



네이버 검색의 Server Driven UI LAPIN

김정기/김명진 NAVER Search FE Solution

CONTENTS

1. 도입 배경
2. 기존 FE 업무 프로세스
3. 시스템 기반의 업무 프로세스
4. Server Driven UI - Lapin
5. 서버/클라이언트 개발
6. 서버 구성/빌드/배포
7. 서버 운영/모니터링
8. 앞으로의 계획

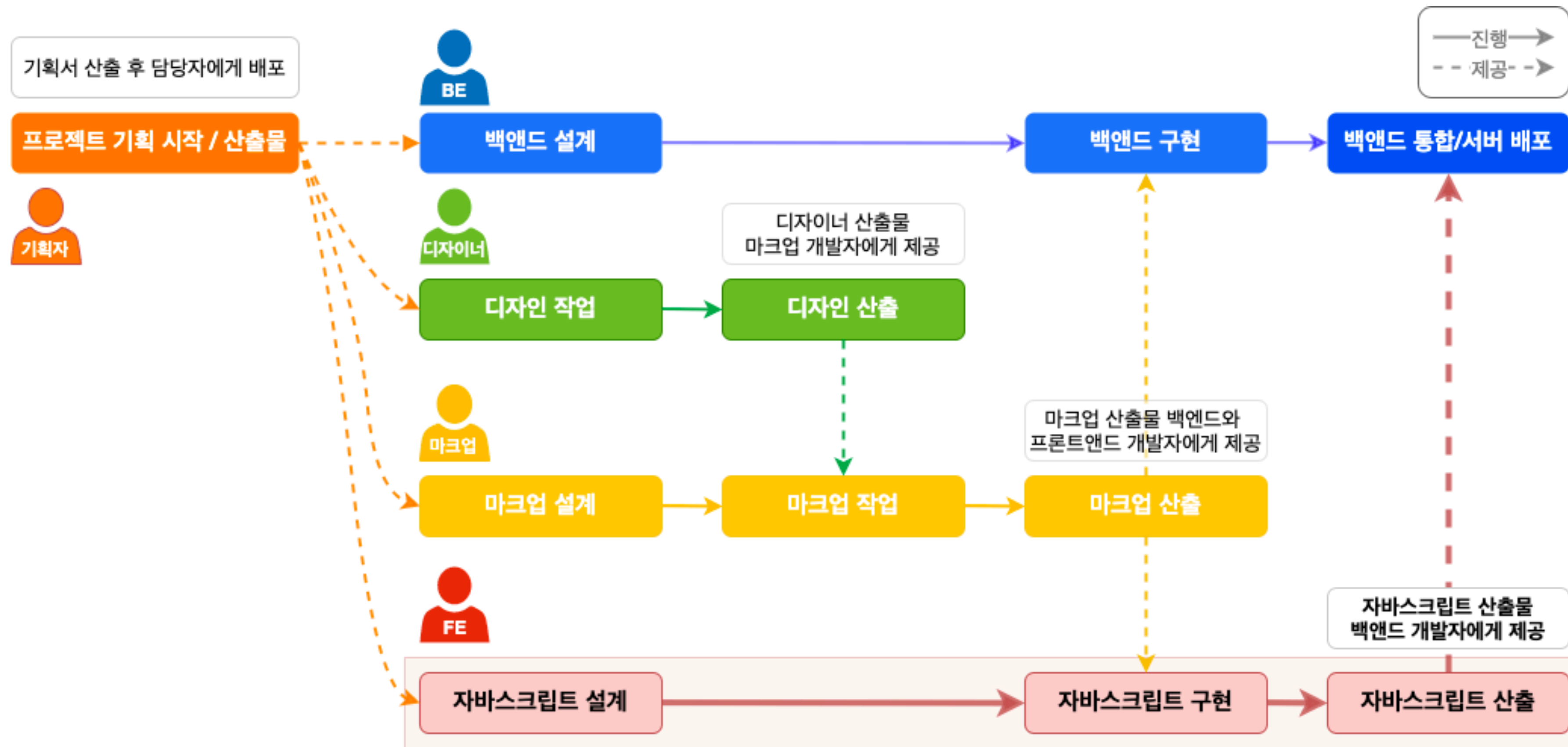
1. 도입 배경

1. 도입 배경

- 적은 인력으로 서비스를 빠르고 안정적으로 운영할 수 있을까?
- 개발/배포 과정 없이 데이터만 준비되면 서비스에 반영할 수 있을까?
- 데이터만 바뀌가며 원하는 화면과 기능을 다양하게 테스트할 수 있을까?

2. 기존 FE 업무 프로세스

2.1 기존 FE 업무 프로세스



출처 <https://d2.naver.com/helloworld/991303>

2.2 기존 FE 업무 한계

커뮤니케이션 증가



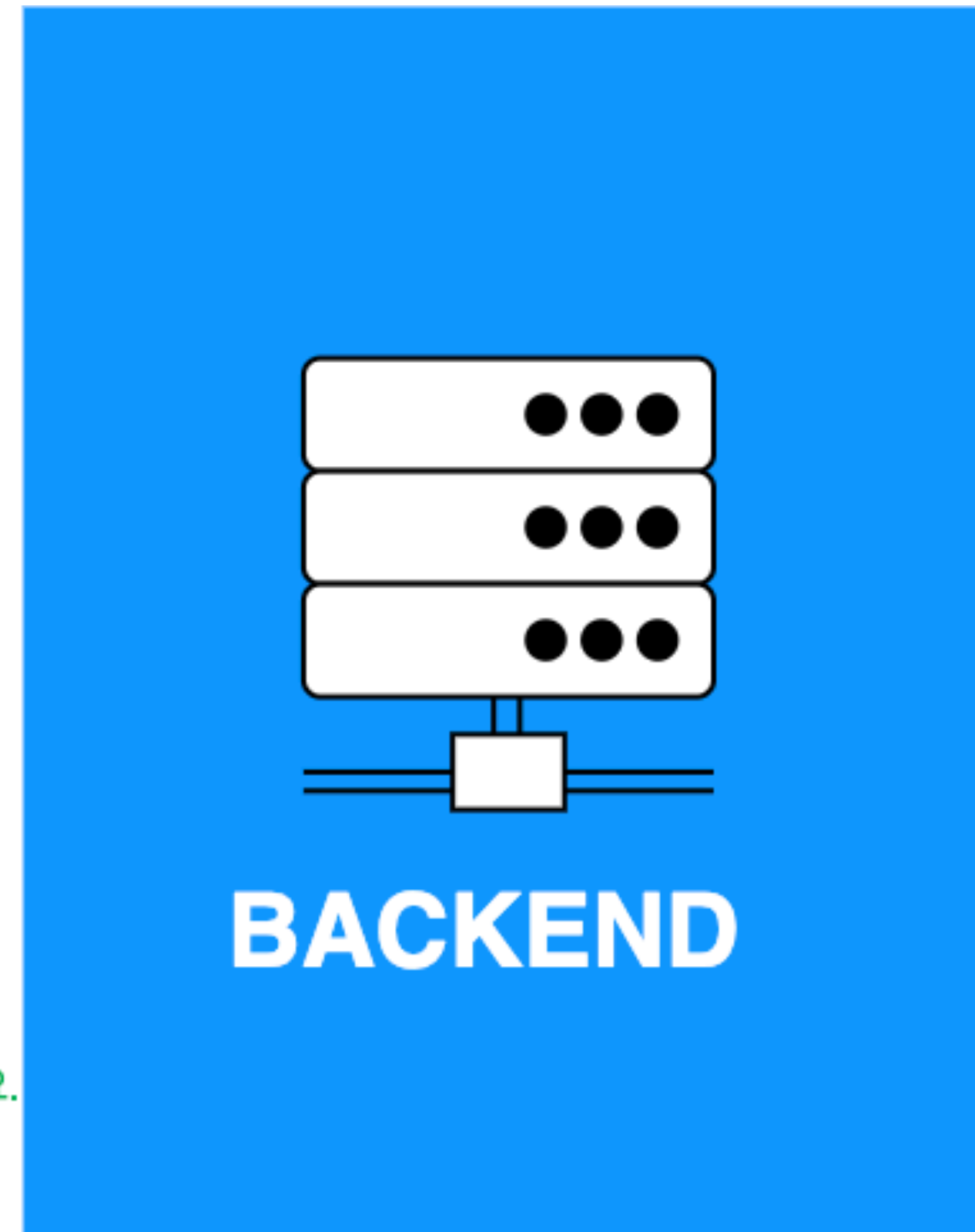
서버에서 초기 HTML 구조는 이렇게 드릴게요.
 클릭 로그는 BE 에서 onclick으로 처리 할게요.
 접근성 대응을 위해서 HTML 추가속성이 필요해요.



API 는 어떻게 할까요?



스크립트 초기화를 위해 태그의 id 를 넣어 주세요.
 플리킹 기능을 위해서는 HTML 추가 태그를 넣어 주세요.
 이미지 레이지 로딩을 위해 data-lazy 속성을 추가해 주세요.



2.2 기존 FE 업무 한계

재사용 범위의 한계



@kgs/ui-components

Accordion@1.0

Sticky@1.0

Tab@1.0

@egjs/components

Flicking

InfiniteGrid

Visible

@kgs/utils

jsonp-promise

lazy-loader

classList

raf

2.2 기존 FE 업무 한계

FE 역할의 한계

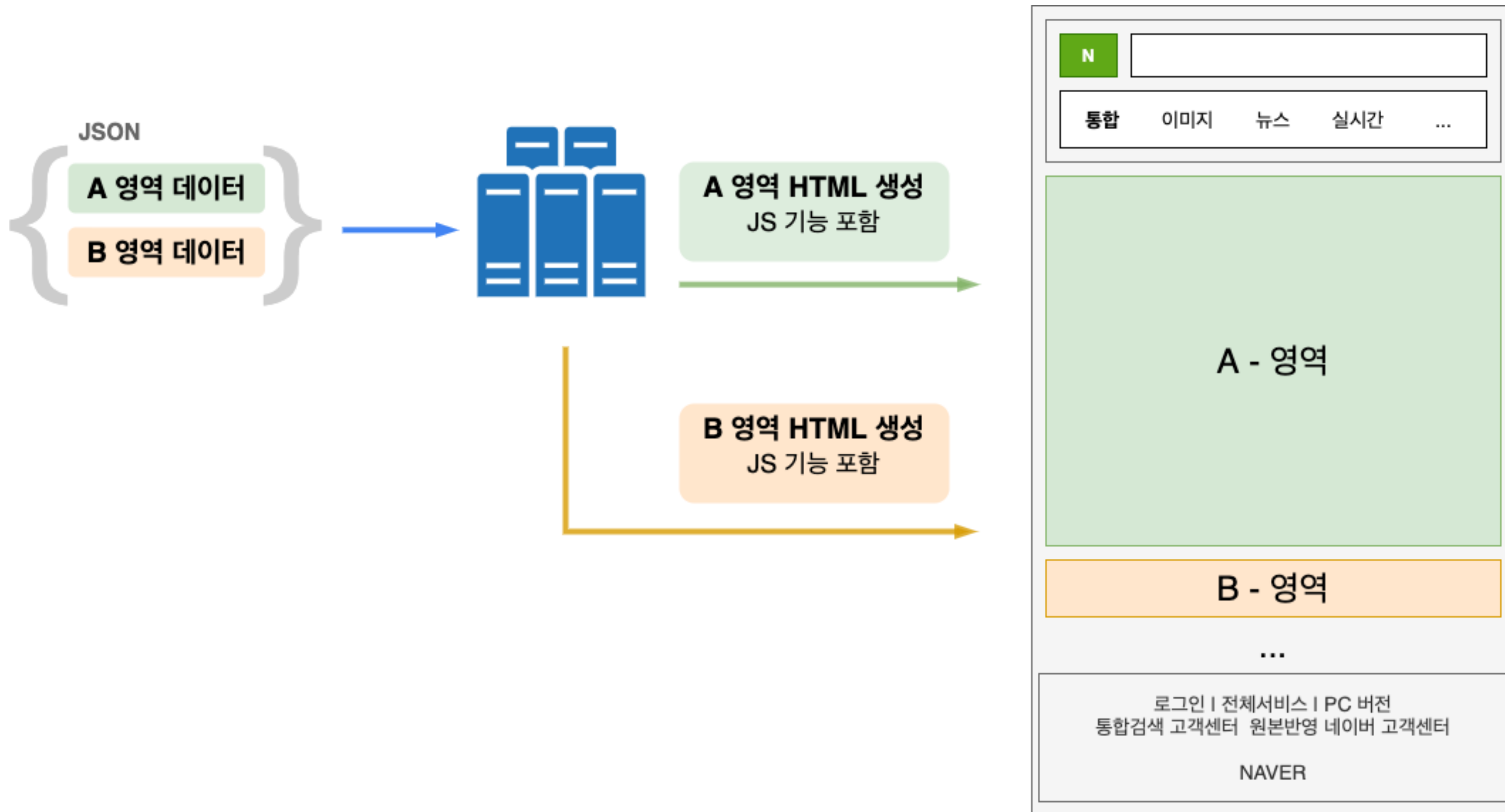
- 브라우저 환경에서만 동작하는 스크립트 개발/테스트만 가능

최적화의 한계

- 서버와 클라이언트 개발자 분리로 코드의 중복 발생
- No Framework로 FE 개발자 능력에 따라 품질 차이 발생

3. 시스템 기반의 업무 프로세스

3.1 시스템 기반의 업무 프로세스



3.1 시스템 기반의 업무 프로세스

- 최소한의 개발/운용 인원으로 운영
- 커뮤니케이션 비용 감소 - API 데이터 구조만 협의
- 제공되는 화면/기능 그대로 사용 시 배포 없이 서비스 가능
- 프레임워크 컴포넌트 개발 방식- 재사용성/성능 최적화/빠른 렌더링

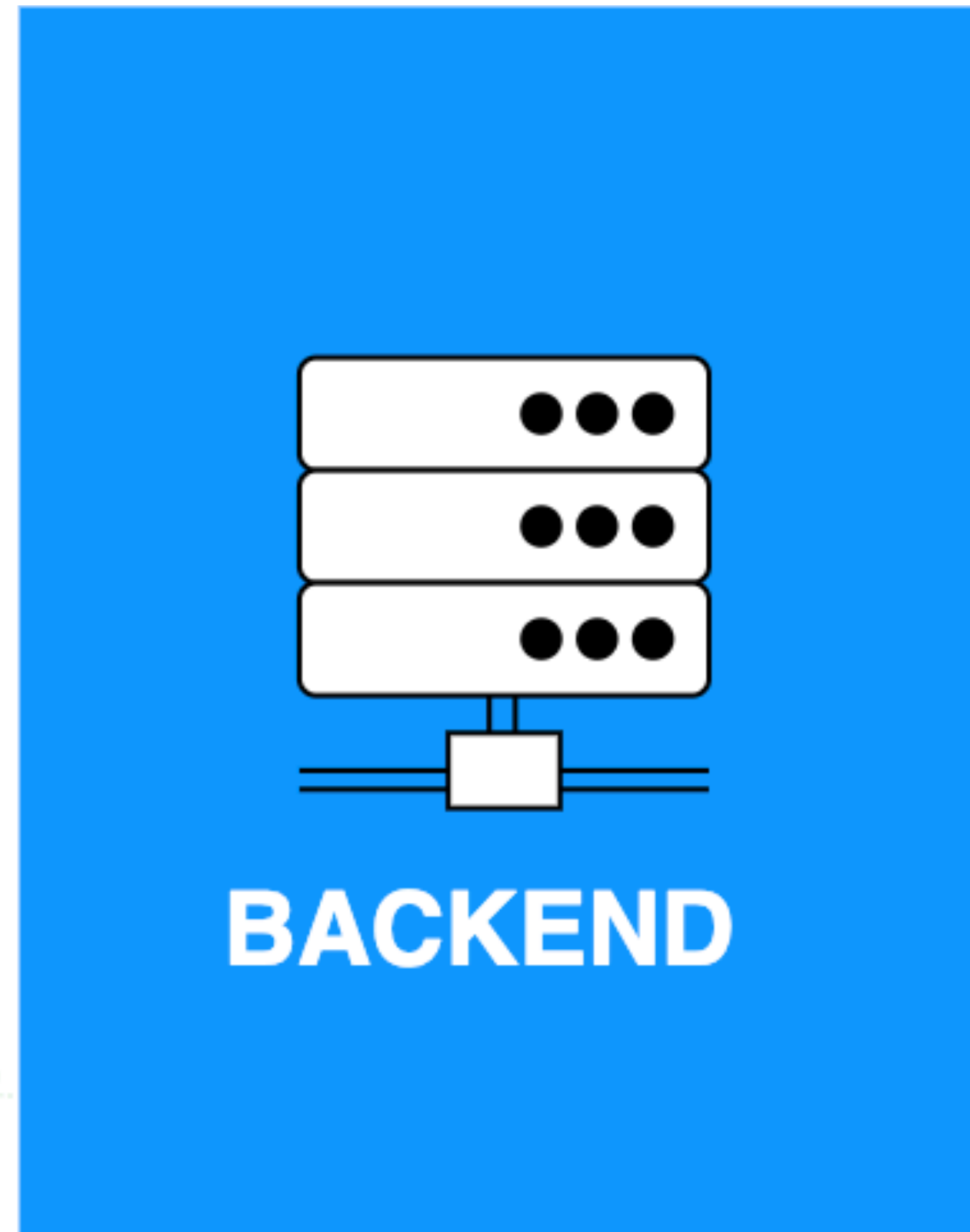
3.2 FE 업무 확장

커뮤니케이션 비용 감소



서버에서 초기 HTML 구조는 이렇게 드릴게요.
클릭 로그는 BE 에서 onclick으로 처리 할게요.
접근성 대응을 위해서 HTML 추가속성이 필요해요.

API 는 어떻게 할까요?



스크립트 초기화를 위해 태그의 id 를 넣어 주세요.
플리킹 기능을 위해서는 HTML 추가 태그를 넣어 주세요.
이미지 레이지 로딩을 위해 data-lazy 속성을 추가해 주세요.

3.2 FE 업무 확장

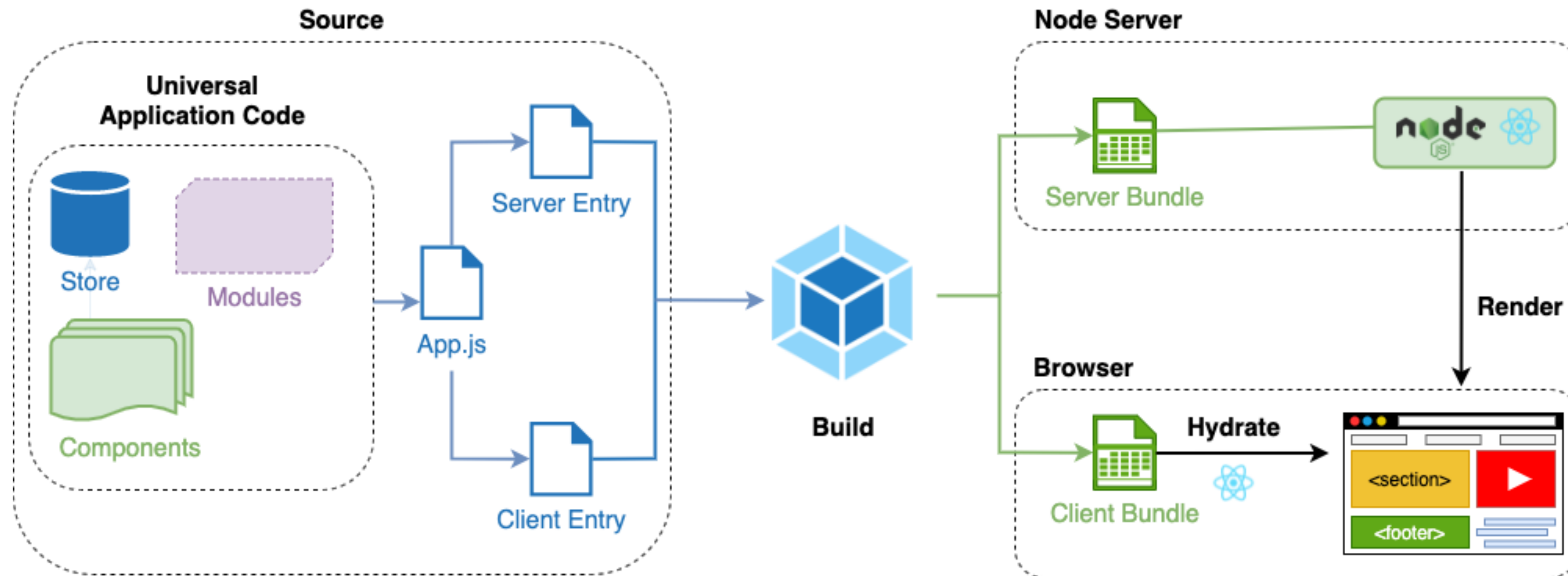
재사용 범위



3.2 FE 업무 확장

Universal Javascript

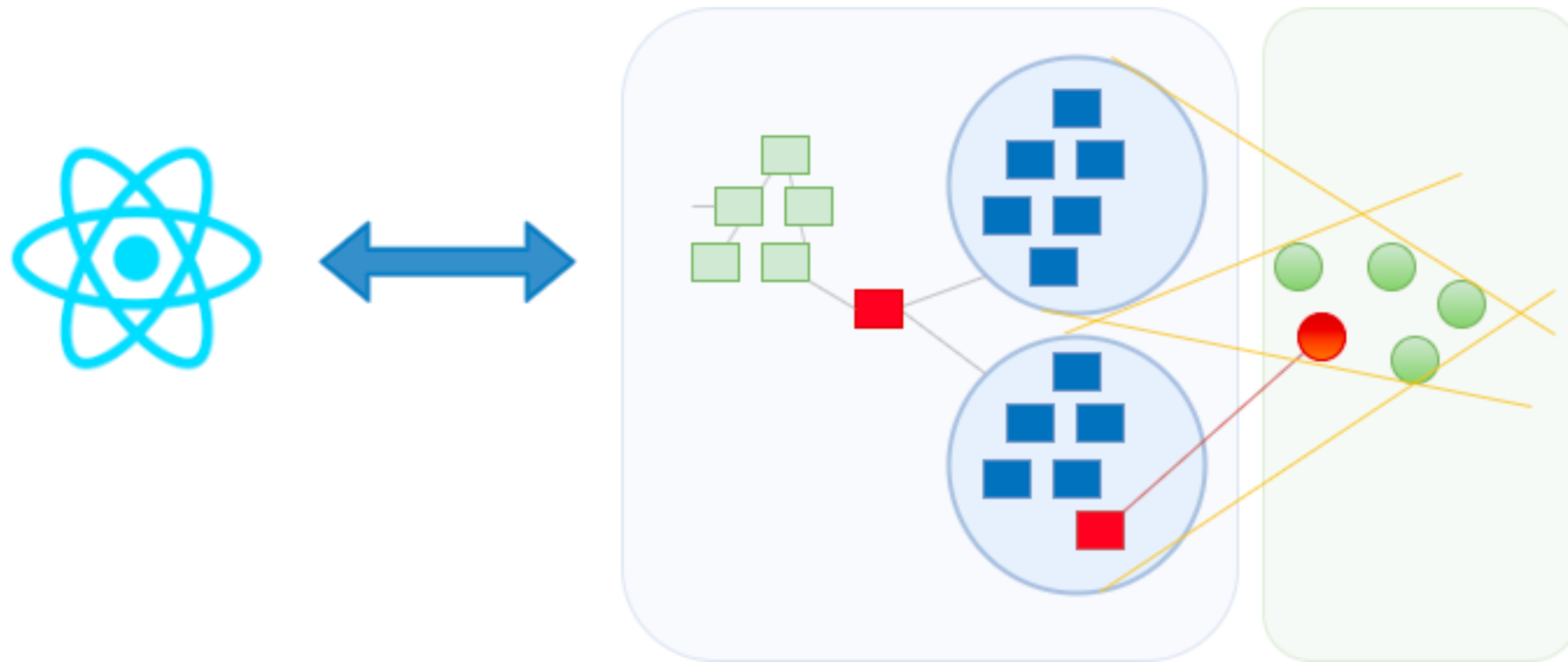
서버/브라우저 환경에서 동작하는 기능을 모두 개발



3.2 FE 업무 확장

Virtual DOM

성능 최적화와 빠른 렌더링 보장

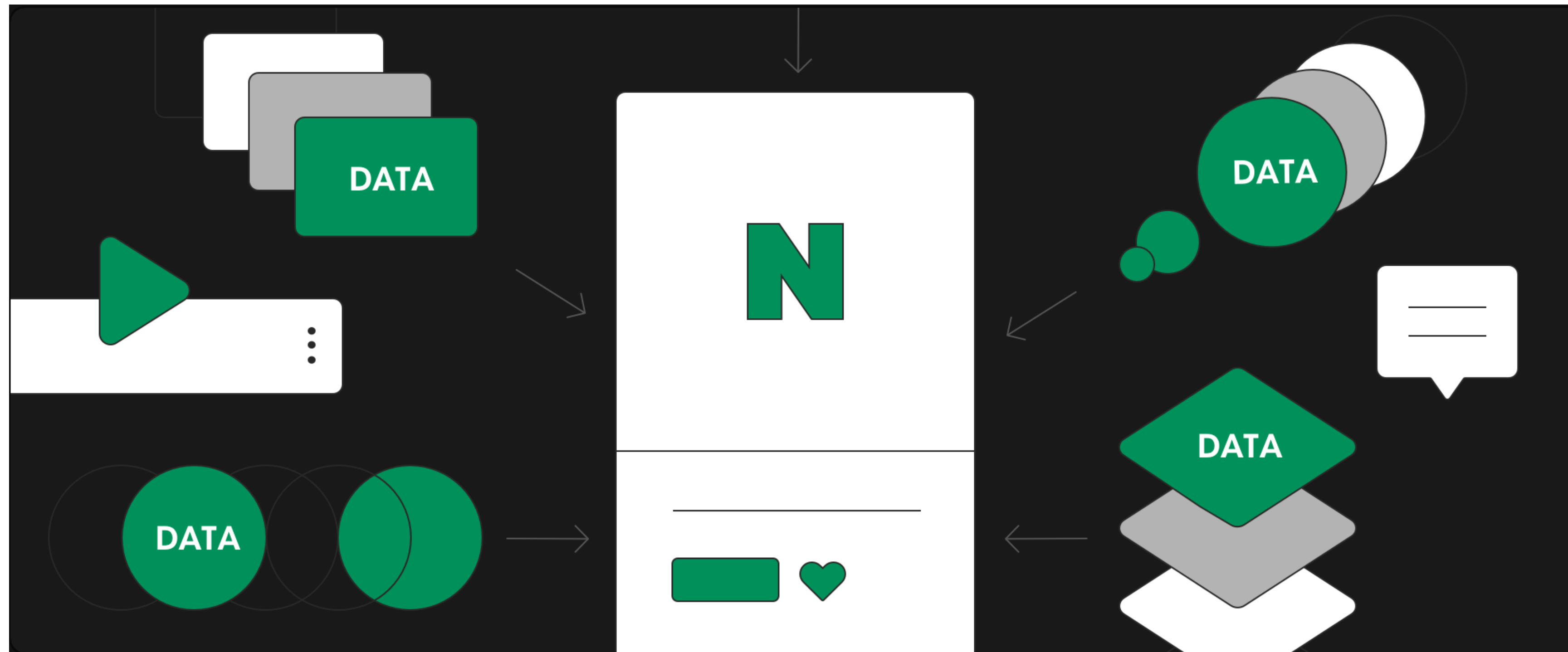


4. Server Driven UI Lapin

4.1 Lapin

네이버 지식베이스 검색의 빠른 개발/운영 프로세스 시스템

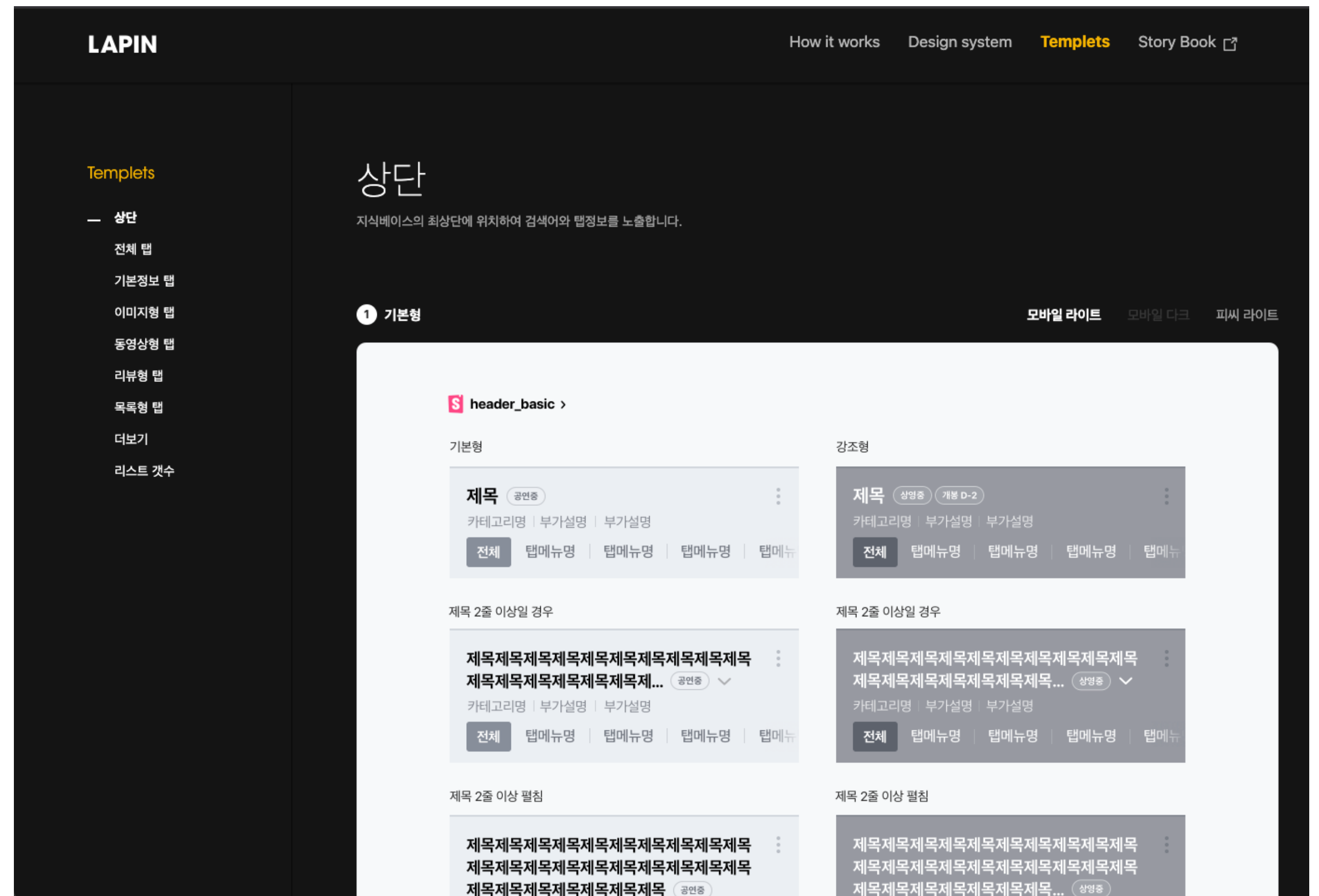
준비된 데이터를 기반으로 검색 결과 화면 생성하며 UI 기능 제공



4.2 Lapin 제공 도구

디자인 템플릿 페이지

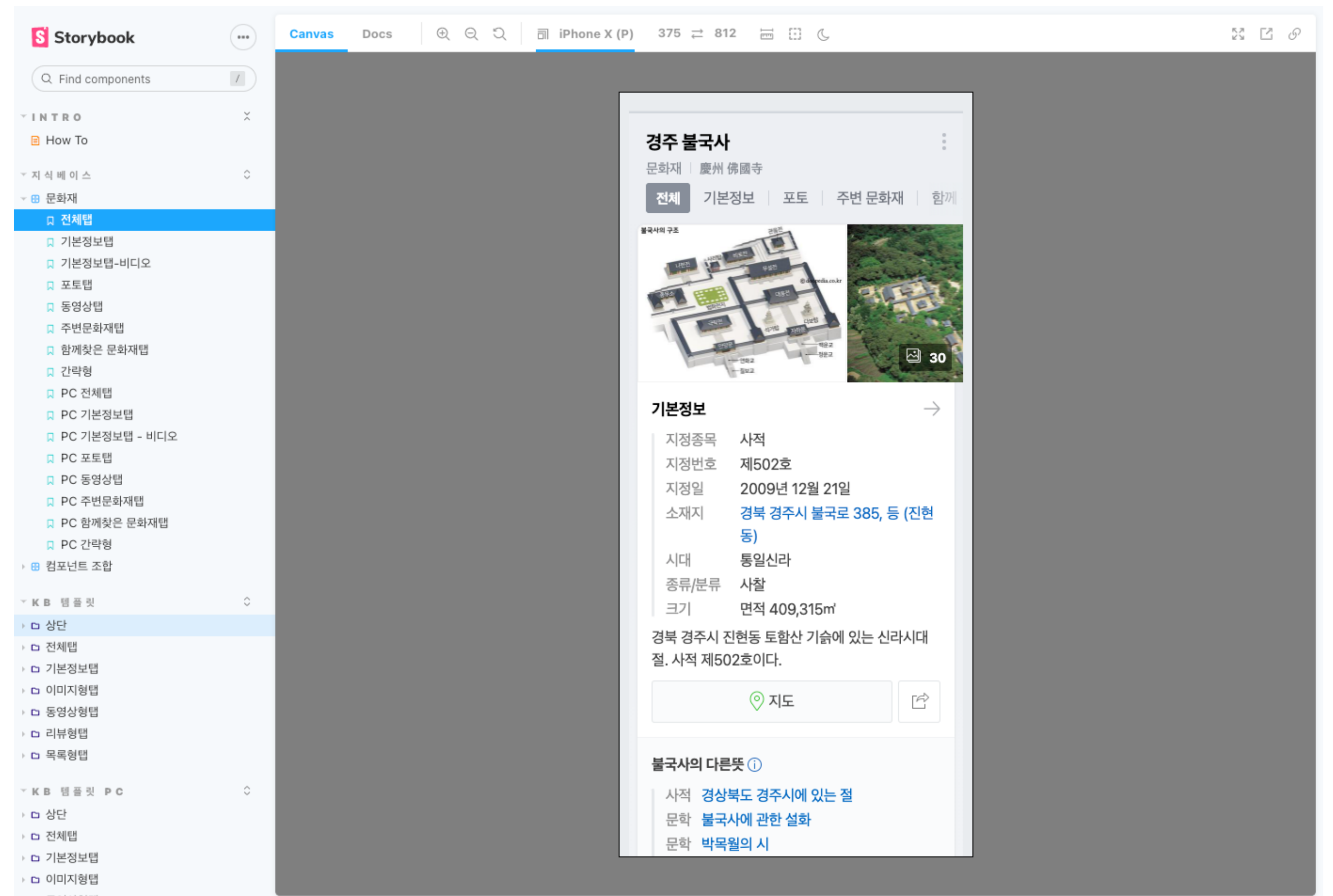
기획자가 원하는 화면 디자인을 선택



4.2 Lapin 제공 도구

스토리북 페이지

기획자가 원하는 화면과 기능을 테스트



4.2 Lapin 제공 도구

검색 결과 시뮬레이터 페이지

실제 화면과 기능을 시뮬레이션

- 모바일/PC 환경 별 화면 테스트
- UMD/AMD 페이지 로딩 방식
- JSON 데이터 직접 변경
- 실시간 검색 쿼리 변경
- HTML 응답 결과 확인

Lapin SSR 시뮬레이터

QueryString

CONTENT_ID: Send Reset

로딩 방식: 통검용-amd Open Popup

렌더링 표시: 통검 화면 모바일

컨검 API: 경주불국사 전체

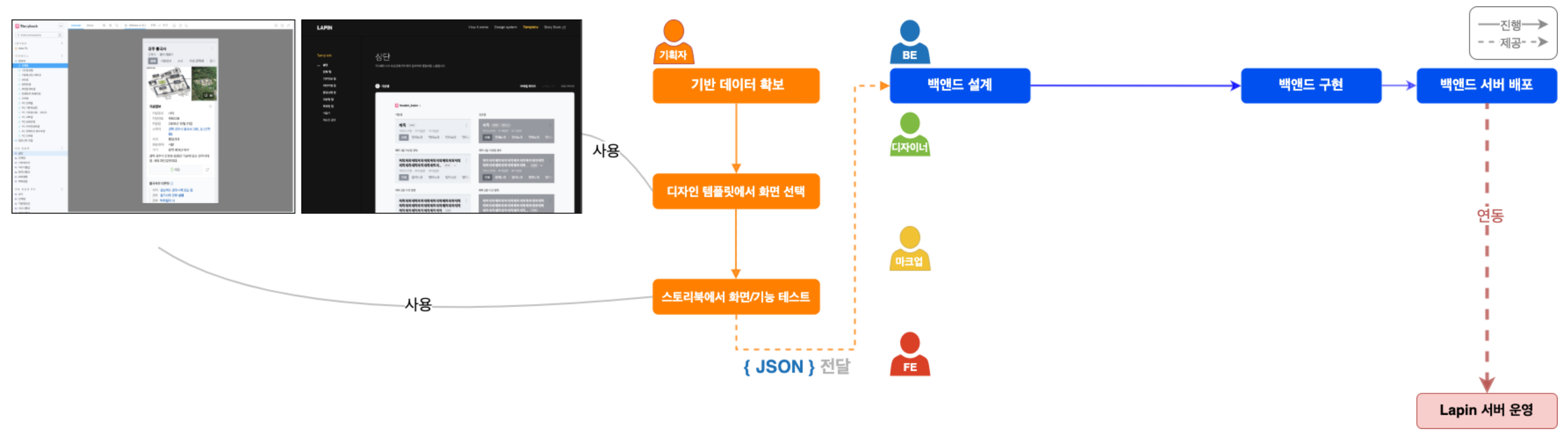
```

{
  "query": "경주불국사",
  "theme": {
    "color": "color_23",
    "case": "case_normal",
    "type": "mcs_common_module"
  },
  "clickLog": {
    "logArea": "nco_xgv*H",
    "gdid": "88214rz3_000000000000"
  },
  "etcInfo": {
    "device": "m"
  },
  "header": {
    "type": "header_basic",
    "data": {
      "title": {
        "text": "경주 불국사",
        "link": "https://m.terms.naver.com/entry.nhn?docid=1104868",
        "extraA": "title"
      }
    }
  },
  "subTitleList": [
    {
      "text": "문화재",
      "link": "",
      "extraA": ""
    }
  ]
}

```

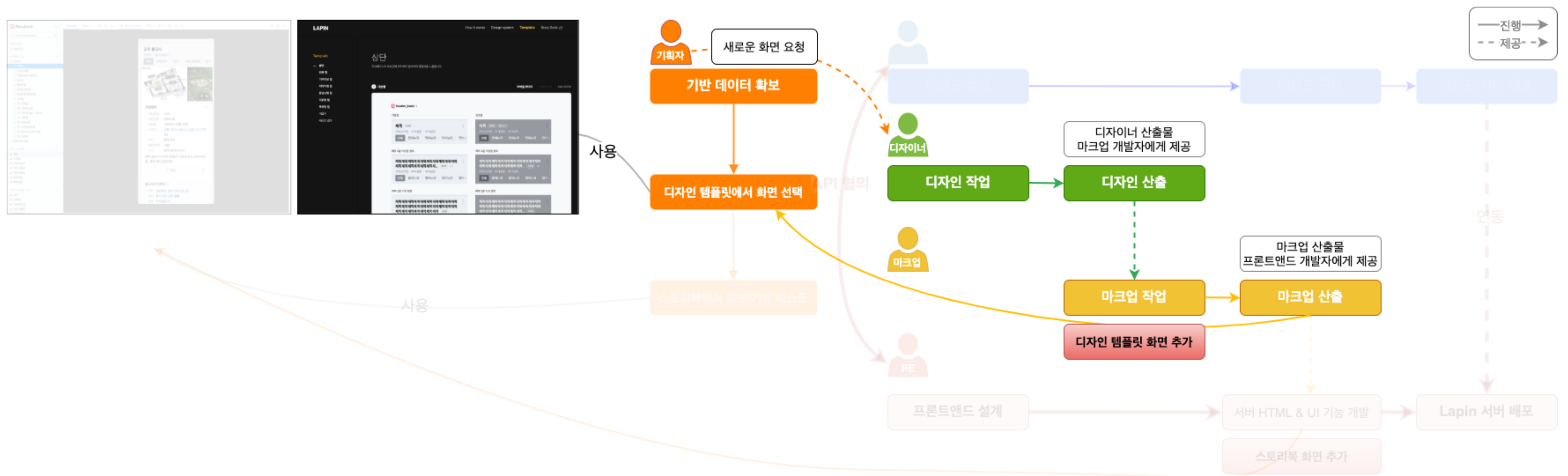
4.2 Lapin 업무 프로세스

case A. 디자인 템플릿에 있는 화면과 기능을 재사용한다면



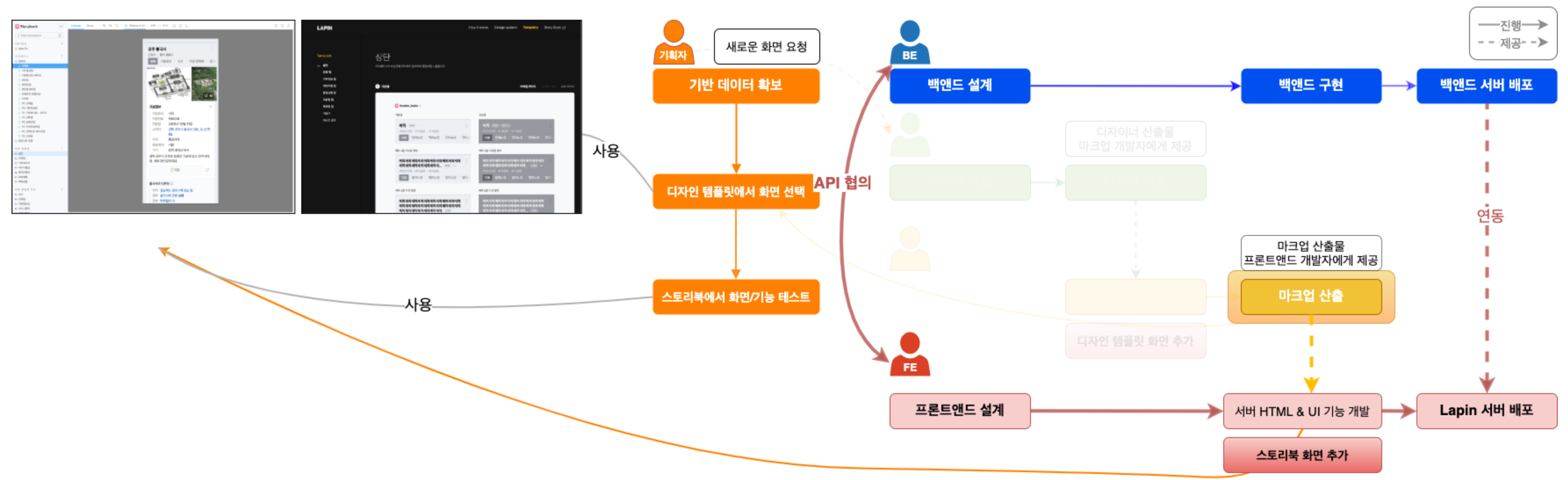
4.2 Lapin 업무 프로세스

case B. 디자인 템플릿에 새로운 화면을 추가한다면 - 디자인



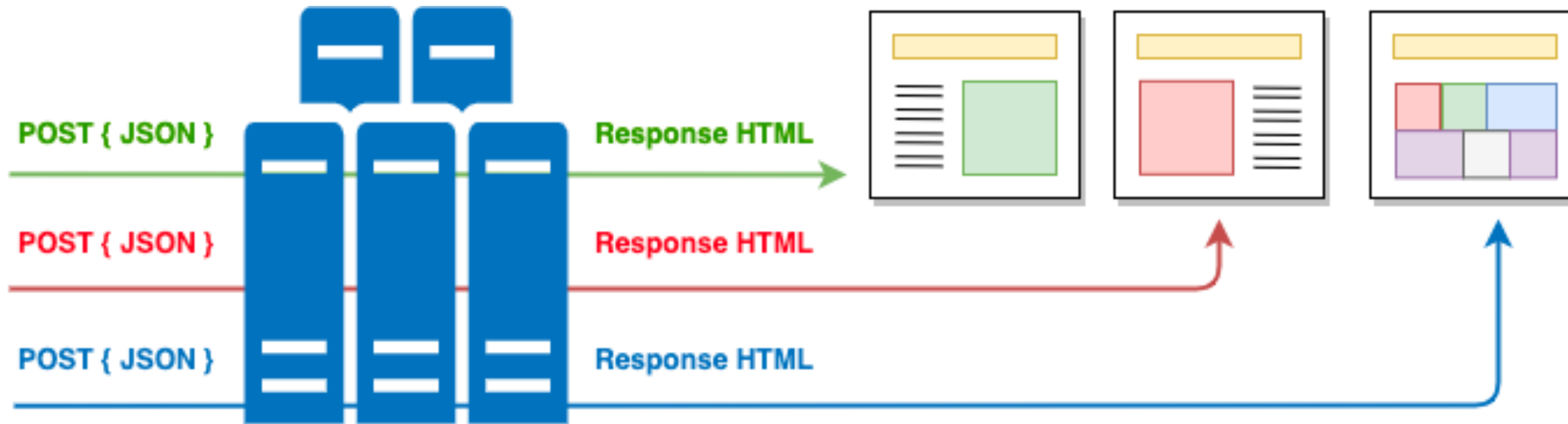
4.2 Lapin 업무 프로세스

case B. 디자인 템플릿에 새로운 화면을 추가한다면 - 개발



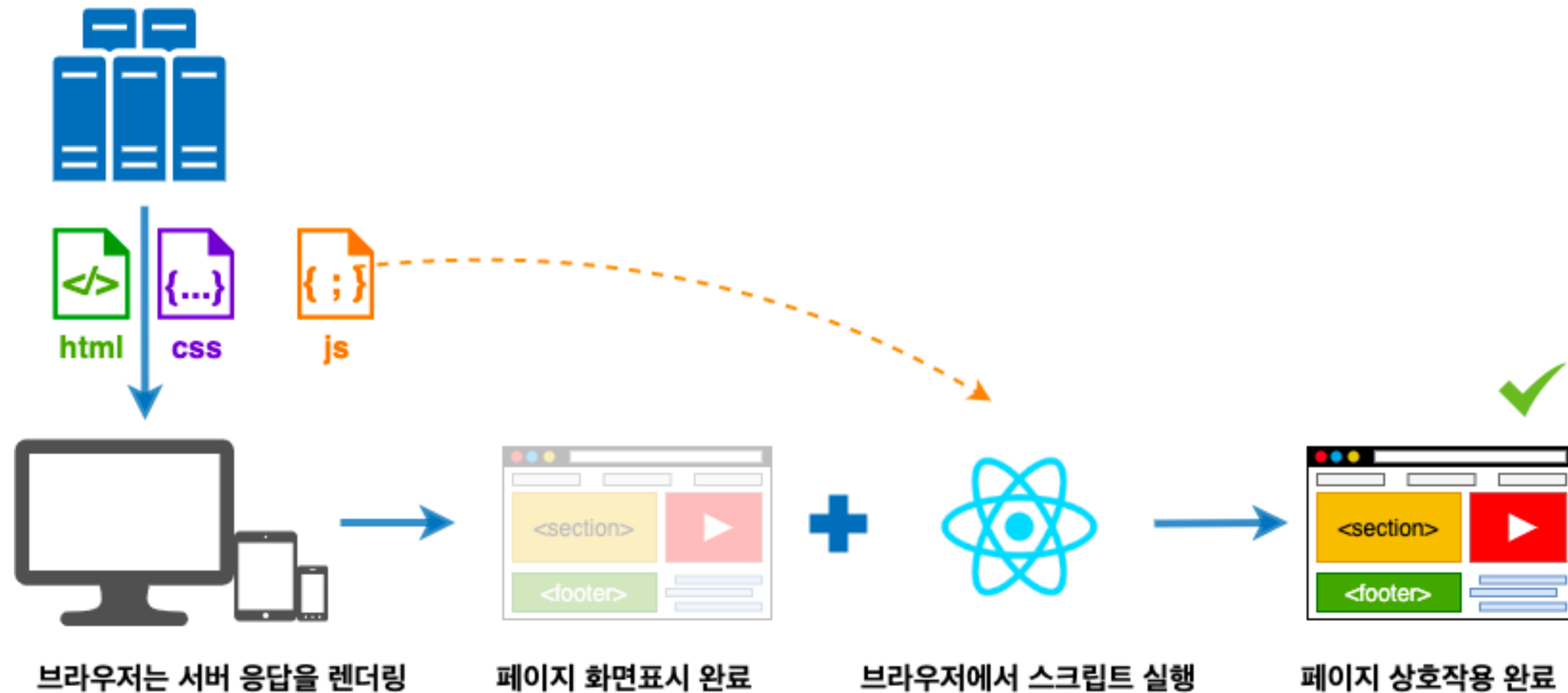
4.3 Lapin 적용된 기술들

SDUI (Server Driven UI)



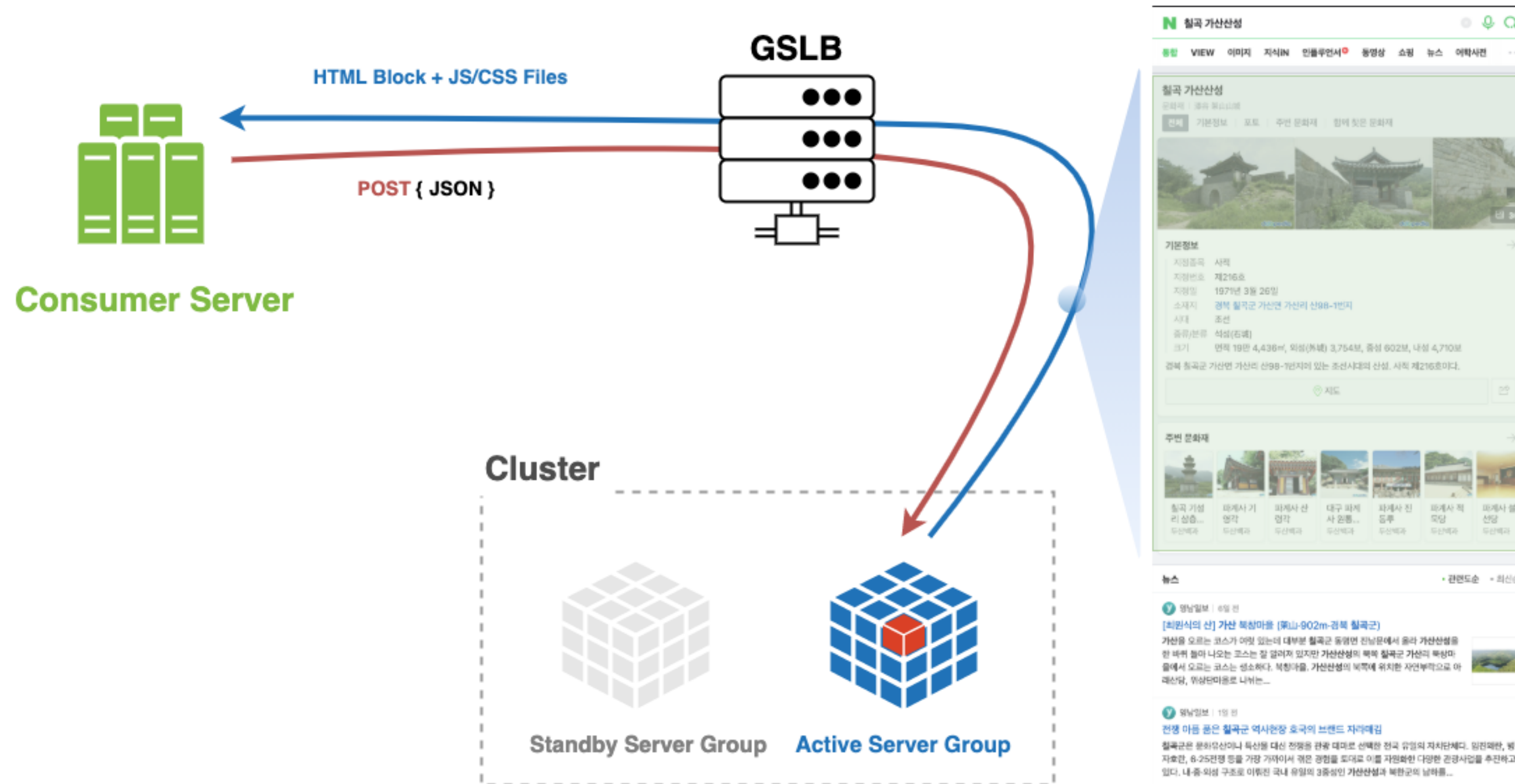
4.3 Lapin 적용된 기술들

SSR (ServerSide Rendering)



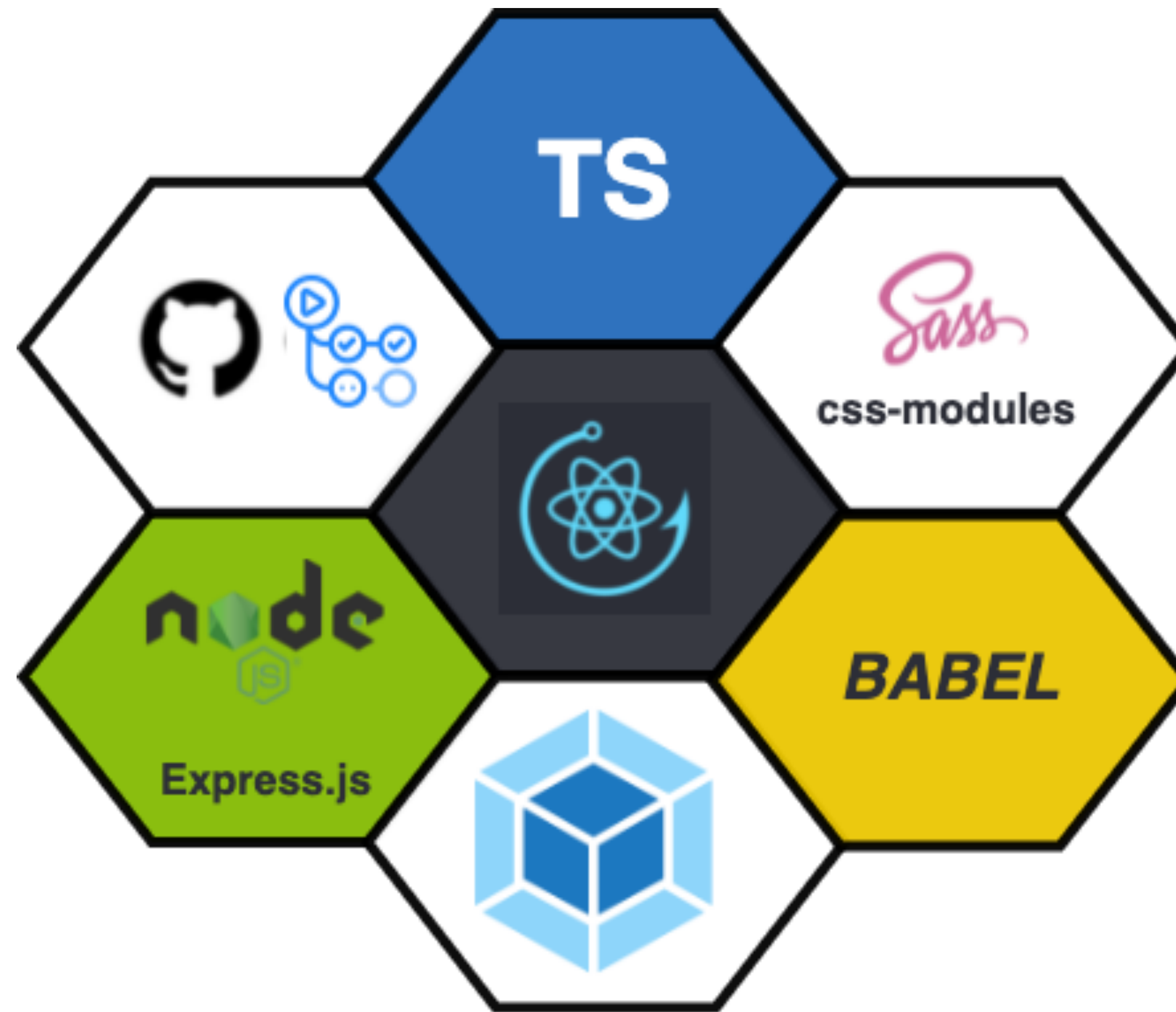
4.3 Lapin 적용된 기술들

MSA (MicroService Architecture)



5. 서버/클라이언트 개발

5.1 기술 스택



5.2 프로젝트 구조

App 컴포넌트 코드

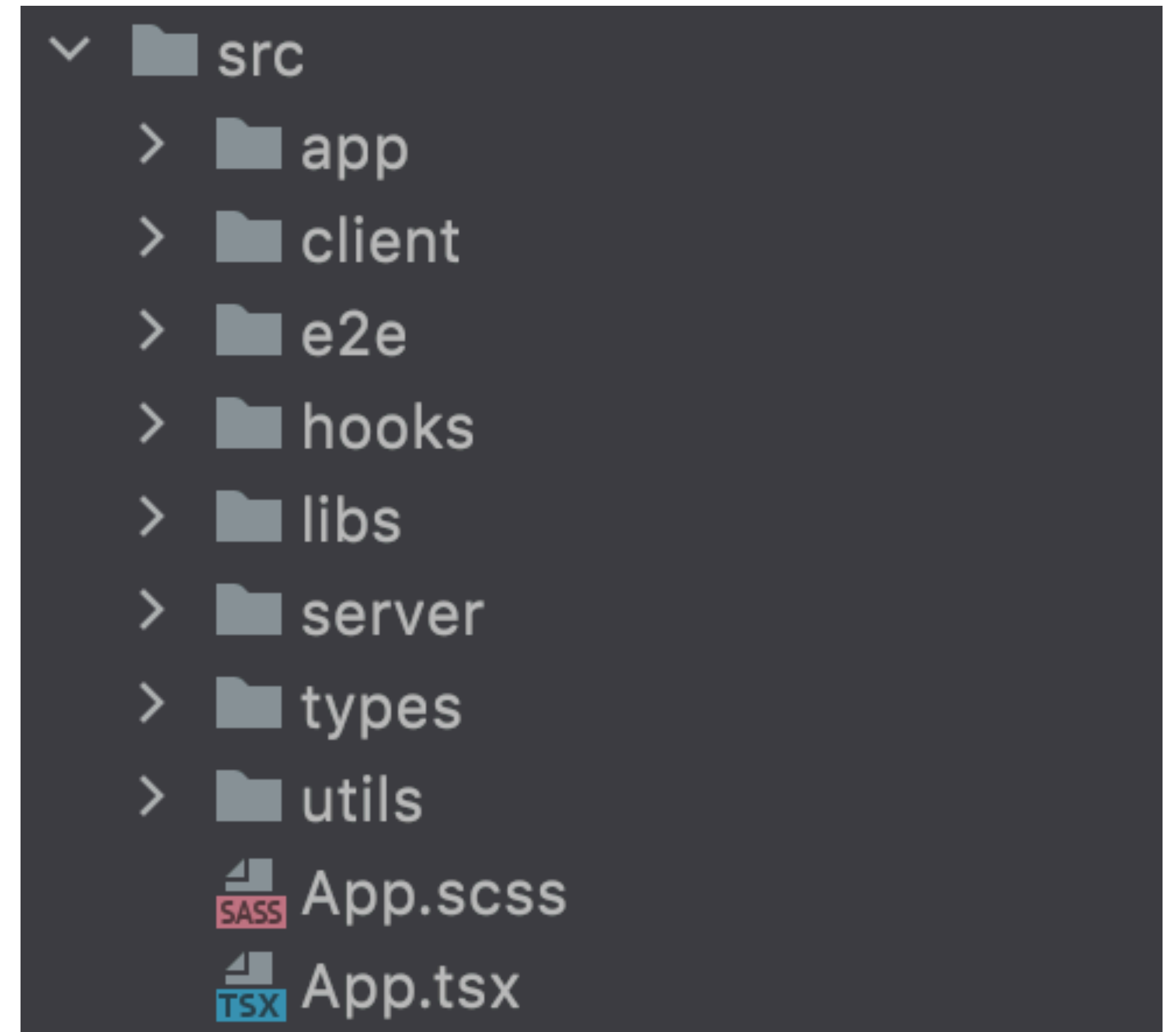
app

플랫폼 코드

client / server

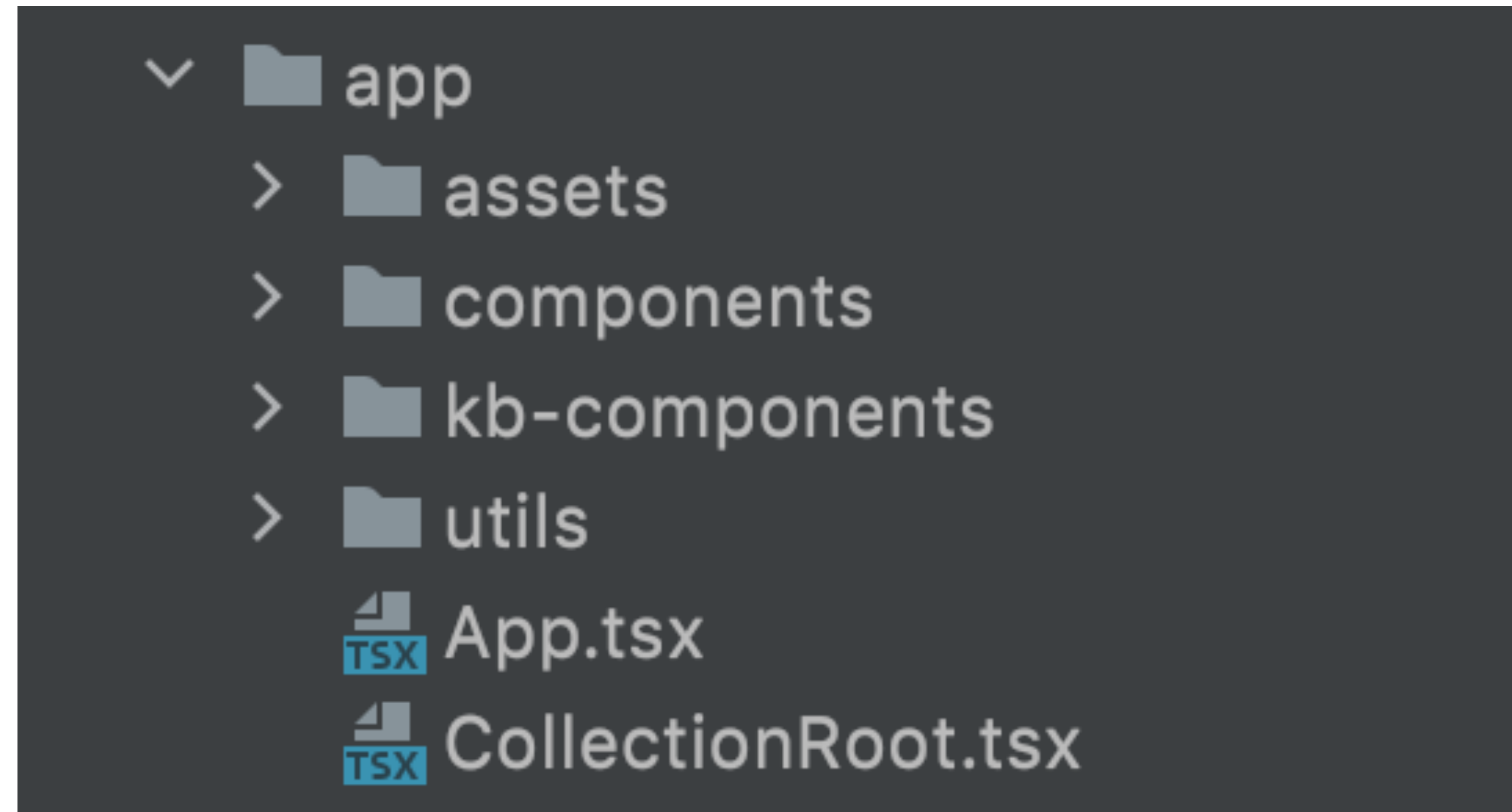
공통 코드

hooks / types / utils / libs

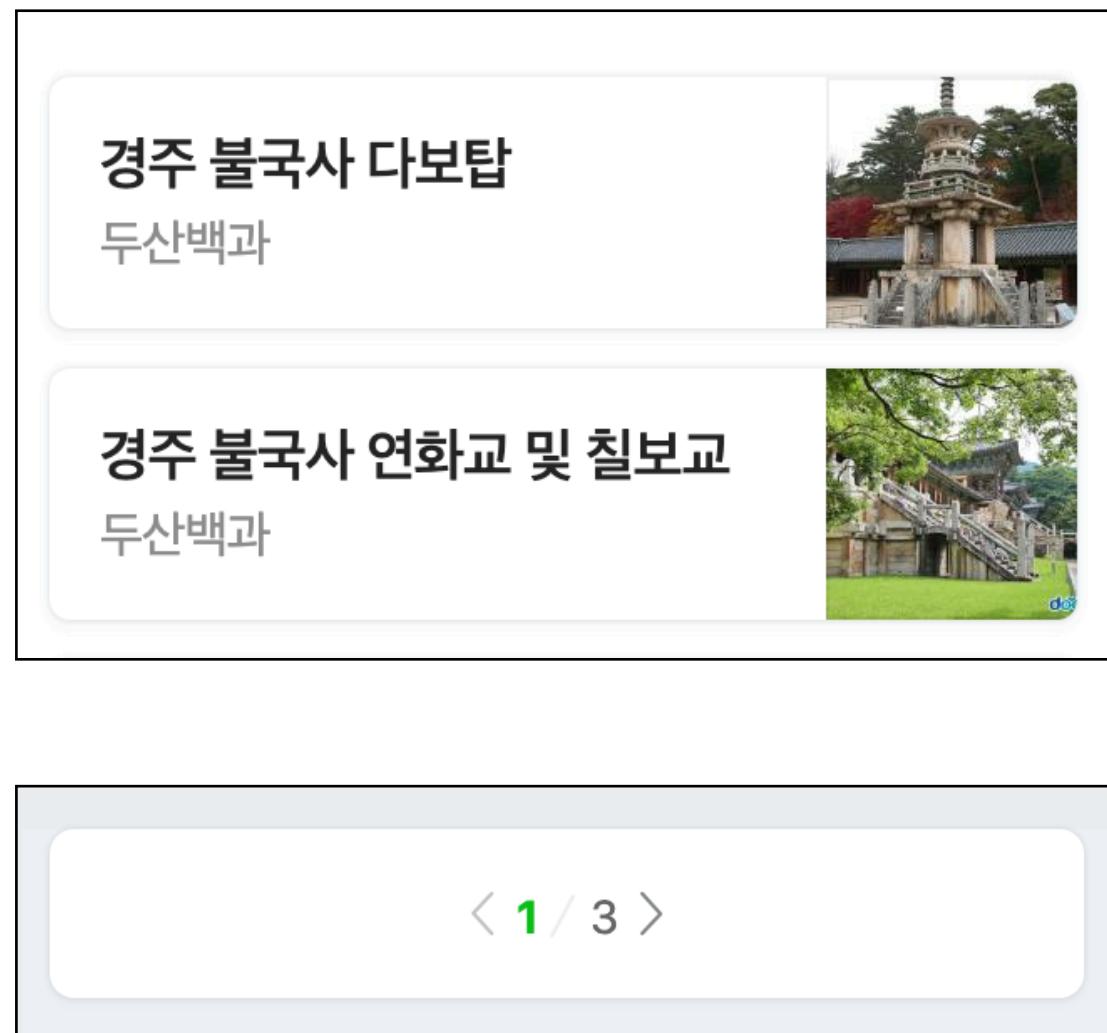


5.2 프로젝트 구조

app/
 components
 kb-components
 utils



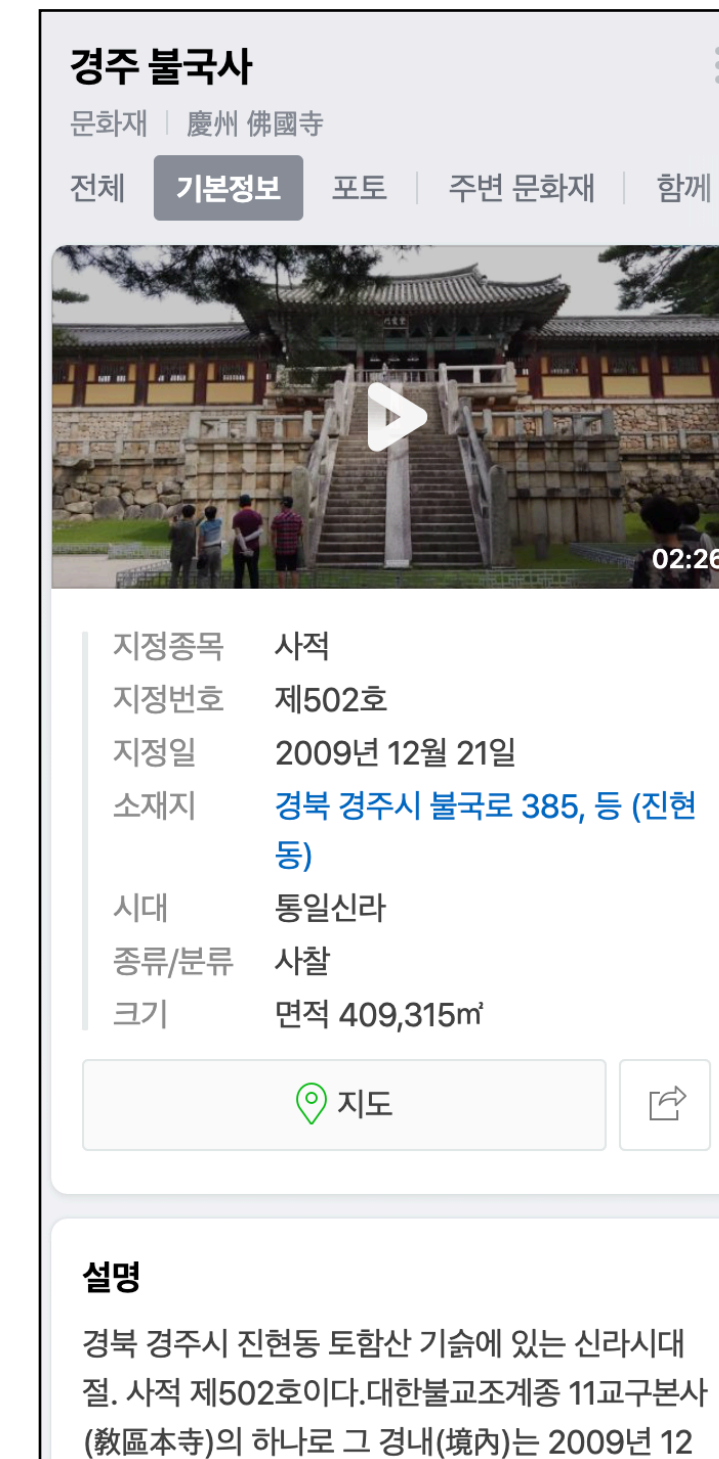
5.2 프로젝트 구조



components



kb-components



Lapin

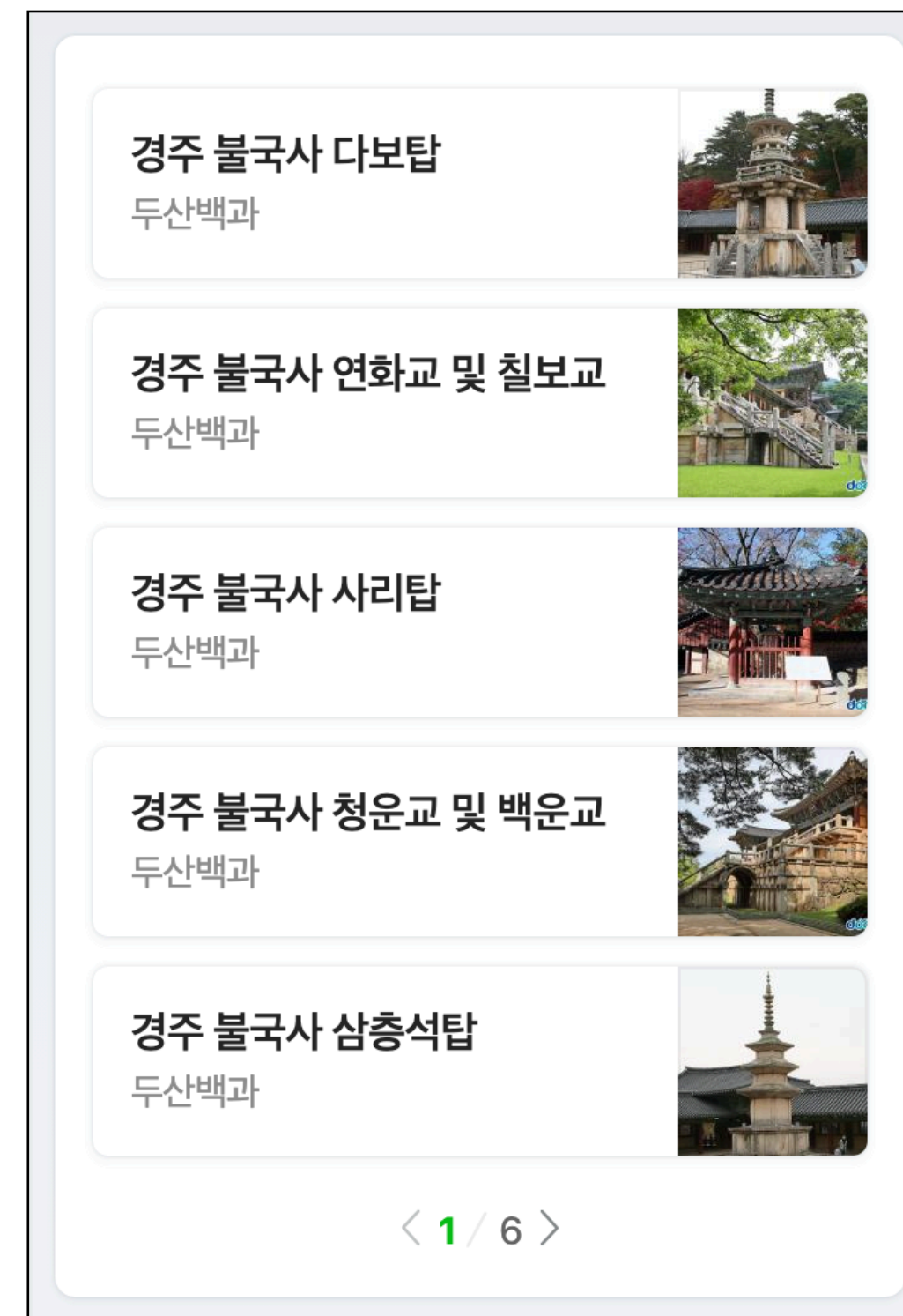
5.2 프로젝트 구조

kb-components

- 카드블럭 형태 UI
- component (pc / mobile)
- test / mock / story

```

    v ListColumn1EmphasisMoreNumberPage
      > __snapshots__
        index.tsx
        ListColumn1EmphasisExtraInfo.stories.tsx
        ListColumn1EmphasisMoreNumberPage.interfaces.ts
        ListColumn1EmphasisMoreNumberPage.mock.ts
        ListColumn1EmphasisMoreNumberPage.stories.tsx
        ListColumn1EmphasisMoreNumberPage.test.tsx
        ListColumn1EmphasisMoreNumberPage.tsx
        PcListColumn1EmphasisExtraInfo.stories.tsx
        PcListColumn1EmphasisMoreNumberPage.stories.tsx
  
```



5.3 데이터 구조

app (공통 데이터)

query, theme, ...

components (카드 UI 데이터)

type, data, ...

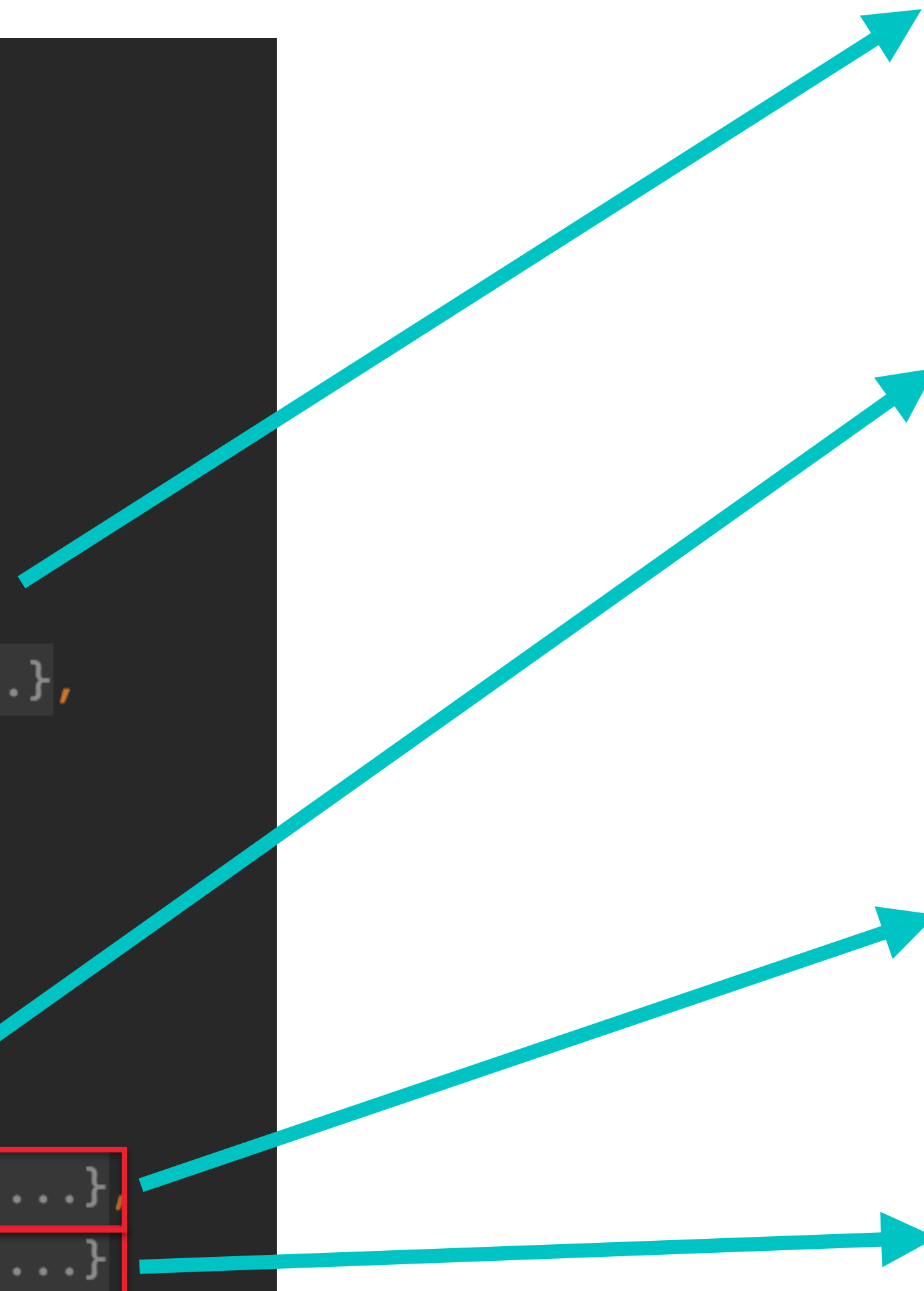
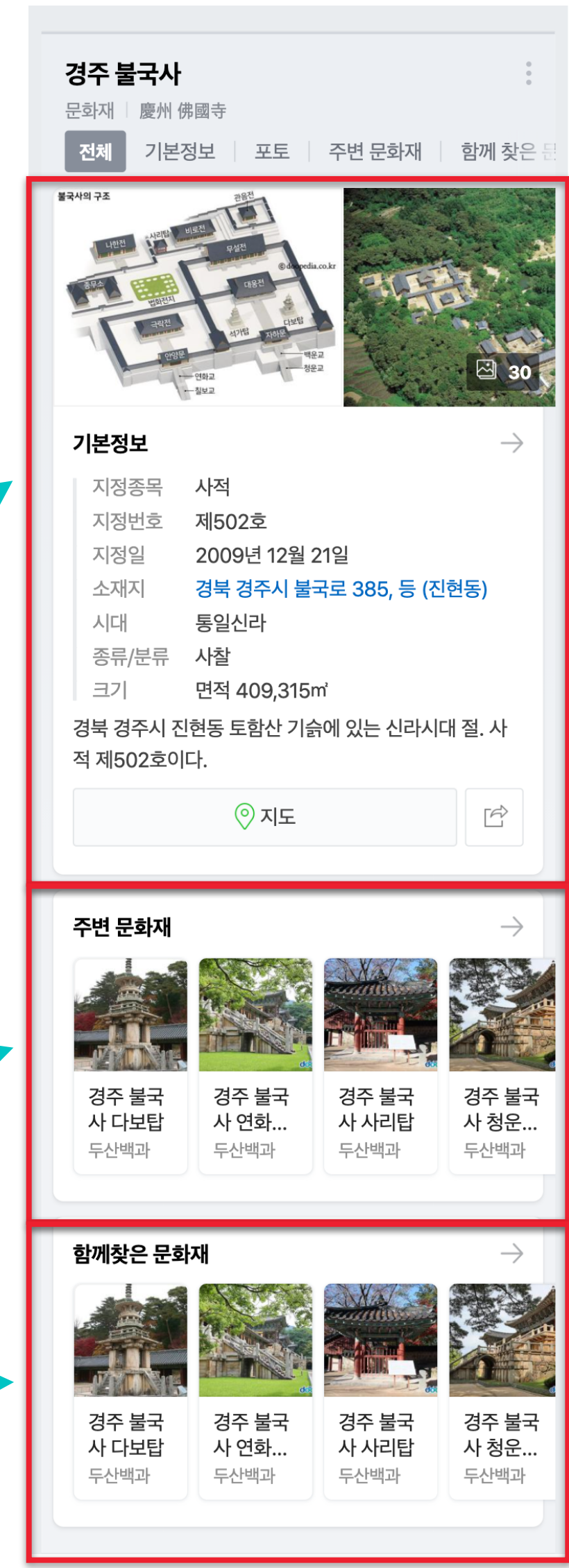
```

{
  "query": "불국사",
  "theme": {"color": "color_23"...},
  "clickLog": {...},
  "etcInfo": {"device": "m"...},
  "header": {
    "type": "header_basic",
    "data": {
      "title": {"text": "경주 불국사"...},
      "subTitleList": [...],
      "tabList": [...]
    }
  },
  "components": [
    {"type": "basic_info"...},
    {"type": "carousel_square_basic"...},
    {"type": "carousel_square_basic"...}
  ]
}

```

5.3 데이터 구조

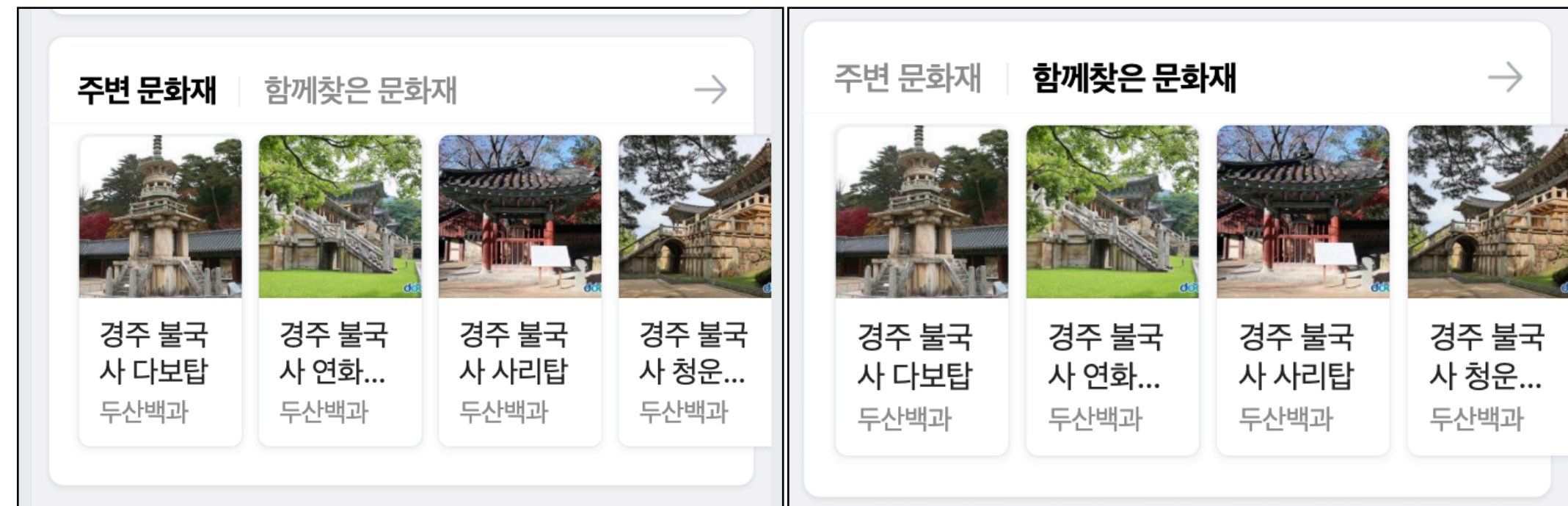
```
{  
  "query": "불국사",  
  "theme": {"color": "color_23"...},  
  "clickLog": {...},  
  "etcInfo": {"device": "m"...},  
  "header": {  
    "type": "header_basic",  
    "data": {  
      "title": {"text": "경주 불국사"...},  
      "subTitleList": [...],  
      "tabList": [...]  
    }  
  },  
  "components": [  
    {"type": "basic_info"...},  
    {"type": "carousel_square_basic"...},  
    {"type": "carousel_square_basic"...}  
  ]  
}
```



5.3 데이터 구조

데이터 구성 시 고려 사항

- 중첩 구조
- 컴포넌트 타입 네이밍



중첩구조 (탭 구조)

```

"components": [
  {"type": "basic_info"...},
  {"type": "carousel_square_basic"...},
  {"type": "carousel_square_basic"...}
]

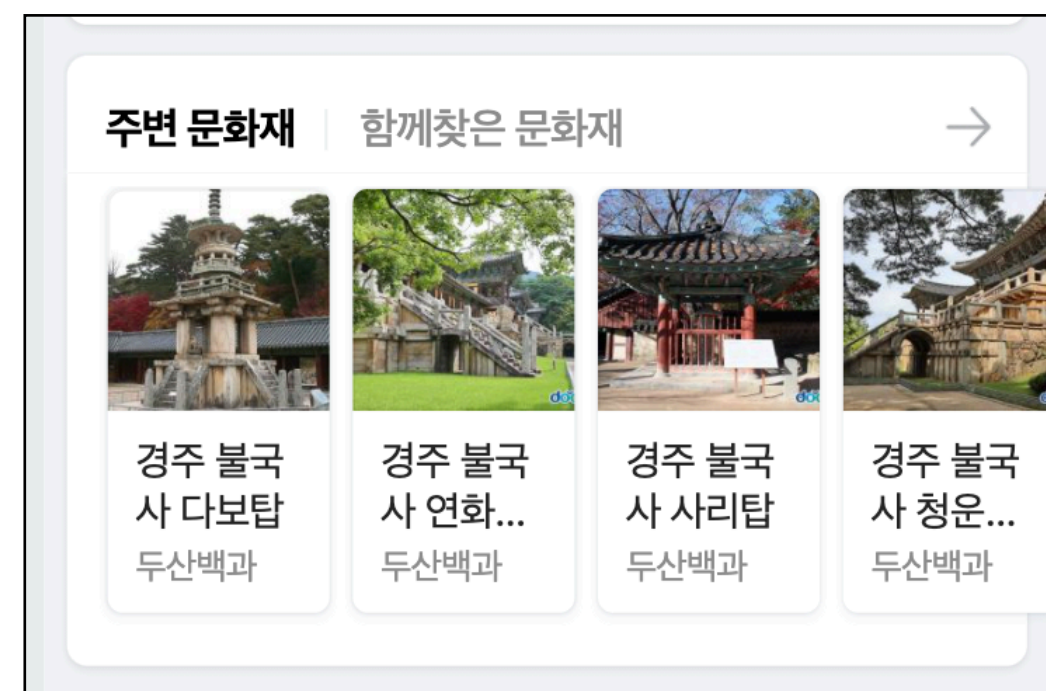
```

컴포넌트 타입 네이밍

5.3 데이터 구조

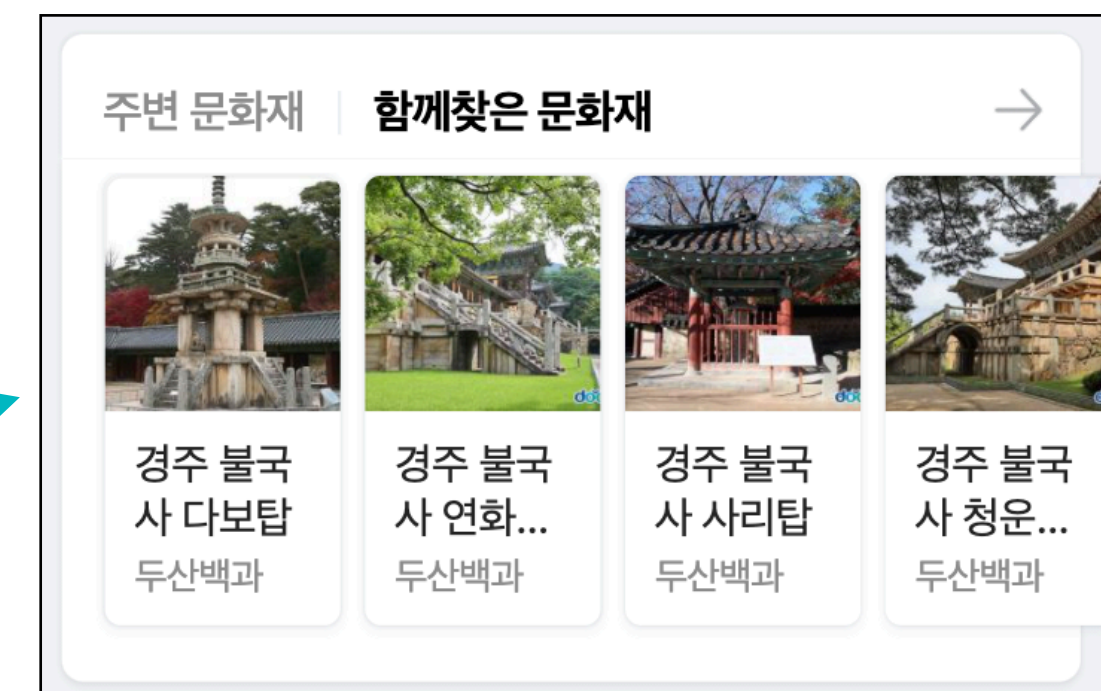
중첩 구조

- components가 조합된 components
- 내부에 components 필드 추가



```

"components": [
  {"type": "basic_info"...},
  {
    "type": "carousel_tab",
    "data": {
      "components": [
        {"type": "carousel_square_basic"...}
        {"type": "carousel_square_basic"...}
      ]
    }
  }
]
  
```



5.3 데이터 구조

중첩 구조

- 기존 데이터 재사용 + 추가 데이터
- union & generic

```

"components": [
  {"type": "basic_info"...},
  {
    "type": "carousel_tab",
    "data": {
      "components": [
        {"type": "carousel_square_basic"...},
        {"type": "carousel_square_basic"...}
      ]
    }
  }
]
  
```

```

export interface CarouselTabComponentData {
  type: "carousel_tab";
  data: {
    components: TypedTabContent[];
  };
}
  
```

```

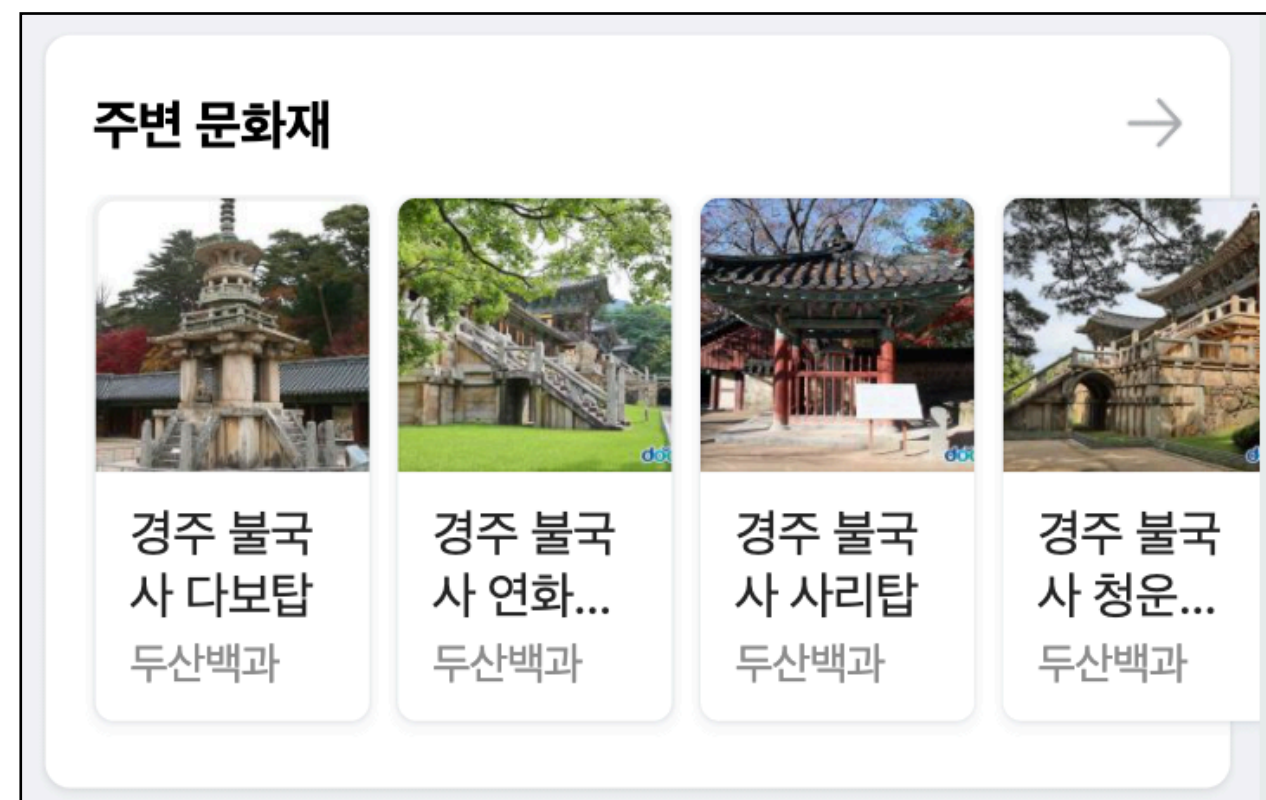
export type TabData<Data> = Data & {
  data: {
    /**
     * 탭 선택여부
     */
    isSelected: boolean;
  };
};

export type TypedTabContent = TabData<
  | CarouselHorizontalBasicComponentData
  | PcCarouselHorizontalBasicComponentData
  | CarouselSquareBasicComponentData
  | PcCarouselSquareBasicComponentData
>;
  
```

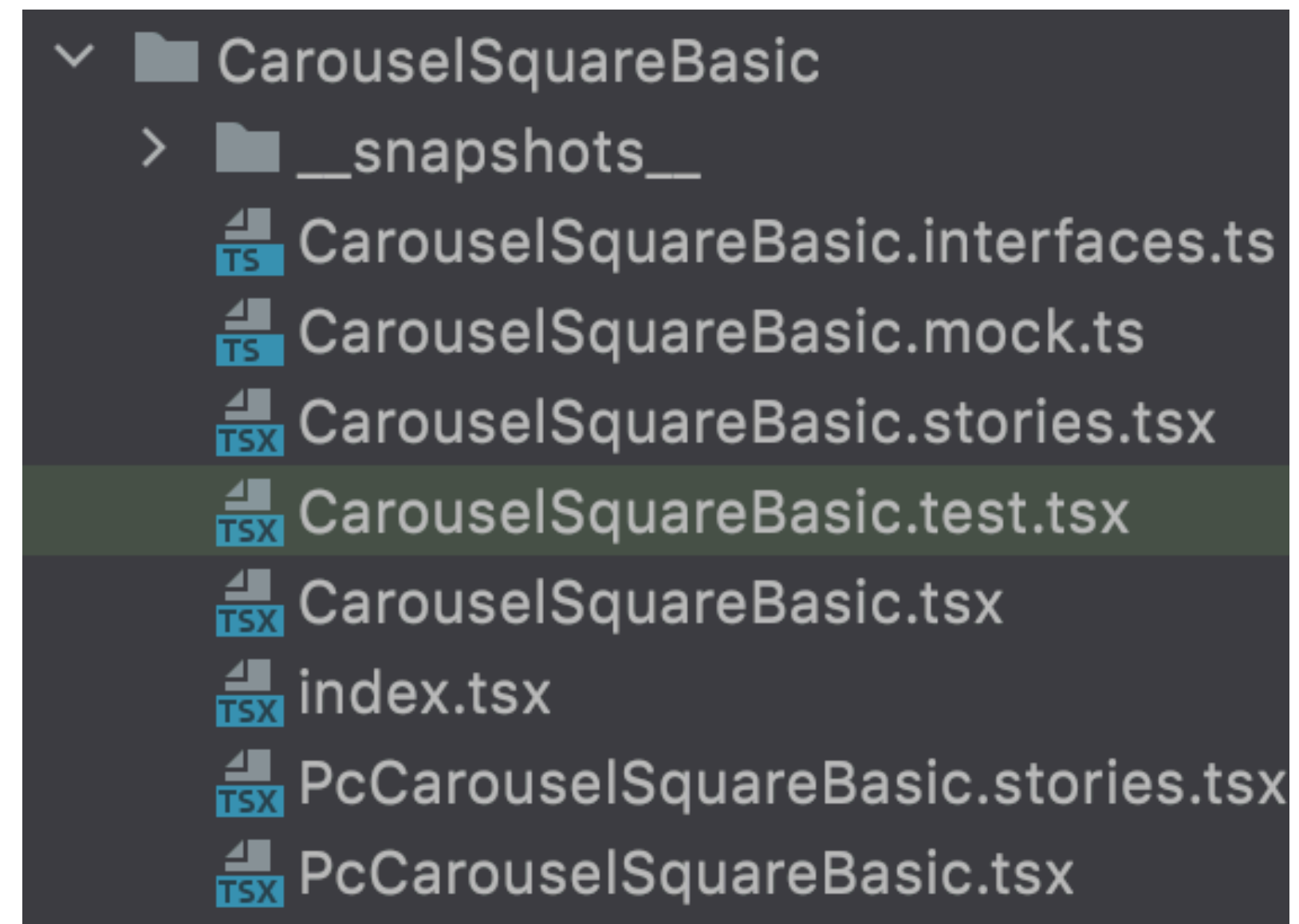
5.3 데이터 구조

컴포넌트 타입 네이밍 기준

- 커뮤니케이션 기준
- UI 형태에 맞게 네이밍
- 네이밍을 파일명에 반영



carousel_square_basic

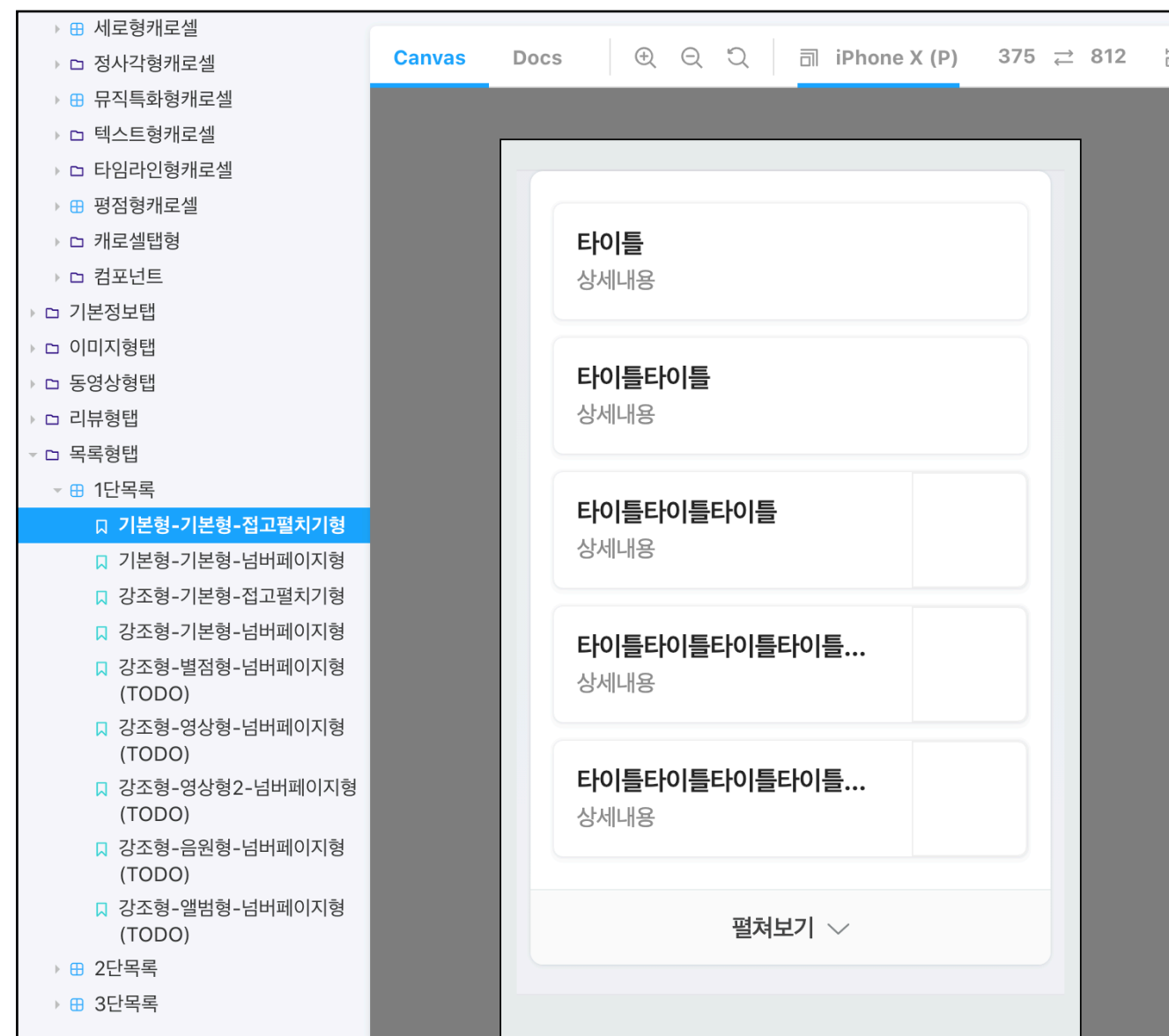


<CarouselSquareBasic />

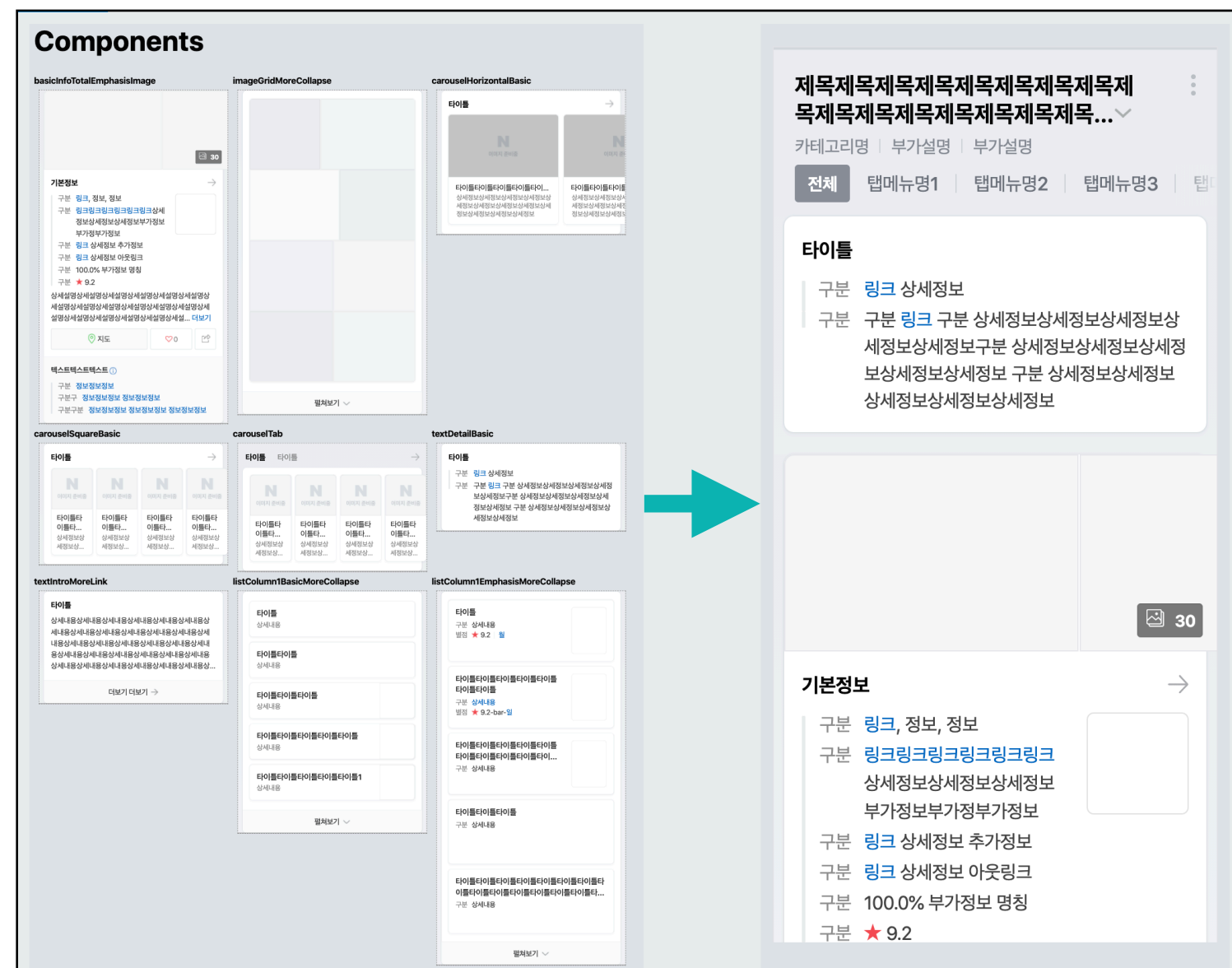
5.3 데이터 구조

데이터 시각화

- 개별 컴포넌트 시각화 => storybook
- 전체 데이터 시각화 => 컴포넌트 빌더



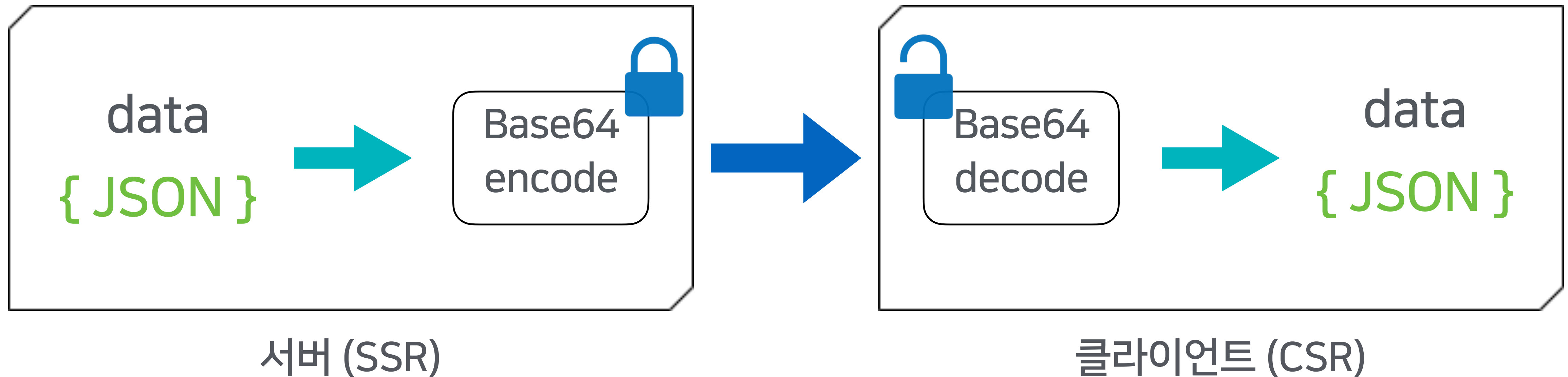
storybook



컴포넌트 빌더

5.4 데이터 처리

데이터 보안 강화



5.4 데이터 처리

서버 데이터 처리

- encodeURIComponent
- base64

```
Buffer.from(JSON.stringify(payload)).toString("base64")
```

```
const cid = findContentId(req.query);  
const uuid = generateUuidVariableLength();  
const encryptedData = encryptData(uuid, req.body);  
res.json( body: {  
  ...getServerAssetsJson().index,  
  entry: { cid, uuid, encryptedData },  
  ...{ html: createServerHtml(cid, uuid, req.body) },  
});
```

5.4 데이터 처리

클라이언트 데이터 처리

- base64 atob
- escape (unicode issue)
- decodeURIComponent

```
const uuid = retrieveUUID();
const data = decryptData(uuid, encryptedData) as SsrRootData;
const renderBy = ENV_IS_SSR ? hydrate : render;
renderBy(
  <StrictMode>
    <App data={data} />
  </StrictMode>,
  rootElement,
);
```

5.5 상태 관리

공통 데이터 context hook

```
function App({ data }: { data: SsrRootData<AppComponentData> }) {
  return (
    <ScriptLogErrorBoundary>
      <EtcInfoContextProvider {...data.etcInfo}>
        <ClickLogContextProvider {...data.clickLog}>
          <RootApp data={data} />
        </ClickLogContextProvider>
      </EtcInfoContextProvider>
    </ScriptLogErrorBoundary>
  );
}
```

```
export function EtcInfoContextProvider(
  props: Partial<SsrRootData["etcInfo"] & { children: React.ReactNode }>
) {
  const { children, device = "m" } = props ?? {};

  return (
    <EtcInfoContext.Provider value={{ device }}>
      {children}
    </EtcInfoContext.Provider>
  );
}
```

5.5 상태 관리

로컬 데이터

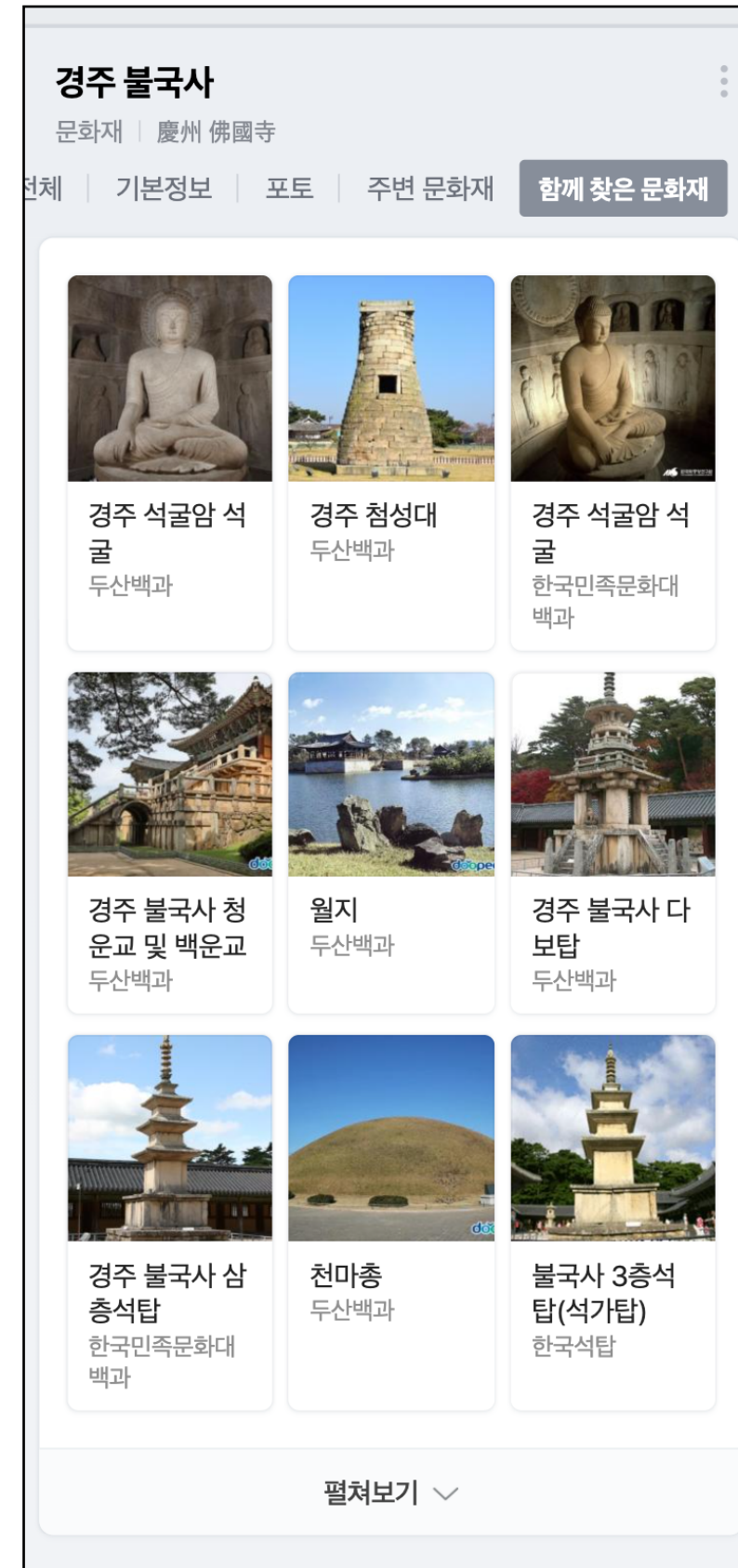
- state hook
- effect hook (외부 주입 데이터 변경시)

```

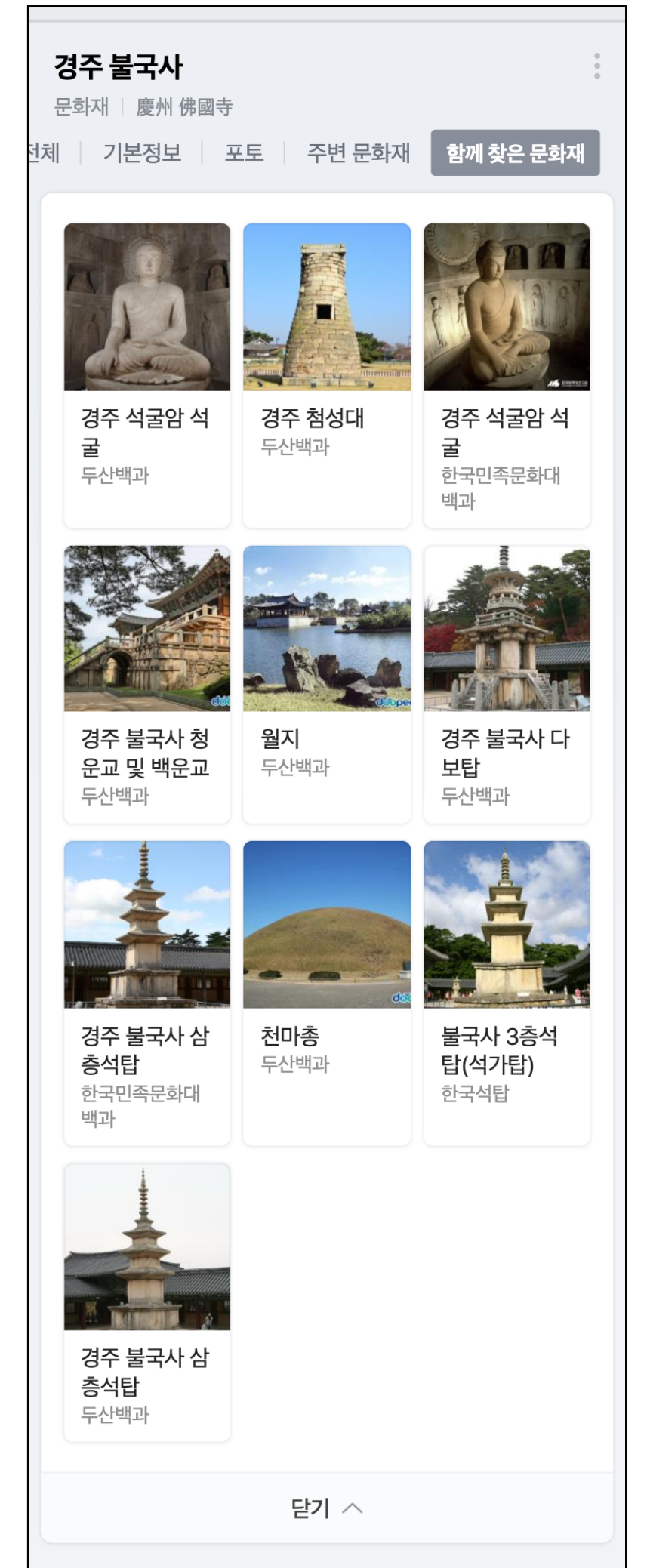
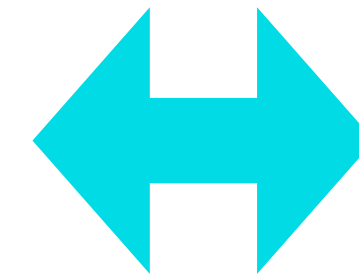
export function useSimpleToggle(data: { isOpen: boolean }) {
  const [isOpen, setToggle] = useState(data.isOpen);
  const onToggle = (e: { opened: boolean }) => {
    setToggle(e.opened);
  };
  useEffect( effect: () => {
    setToggle(data.isOpen);
  }, deps: [data.isOpen]);

  return [isOpen, onToggle] as const;
}

```



접힌 상태



펼쳐진 상태

5.6 실행 환경

실행 환경

- Node.js 14+ | IE 10+ | Android 4.4+, IOS 10+
- 서버/클라이언트 전역 객체 처리
- 디바이스 분기 처리

```
const performance = globalThis.performance;
const navigationTiming = performance?.getEntriesByType("navigation")?.[0];
if (getObjectType(navigationTiming) === "PerformanceNavigationTiming") {
  return navigationTiming as PerformanceNavigationTiming;
}
```

공통 전역 객체 (globalThis)

```
const etcInfo = useEtcInfoContext();

if (etcInfo.device === "pc") {
  return <PcCarouselSquareBasic {...(
} else {
  return <CarouselSquareBasic {...(pr
}
```

환경별 분기 처리

5.6 실행 환경

Micro Frontend 환경

- 네이버 검색 질의에 따라 동적으로 화면 배치 및 추가

구현 목표

- 외부 Side effect 최소화
- Global Context 수정 최소화

5.7 개발 환경

Webpack / Babel Loader (client)

- 외부 의존성도 트랜스파일링
- React (global Map/Set 의존성)

```
{
  test: /\. (js|ts)x?$/,
  exclude: file => /node_modules([\w\\])(@babel|core-js)/.test(file),
  use: {
    loader: "babel-loader",
    options: {
      configFile: path.resolve(__dirname, ".babelrc.client.js"),
    },
  },
},
```


5.7 개발 환경

Webpack / Babel Loader (client)

- 외부 의존성도 트랜스파일링
- babel 패키지 중복 트랜스파일링 주의

```
{  
  test: /\.js|ts$/,  
  exclude: file => /node_modules([\w\@babel|core-js])/ test(file),  
  use: {  
    loader: "babel-loader",  
    options: {  
      configFile: path.resolve(__dirname, ".babelrc.client.js"),  
    },  
  },  
},
```

5.7 개발 환경

Babel Presets (client)

- 외부 의존성 트랜스파일링으로 인한 충돌 방지

```
const presets = [  
  "@babel/preset-react",  
  "@babel/preset-typescript",  
  ["@babel/preset-env", {  
    "loose": true,  
    "modules": "cjs",  
    "useBuiltIns": false  
  }],  
];
```

5.7 개발 환경

Babel Plugins (client)

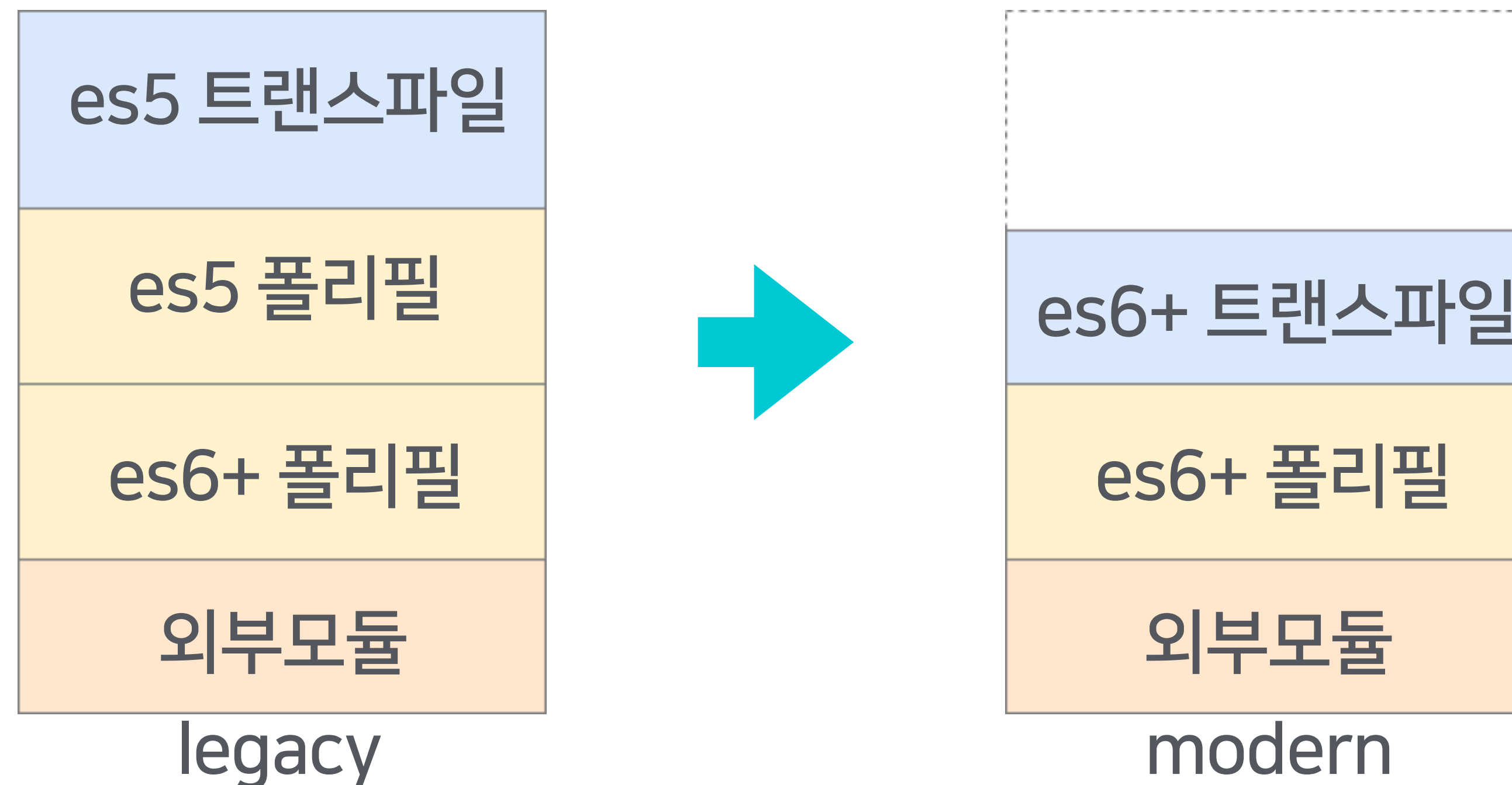
- 전역 오염 없는 폴리필 추가

```
const plugins = [  
  ["@babel/plugin-proposal-class-properties", {"loose": true}],  
  [  
    "@babel/plugin-transform-runtime",  
    {  
      "absoluteRuntime": false,  
      "corejs": {  
        "version": 3,  
        "proposals": true,  
      },  
      "helpers": true,  
      "regenerator": true,  
    }  
  ]  
];
```

5.8 Legacy / Modern 번들

최신 브라우저에서 최적화된 번들 전달

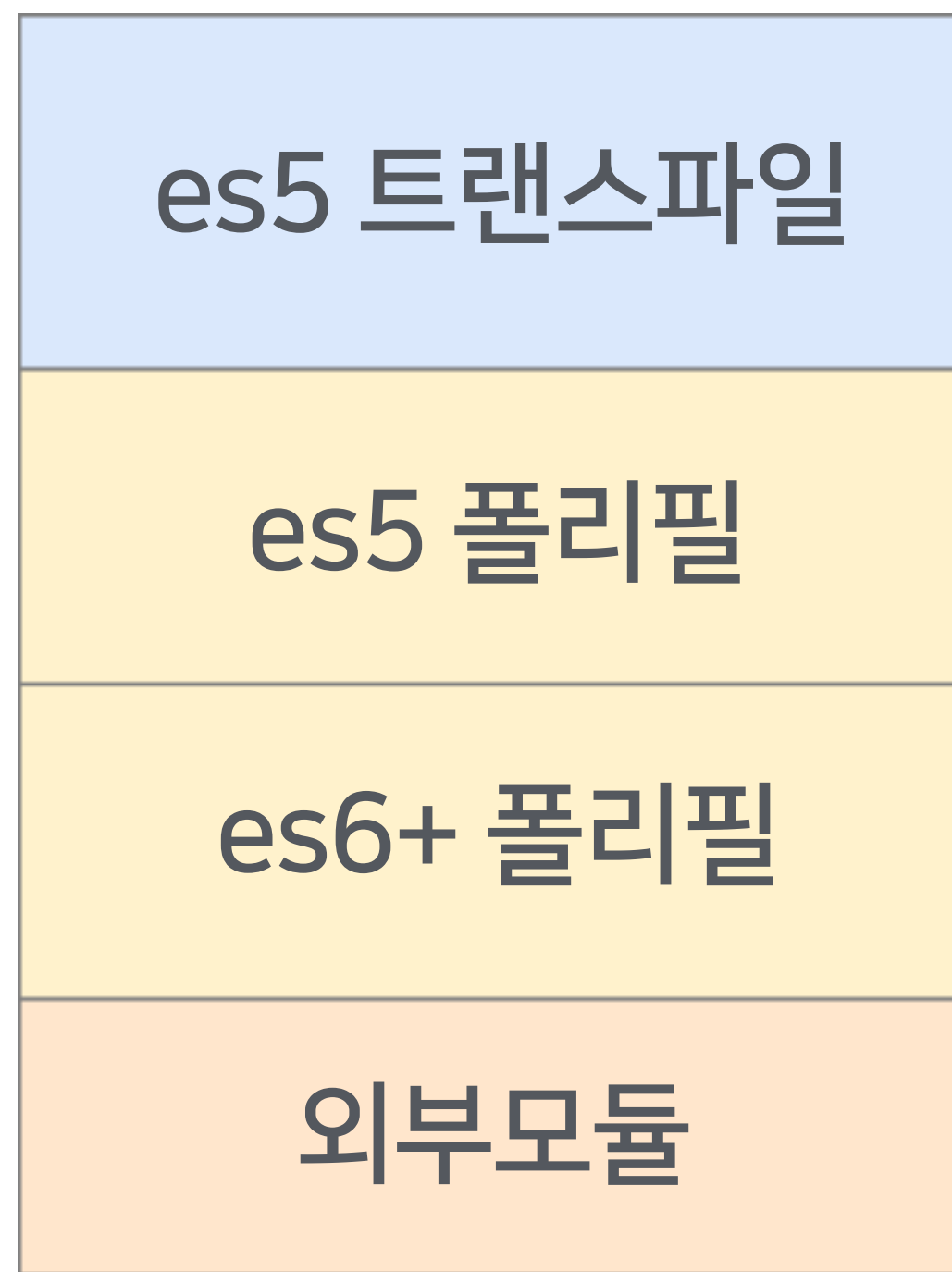
- 불필요한 폴리필, 트랜스파일링 제거
- 크기 감소, 다운로드 & 실행시간 감소



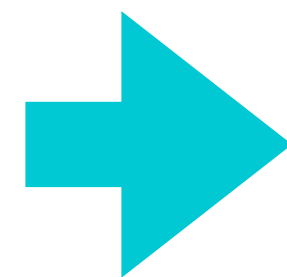
5.8 Legacy / Modern 번들

추가 최적화

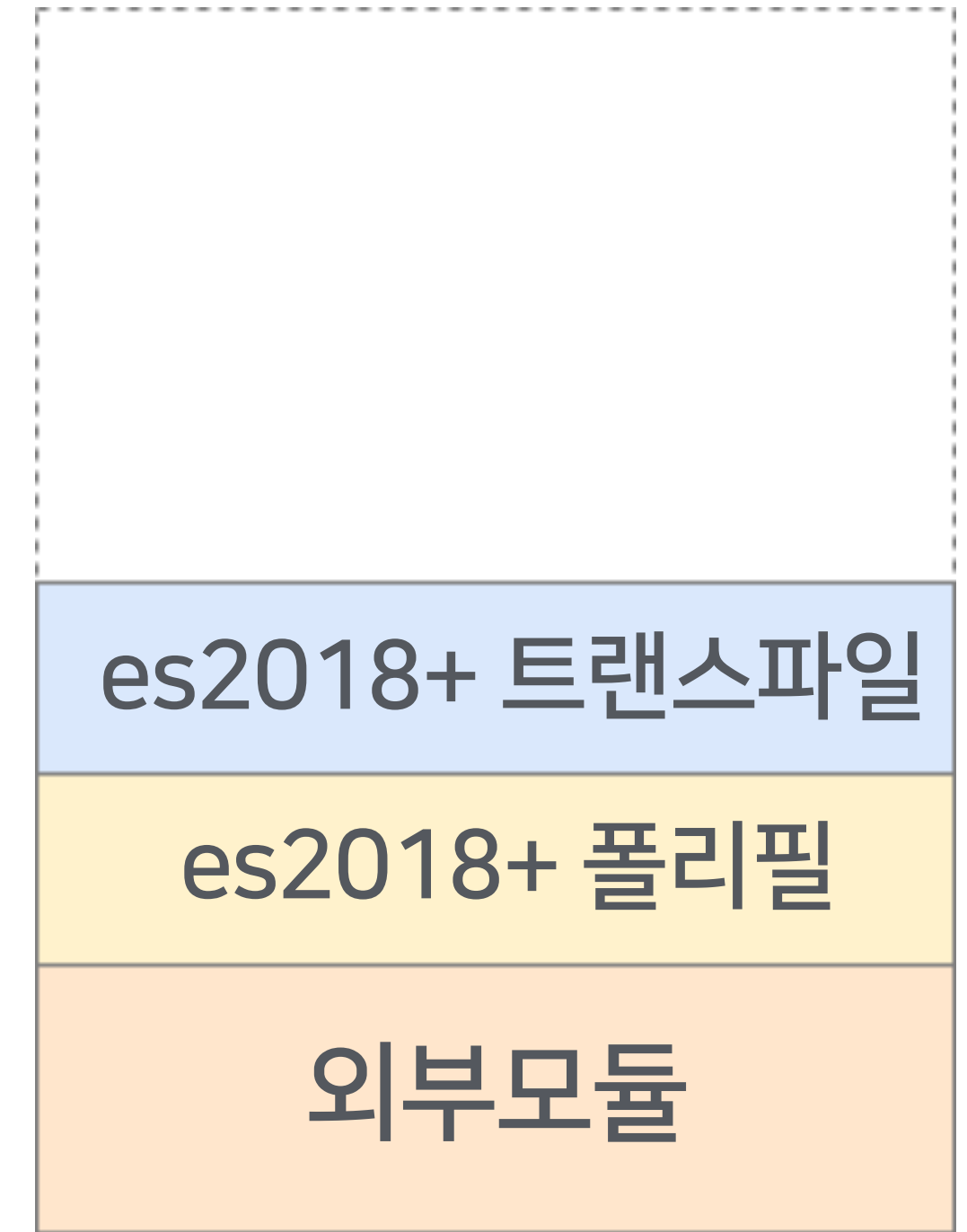
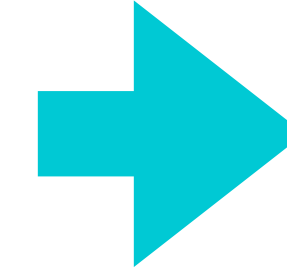
ES2018 스펙 지원여부로 최신브라우저 판단



legacy



modern



modern (추가 최적화)

5.8 Legacy / Modern 번들

Babel 설정 - Modern 번들 (Preset)

- targets: object-spread (es2018)

```
const targets = {  
  "browsers": [  
    'Chrome >= 61',  
    'Firefox >= 60',  
    'Edge >= 79',  
    'Safari >= 11.1',  
    'iOS >= 11.3',  
  ],  
};
```

```
module.exports = {  
  browserslistConfigFile: false,  
  targets: {"esmodules": true, ...targets},  
  presets: [  
    "@babel/preset-react",  
    "@babel/preset-typescript",  
    ["@babel/preset-env", {  
      "loose": false,  
      "modules": false,  
      "useBuiltIns": false,  
      ignoreBrowserslistConfig: true,  
      targets,  
    }],  
  ],  
};
```

5.8 Legacy / Modern 번들

Babel 설정 - Modern 번들 (Plugin)

- babel-polyfills: 전역 오염 X + 환경에 알맞는 폴리필 추가

```
plugins: [  
  "@babel/plugin-proposal-class-properties",  
  ["polyfill-corejs3", {  
    "method": "usage-pure",  
    ignoreBrowserslistConfig: true,  
    targets,  
  }],  
  ["@babel/plugin-transform-runtime", {  
    "absoluteRuntime": false,  
    "corejs": false,  
    "helpers": true,  
    "regenerator": false,  
  }]  
],  
};
```

babel-polyfills
+ corejs3

5.8 Legacy / Modern 번들

Webpack 설정 - Modern 번들

- 외부 의존성 트랜스파일링 제거
- 환경변수 주입 (필요한 경우)

```
if (!ENV_ES_NEXT) {
  require("intersection-observer");
  require("mutationobserver-polyfill");
}
```

```
module: {
  rules: [
    {
      test: /\.?(js|ts)x?$/,
      exclude: isWebEnvEsNext ? /node_modules/
      : file => /node_modules(?:[/\\])(@babel|core-js)/.test(file),
      loader: "babel-loader",
      options: {
        configFile: path.resolve(
          __dirname,
          isWebEnvEsNext ? ".babelrc.client.es.js" : ".babelrc.client.js",
        ),
      },
    }
  ]
},
plugins: [
  new webpack.DefinePlugin({
    ENV_ES_NEXT: isWebEnvEsNext,
  }),
]
```


5.8 Legacy / Modern 번들

번들 로딩

- 스펙 지원 여부로 분기

```
function checkSpec() {  
  try { var a = {}; a = new Function("var a = { ...{a:1} }; return a;")(); return a.a === 1;}  
  catch (e) {return false;}  
}  
var jsFileUrl;  
if (checkSpec()) {  
  jsFileUrl = "${modern-bundle}";  
} else {  
  jsFileUrl = "${legacy-bundle}";  
}
```

5.8 Legacy / Modern 번들

번들 로딩

- user-agent 분기

```
router.post( path: "/", handlers: (req: Request, res: Response) => {  
  const userAgent = req.headers["user-agent"];  
  
  const html = isModernBrowser(userAgent) ? modernJShtml : legacyJShtml;  
  
  res.send(html);  
});
```

5.8 Legacy / Modern 번들

파일크기 약 19% 감소 (gzip)

163.25 KB => 132.38 KB

실행시간 약 23% 감소 (throttle)

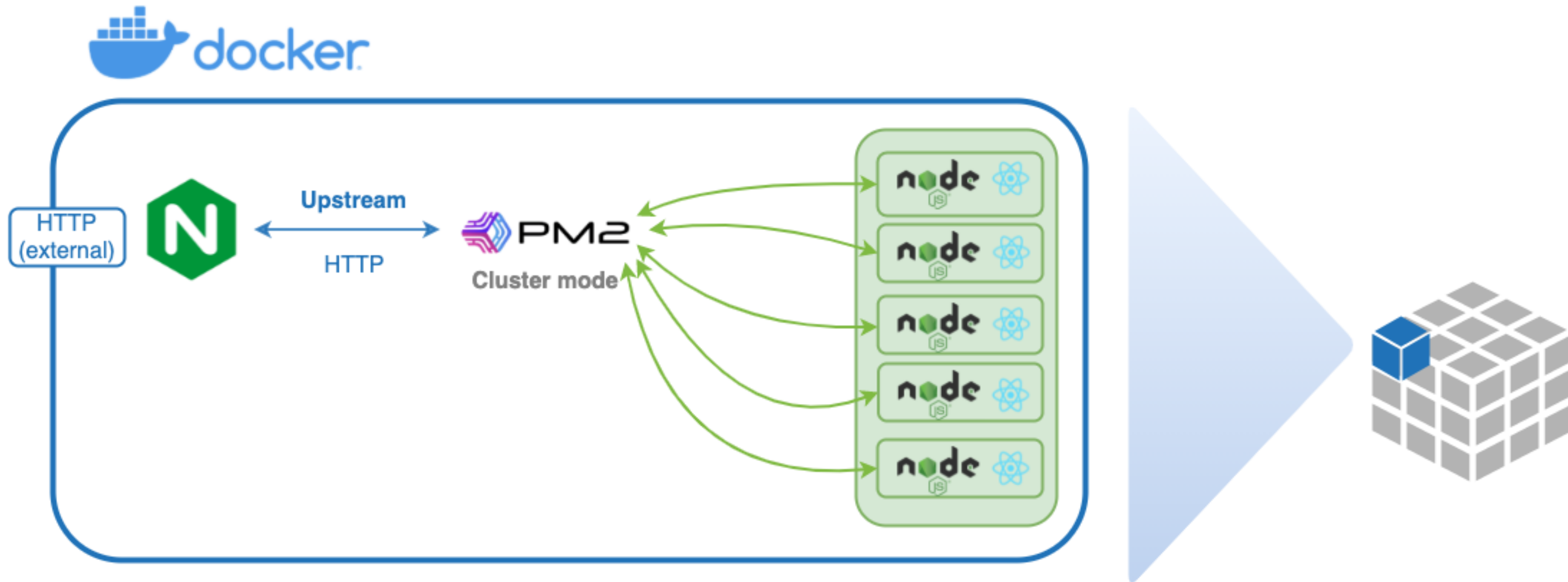
4473 ms => 3420 ms

size-limit report				
Path	Size (gzip)	Loading Time (slow 3G)	Running Time (snapdragon 410)	Total Time
index.js	163.25 KB	3265 ms	1207 ms	4473 ms
index.es.js	132.38 KB	2648 ms	772 ms	3420 ms

6. 서버 구성/빌드/배포

6.1 서버 구성

Docker 컨테이너 내부



6.1 서버 구성



가볍고 높은 가용성과 빠른 성능의 웹서버

- Gzip on
- L7 check
- Reverse proxy (pm2 서버 연동)
- Access/Error logs (SEER Collector 모니터링)

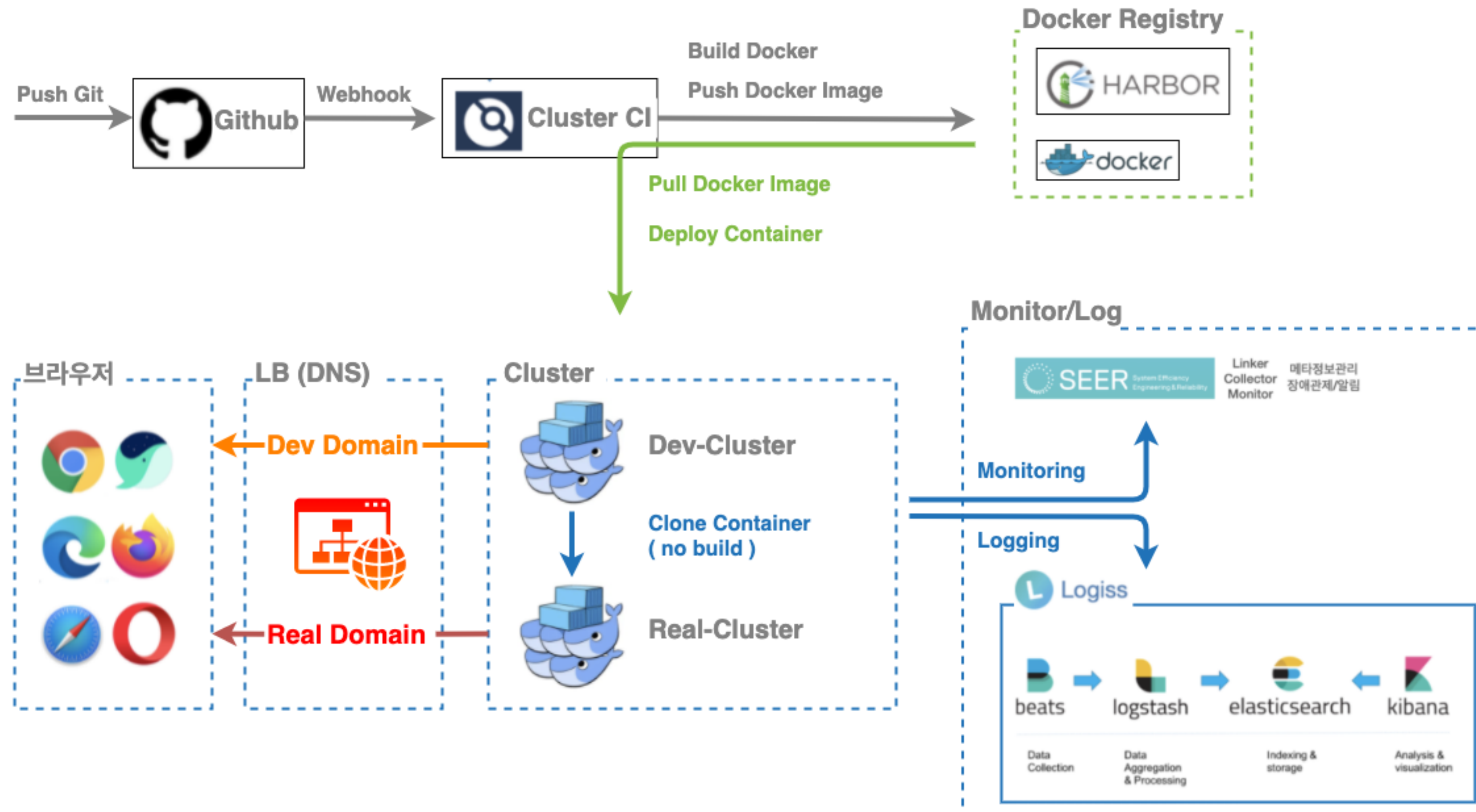
6.1 서버 구성



node.js 멀티 프로세스 관리

- 실시간 모니터링 도구 지원 (pm2 monit)
- 클러스터 모드를 설정하여 CPU 최대 활용 (트래픽 분산)
- 프로세스 비정상 종료 시, 특정 조건 (예: MEM 80%) 발생 시 재시작

6.2 서버 빌드/배포



7. 서버 운영/모니터링

서버 운영/모니터링

- 사전검증
- 배포/롤백
- AutoScale
- 모니터링
- 장애대응

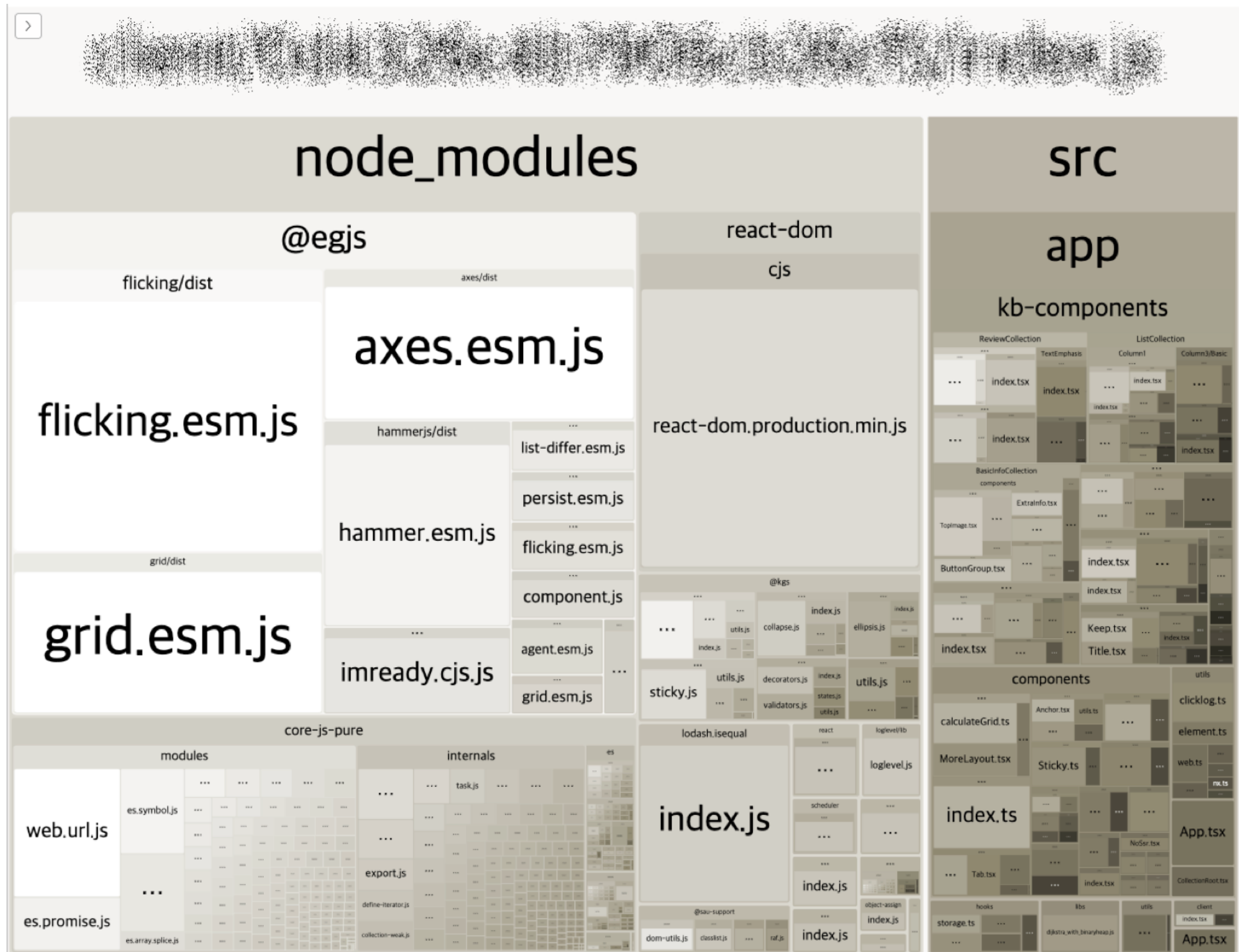
7.1 사전 검증

배포 전 Github Actions 이용한 코드 검증/테스트/복잡도 리포트

The screenshot displays the Github Actions interface. On the left, a sidebar lists various workflows such as 'Create Release Note', 'Client Bundle Check for PR', 'Push Develop', 'Open Pull Request', and 'Close Pull Request'. The main area shows a list of workflow runs under the heading 'All workflows' with 599 results. The top entry is 'env: 클라이언트 es5 / es-next 번들 분리' (Client Bundle Check for PR #46), which is in a successful state. Below this, a detailed view of the 'build-n-publish' job is shown, indicating it succeeded 18 hours ago in 2m 42s. The job steps include: Set up job (2s), Run actions/checkout@v2 (5s), setup node (1s), install yarn (1s), install node_modules (14s), build application (47s), check and storybook build (1m 2s), report coverage to Github (0s), git config user & remote set-url (0s), publish storybook pr branch (6s), test e2e storybook pr (22s), check pr head branch was deleted (1s), report storybook url to commit check (0s), report to works when workflows failed (0s), Post Run actions/checkout@v2 (1s), and Complete job (0s).

7.1 사전 검증

배포 전 Github Actions 이용한 코드 검증/테스트/복잡도 리포트



All files

73.5% Statements 1556/2117 66.69% Branches 817/1225 74.15% Functions 522/704 74% Lines 1511/2042

Press n or j to go to the next uncovered block, b, p or k for the previous block.

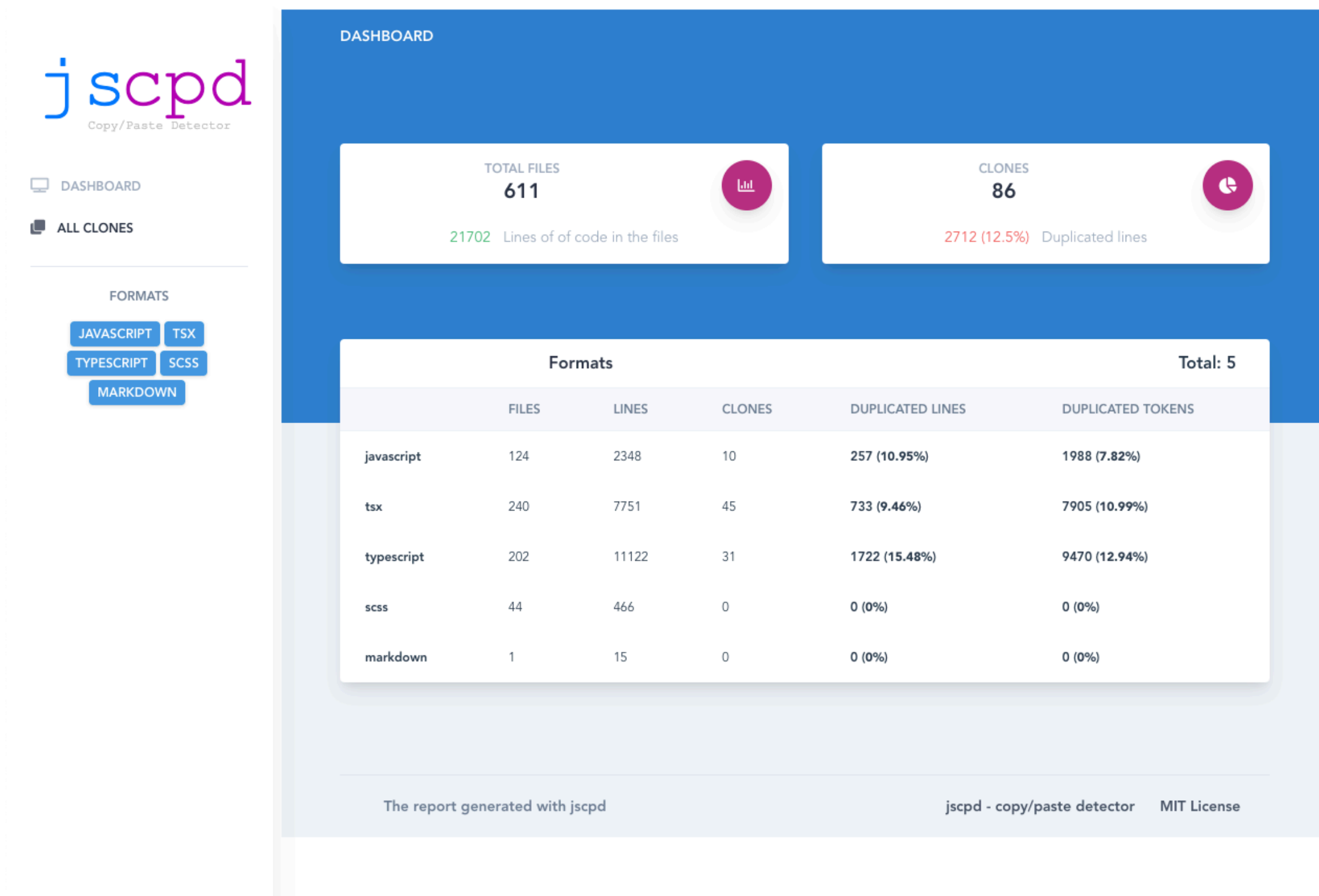
File	Statements	Branches	Functions
src	50%	2/4	100%
src/app	65.91%	29/44	48.57%
src/app/components/Anchor	100%	2/2	100%
src/app/components/ClickLog	100%	48/48	94.87%
src/app/components/Collapse	96.77%	30/31	84.21%
src/app/components/ComponentTitle	100%	4/4	100%
src/app/components/DescriptionList	100%	7/7	100%
src/app/components/DescriptionList/Entry	88.89%	24/27	88.46%
src/app/components/Ellipsis	95.24%	40/42	90.48%
src/app/components/Error	50%	8/16	75%
src/app/components/Flicking	84.62%	22/26	37.5%
src/app/components/Image	100%	8/8	100%
src/app/components/Layout/JustifiedLayout	88.21%	172/195	74.31%
src/app/components/MediaViewer	18.9%	24/127	11.11%
src/app/components/MoreCollection	0%	0/0	0%
src/app/components/MoreCollection/MoreCollapse	100%	5/5	100%
src/app/components/MoreCollection/MoreLink	100%	1/1	100%
src/app/components/MoreCollection/MoreOnce	75%	3/4	100%
src/app/components/MoreCollection/MoreOnceLink	80%	4/5	40%
src/app/components/MoreCollection/MoreSlideboard	100%	1/1	100%
src/app/components/MoreCollection/components/MoreButton	100%	1/1	80%

Test Reports

bundle analyzer develop test coverage develop duplicated code develop complexity develop

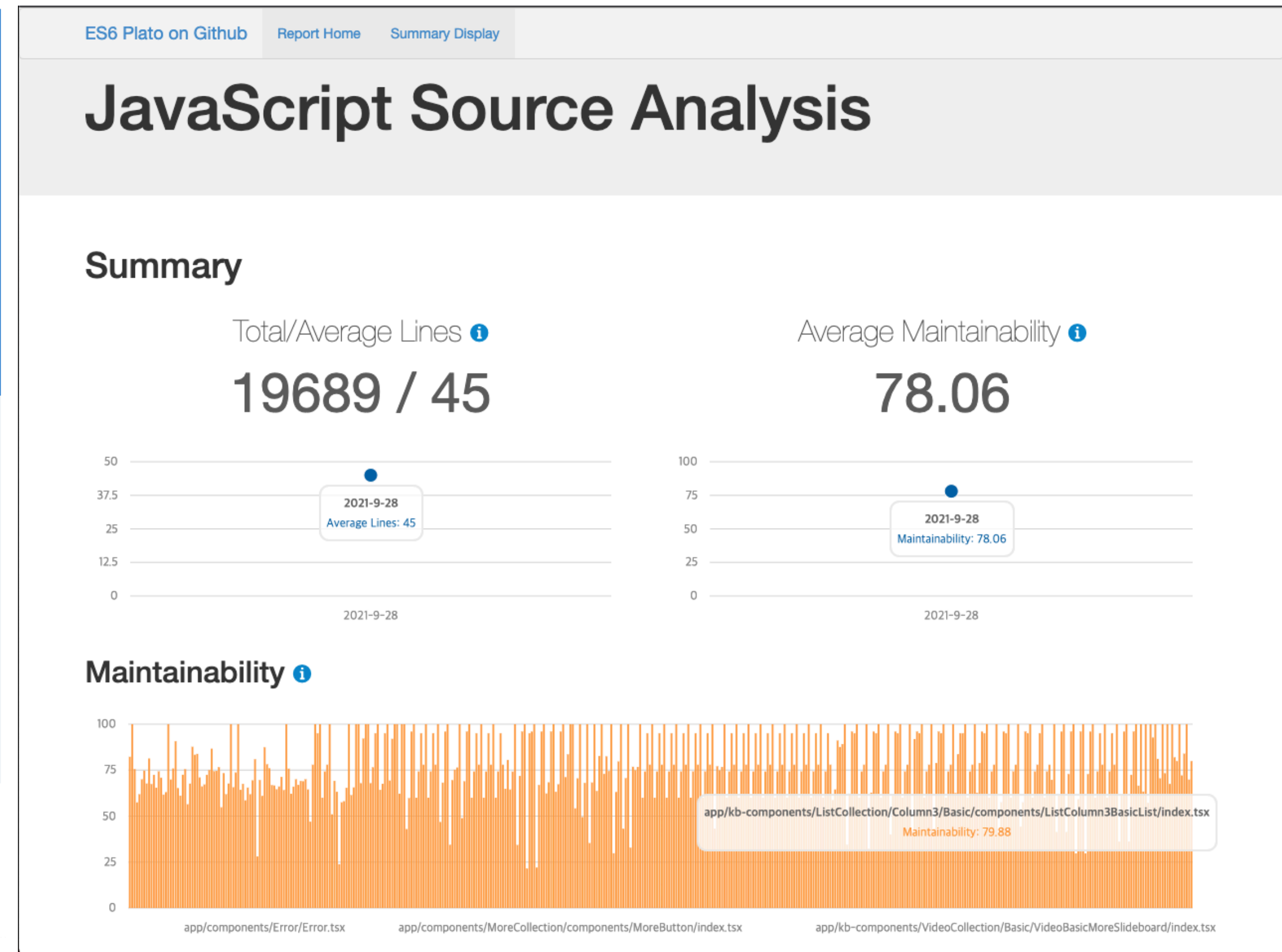
7.1 사전 검증

배포 전 Github Actions 이용한 코드 검증/테스트/복잡도 리포트



Test Reports

bundle analyzer develop test coverage develop duplicated code develop complexcity develop

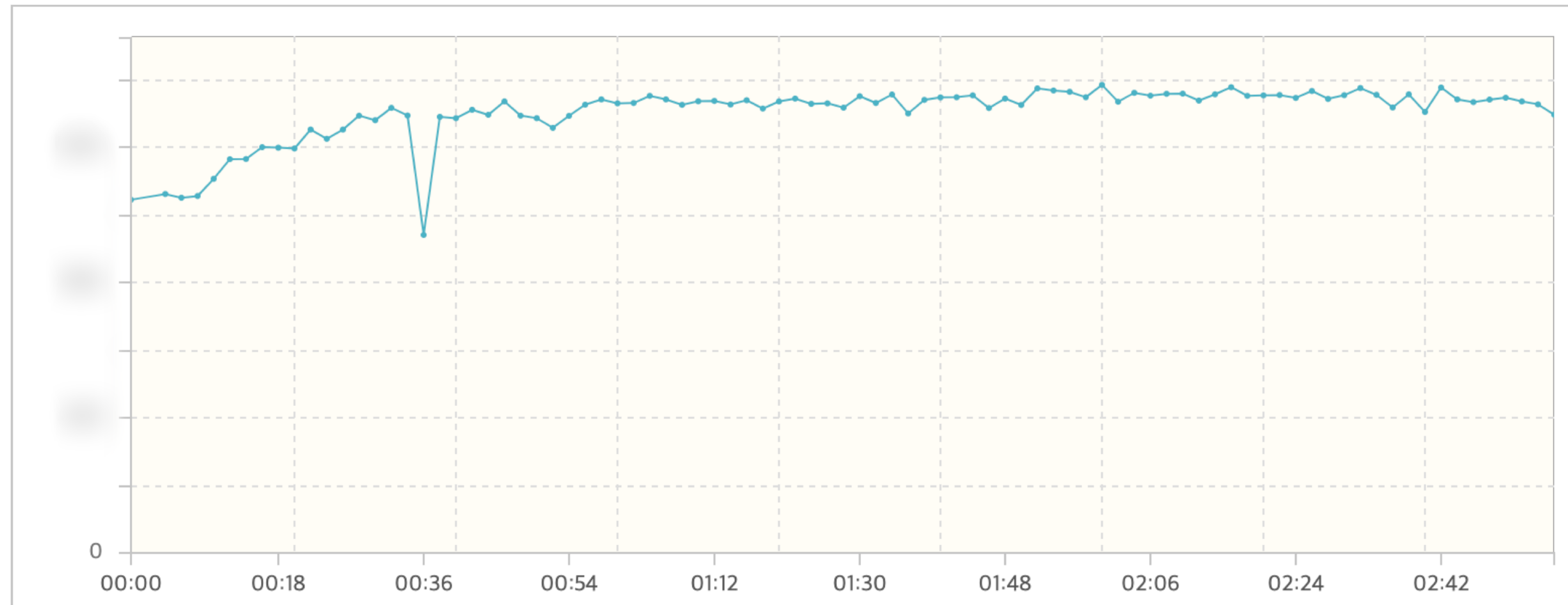


7.1 사전 검증

배포 후 nGrinder 성능/부하 테스트로 최대 TPS 산출

TPS 그래프

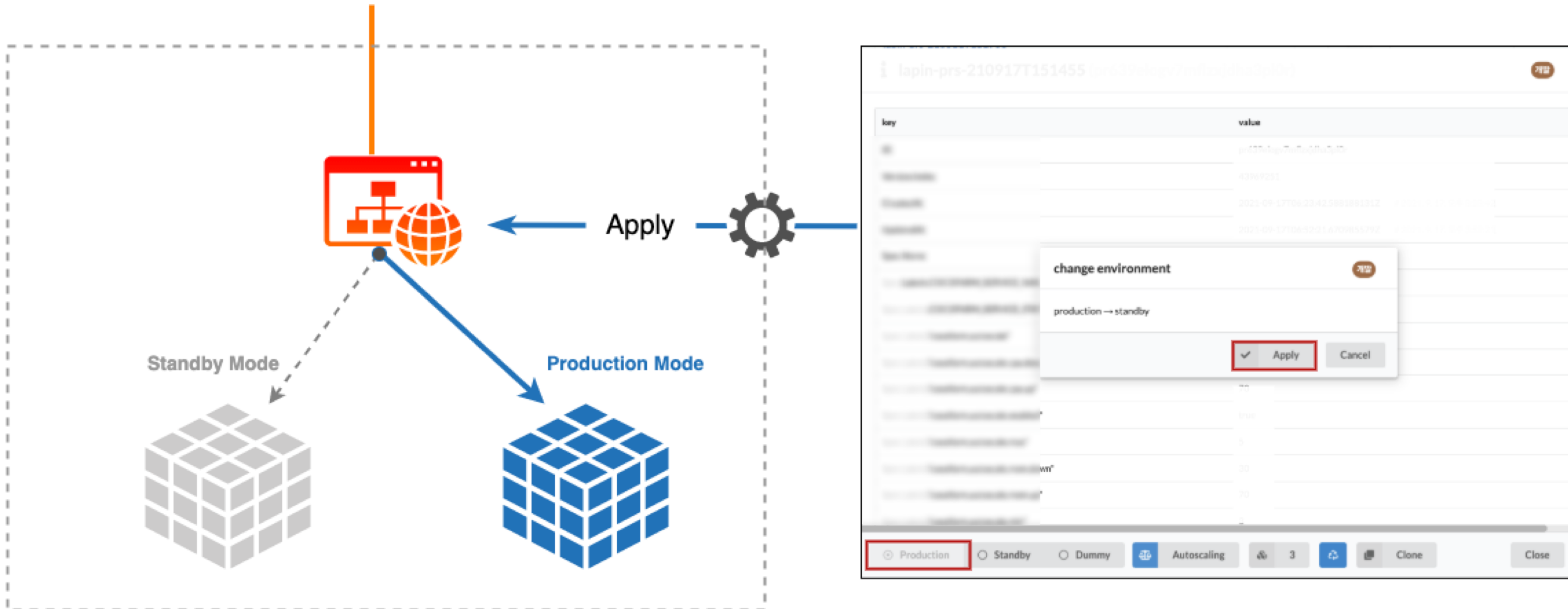
상세 보고서



<https://naver.github.io/ngrinder/>

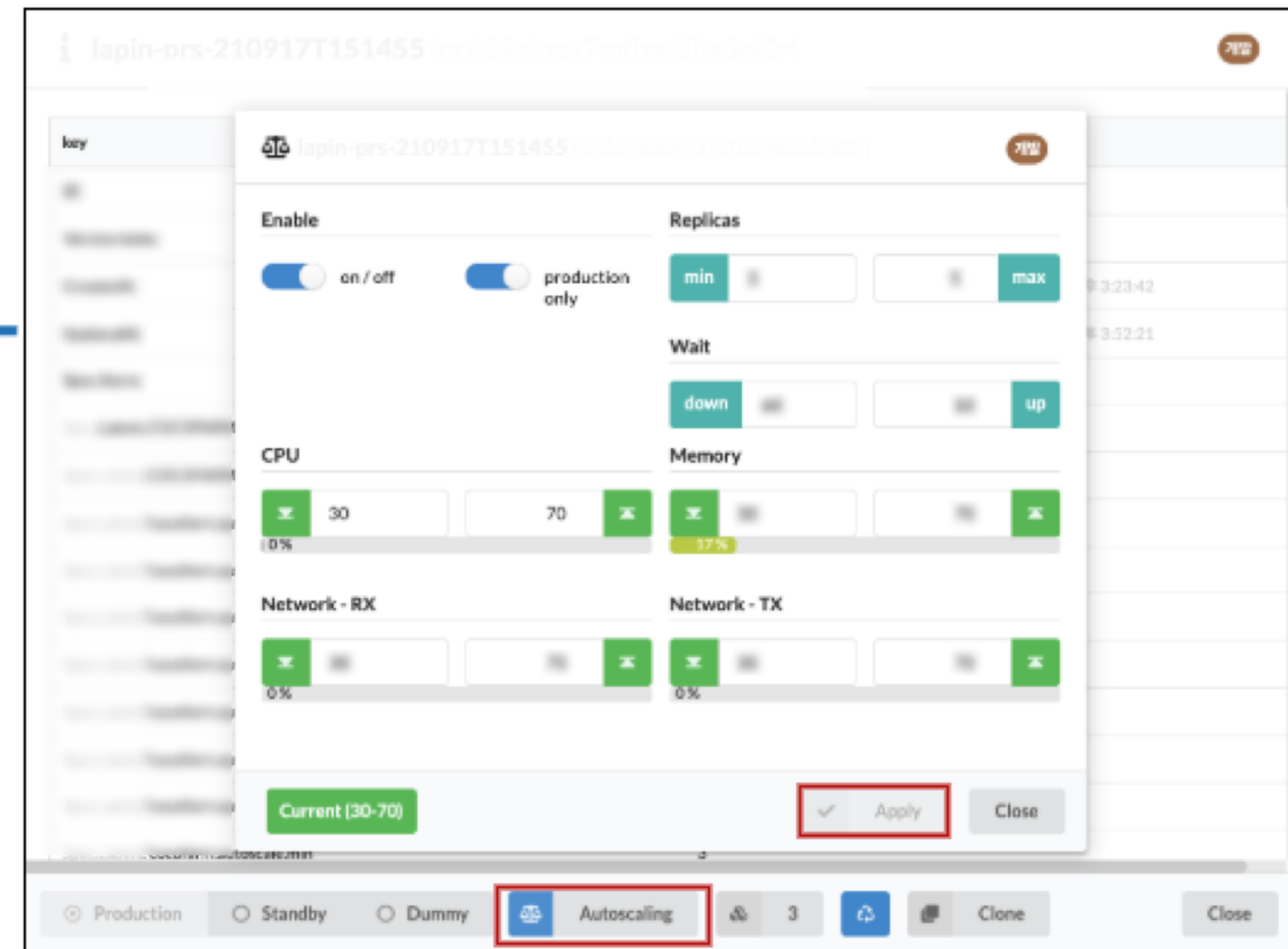
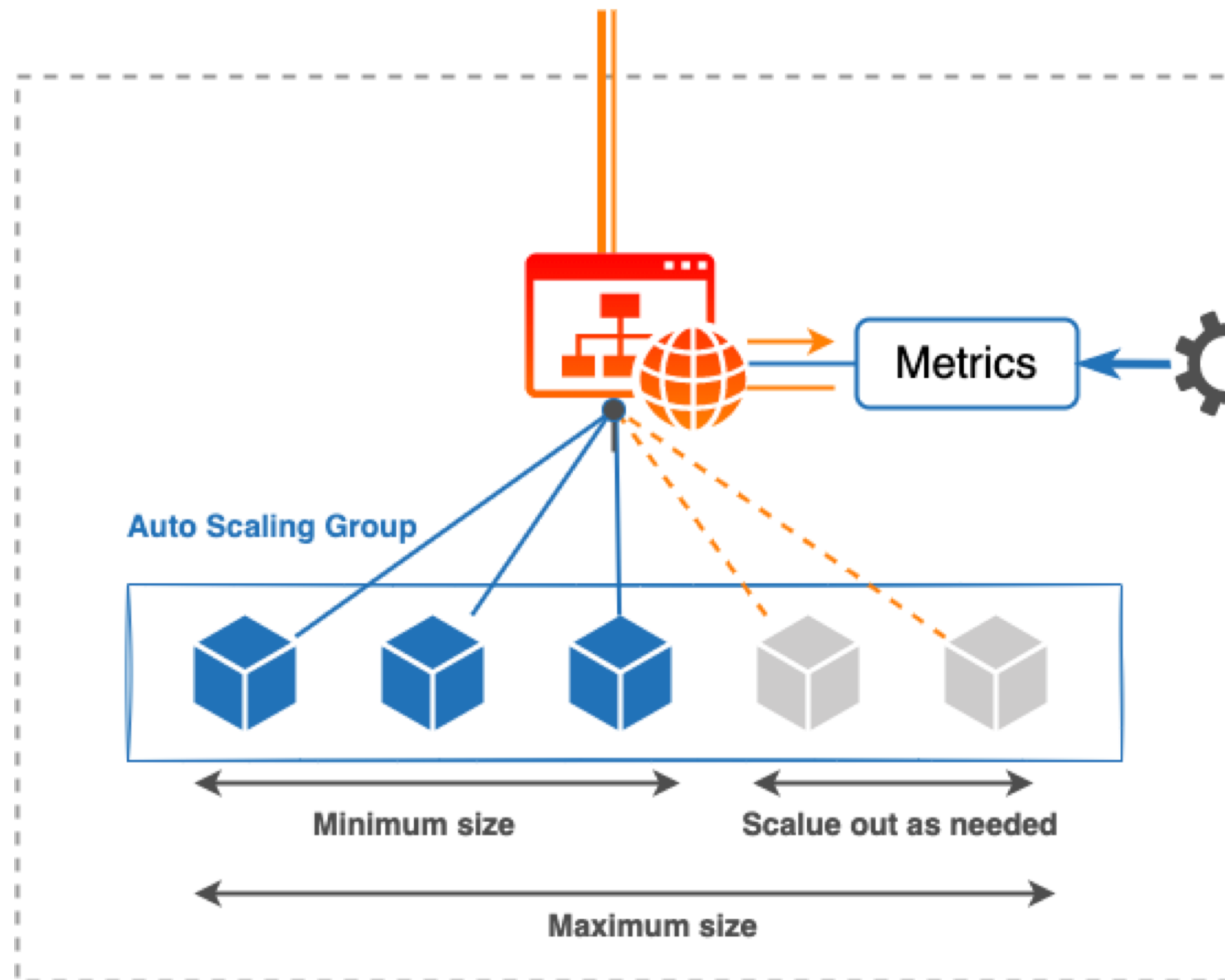
7.2 배포/롤백

웹 관리화면에서 모드 변경 후 실시간 서비스 반영/롤백



7.3 AutoScale

웹 관리화면에서 설정 변경 후 실시간 AutoScale 적용



7.4 모니터링 - 서버

SEER Collector

- 검색서비스 메타정보 관리, 시스템 로그 수집/장애알림
- CPU/MEM/DISK/NETWORK 등 서버 가용 정보 모니터링

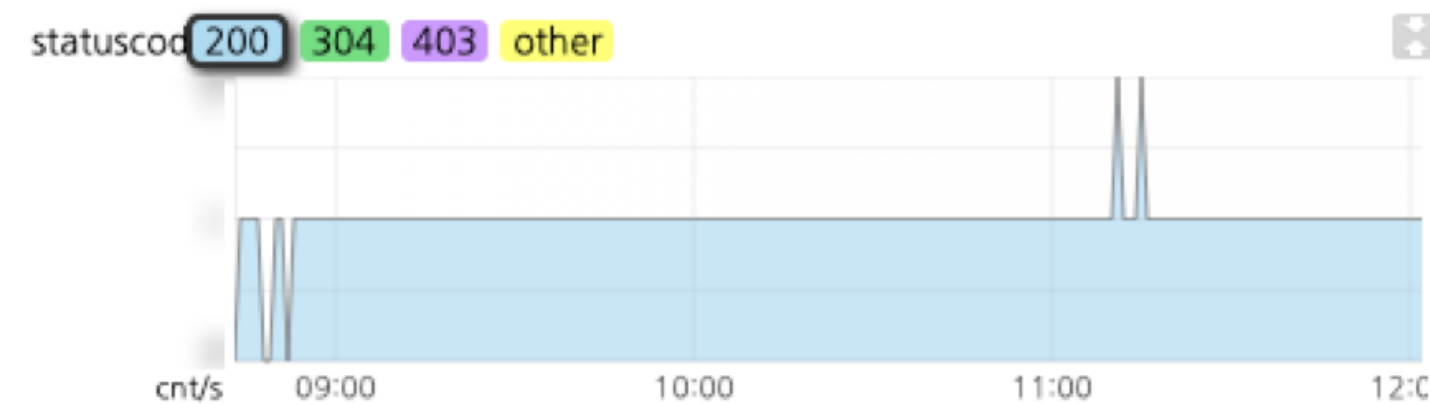
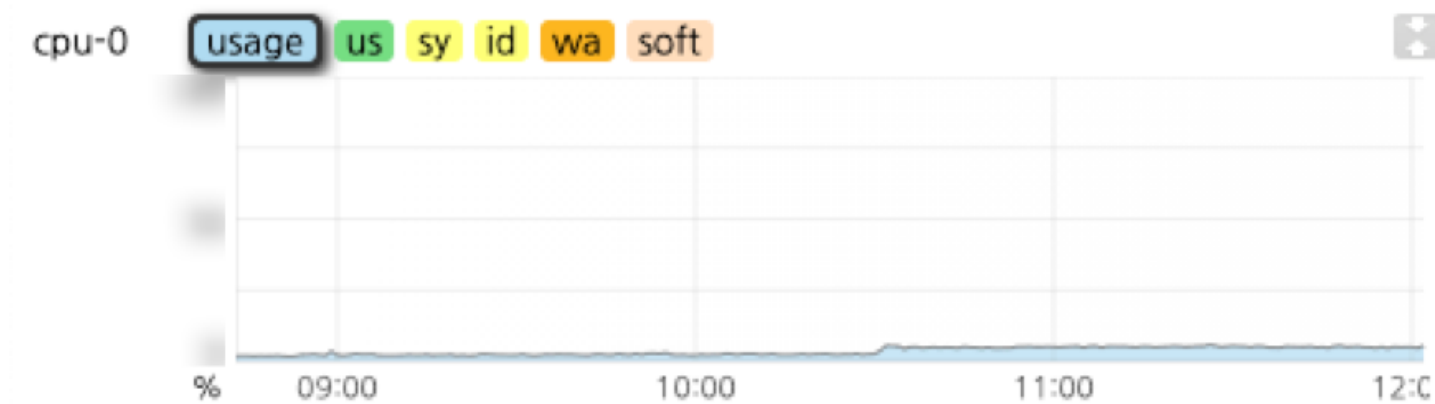
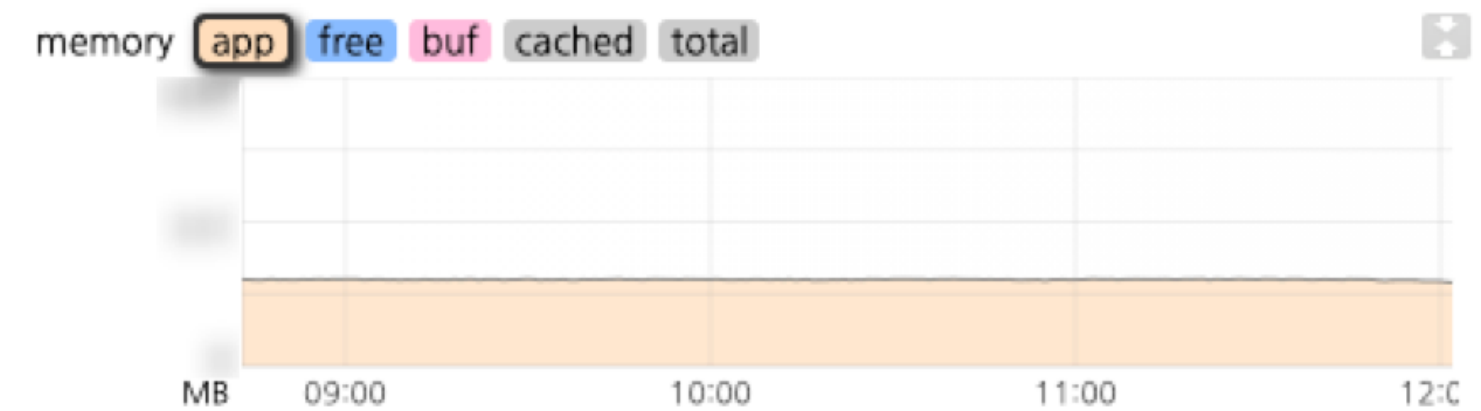
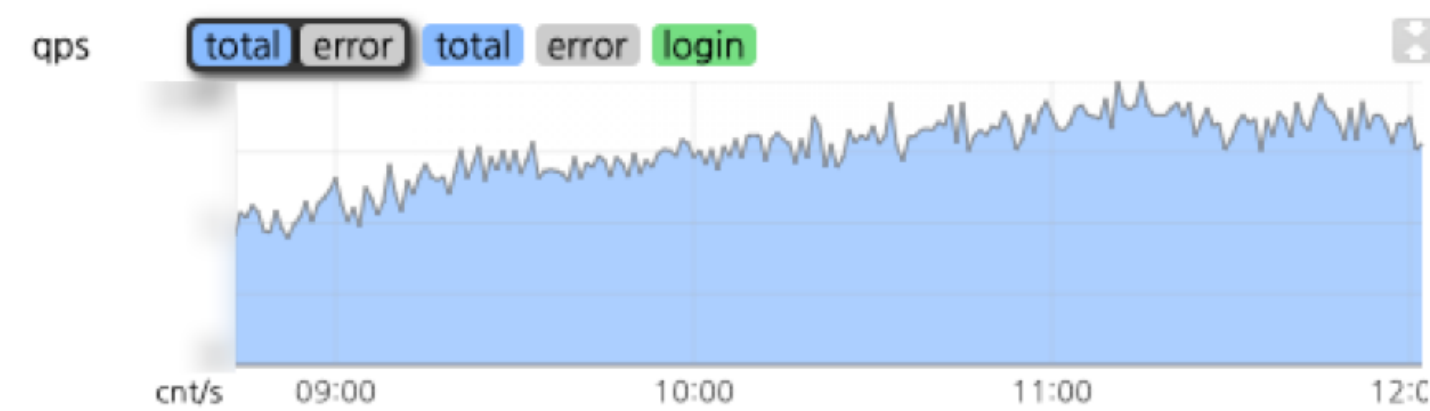
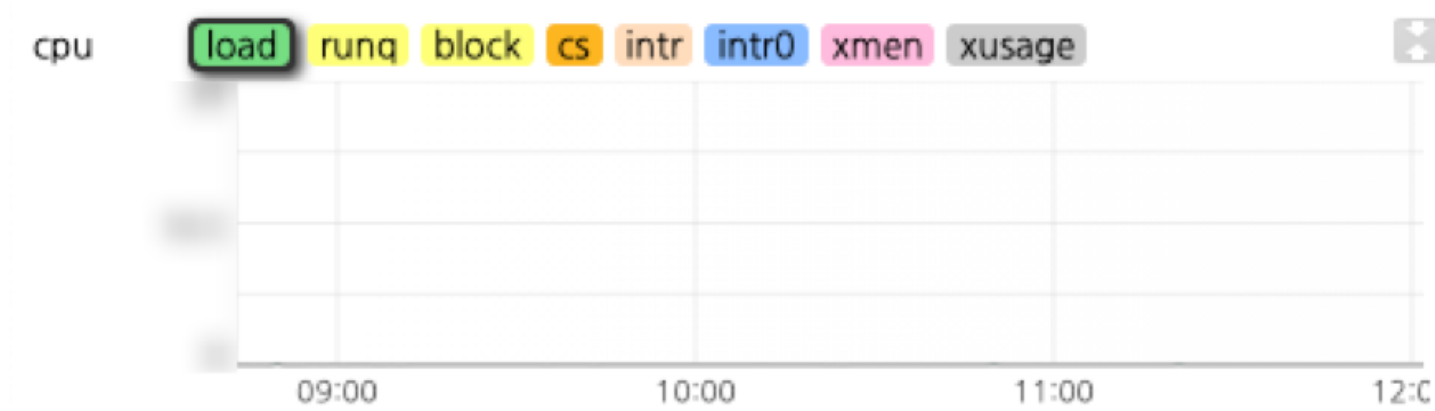
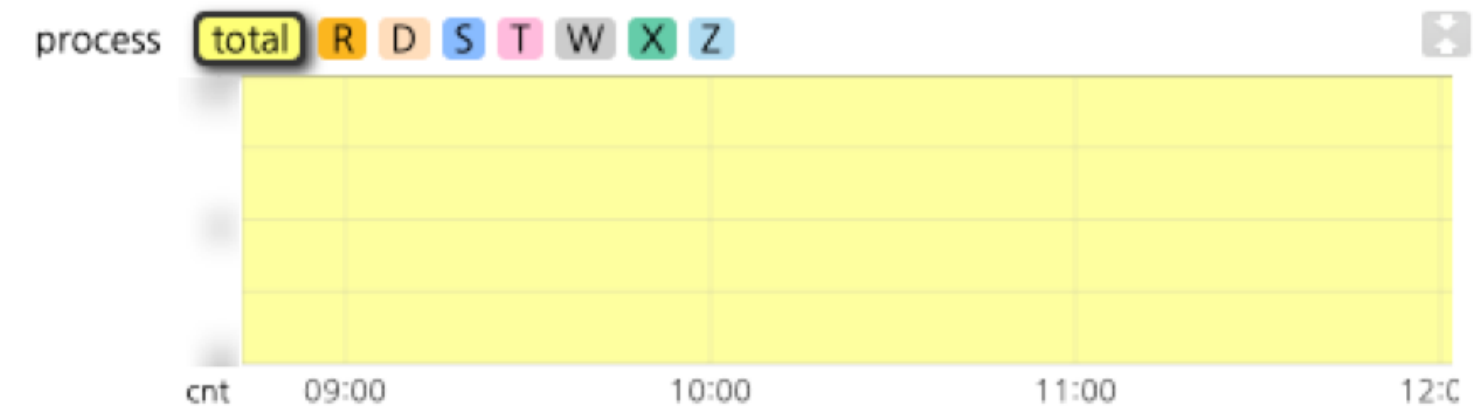
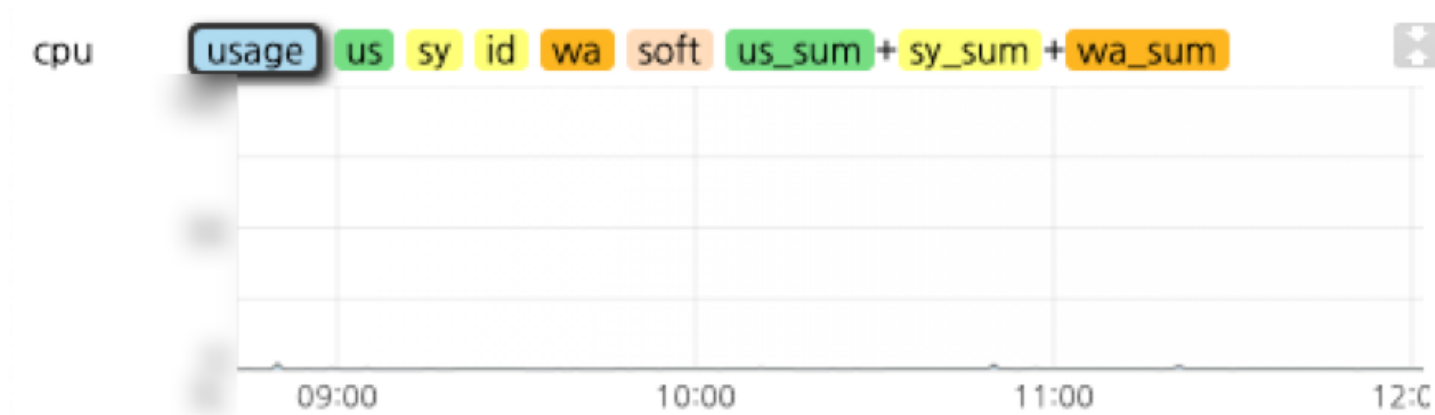
Logiss

- ELK 기반의 사내 통합 로그 검색 시스템
- 서버 액세스/에러 로그 저장/조회

7.4 모니터링 - 서버

SEER Collector

- 서버 가용정보 확인



7.4 모니터링 - 서버

Logiss

- 서버 접근/에러 로그 조회



7.4 모니터링 - 브라우저

Sentry

- 브라우저 스크립트 에러 수집
- 브라우저 환경, 에러 발생 시나리오, 에러 상세 정보를 조회

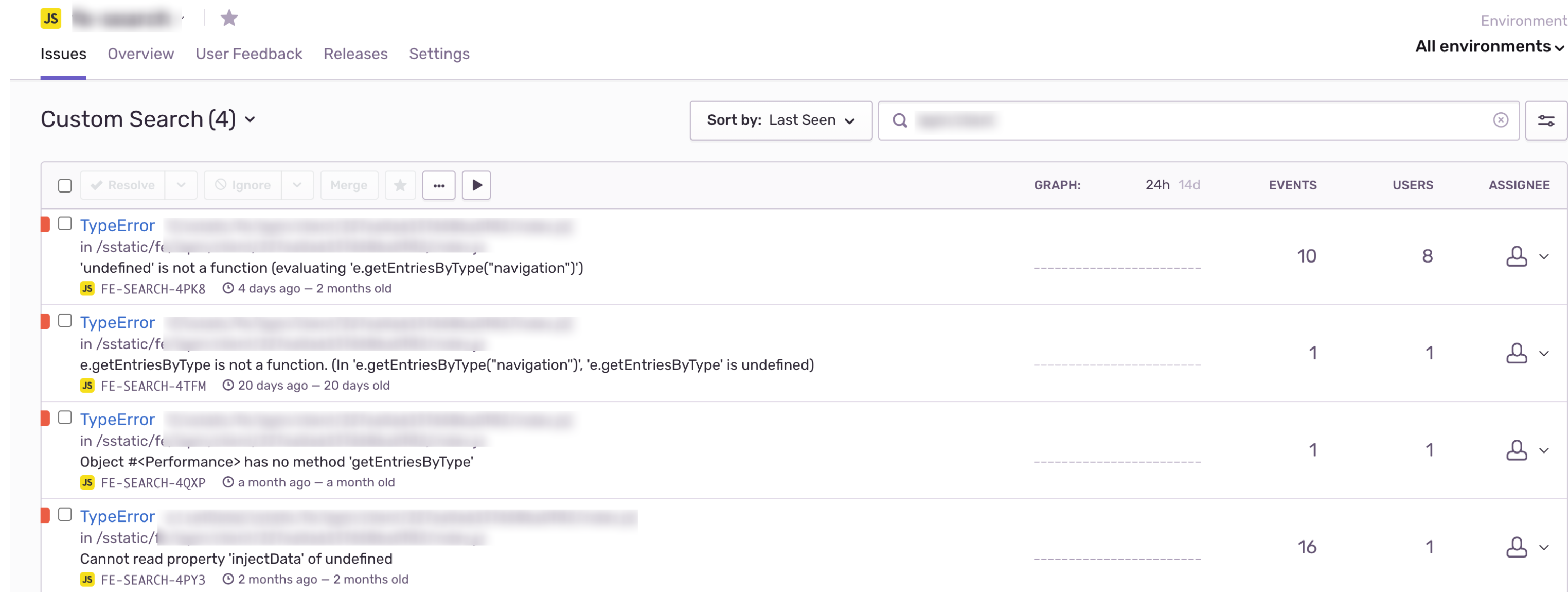
Nelo2

- 디자인 템플릿 화면 로그를 실시간 수집
- 화면 템플릿을 제거하거나 개선하는 용도로 활용

7.4 모니터링 - 브라우저

Sentry

- 브라우저 스크립트 에러 수집



The screenshot shows the Sentry web interface for monitoring browser errors. At the top, there are navigation links: Issues, Overview, User Feedback, Releases, and Settings. The environment is set to 'All environments'. A search bar is visible with 'Sort by: Last Seen' and a search icon. Below the search bar, there are action buttons: Resolve, Ignore, Merge, and a play button. The main content is a table of error events.

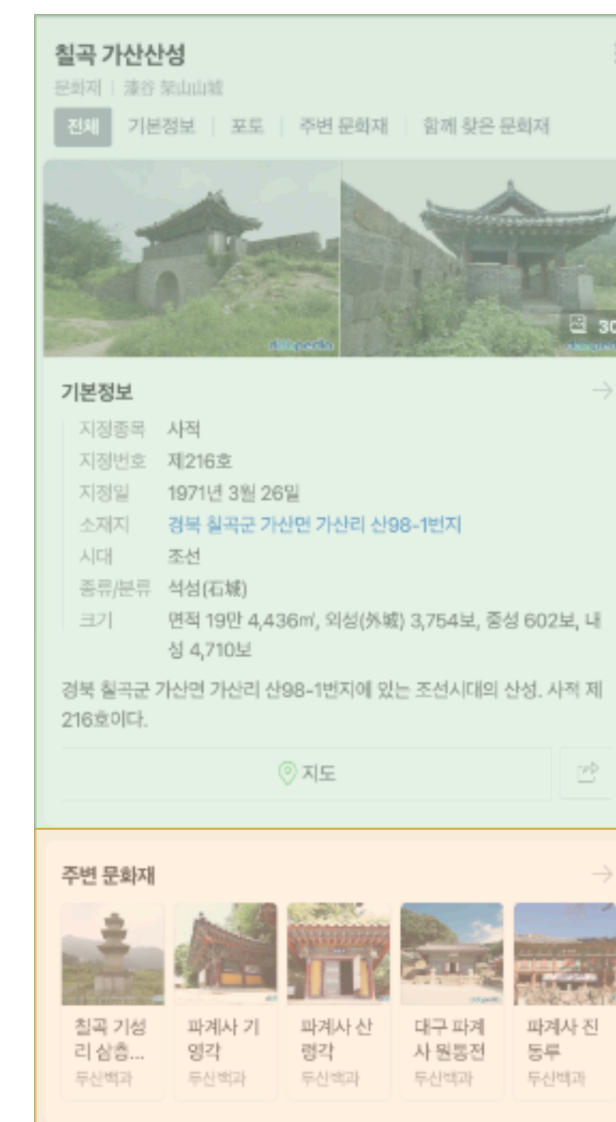
	GRAPH:	24h	14d	EVENTS	USERS	ASSIGNEE
<input type="checkbox"/> TypeError in /static/fe... 'undefined' is not a function (evaluating 'e.getEntriesByType("navigation")') JS FE-SEARCH-4PK8 4 days ago - 2 months old	-----			10	8	👤
<input type="checkbox"/> TypeError in /static/fe... e.getEntriesByType is not a function. (In 'e.getEntriesByType("navigation"); 'e.getEntriesByType' is undefined) JS FE-SEARCH-4TFM 20 days ago - 20 days old	-----			1	1	👤
<input type="checkbox"/> TypeError in /static/fe... Object #<Performance> has no method 'getEntriesByType' JS FE-SEARCH-4QXP a month ago - a month old	-----			1	1	👤
<input type="checkbox"/> TypeError in /static/fe... Cannot read property 'injectData' of undefined JS FE-SEARCH-4PY3 2 months ago - 2 months old	-----			16	1	👤

<https://sentry.io/>

7.4 모니터링 - 브라우저

Nelo2

- 디자인 템플릿 화면 실제 사용 로그 수집



7.5 장애 대응

pm2 설정에 따라서

- 서버 프로세스 에러로 강제 종료 발생 시 빠르게 인스턴스 재시작
- 서버 프로세스 사용 최대 메모리 초과 시 빠르게 인스턴스 재시작

```
module.exports = {
  apps: [{
    name: "app",
    script: "dist/server/index.bundle.js",
    max_restarts: 5, // restart max count when node down
    exec_mode: "cluster", // use cluster for node multi process (worker)
    max_memory_restart: '300M', // each instance, restart node when reach max memory
    ...
  }],
};
```

7.5 장애 대응

메모리릭 발생 시

- 필요 시 메모리 덤프 (스냅샷) 생성

```
process.on("SIGUSR2", () => {
  const snapshotStream = v8.getHeapSnapshot();
  snapshotStream.pipe(
    fs.createWriteStream(
      `${process.cwd()}/var/app${NODE_APP_INSTANCE}-${Date.now()}.heapsnapshot`
    )
  );
});
```

```
node_pid=$(pgrep '^node\s+(/|\w)+index.bundle.js' | head -1)
kill -USR2 $node_pid
```


8. **앞으로의 계획**

8. 앞으로의 계획

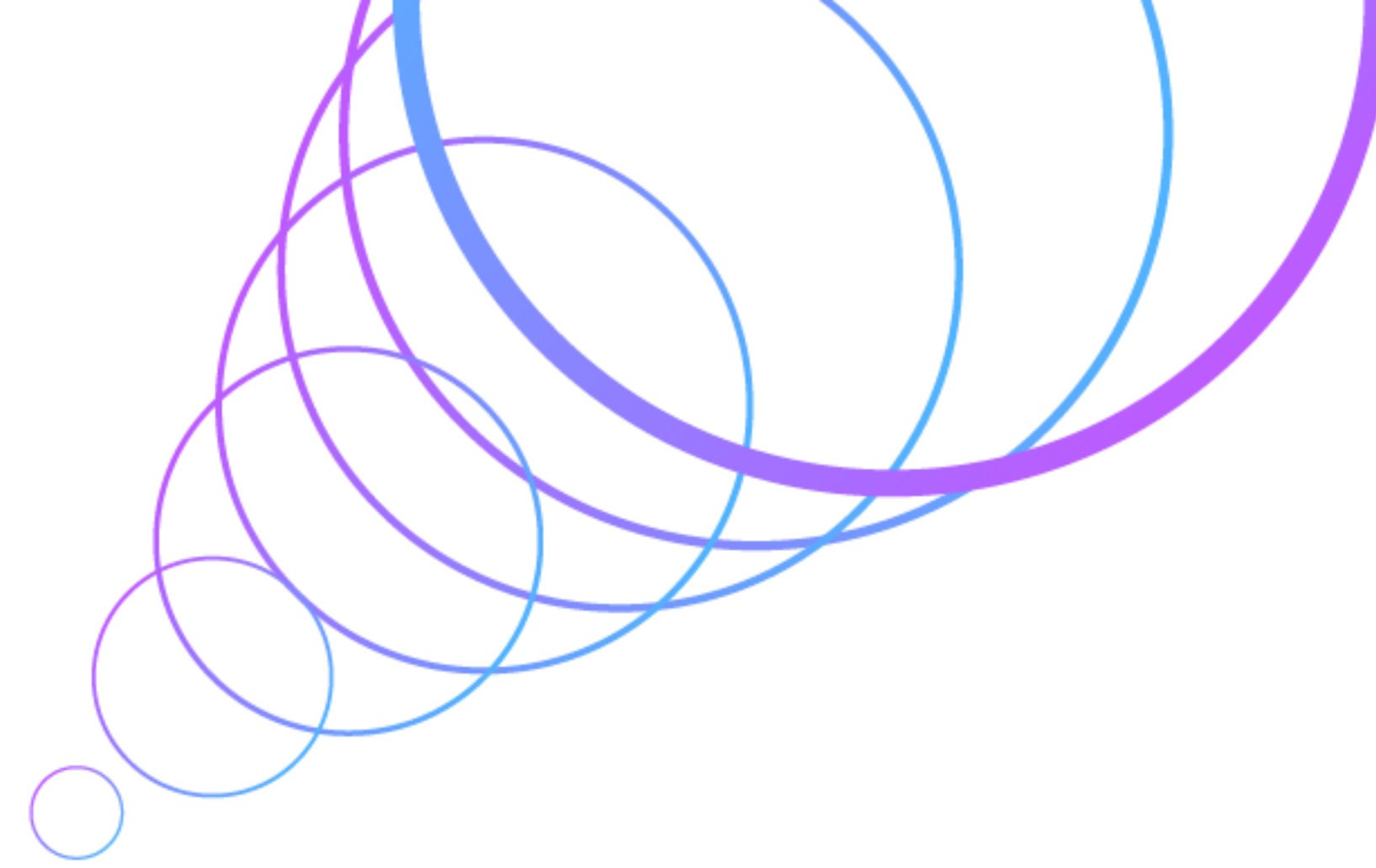
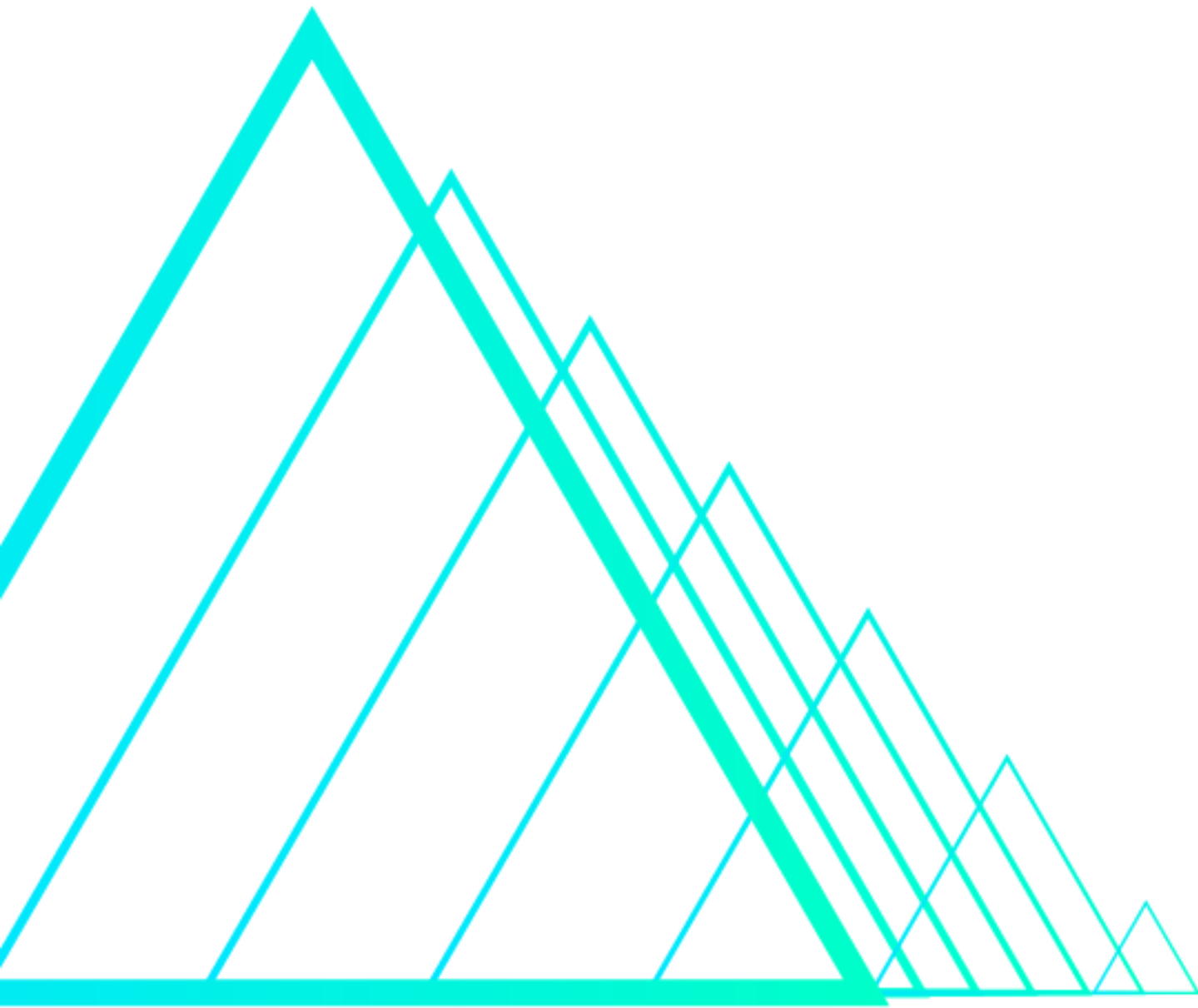
- 미구현 화면 컴포넌트 추가
- 화면 컴포넌트 빌더 기능 개선
- 디자인 시스템의 개선 및 고도화
- 디자인 템플릿 페이지와 스토리북 연동

채용 공지



네이버 검색 부스로 오세요

- 11월 26일은 발표자 Q/A 및 채용 상담 가능합니다.



Thank You

