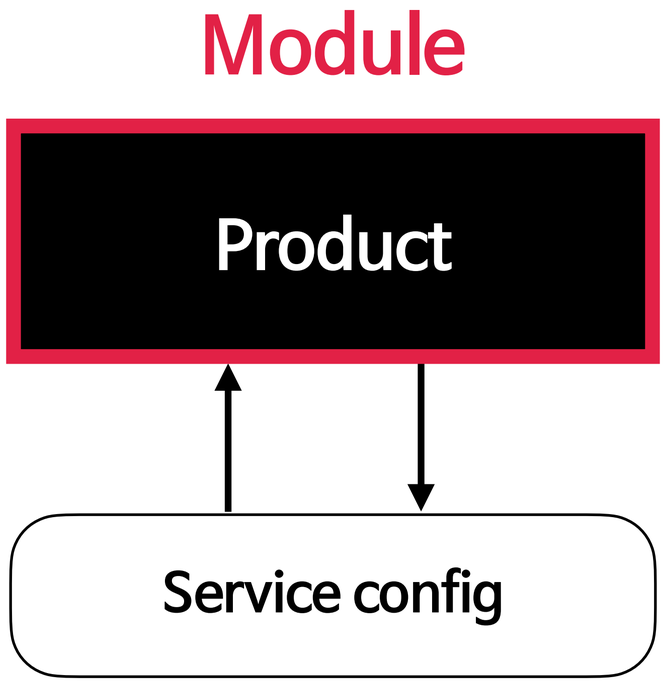
JSConfKorea 발표자 모집

JSConf Korea

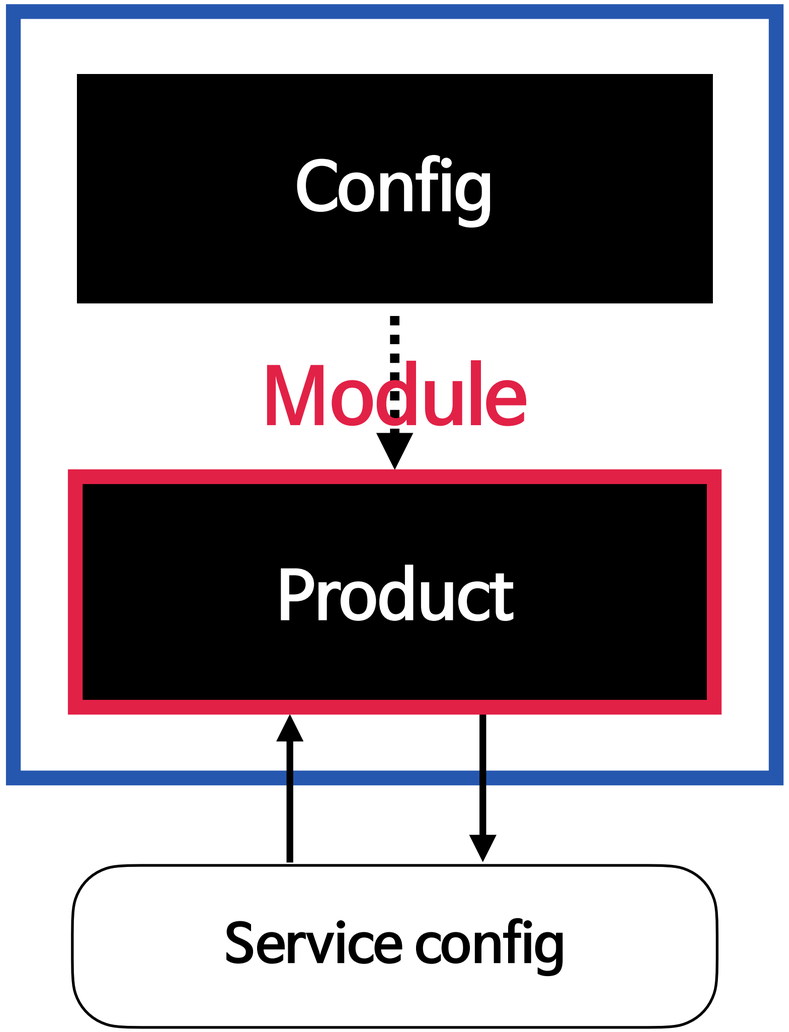
```
{
 logo: {
  text: 'JSConf₩nKorea2022}
  style: { fontColor: 'white' },
  onClick: () => {
   location.href = 'https://2022.jsconf.kr'
  }
 }
 title: {
  useFeature: true
 },
 profile:  {
  useFeature: true
 },
 subscribe: {
  useFeature: false
 }
}
```

Module

Module (2/2)



Wrapped Module

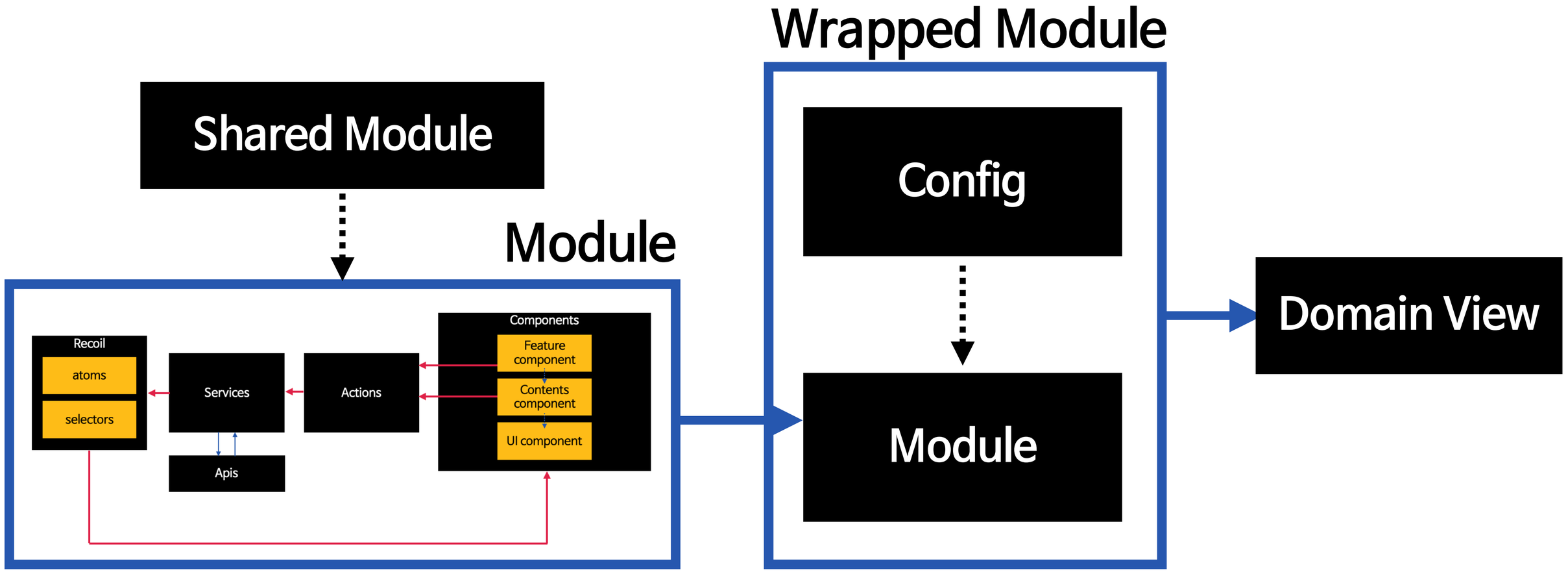


Wrapped Module

Domain View

TOTAL STRUCTURE

TOTAL STRUCTURE




Result


Module


Structure without
considering multiple services


Structure change


Structure without
considering scalability


WHY


JSConf
Korea2020
구독
JSCONF
KOREA
2020
coming soon
개발자A 좋아요
개발자B 유용하네요
개발자C 기대됩니다
1DAY
JSConf
Korea2020
구독
JSCONF
KOREA
2020
coming soon
개발자A 좋아요
개발자B 유용하네요
개발자C 기대됩니다
개발자D 온라인은 어디서 보나요?
개발자B https://jsconf.kr 참고하세요
개발자D 감사합니다.
개발자E 오프라인도 해요 ?
관리자 이번행사는 온라인으로만 진행됩니다.

JSConf
Korea2020

JSConf
Korea
2019

구독

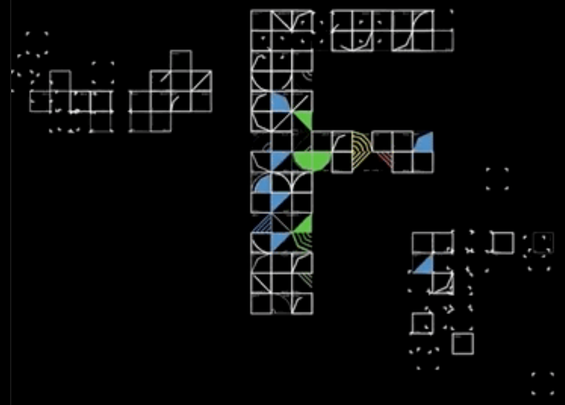
JSConf
Korea2021

JS

JSConf Korea

구독

JSConf
Korea2022



JSConfKorea 발표자 모집

JSConf Korea

BENEFITS

Increased productivity

Fewer Mistakes

Multiple

#설계 잘한

서버 배포 순서 고려 안해도 되어서 좋아요

스펙 문서 = 스토리북

플랫폼제공

화면 작업할 때 스토리북만 있으면 만사 ok

그냥+쉬고+싶다.

리팩토링

모듈화 된 프로덕트들로 페이지 구성 가능

비개발자와 협업하기 쉬워서 좋아요

#온.근.한

모듈 의상실 룩북

모듈화 된 설계로 업무가 빨라졌어요 😀

A/B 테스트 레이아웃 구성의 자유도가 높아졌어요

#F/U

변경하기 쉬워요

프로토 타이핑이 제일 쉬웠어요

#다_해주는_개발자

[참조] 함께 일하는 사람들의 간증

Work life balance

THANK YOU :)

Instagram/twitter: @snooje
E-mail: sweetzhzh@gmail.com