

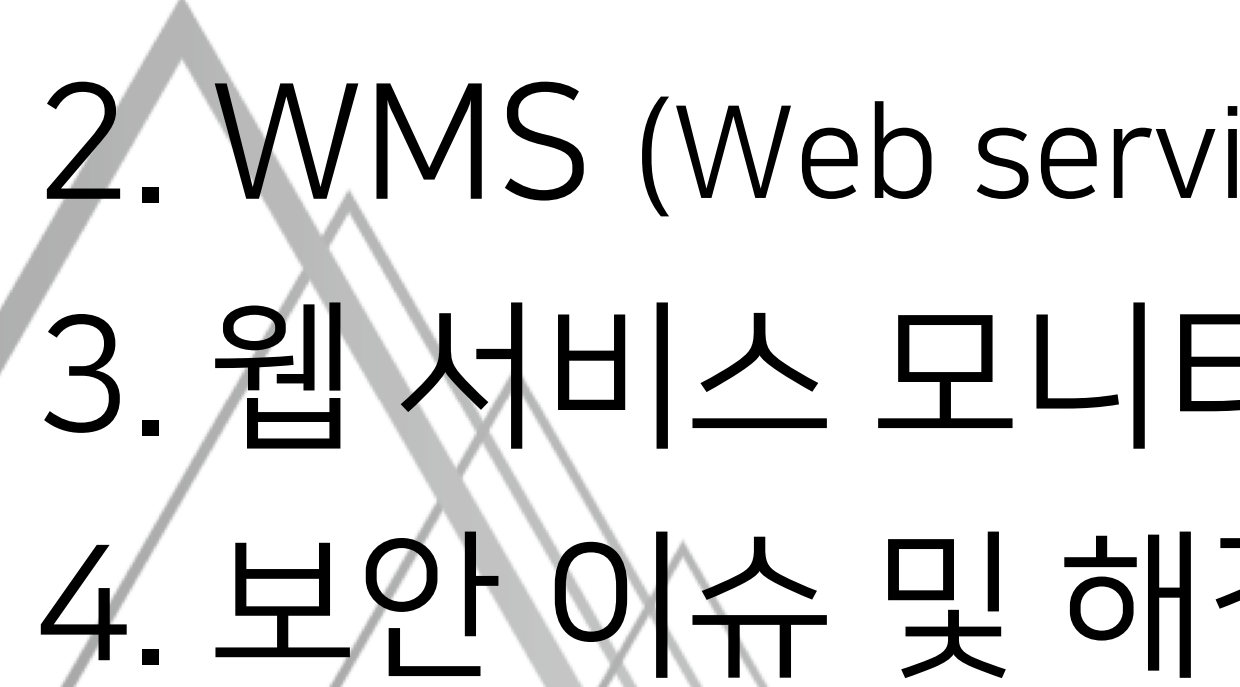



24/7 웹 서비스 장애 감지를 위한 모니터링 시스템 개발기

전현준 PaaS Dev2, NAVER Cloud

CONTENTS



1. 개발 배경 (기존 시스템 문제 및 해결방안)
 2. WMS (Web service Monitoring System) 소개 및 구조
 3. 웹 서비스 모니터링 기능 구현하기
 4. 보안 이슈 및 해결
- 
- 

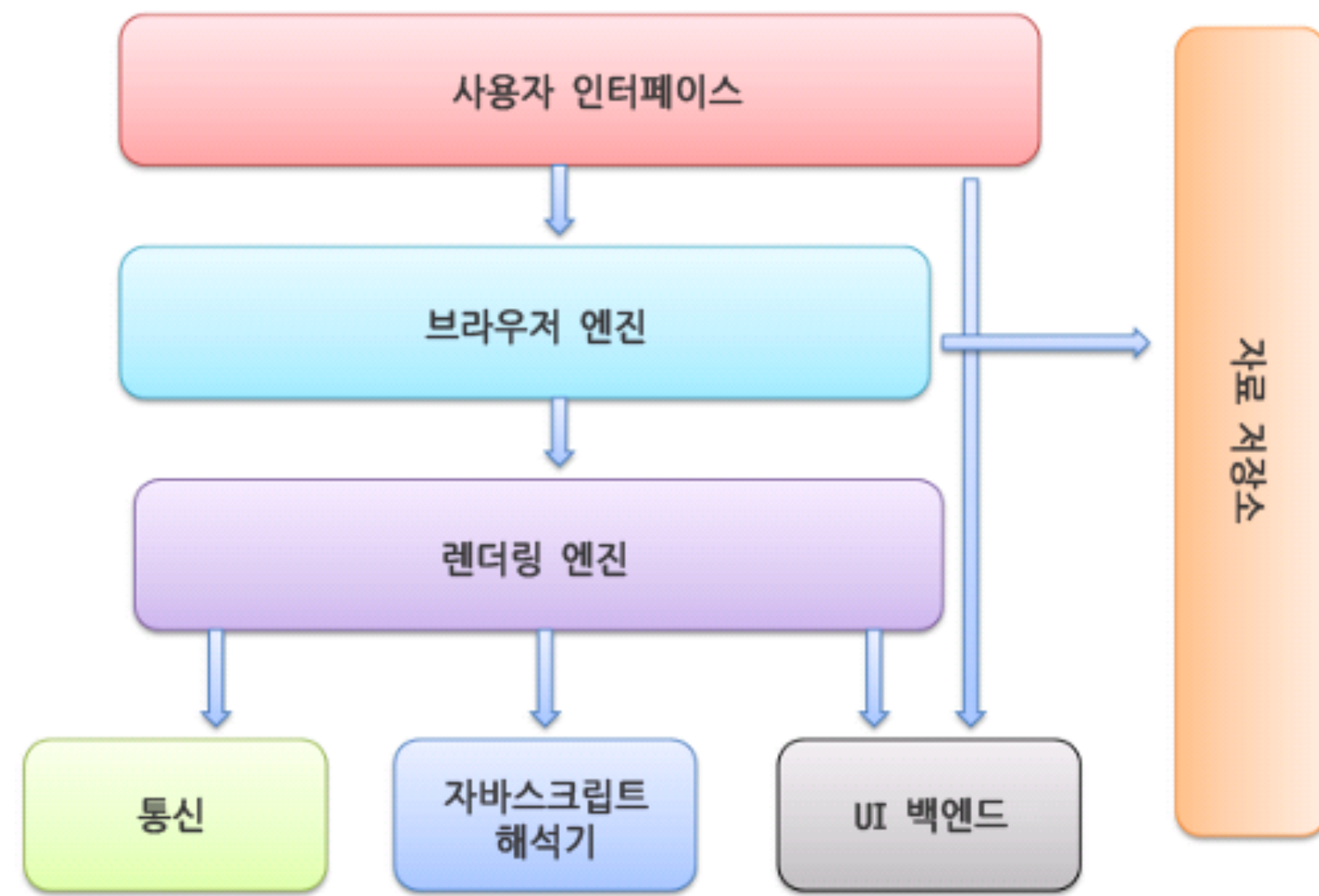
1. 개발 배경

(기존 시스템 문제 및 해결방안)

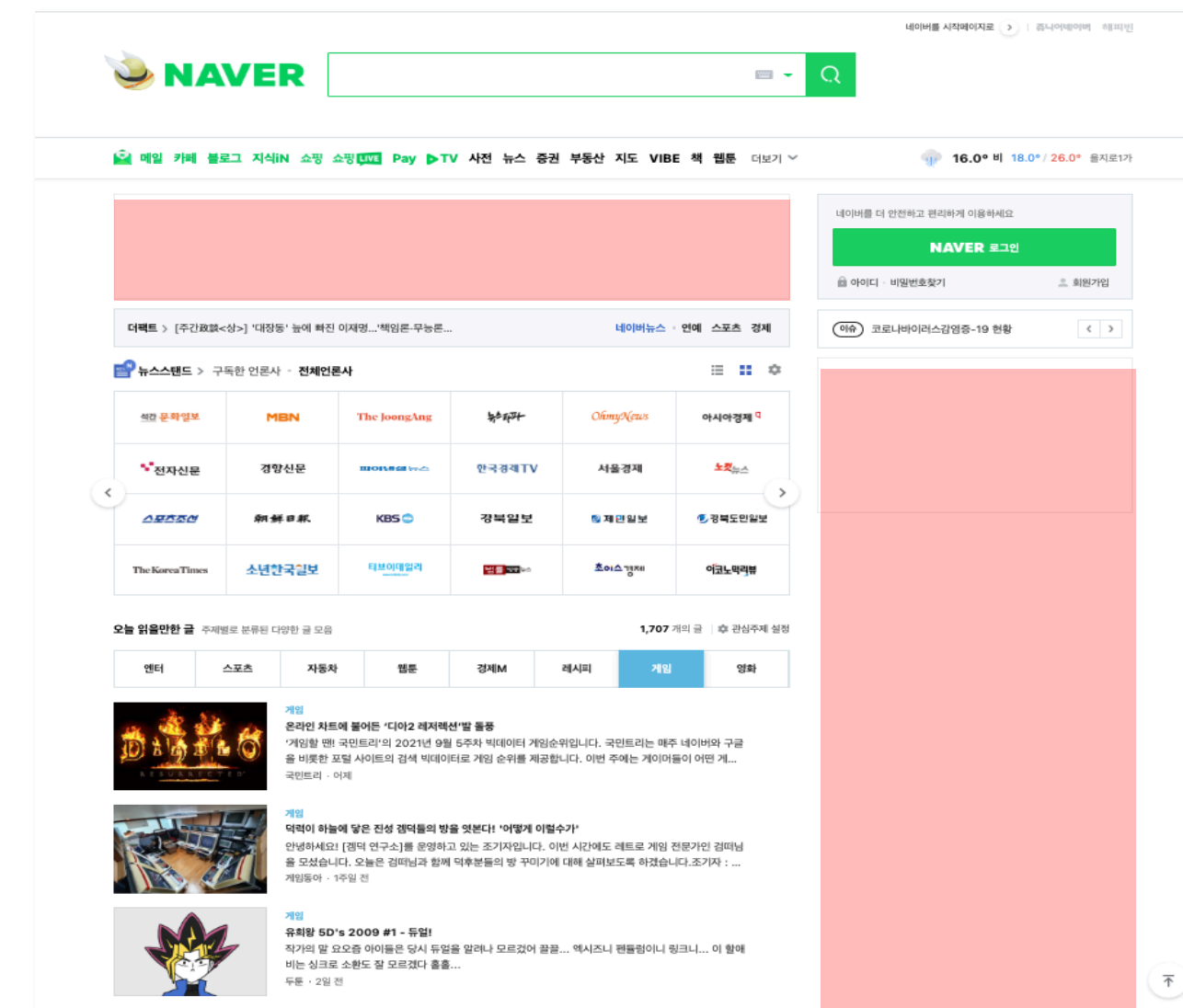
1.1 기존 시스템 문제 (1)

HTTP Client + Rendering engine

- 접속 상태는 빠르게 확인할 수 있지만 감지하지 못하는 영역 존재
- 자바스크립트 해석기가 없어 동적 콘텐츠 및 기능이 실행되지 않음



브라우저의 주요 구성 요소



HTML/CSS만 렌더링된 화면

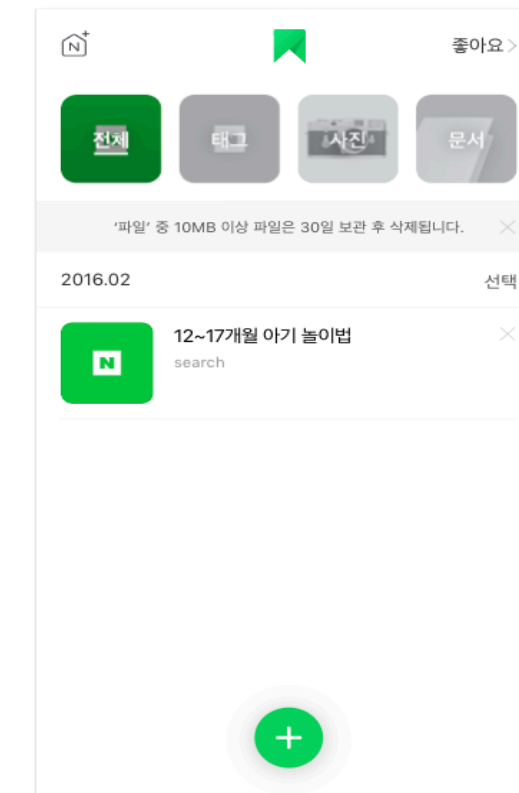
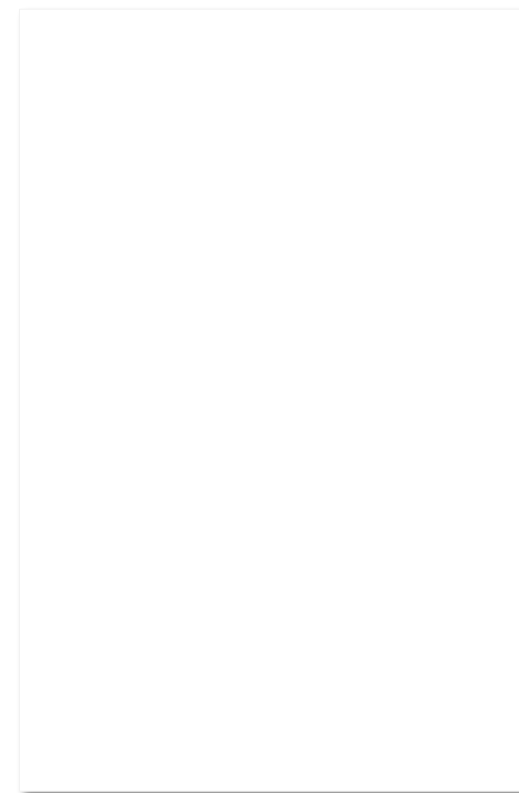
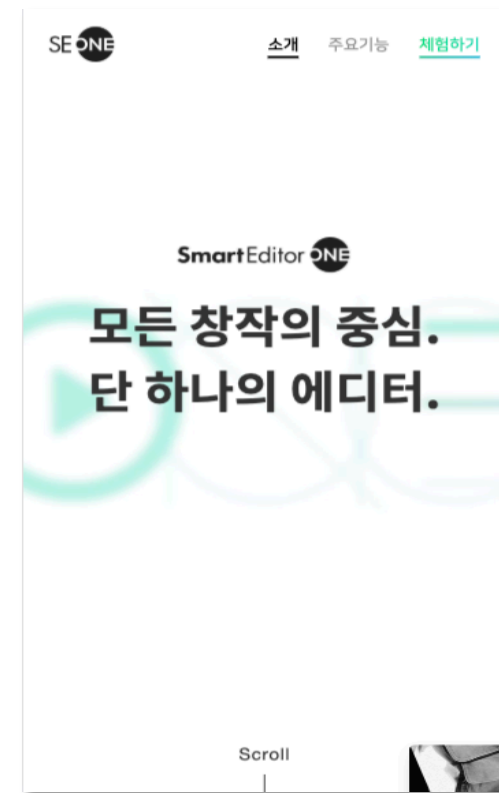
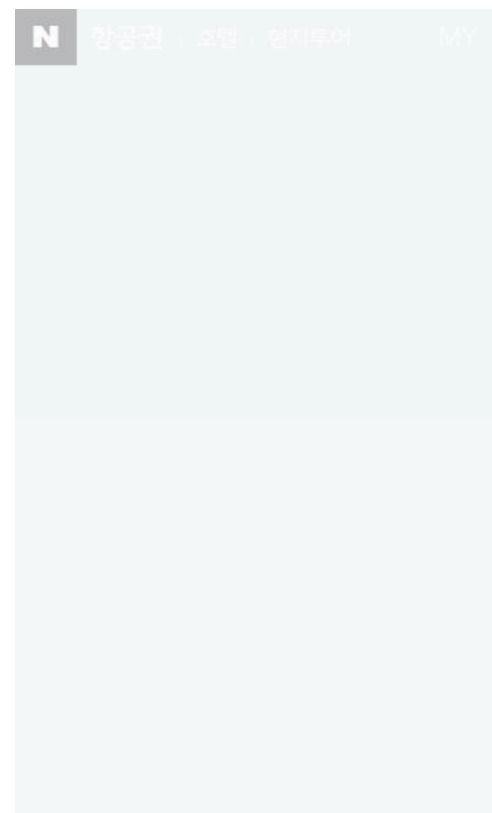
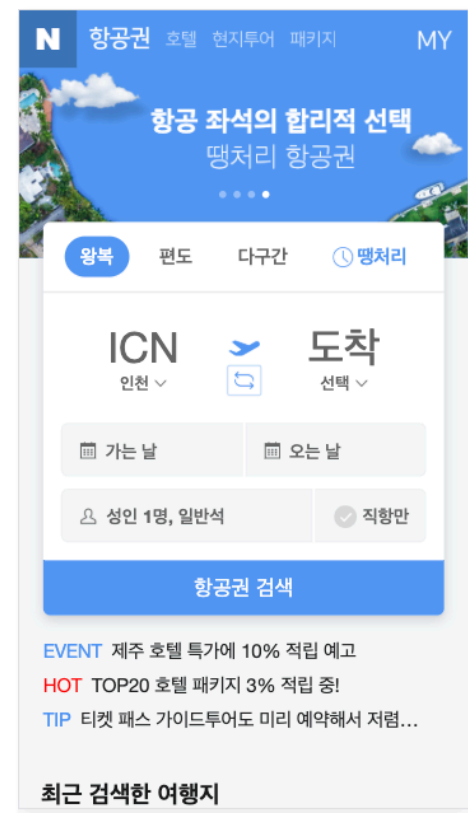
1.1 기존 시스템 문제 (2)

Headless browser.. PhantomJS.. 자주 이루어지지 않는 업데이트

- 브라우저 버전이 낮아 ES6 이상으로 개발된 페이지는 렌더링이 되지 않음
- 일반적인 브라우저에서 발생하지 않는 자바스크립트 오류가 검출됨

왼쪽 : 크롬

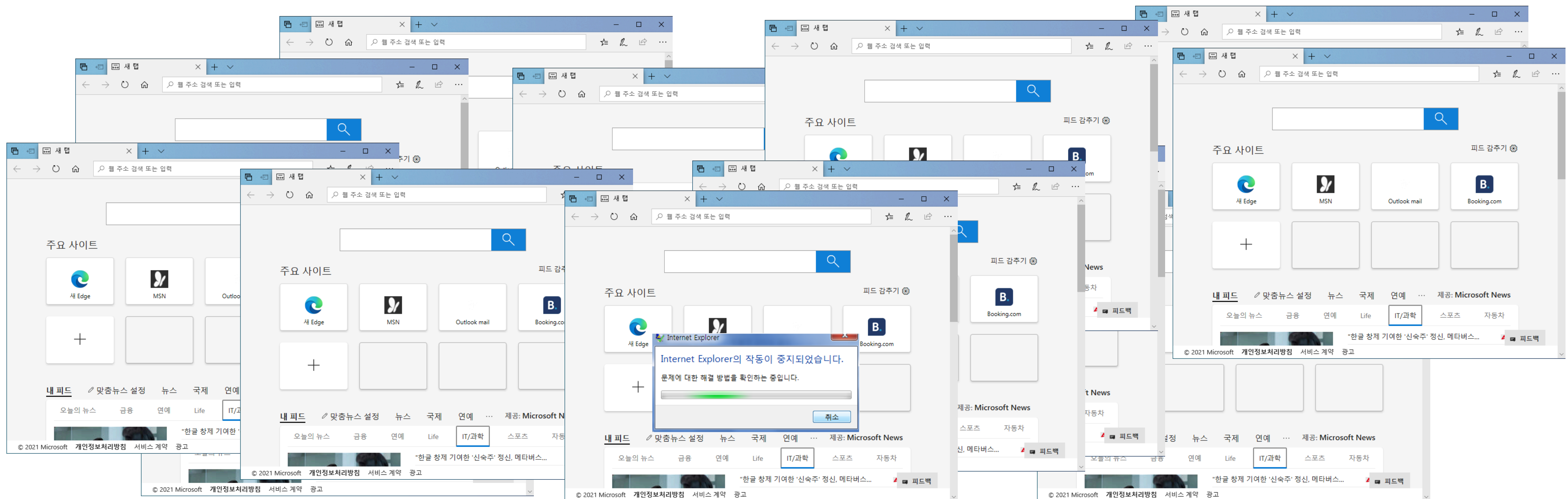
오른쪽 : 기존 시스템 모니터링 스크린샷



1.3 어떻게 해결할까

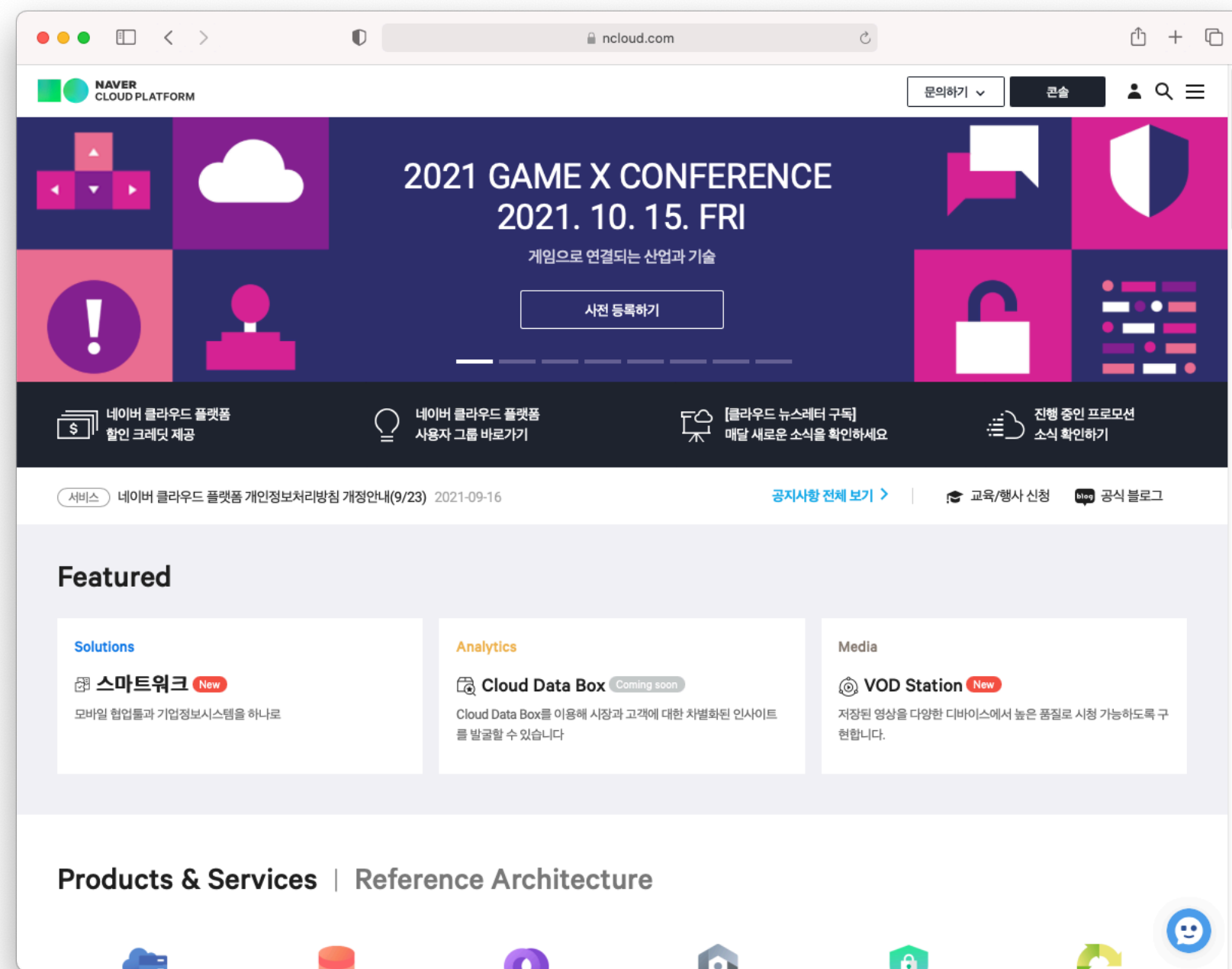
브라우저 사용하면 안되나요?

- 1분에 약 2만 4천 여개 URL과 기능 테스트를 포함한 시나리오를 실행해야 한다
- GUI 환경의 브라우저는 불필요한 리소스가 많다



1.3 어떻게 해결할까

Headless browser = GUI가 없는 웹 브라우저

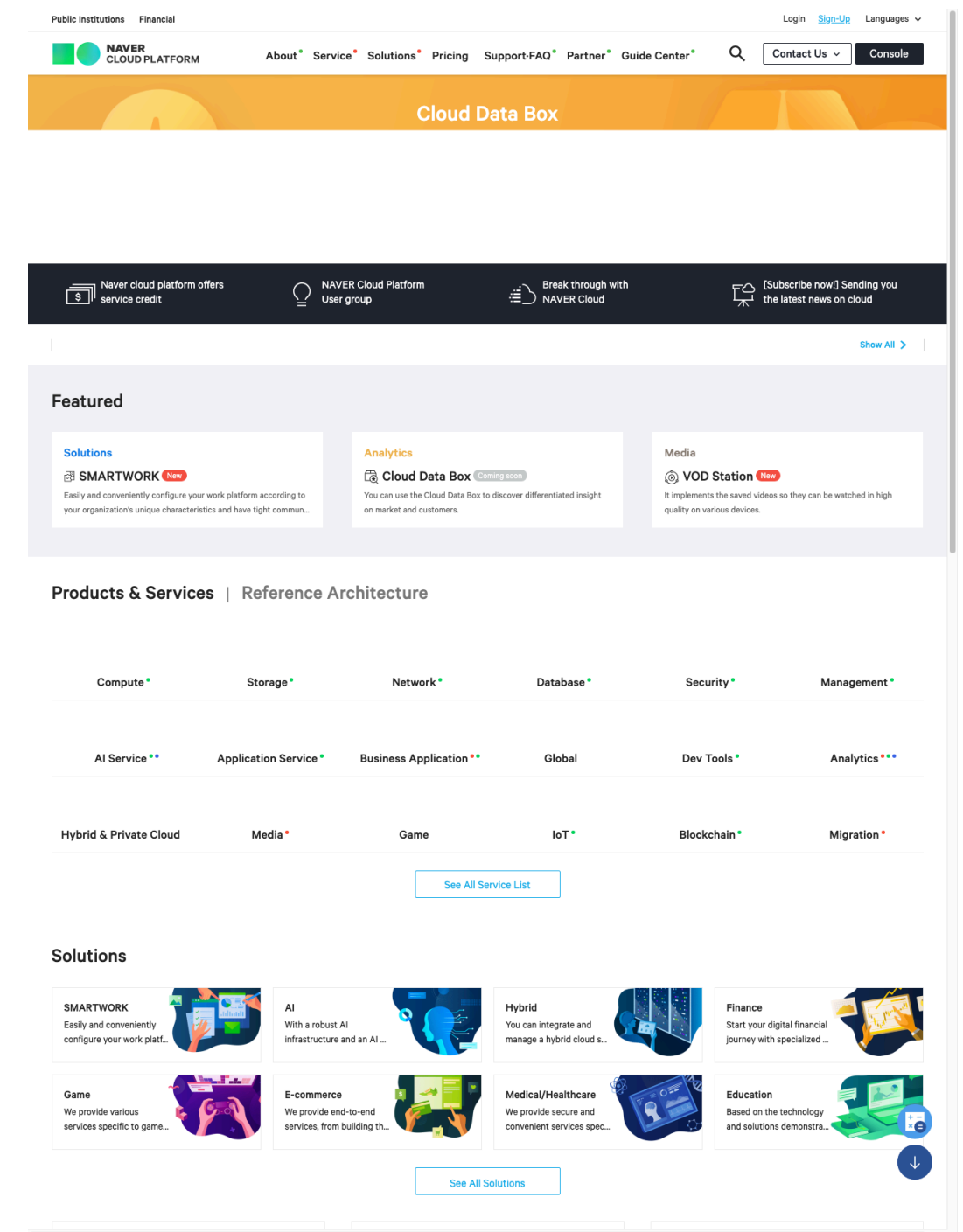


GUI 환경에서의 브라우저

```

Chrome
--headless
--disable-gpu
--screenshot
--window-size=1280,1696
https://www.ncloud.com
  
```

크롬 Headless 모드 실행

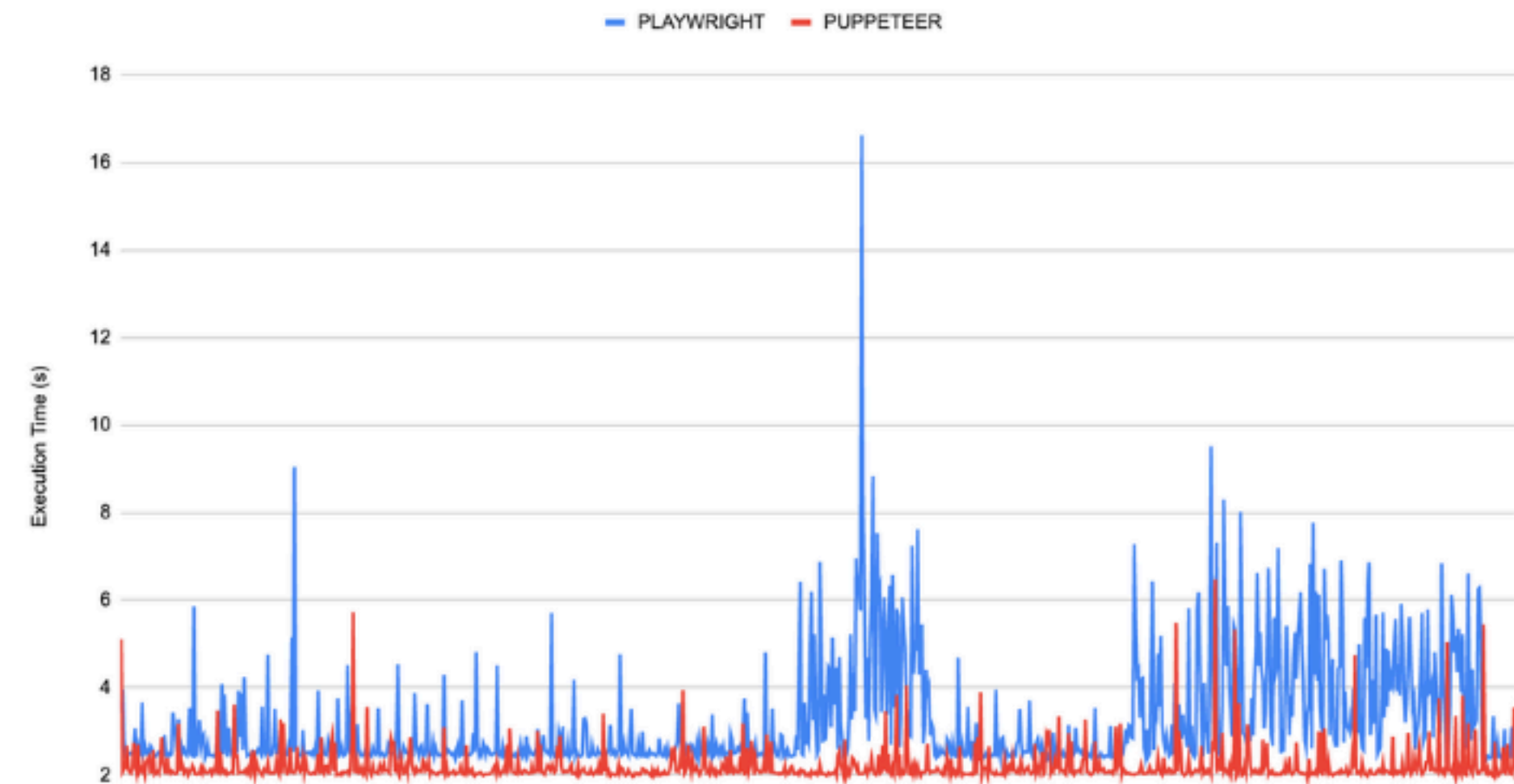


screenshot

1.3 어떻게 해결할까

Headless browser 어떤걸 사용해야 하나

	HEADLESS			
	PLAYWRIGHT	PUPPETEER	WDIO SELENIUM	WDIO DEVTOOLS
Mean (s)	3.194	2.218	3.656	3.634
SD (s)	1.230	0.412	0.653	0.466
Cv	0.385	0.186	0.179	0.128
P95 (s)	5.783	2.883	4.692	4.416



출처 : <https://blog.checklyhq.com/puppeteer-vs-selenium-vs-playwright-speed-comparison/>

- 벤치마크 결과 Puppeteer가 가장 빠르고 편차 없이 안정적으로 결과를 수집
- 하지만 각 라이브러리 마다 장단점이 있으니 목적에 따라 선택 필요

1.4 테스트 시나리오 관리의 어려움

문제1) 코드로 되어 있어 언어에 대한 이해 필요

문제2) Element을 제어 하려면 CSS Path를 알아야함

문제3) 시나리오가 어떤 목적으로 만들어 졌는지 파악이 어려움

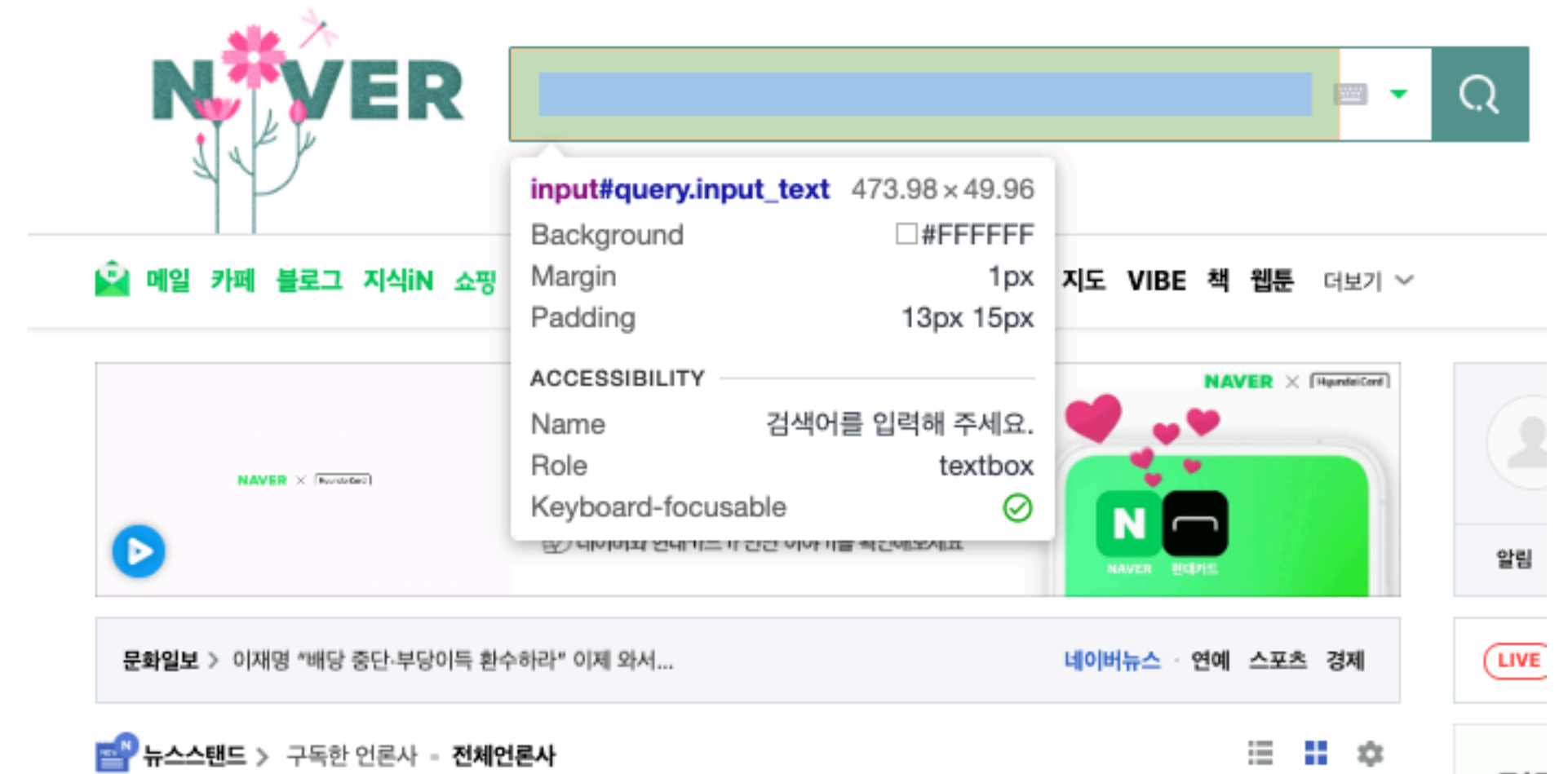
```

Action()
{
  // save the value of "session_id" cookie to "Session_
  web_req_save_params("Session_ID",
    "LB=Set-Cookie: session_id=",
    "RB=",
    "Ord=1",
    "RelFrameId=1",
    "Search=Headers",
    "IgnoreRedirections=Yes",
    LAST);

  web_url("login.php",
    "URL=www.pkrcurrency.com/login.php",
    "TargetFrame=",
    "Resource=0",
    "RecContentType=text/html",
    "Referer=",
    "Snapshot=t3.inf",
    "Mode=HTML",
    EXTRARES,
    "URL=/media/graphics/top_banner_bg.png", ENDITEM,
  );
}
  
```

```

27 retrieve = function() {
28   var url;
29   if (urls.length > 0) {
30     url = urls.shift();
31     urlIndex++;
32     page = webpage.create();
33     page.viewportSize = {
34       width: 800,
35       height: 600
36     };
37     page.settings.userAgent = "Phantom.js bot";
38     return page.open("http://" + url, function(status) {
39       var file;
40       file = getFilename();
41       if (status === "success") {
42         return window.setTimeout(function() {
43           page.render(file);
44           return next(status, url, file);
45         }, 200);
46       } else {
47         return next(status, url, file);
48       }
49     });
50   } else {
51     return callbackFinal();
52   }
53 };
54 return retrieve();
55 };
56
57 arrayOfUrls = null;
58
59 if (system.args.length > 1) {
60   arrayOfUrls = Array.prototype.slice.call(system.args, 1);
61 } else {
62   console.log("Usage: phantomjs render_multi_urls.js [domain.name1, domain.name2, ...]");
63   arrayOfUrls = ["www.google.com", "www.bbc.co.uk", "phantomjs.org"];
64 }
65
66 RenderUrlsToFile(arrayOfUrls, (function(status, url, file) {
67   if (status !== "success") {
68     return console.log("Unable to render '" + url + "'");
69   } else {
70     return console.log("Rendered '" + url + "' at '" + file + "'");
71   }
72 })), function() {
73   return phantom.exit();
74 });
  
```



1.4 운영 환경을 어떻게 개선해야 할까

문제1) 코드로 되어 있어 언어에 대한 이해 필요

- 브라우저 사용 경험만으로 시나리오를 작성 할 수 있도록 고도화

문제2) Element을 제어 하려면 CSS Path을 알아야함

- Inspect 기능을 시나리오 작성할 때 적용하여 CSS Path을 자동으로 검출

문제3) 시나리오가 어떤 목적으로 만들어 졌는지 파악이 어려움

- 시나리오 각 스텝마다 스크린샷을 남기고 제어하려는 Element을 강조

1.4 운영 환경을 어떻게 개선해야 할까

Classic / Web service Monitoring System / Web Monitoring

서비스 등록 URL의 상태를 테스트하고 주기적인 모니터링이 가능하도록 등록할 수 있습니다.

1 스텝 작성 2 서비스 설정 3 서비스 등록

테스트 환경 선택
스텝을 테스트할 환경을 선택 합니다.

모니터링 유형 • URL SCENARIO
모니터링 하실 유형을 선택하세요.

서비스 유형 • PC MOBILE
모니터링 하실 서비스 유형을 선택하세요.

지역 선택 • 국내 홍콩 일본 싱가포르 미국(서부) 독일
어느 환경에서 모니터링 할지 국가를 선택하세요.

스텝 작성 스텝 추가 ▼
스텝을 작성하고 테스트를 진행합니다.

URL 접속 ✓ 입력한 URL에 접속 합니다. 옵션 설정을 통해 Request Header, Body를 변경할 수 있습니다.

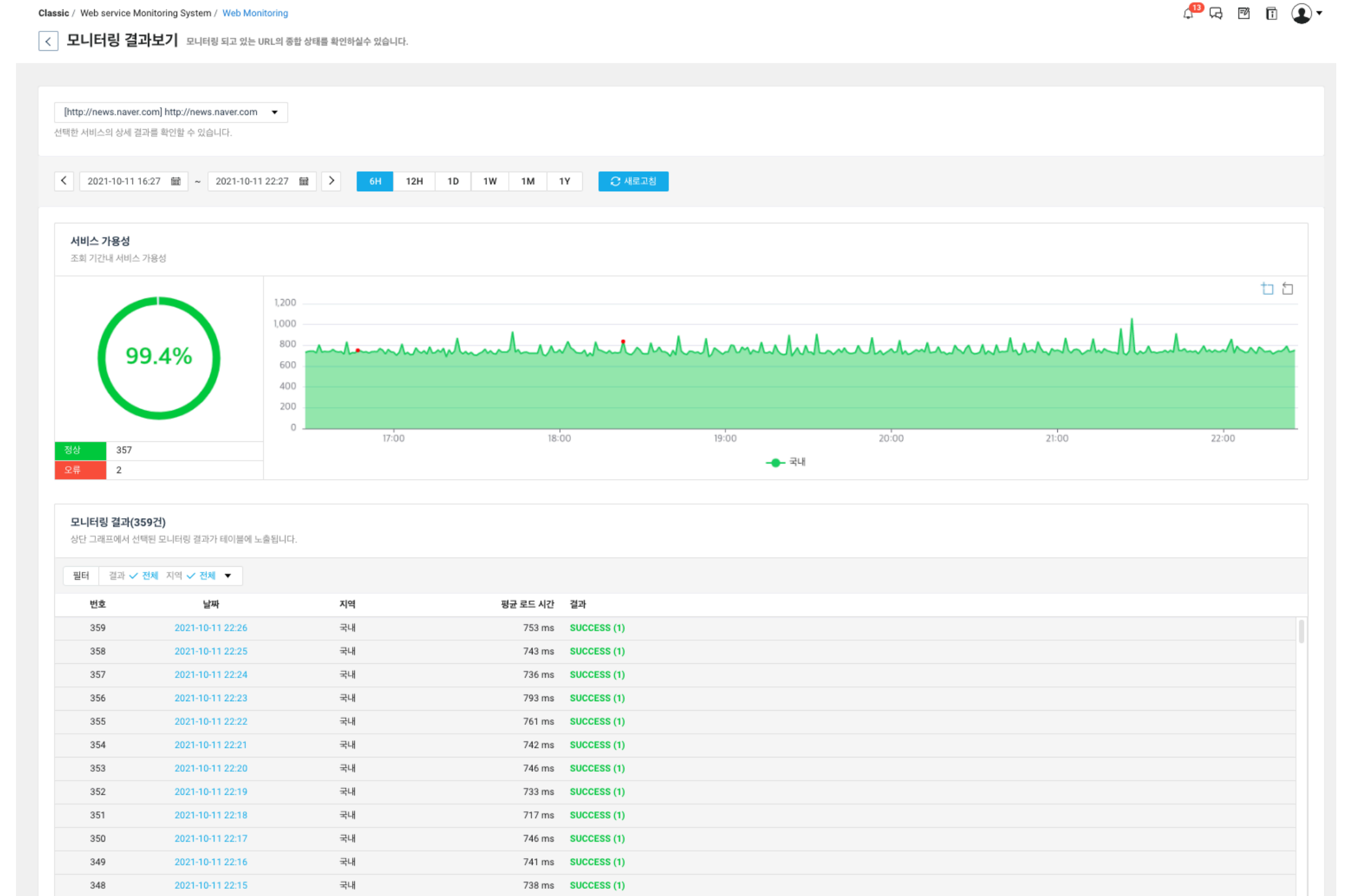
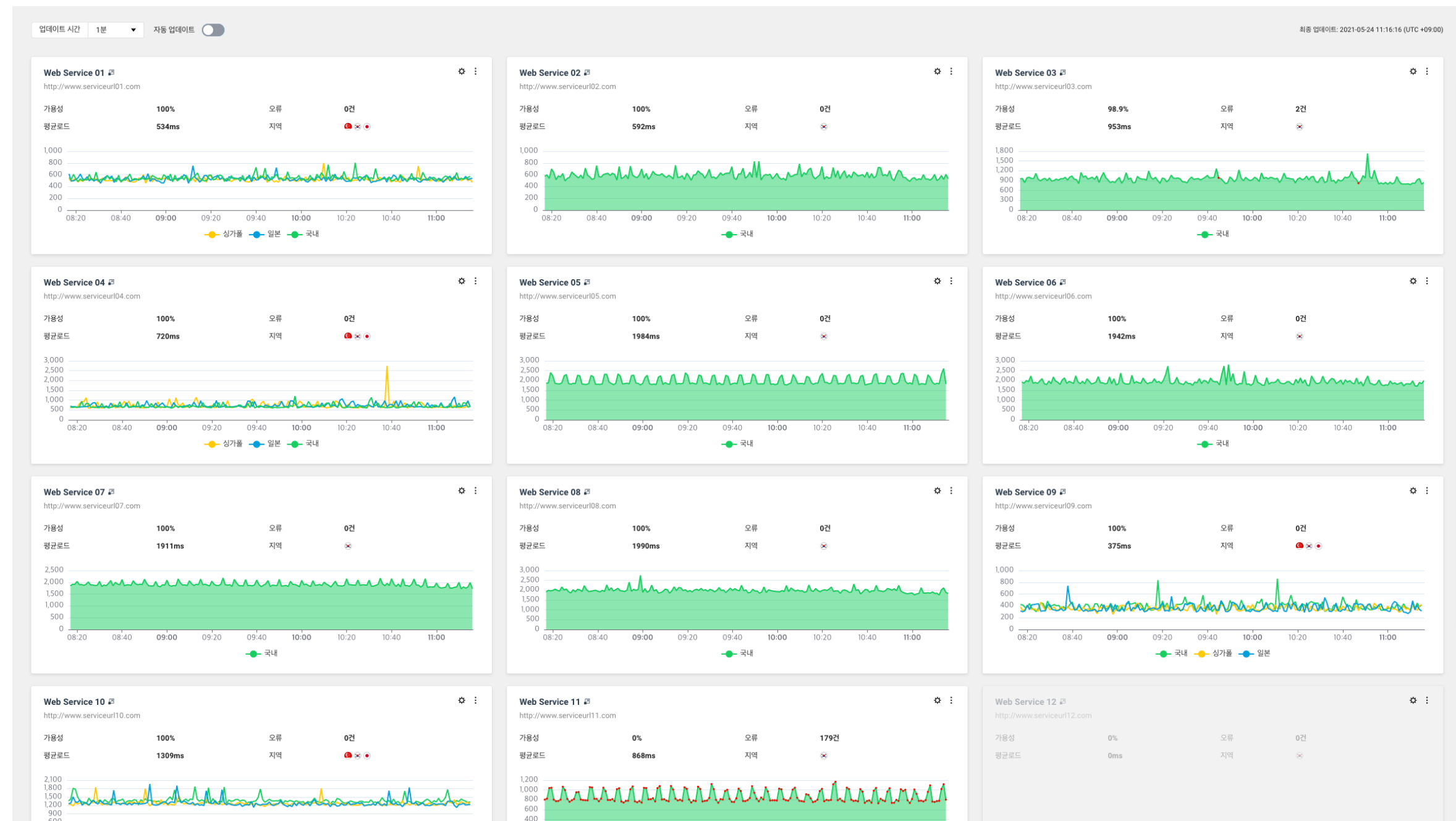
URL GET http://www.example.com 옵션

테스트 시작

2. WMS (Web service Monitoring System) 소개 및 구조

2.1 WMS(Web service monitoring system) 소개

서비스 별 모니터링 상태를 확인하고 상세 결과를 확인
 - 글로벌 모니터링 지원을 위해 해외에 에이전트를 구축하여 제공



네이버 클라우드 (Web service Monitoring System 상품 대시보드)

모니터링 결과 페이지

2.1 WMS(Web service monitoring system) 소개

테스트 시나리오의 각 스텝별 상세 결과와 Waterfall 및 스크린샷 제공

Classic / Web service Monitoring System / Web Monitoring

모니터링 결과보기 < 모니터링 되고 있는 URL의 종합 상태를 확인하실 수 있습니다.

모니터링 결과 상세

결과 목록
모니터링 결과 목록입니다.

- 2021-10-06 09:01 SUCCESS
- 2021-10-06 09:02 SUCCESS
- 2021-10-06 09:03 ERROR
- 2021-10-06 09:04 SUCCESS
- 2021-10-06 09:05 SUCCESS

에러 로그
모니터링 오류로그

시간	스텝 이름	유형	로그	Waterfall	이미지
2021-10-06 09:03	URL Connection	URL	404 - Not Found https://mimgnews.pstatic.n...		

모니터링 결과 상세
서비스 모니터링 결과를 확인할 수 있습니다.

URL Connection	841 ms	ERROR
Mouse click	96 ms	SUCCESS
Text input	250 ms	SUCCESS
Validation text	14 ms	SUCCESS

닫기

8	2021-10-06 08:55	국내	760 ms	SUCCESS (5)
7	2021-10-06 08:50	국내	749 ms	SUCCESS (5)
6	2021-10-06 08:45	국내	756 ms	SUCCESS (5)
5	2021-10-06 08:40	국내	758 ms	SUCCESS (5)

모니터링 상세 결과

Classic / Web service Monitoring System / Web Monitoring

모니터링 결과보기 < 모니터링 되고 있는 URL의 종합 상태를 확인하실 수 있습니다.

모니터링 결과 상세

Waterfall

기종: 2021/10/11 18:23 GMT+0900

URL	Method	Status	Type	Size	Time
https://news.naver.com	GET	200	document	224 KB	126 ms
https://news.naver.com	GET	200	stylesheet	270 KB	24 ms
https://ssl.gstatic.net/static/news/pnews/resources/20211006_191430/css/news.css	GET	200	stylesheet	237 KB	24 ms
https://ssl.gstatic.net/static/news/pnews/resources/20211006_191430/js/news.jvnav.js	GET	200	script	95 KB	25 ms
https://ssl.gstatic.net/static/news/pnews/resources/20211006_191430/js/news.service.js	GET	200	script	470 KB	18 ms
https://ssl.gstatic.net/static/news/pnews/resources/20211006_191430/css/common.css	GET	200	stylesheet	137 KB	23 ms
https://imgnews.pstatic.net/image/021/2021/10/11/005649554_001_2021101111226759.jpg?type=300_190	GET	200	image	15 KB	279 ms
https://imgnews.pstatic.net/image/021/2021/10/11/000725243_001_2021101117241349.jpg?type=300_190	GET	200	image	27 KB	218 ms
https://imgnews.pstatic.net/image/origin/022/2021/10/11/3627203.jpg?type=220_140	GET	200	image	13 KB	113 ms
https://imgnews.pstatic.net/image/origin/056/2021/10/11/1135119.jpg?type=220_140	GET	404	image	0 B	113 ms

확인

모니터링 중

URL CG Mouse Text Input Valid

경계

닫기

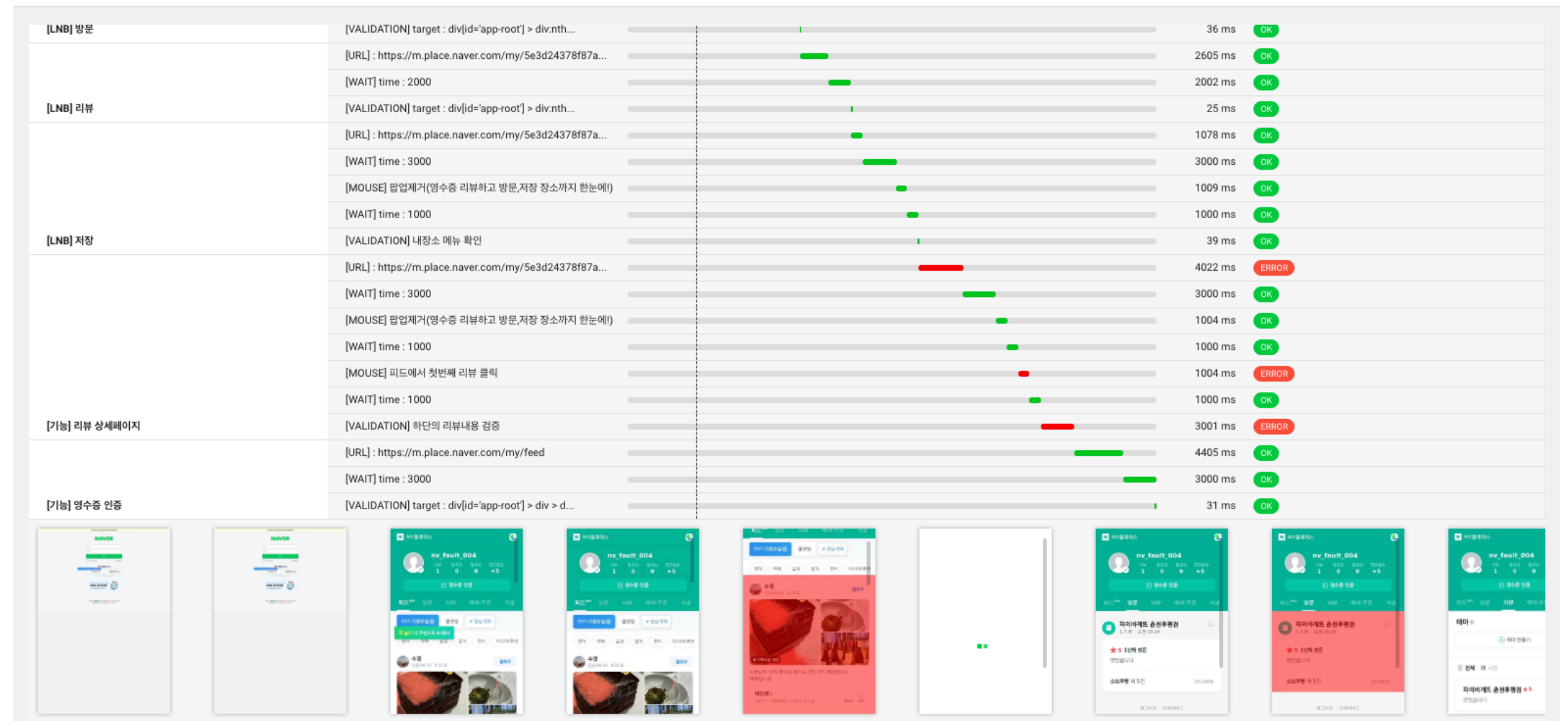
Waterfall & 스크린샷

2.1 WMS(Web service monitoring system) 소개

서비스에서 제공하는 전체 콘텐츠를 한눈에 볼 수 있도록 시나리오 운영

WMS 2.0

[HOME] 홈/로그인 (개인)				
[HOME][Mobile] 홈				
[LNB] 교육/행사 신청				
[LNB] 서비스				
[Sub] 소개 > 고객센터				x
[Sub] 소개 > 레퍼런스 아키텍처				
[Sub] 소개 > 서비스소개				
[Sub] 요금 > 요금계산기				
[Sub] 요금 > 요금소개				
[Sub] Hybrid & Private Cloud > VMware on Ncloud				
[Sub] 가이드센터 > API 참조서	x	x		x
[Sub] 가이드센터 > 설명서(구 사용자 가이드)	x	x		x
[Sub] 가이드센터 > 쉬운시작 가이드				
[Sub] 가이드센터 > 쉬운시작 가이드 > Object Storag...	x			
[Sub] 가이드센터 > 쉬운시작 가이드 > 리눅스 서버 사...				x
[Sub] 가이드센터 > 쉬운시작 프로젝트				
[Sub] 가이드센터 > 쉬운시작 프로젝트 > Jenkins 상...				
[Sub] 가이드센터 > 쉬운시작 프로젝트 > Object Stora...				
[Sub] 가이드센터 > 용어사전				
[Sub] 고객지원 > 공지사항				
[Sub] 고객지원 > 교육포탈				
[Sub] 고객지원 > 교육포탈 > 교육 카탈로그				
[Sub] 고객지원 > 교육포탈 > 교육신청				
[Sub] 고객지원 > 교육포탈 > 기술자격증				
[Sub] 고객지원 > 교육포탈 > 문의하기				
[Sub] 고객지원 > 교육포탈 > 소개				
[Sub] 고객지원 > 교육포탈 > 온라인교육				
[Sub] 고객지원 > 나의문의내역				
[Sub] 고객지원 > 문의하기				
[Sub] 고객지원 > 문의하기 > 영업문의				
[Sub] 고객지원 > 자주하는질문				
[Sub] 마이페이지 > 결제수단관리 - 개인계정로그인				
[Sub] 마이페이지 > 나의문의내역 - 개인계정로그인				
[Sub] 마이페이지 > 나의문의내역 - 서브계정로그인				
[Sub] 마이페이지 > 알림관리 - 서브계정로그인				
[Sub] 마이페이지 > 알림관리 > 추가하기버튼 - 개인계...				
[Sub] 마이페이지 > 알림관리 개인계정로그인				

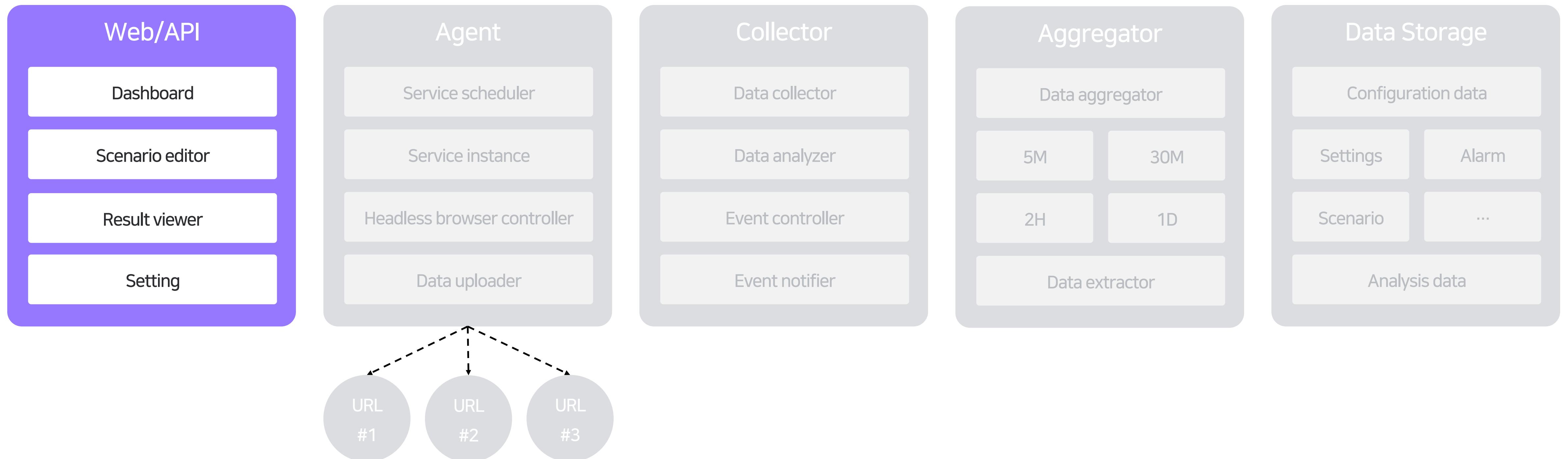


수집된 테스트 결과 상세 정보 조회 화면

2.2 WMS 구조 (Web/API)

테스트 시나리오 관리 및 모니터링 결과 제공

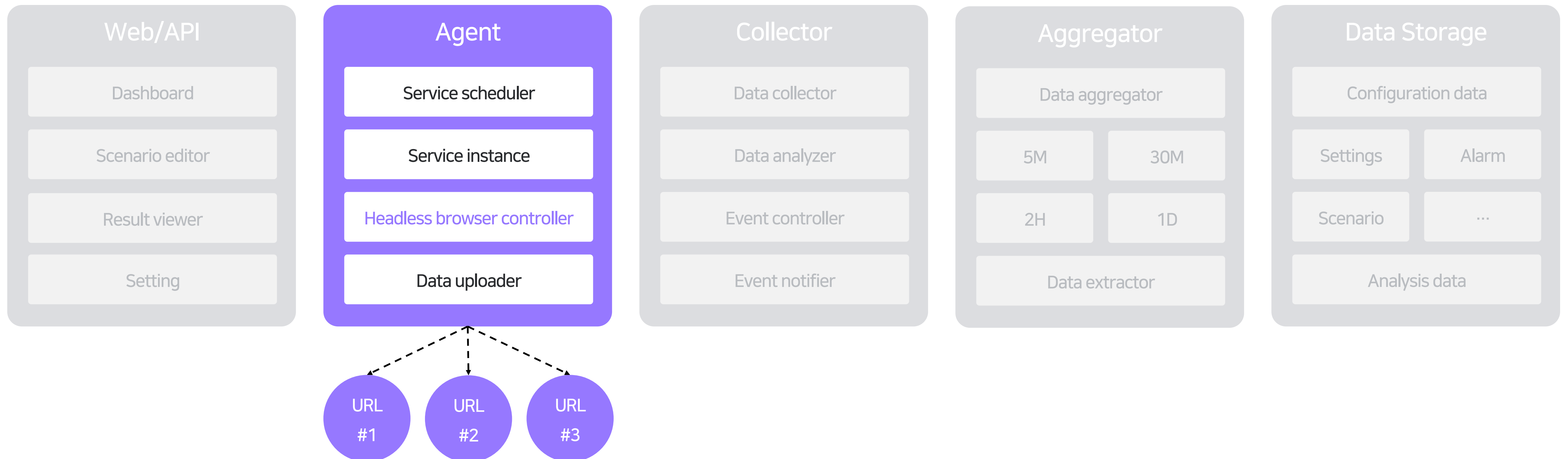
- 사용자는 Web을 통해 테스트 시나리오를 실행해 보고 생성 및 관리
- Dashboard를 통해 오류 감지 현황을 파악하고 Result viewer에서 상세 정보 조회



2.2 WMS 구조 (Agent)

Headless browser를 제어하며 등록된 테스트 시나리오를 실행

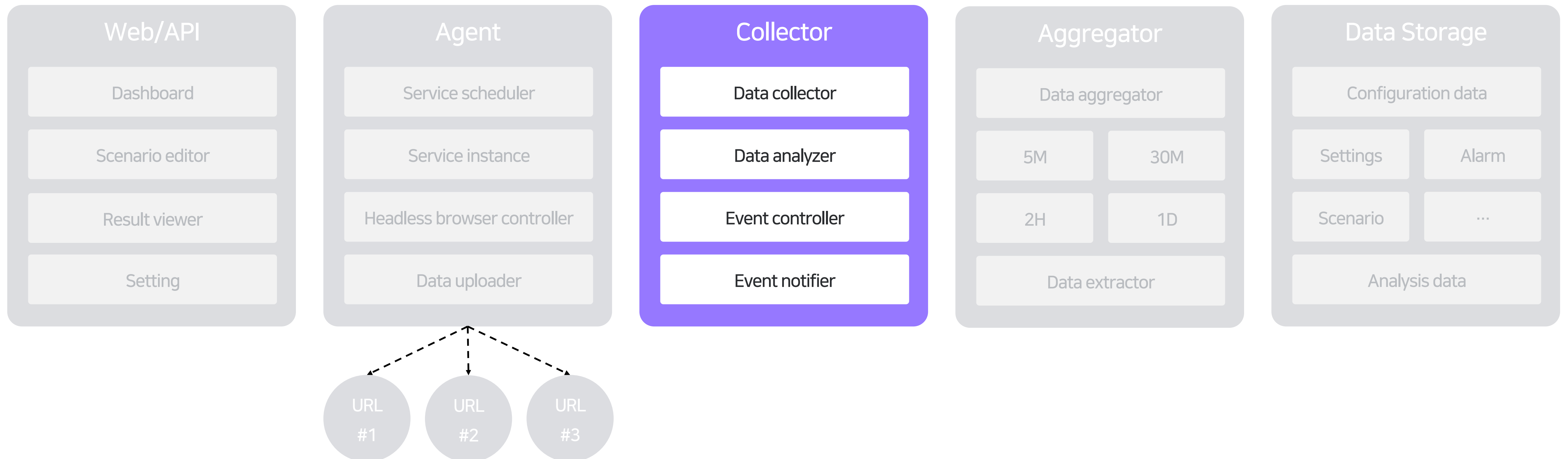
- 테스트 결과로 오류가 감지된 페이지 스크린샷, 네트워크 정보 (HAR), 오류 로그 등을 생성 하며 해당 데이터는 Collector로 전송



2.2 WMS 구조 (Collector)

테스트 결과를 수신하여 각 목적에 맞게 데이터 분류 및 저장

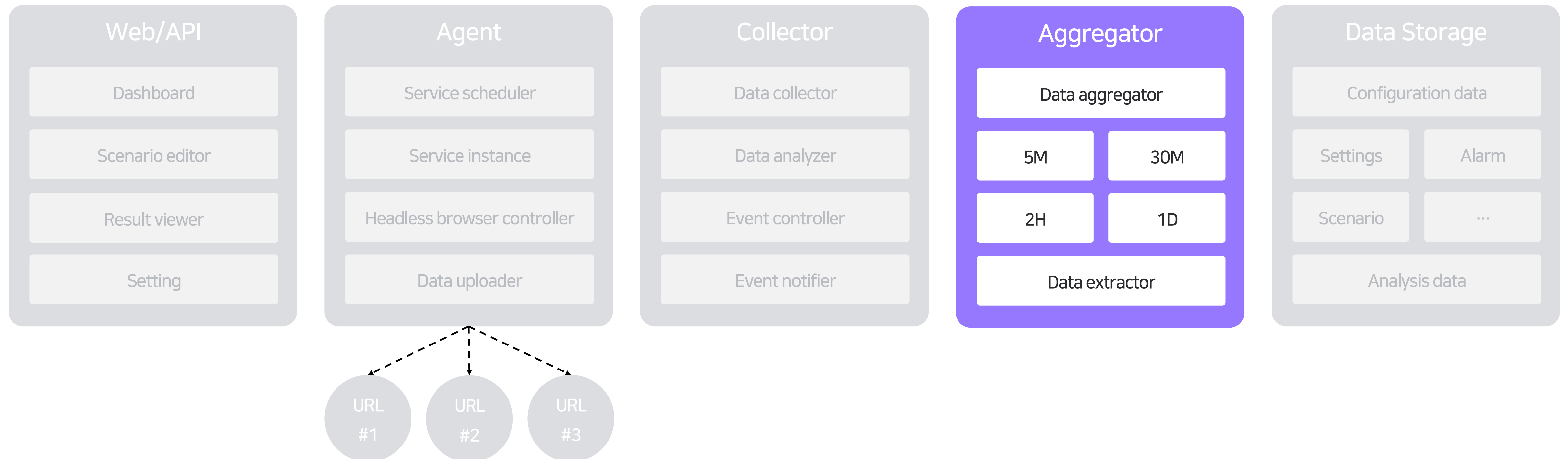
- 장애 감지 스크린샷, 네트워크 정보 (HAR), 테스트 시나리오 결과를 저장
- 시계열 데이터는 mongodb에, 스크린샷 및 네트워크 정보는 파일로 저장



2.2 WMS 구조 (Aggregator)

추세 및 추이 분석을 위한 데이터 Aggregation

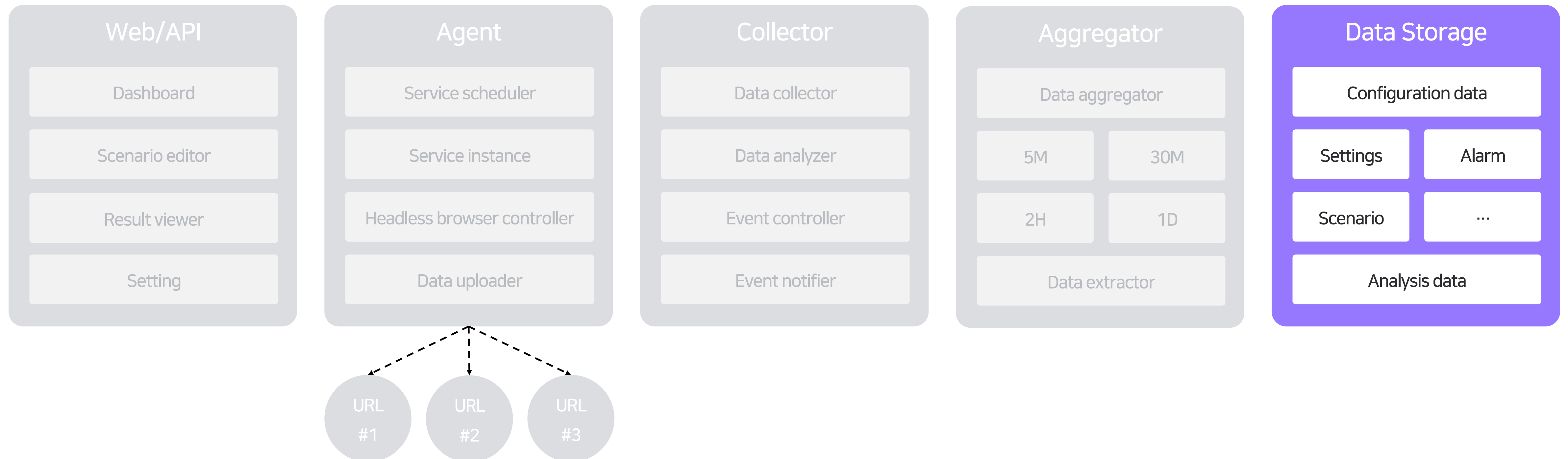
- 실시간 분석 뿐만 아니라 서비스 성능 추이를 분석하기 위함
- 예) 특정 시간대에 주기적으로 오류가 감지됨, 사용자 증가에 따른 서비스 성능 추이



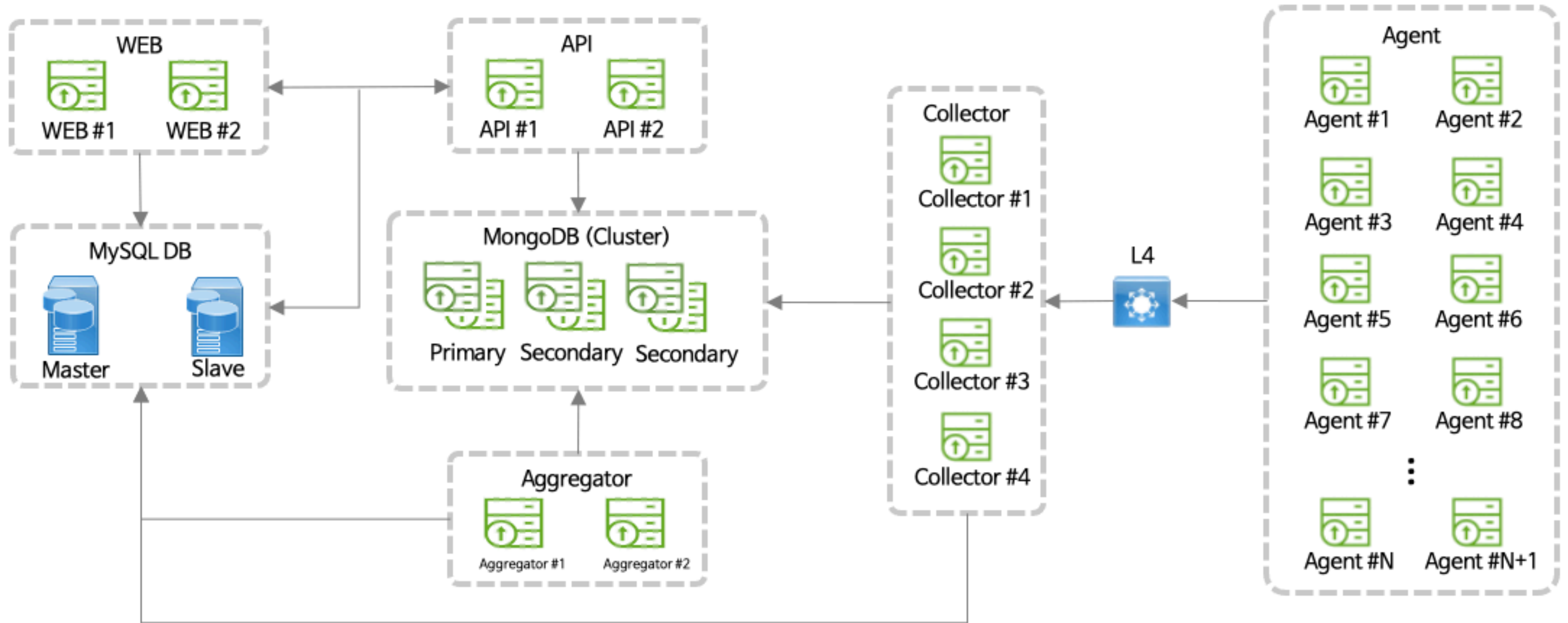
2.2 WMS 구조 (Data Storage)

MySQL과 Mongodb 활용

- 사용자, 시나리오, 설정 값등은 관계형 데이터는 MySQL에 저장
- 시계열 데이터는 Mongodb에 저장하고 Cluster 구성하여 확장성 고려



2.3 인프라 구조

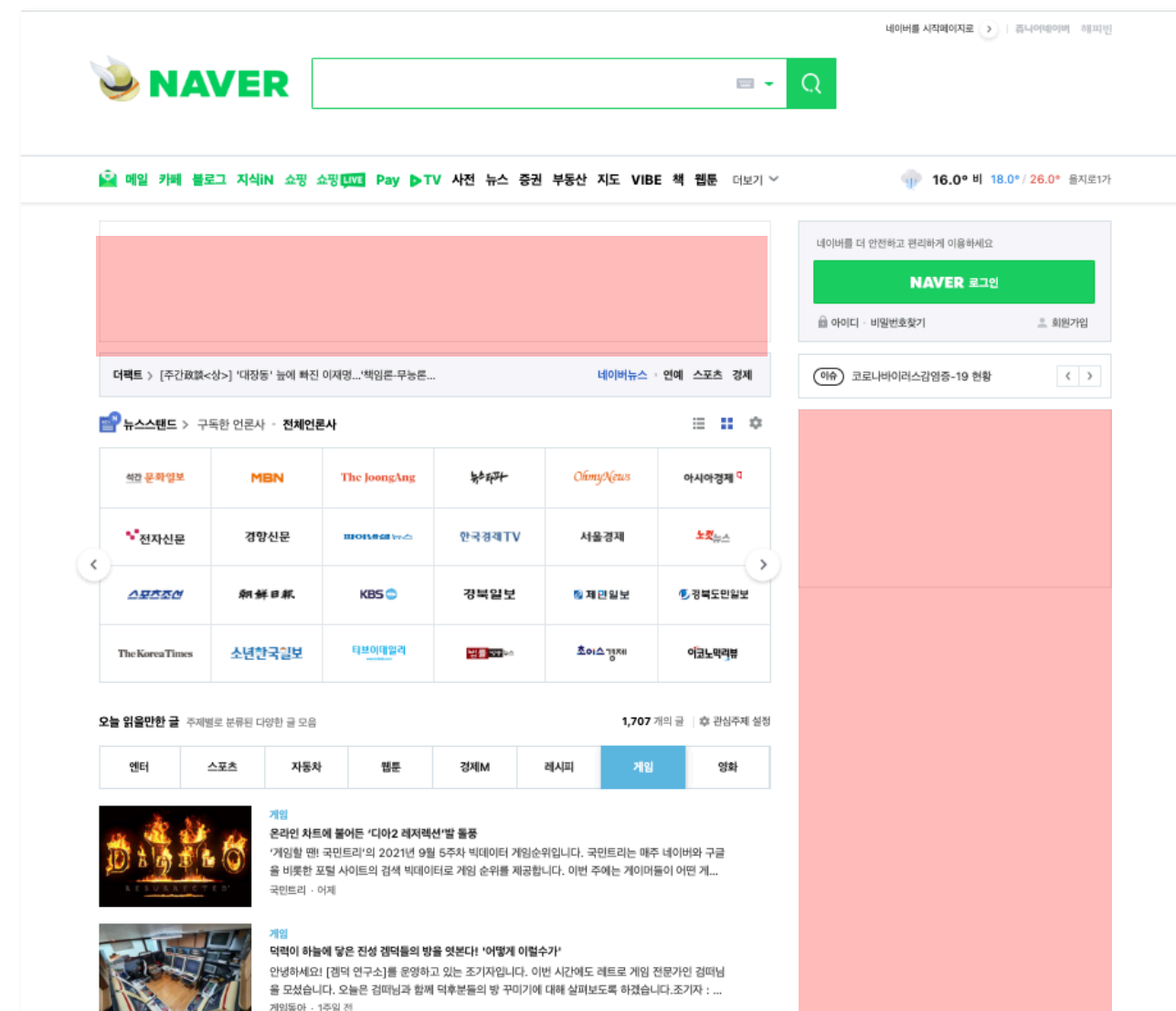


3. 웹 서비스 모니터링 기능 구현하기

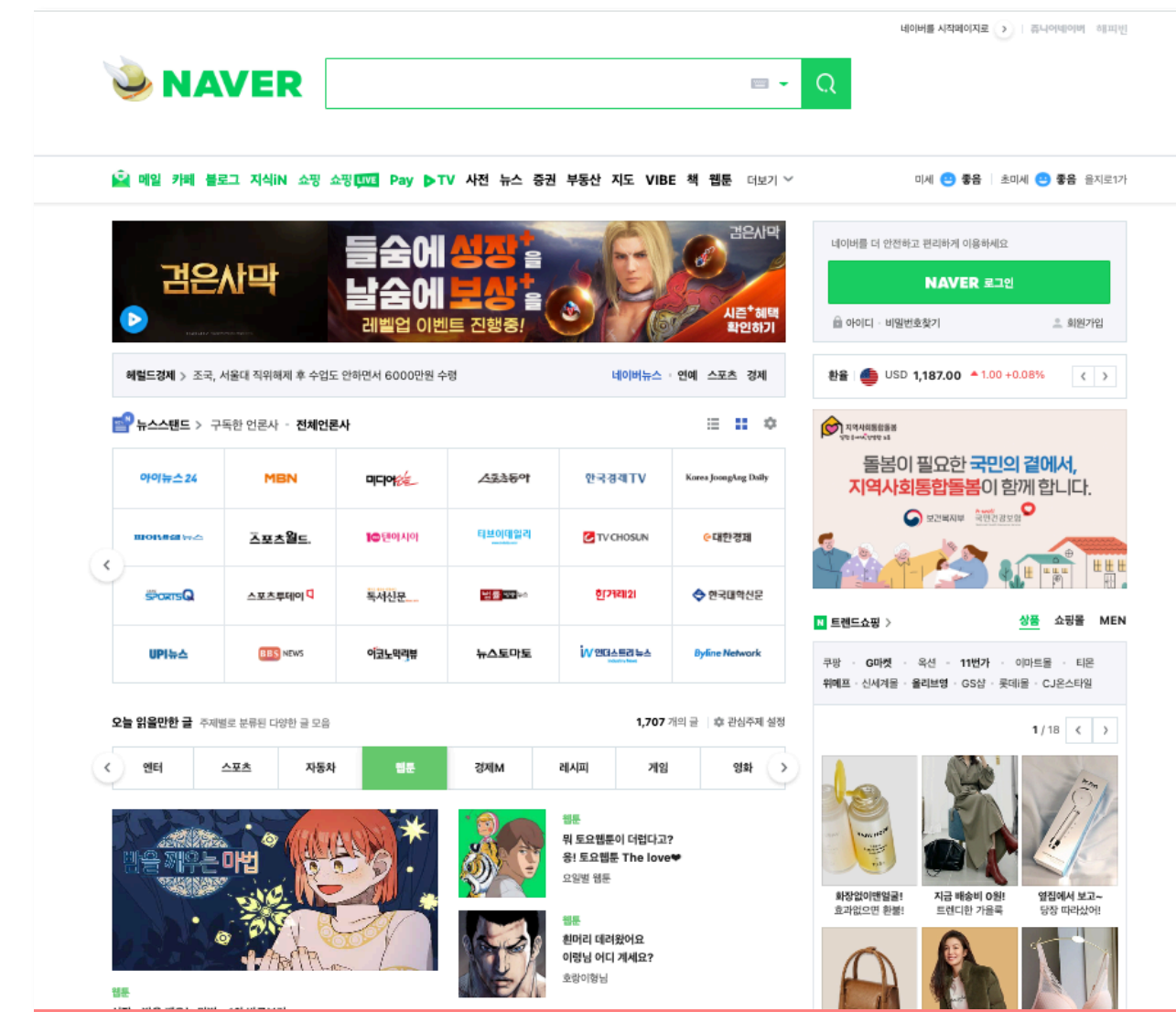
3.1 Puppeteer?

Puppeteer?

- Puppeteer 라이브러리는 Chrome 또는 Chromium의 렌더링 엔진을 사용하고 있다
- 기본적으로 Headless로 동작하며 non-headless로도 실행할 수 있다
- Headless browser는 GUI가 없는 브라우저로 CLI기반으로 동작한다



BeautifulSoup & Scrapy



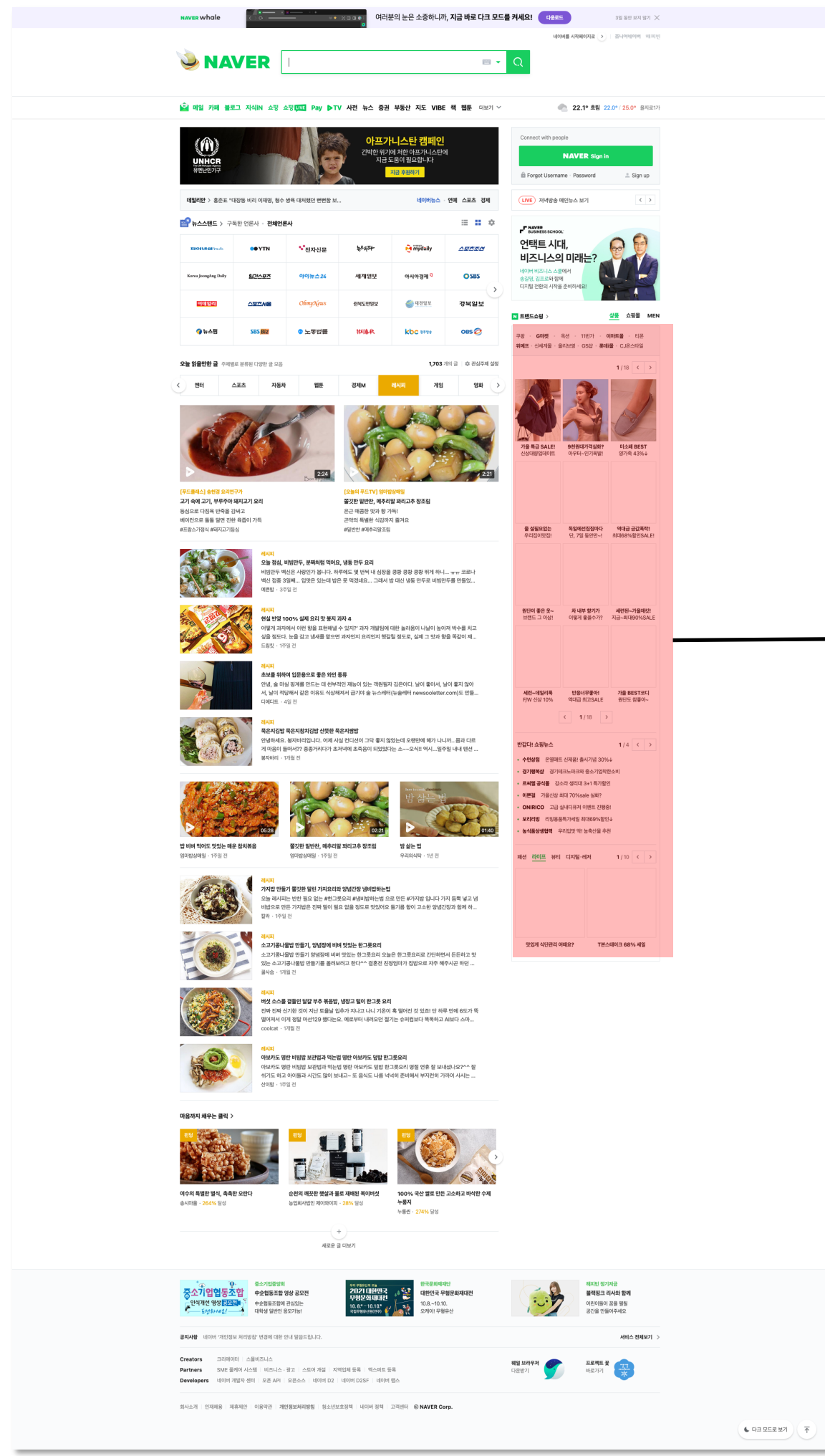
Puppeteer

3.2 구현하기 - URL 접속

```
const puppeteer = require('puppeteer');

(async () => {
  const browser = await puppeteer.launch();
  const page = await browser.newPage();
  await page.setViewport({ width: 1920, height: 1080 });
  await page.goto('https://www.naver.com');
  await page.screenshot({ path: 'sample.png', fullPage: true });
  await browser.close();
})();
```

서비스에 접속하고 스크린샷까지 결과물을 얻었다
 하지만 우측에 로드되지 않은 이미지들이 있다
 왜 그럴까?



sample.png

3.3 구현하기 - URL 접속 (완료 조건 정의)

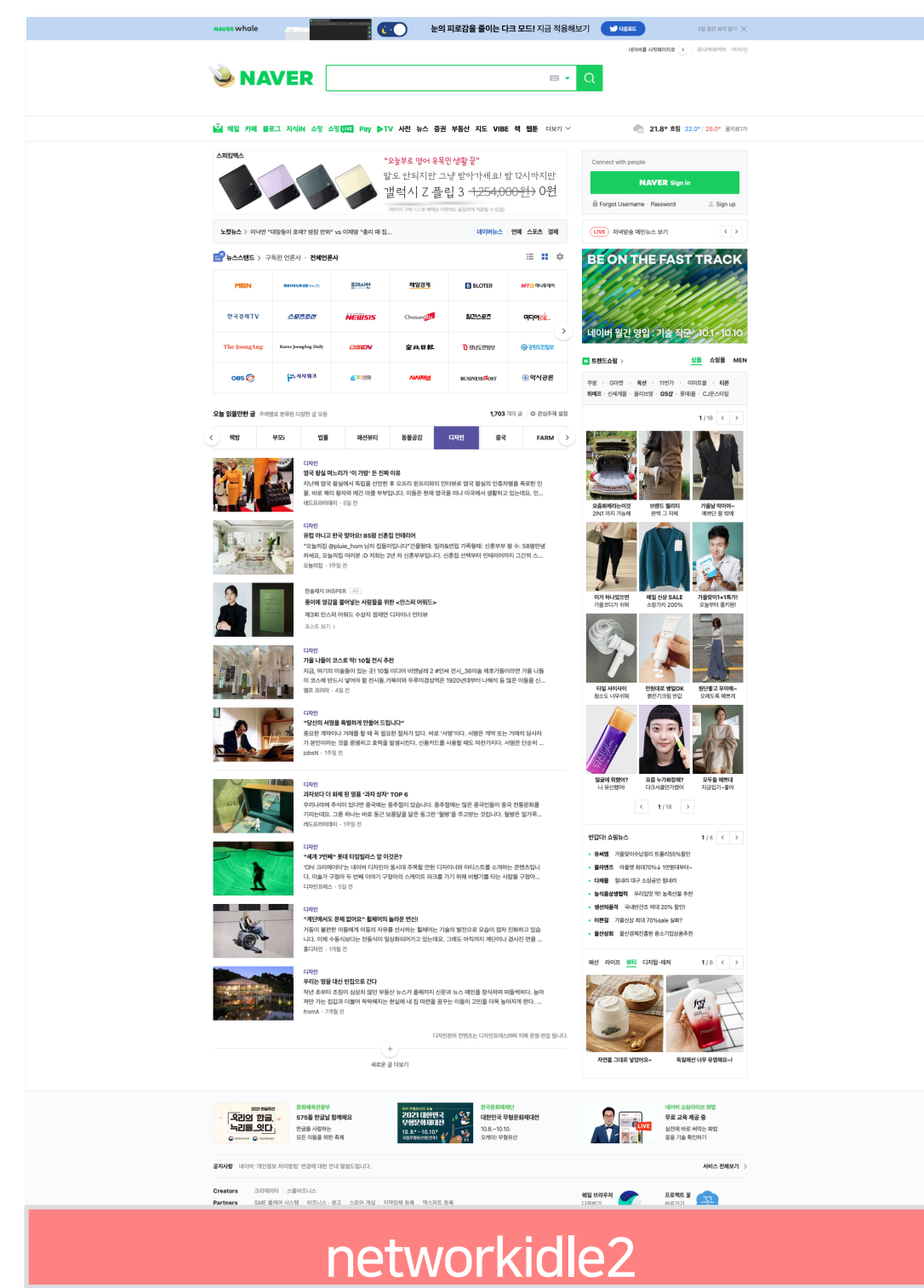
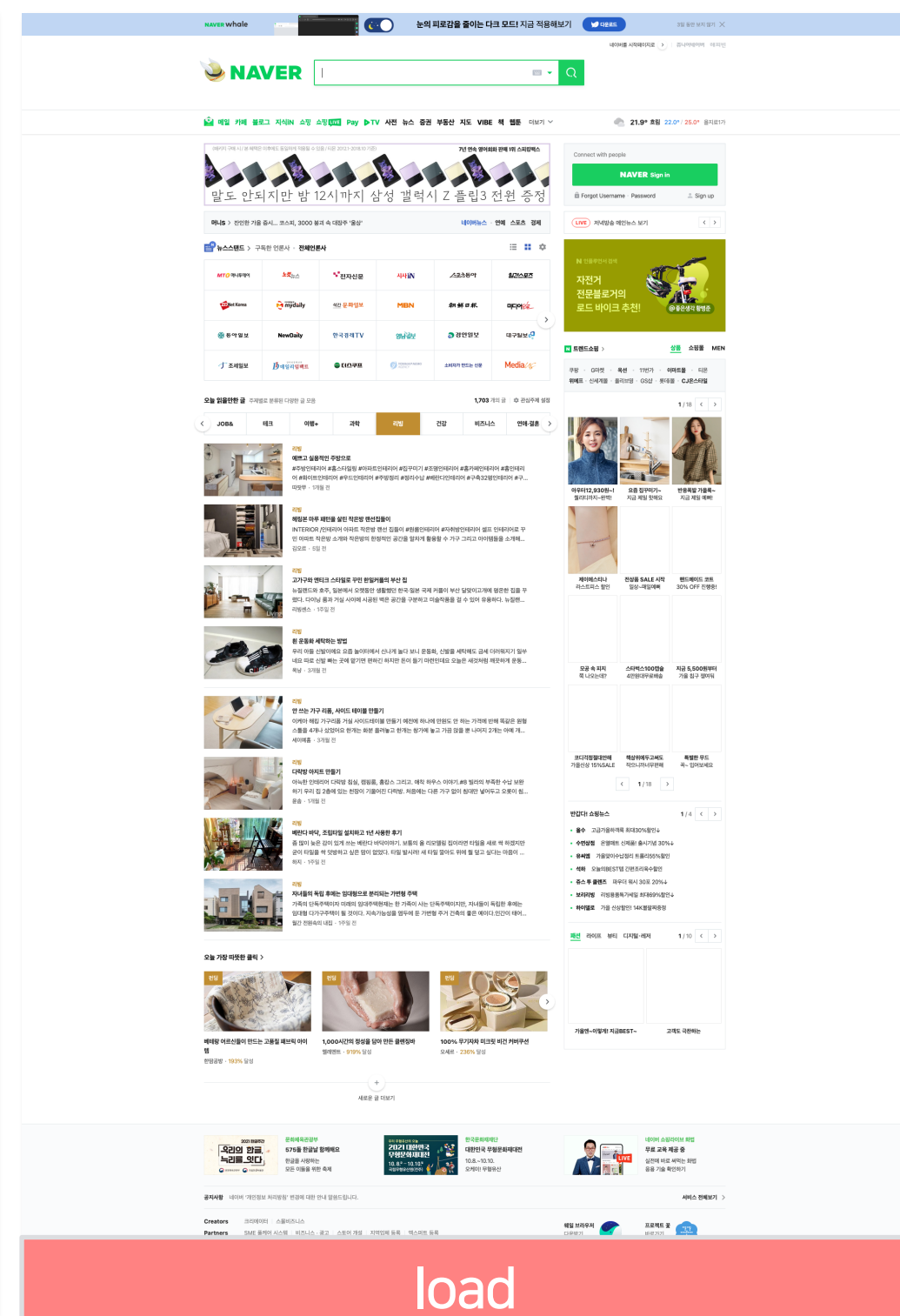
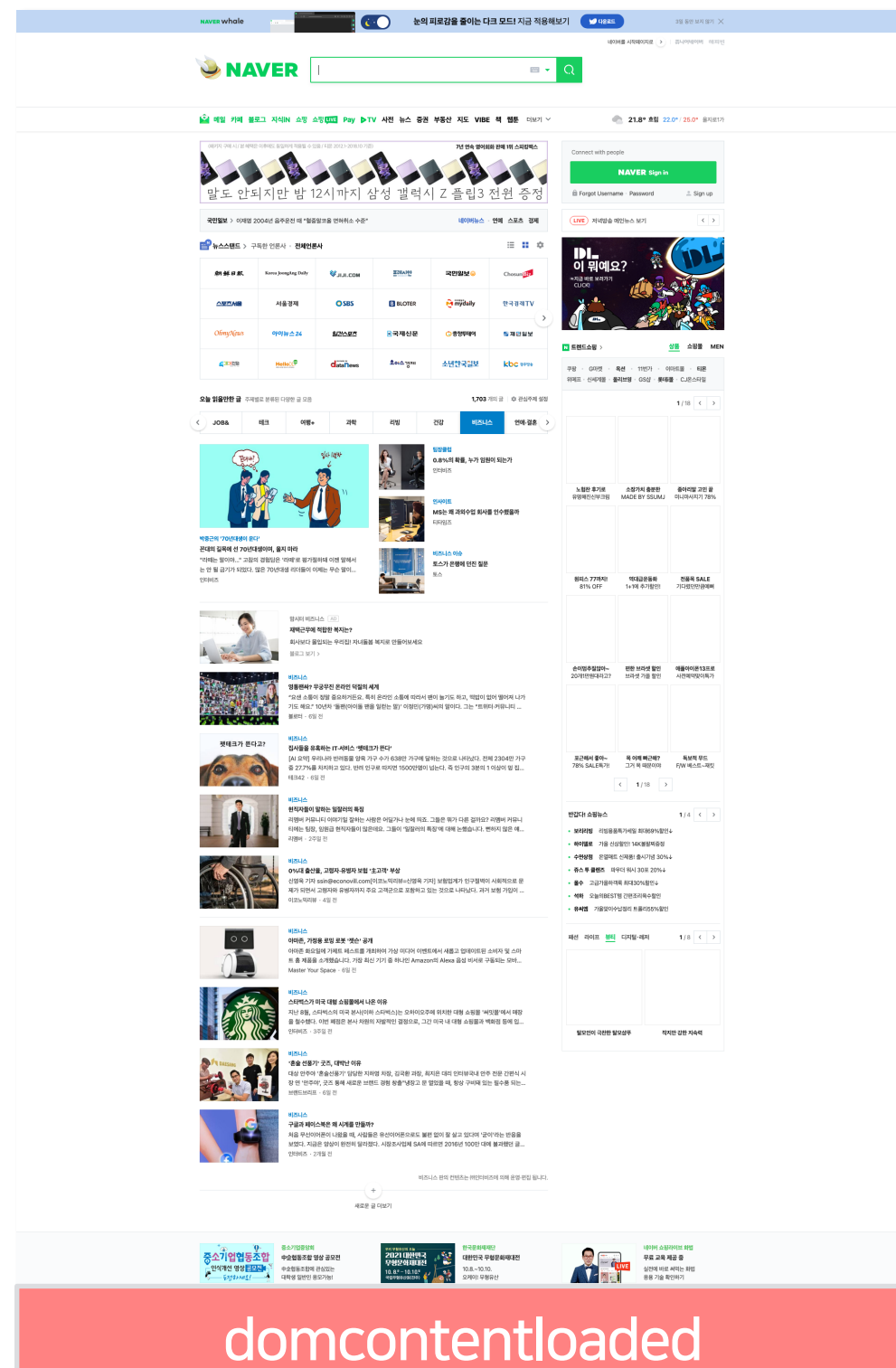
```
await page.goto('https://www.naver.com', {waitUntil: domcontentloaded});
await page.goto('https://www.naver.com', {waitUntil: load});
await page.goto('https://www.naver.com', {waitUntil: networkidle2});
await page.goto('https://www.naver.com', {waitUntil: networkidle0});
```

domcontentloaded
- HTML이 로드되고 DOM 트리가 생성완료 되었을 때

load
- HTML, DOM이 생성되고 외부 리소스도 모두 로드 되었을 때

networkidle2
- 500ms 이내에 완료되지 않은 요청이 2개 이하일 때

networkidle0
- 500ms 이내에 완료되지 않은 요청이 0개 일 때



3.4 구현하기 - URL 접속 (오류 감지)

잘못된 URL 접속 시도, 접속 지연 등 사이트 접속 장애 감지

- 서버로부터 응답이 없는 접속 오류 발생 시 예외 처리를 통해 감지해야 함

```
await page.goto('https://www.nave1r.com', {waitUntil: networkidle2, timeout: 1});
```



```
await page.goto('https://www.nave1r.com',
    {waitUntil: networkidle2, timeout: 1})
    .catch(error => {
        const resultDto = new resultDto();
        resultDto.setSuccessYn("N");
        resultDto.setErrorType("URL");
        resultDto.setErrorMessage(error.message);
    });
```

```
(node:58953) UnhandledPromiseRejectionWarning: Error: net::ERR_NAME_NOT_RESOLVED at https://www.nave1r.com
at navigate (/Users/user/devview/node_modules/puppeteer/lib/cjs/puppeteer/common/FrameManager.js:115:23)
at processTicksAndRejections (internal/process/task_queues.js:97:5)
at async FrameManager.navigateFrame (/Users/user/devview/node_modules/puppeteer/lib/cjs/puppeteer/common/FrameManager.js:90:21)
at async Frame.goto (/Users/user/devview/node_modules/puppeteer/lib/cjs/puppeteer/common/FrameManager.js:417:16)
at async Page.goto (/Users/user/devview/node_modules/puppeteer/lib/cjs/puppeteer/common/Page.js:825:16)
at async /Users/user/devview/example.js:11:5
(node:58953) UnhandledPromiseRejectionWarning: Unhandled promise rejection. This error originated either by throwing inside of an async function
without a catch block, or by rejecting a promise which was not handled with .catch(). To terminate the node process on unhandled promise reject
ion, use the CLI flag '--unhandled-rejections=strict' (see https://nodejs.org/api/cli.html#cli_unhandled_rejections_mode). (rejection id: 1)
(node:58953) [DEP0018] DeprecationWarning: Unhandled promise rejections are deprecated. In the future, promise rejections that are not handled w
ill terminate the Node.js process with a non-zero exit code.
```

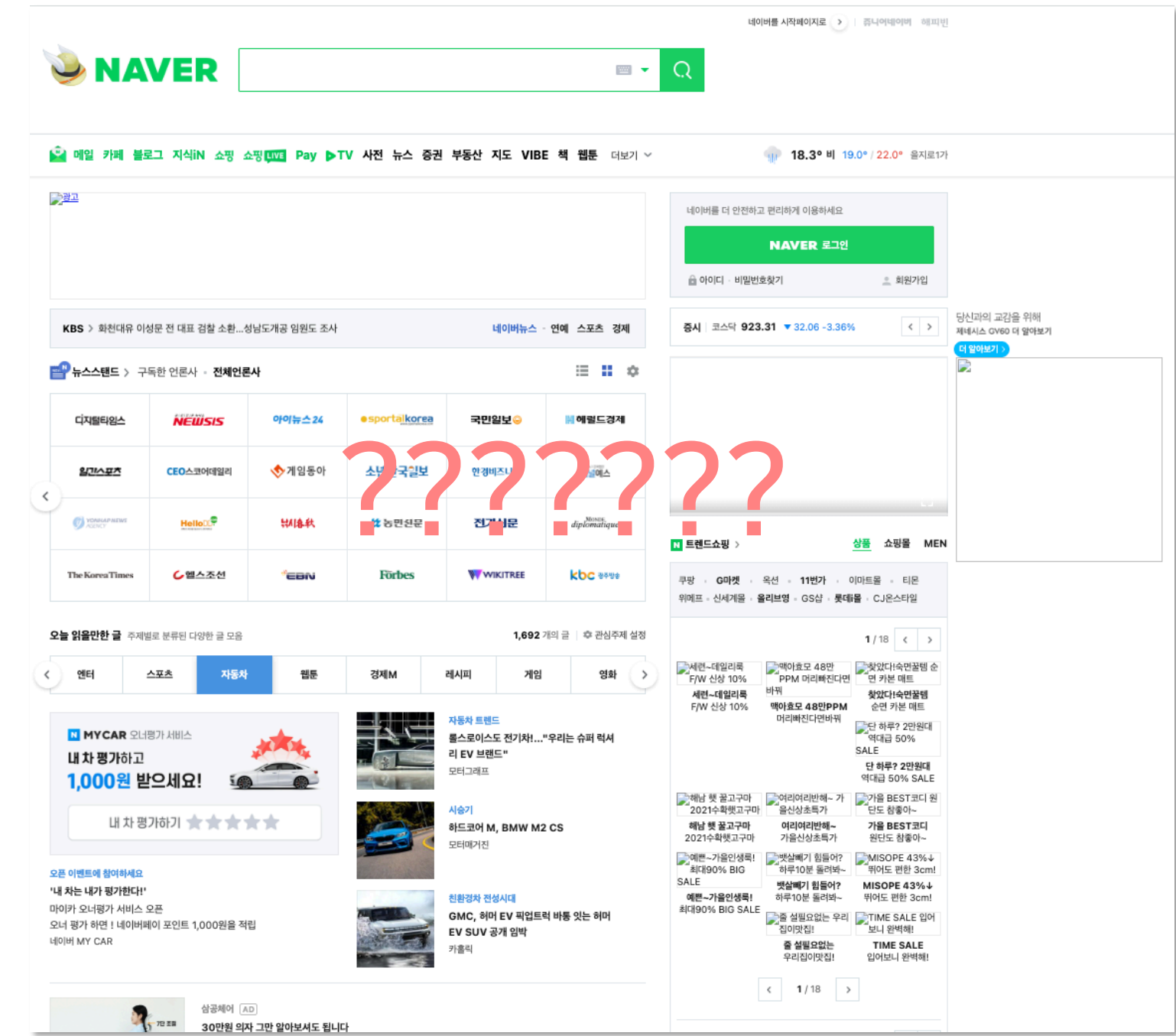
```
(node:59385) UnhandledPromiseRejectionWarning: TimeoutError: Navigation timeout of 1 ms exceeded
at /Users/user/devview/node_modules/puppeteer/lib/cjs/puppeteer/common/LifecycleWatcher.js:106:111
at async FrameManager.navigateFrame (/Users/user/devview/node_modules/puppeteer/lib/cjs/puppeteer/common/FrameManager.js:90:21)
at async Frame.goto (/Users/user/devview/node_modules/puppeteer/lib/cjs/puppeteer/common/FrameManager.js:417:16)
at async Page.goto (/Users/user/devview/node_modules/puppeteer/lib/cjs/puppeteer/common/Page.js:825:16)
at async /Users/user/devview/example.js:11:5
(node:59385) UnhandledPromiseRejectionWarning: Unhandled promise rejection. This error originated either by throwing inside of an async function
without a catch block, or by rejecting a promise which was not handled with .catch(). To terminate the node process on unhandled promise reject
ion, use the CLI flag '--unhandled-rejections=strict' (see https://nodejs.org/api/cli.html#cli_unhandled_rejections_mode). (rejection id: 1)
(node:59385) [DEP0018] DeprecationWarning: Unhandled promise rejections are deprecated. In the future, promise rejections that are not handled w
ill terminate the Node.js process with a non-zero exit code.
```

3.5 구현하기 - 4xx~5xx 오류 감지

이제 서비스 접속은 모니터링 할 수 있다 하지만 부족하다

- page.goto()는 navigation하는 URL에 대해서만 응답 코드를 반환한다
- 리소스 요청에 대한 오류는 response EventEmitter를 통해 구현해야 한다

page.goto("https://www.naver.com") → 응, 정상 →



3.5 구현하기 - 4xx~5xx 오류 감지

```

...
page.on('response', async res => {
  const httpStatusCode = res.status();
  const url = res.url();
  if (httpStatusCode >= 400) {
    const resultDto = new resultDto();
    resultDto.setSuccessYn("N");
    resultDto.setErrorType("URL");
    resultDto.setErrorMessage(` ${httpStatusCode} - ${url} `);
  }
});
...

```

Request 와 Response 정보는 EventEmitter를 통해 얻을 수 있다.
 이 정보를 활용하여 404 - Not found, 500 - Internal server error 등
 Http status code가 error인 경우를 검증할 수 있다.

```

[200] https://www.naver.com/
[200] https://ssl.pstatic.net/sstatic/search/pc/css/sp_autocomplete_210318.css
[200] https://ssl.pstatic.net/tveta/libs/assets/js/pc/main/min/pc.veta.core.min.js
[200] https://pm.pstatic.net/dist/css/nmain.20210826.css
[200] https://ssl.pstatic.net/tveta/libs/glad/prod/gfp-core.js
[200] https://pm.pstatic.net/dist/js/nmain.3aead730.js?o=www
[200] https://pm.pstatic.net/dist/js/external.5d9e21ac.js?o=www
[200] https://s.pstatic.net/static/newsstand/2020/logo/light/0604/076.png
[200] https://s.pstatic.net/static/newsstand/2020/logo/light/0604/018.png
[200] https://s.pstatic.net/static/newsstand/2020/logo/light/0604/308.png
[200] https://s.pstatic.net/static/newsstand/up/2020/0610/nsd151458769.png
[200] https://s.pstatic.net/static/newsstand/2020/logo/light/0604/904.png
[200] https://s.pstatic.net/static/newsstand/2020/logo/light/0604/330.png
[200] https://static-whale.pstatic.net/main/img_darkmode_v4@2x.png
[200] https://static-whale.pstatic.net/main/sprite-20210713@2x.png
[200] https://pm.pstatic.net/dist/js/vendors~more~nmain~sidebar_notice.031bbbfd.js?o=www
[200] https://pm.pstatic.net/dist/js/preload.59a14d58.js?o=www
[200] https://pm.pstatic.net/dist/js/vendors~search.30b5d61b.js?o=www
[200] https://s.pstatic.net/static/www/img/uit/2021/sp_weather_time_5f2bbb.png
[200] https://pm.pstatic.net/dist/js/more~nmain~sidebar_notice.3987c2c7.js?o=www
[200] https://pm.pstatic.net/dist/js/polyfill_async.942eb7d6.js?o=www
[200] https://s.pstatic.net/static/newsstand/2020/logo/light/0604/277.png
[200] https://s.pstatic.net/static/newsstand/2020/logo/light/0604/327.png
[200] https://s.pstatic.net/static/newsstand/2020/logo/light/0604/009.png
[200] https://s.pstatic.net/static/newsstand/2020/logo/light/0604/057.png
[200] https://s.pstatic.net/static/newsstand/2020/logo/light/0604/293.png
[200] https://s.pstatic.net/static/newsstand/2020/logo/light/0604/038.png
[200] https://s.pstatic.net/static/newsstand/2020/logo/light/0604/002.png
[200] https://s.pstatic.net/static/newsstand/2020/logo/light/0604/032.png
[200] https://s.pstatic.net/static/newsstand/up/2021/0824/nsd115034872.png
[200] https://pm.pstatic.net/dist/js/search.6d032745.js?o=www
[200] https://s.pstatic.net/static/newsstand/2020/logo/light/0604/088.png
[200] https://s.pstatic.net/static/newsstand/2020/logo/light/0604/336.png
[200] https://s.pstatic.net/static/newsstand/2020/logo/light/0604/817.png
[200] https://s.pstatic.net/static/newsstand/2020/logo/light/0604/135.png
[200] https://s.pstatic.net/static/newsstand/up/2020/1228/nsd102538785.png
[200] https://s.pstatic.net/static/newsstand/2020/logo/light/0604/811.png
[200] https://s.pstatic.net/static/newsstand/2020/logo/light/0604/108.png
[200] https://s.pstatic.net/static/newsstand/2020/logo/light/0604/094.png
[200] https://s.pstatic.net/static/newsstand/2020/logo/light/0604/951.png
[200] https://s.pstatic.net/static/www/mobile/edit/20210929/cropImg_196x196_74817712174882549.jpeg
[200] https://s.pstatic.net/static/www/mobile/edit/20210929/cropImg_196x196_74817729437746617.jpeg
[200] https://s.pstatic.net/static/www/mobile/edit/20211005/cropImg_728x360_75353340594820619.png
[200] https://s.pstatic.net/static/www/mobile/edit/20210929/cropImg_196x196_74817750281560327.jpeg

```

3.6 구현하기 - 접속 지연 감지

서비스 접속은 정상 하지만 일부 요청에 지연이 확인된다

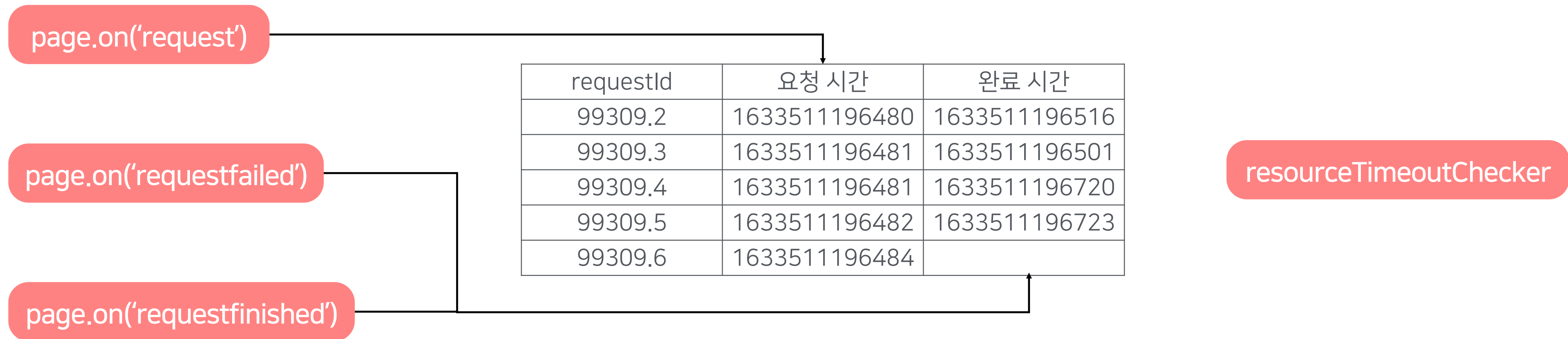
- 예시와 같이 일부 리소스 요청에 지연이 있는 경우 어떤 기능을 사용함에 있어 사용자 불편이 발생할 수 있고 지속될 경우 서비스 장애로 이어지는 경우도 있다

 icon-flag-de.png	GET	200	png	app-66c25e715...	1.1 kB	20 ms
<input type="checkbox"/> ace-builds.4a88ea03b6d954f7027d.js	GET	200	javascript	monitoring	433 kB	144 ms
<input type="checkbox"/> monitoringDetail.30adb08f2db09e68991e.js	GET	200	javascript	monitoring	1.0 MB	187 ms
<input type="checkbox"/> monitoringRegitster.a50887b8090bf85503a7.js	GET	200	javascript	monitoring	94.5 kB	70 ms
<input type="checkbox"/> subscription.d8c51d52c56a71073b8f.js	GET	200	javascript	monitoring	15.6 kB	41 ms
<input type="checkbox"/> m?u=https%3A%2F%2Fconsole.ncloud.com%2F...	GET	200	gif	wcslog.js:1	580 B	93 ms
<input checked="" type="checkbox"/> productContentPage.47a5e19b7cd0ab686319.js	GET	200	script	manifest.547e6f...	176 kB	36 ms
<input type="checkbox"/> productContentPage.47a5e19b7cd0ab686319.js	GET	200	javascript	jquery-e86504b...	176 kB	31 ms
<input type="checkbox"/> ico-check-white.svg	GET	200	svg+xml	app-66c25e715...	513 B	7 ms
<input type="checkbox"/> ico-extended-list-close.svg	GET	200	svg+xml	app-66c25e715...	575 B	7 ms
<input type="checkbox"/> recent	GET	200	xhr	vendor.cda21b2...	5.0 kB	42 ms
<input type="checkbox"/> new-count	GET	200	xhr	vendor.cda21b2...	1.6 kB	45 ms
<input checked="" type="checkbox"/> 0713f05433	GET	200	script	monitoring:391	844 B	8 ms
<input type="checkbox"/> favicon.ico	GET	200	x-icon	Other	2.4 kB	10 ms
<input checked="" type="checkbox"/> 0713f05433?url=https%3A%2F%2Fconsole.nclou...	GET	200	script	monitoring:391	367 B	9 ms
<input checked="" type="checkbox"/> ico-checked.svg	GET	200	svg+xml	app-66c25e715...	527 B	8 ms
<input type="checkbox"/> notification?articleOpenTypeCode=NANG&bbsArtic...	GET	200	xhr	vendor.9e2847a...	1.9 kB	157 ms
<input type="checkbox"/> members	GET	200	xhr	vendor.9e2847a...	2.2 kB	191 ms
<input checked="" type="checkbox"/> webhooks	GET	200	xhr	vendor.9e2847a...	1.9 kB	5.2 s
<input type="checkbox"/> dashboard	GET	200	xhr	vendor.9e2847a...	58.0 kB	53 ms

3.6 구현하기 - 접속 지연 감지

request, requestfailed/requestfinished 활용하여 지연 감지

- 서비스 접속에 대한 timeout은 page.goto()에 timeout 설정을 통해 가능
- 요청 하나하나에 대한 감시 설정 기능은 없으므로 EventEmitter를 활용해 구현



3.7 구현하기 - 자바스크립트 오류 감지

자바스크립트 오류는 'pageerror'을 통해 감지한다

- 하지만 evaluate로 실행된 오류는 'pageerror'에서 감지 되지 않으므로 예외처리를 통해 오류를 감지해야 한다

✖ Uncaught TypeError: Cannot read property 'addEventListener' of null
at docs.js:292

```
page.on('pageerror', (error) => {
  console.log(`pageerror : ${error.message}`);
  const resultDto = new resultDto();
  resultDto.setSuccessYn("N");
  resultDto.setErrorType("JAVASCRIPT");
  resultDto.setErrorMessage(`${httpStatusCode} - ${url}`);
});

await page.evaluate(() => {
  getValue();
}).catch(error => { /* */});
```

```
devview user$ npm start

> devview@0.0.0 start
> node example.js

pageerror : TypeError: Cannot read property 'addEventListener' of null
at https://www.elastic.co/guide/static/docs.js:292:49767
evaluate : Evaluation failed: ReferenceError: getValue is not defined
at __puppeteer_evaluation_script__:2:7
```

3.8 구현하기 - 기능 장애 감지

클릭, 텍스트 입력 등의 사용자 이벤트 정의를 통한 서비스 기능 장애 감지

- 검색어 입력 -> 검색 버튼 클릭 -> 검색 결과 확인

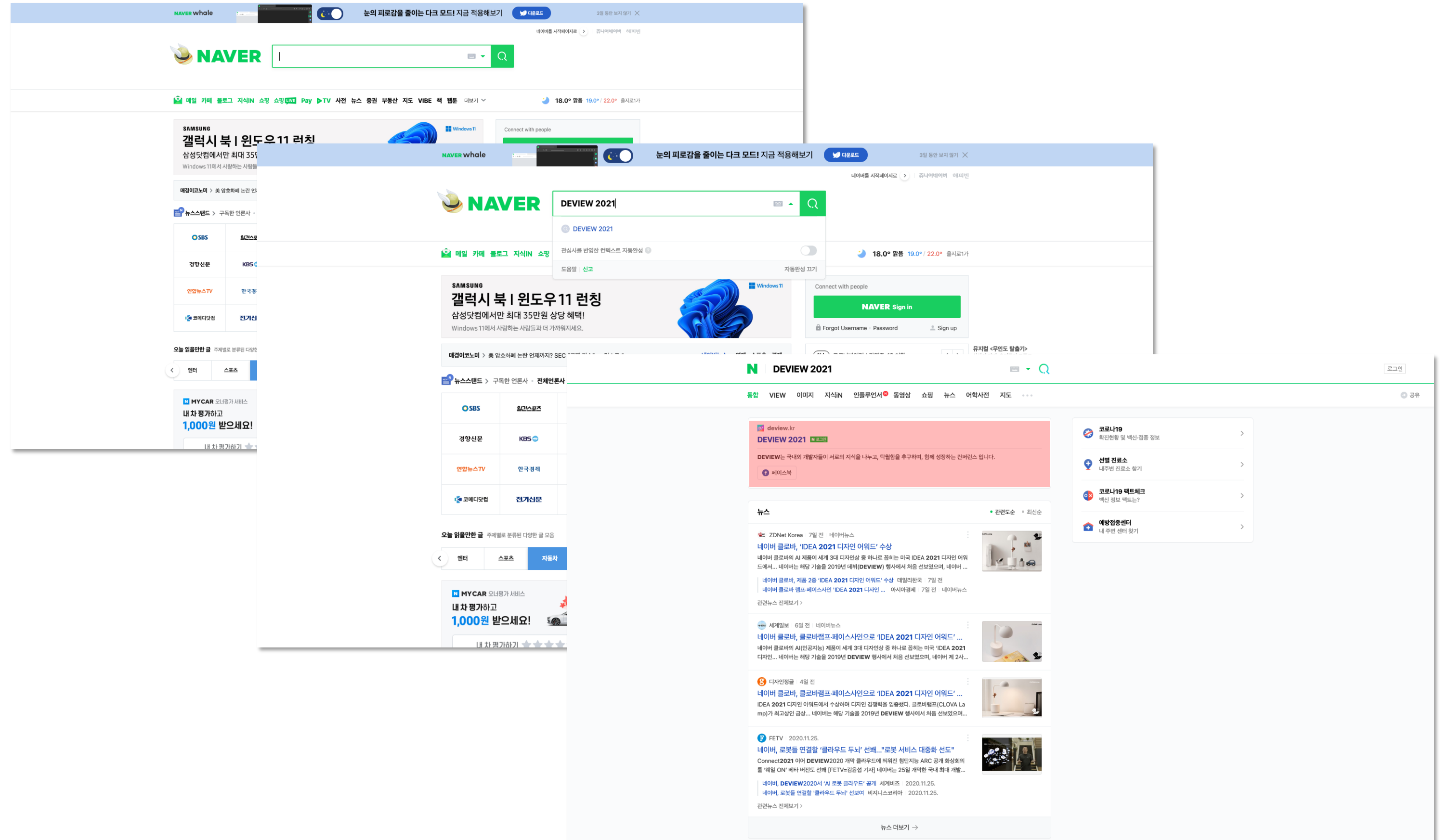
```

await page.goto('https://www.naver.com', {
  waitUntil: 'networkidle2'
}).catch(error => { ... });

await page.type("#query", "DEVIEW 2021").catch(error => {
  ...
});

await page.click("button[id='search_btn']").catch(error => {
  ...
});

await page.waitForSelector("_nsiteTop").catch(error => {
  const resultDto = new resultDto();
  resultDto.setSuccessYn("N");
  resultDto.setErrorType("OBJECT");
  resultDto.setErrorMessage("Element not found - _nsiteTop");
});
  
```



3.9 구현하기 - 요청 예외처리

Request interception을 활용한 예외처리

- continue, abort을 통해 request를 제어 할 수 있음
- 서비스 장애 영향이 없는 오류를 예외 처리할 수 있도록 필터 기능으로 활용

```
await page.setRequestInterception(true);
page.on('request', req => {
  if (req.url().indexOf(filterUrlList) >= 0) {
    req.abort();
  } else {
    req.continue();
  }
});
```

3.10 구현하기 - 요청 정보 변경

Request interception을 활용한 요청 정보 변경

- Method, Header, Post data등을 추가 또는 변경하여 요청
- 접속하기 위해 Header에 특별한 값이 있어야 접속이 되거나 API 호출 등이 활용

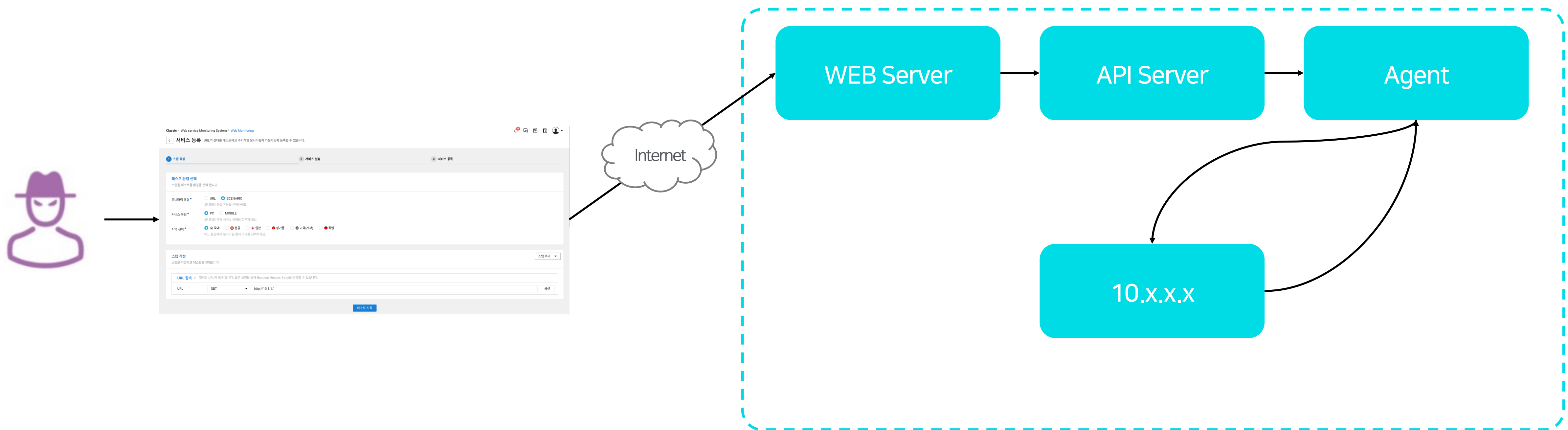
```
await page.setRequestInterception(true);
page.on('request', req => {
  if (req.url() == navigationUrl) {
    req.continue({
      method : 'POST',
      postData : JSON.stringify(data),
      headers : { 'key' : 'value' }
    });
  } else {
    req.continue();
  }
});
```

4. 보안 이슈 및 해결

4.1 SSRF 보안이슈

SSRF(Server Side Request Forgery) 보안이슈 문제 및 해결

- 외부에 공개되어 있는 Web 서버를 통해 내부 Private IP 대역에 우회 접근
- 네트워크 설정 또는 격리된 망으로 서버 이전 등으로 해결할 수 있지만
- Application 에서 해결할 수 있는 방안을 검토



4.1 SSRF 보안이슈

SSRF(Server Side Request Forgery) 보안이슈 문제 및 해결

- 에이전트에서 요청 하는 모든 Request를 검증

Agent

```
const regPrivatelp = /^(127W.)(10W.)(172W.1[6-9]W.)(172W.2[0-9]W.)(172W.3[0-1]W.)(192W.168W.)/i;
page.on('request', req => {
  const domain = this.getDomain(req);
  const ipList = await this.resolve(domain);

  for (let ip of ipList) {
    if (regPrivatelp.test(ip)) {
      privatelpYn = true;
      privatelpLookupList.set(domain, true);
    }
  }
});
...
If (privatelpYn) req.abort();
else req.continue();
```

```
async resolve(domain){
  return new Promise((resolve, reject) => {
    dns.resolve(domain, (err, address) => {
      if(err) reject(err);
      resolve(address);
    });
  });
};
-> dns.resolve => [1.1.1.1, 10.1.1.1]
-> dns.lookup => 1.1.1.1
```

4.2 SSRF 보안이슈 - DNS rebinding

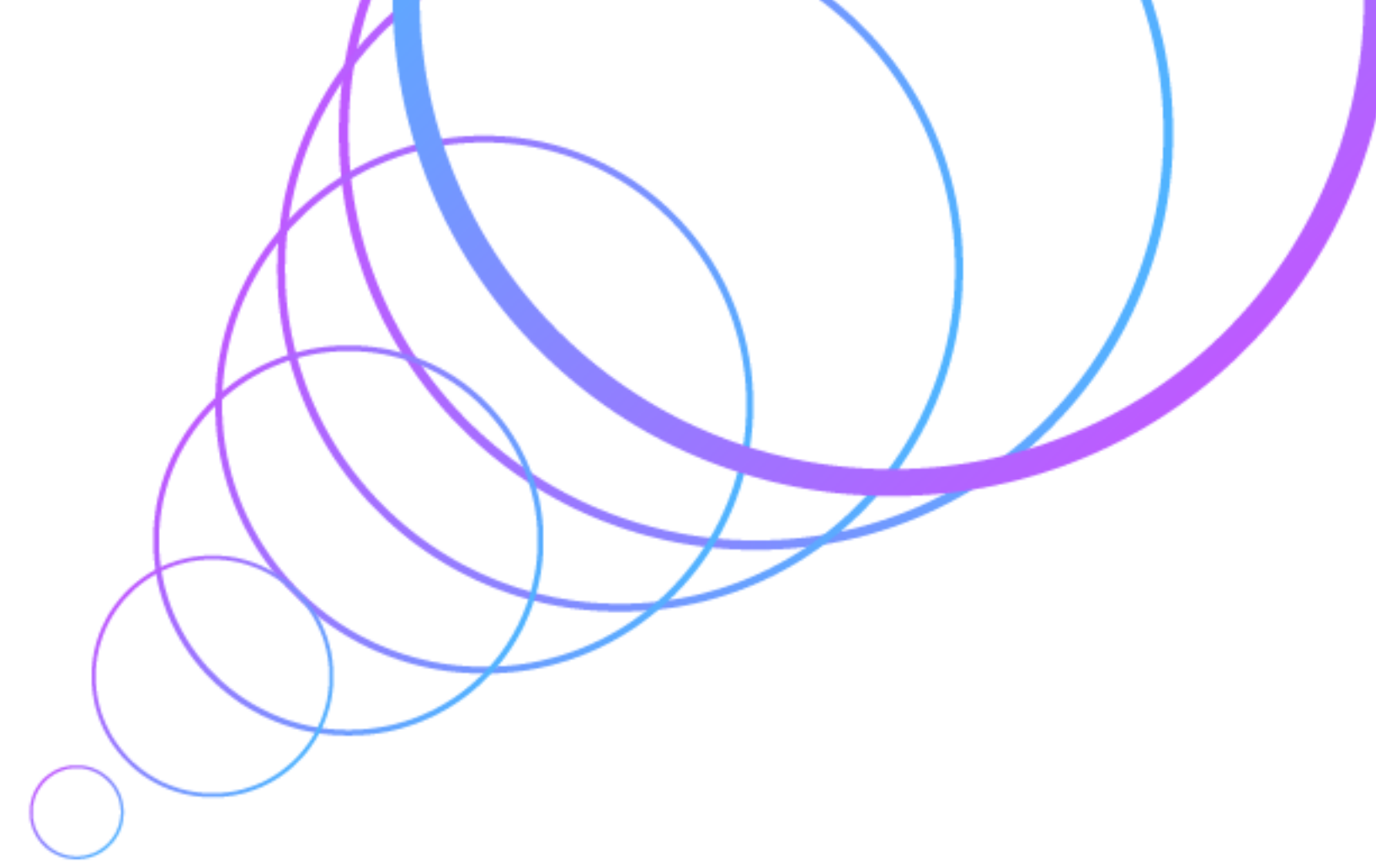
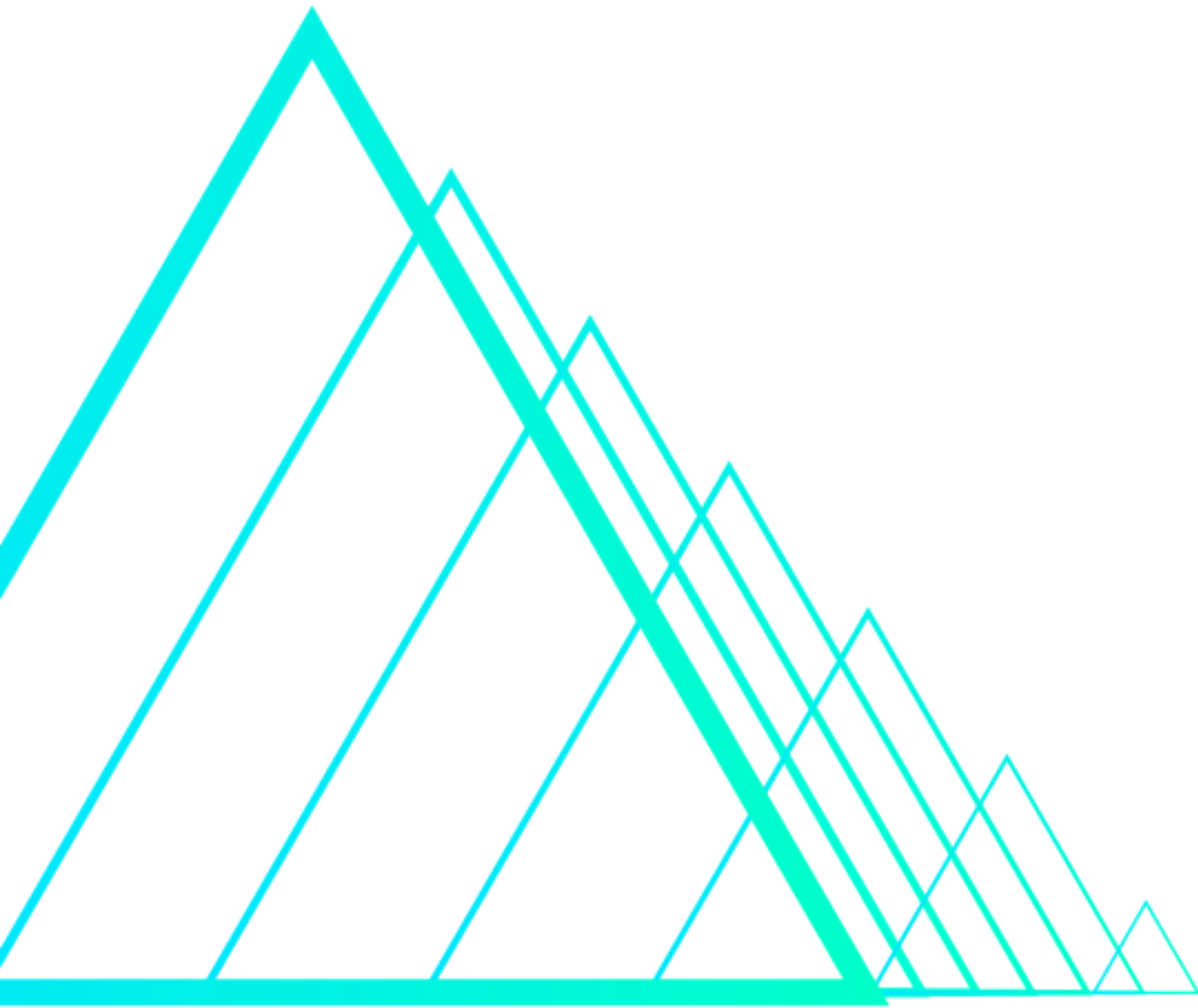
DNS rebinding? 보안이슈 문제 및 해결

- 검증 로직을 Pass하고 실제 req.continue()가 진행될 때 DNS rebinding을 통해 10.x.x.x 대역으로 변경하여 요청하여 검증 로직 우회

Agent

```
const regPrivatelp = /^(127\W.)(10\W.)(172\W.1[6-9]\W.)(172\W.2[0-9]\W.)(172\W.3[0-1]\W.)(192\W.168\W.)/i;
page.on('request', req => {
  const domain = this.getDomain(req);
  const ipList = await this.resolve(domain);
  ...
  });
  ...
  If (privatelpYn) req.abort();
  else req.continue();
```

```
page.on('response', async response => {
  const regPrivatelp = /^(127\W.)(10\W.)(172\W.1[6-9]\W.)(172\W.2[0-9]\W.)(172\W.3[0-1]\W.)(192\W.168\W.)/i;
  if (regPrivatelp.test(response.remoteAddress().ip)) {
    await this.setForbiddenErrorLog(` 403 - ${HttpStatusCode[403].msg}
    ${response.url()} `);
  }
});
```



Thank You

