Monorepo with lerna

한승호 @emmental

한승호

에멘탈 CTO

shilpe0304

Index

Monorepo with Node.js

노드 개발환경에 효율적인 모노레포

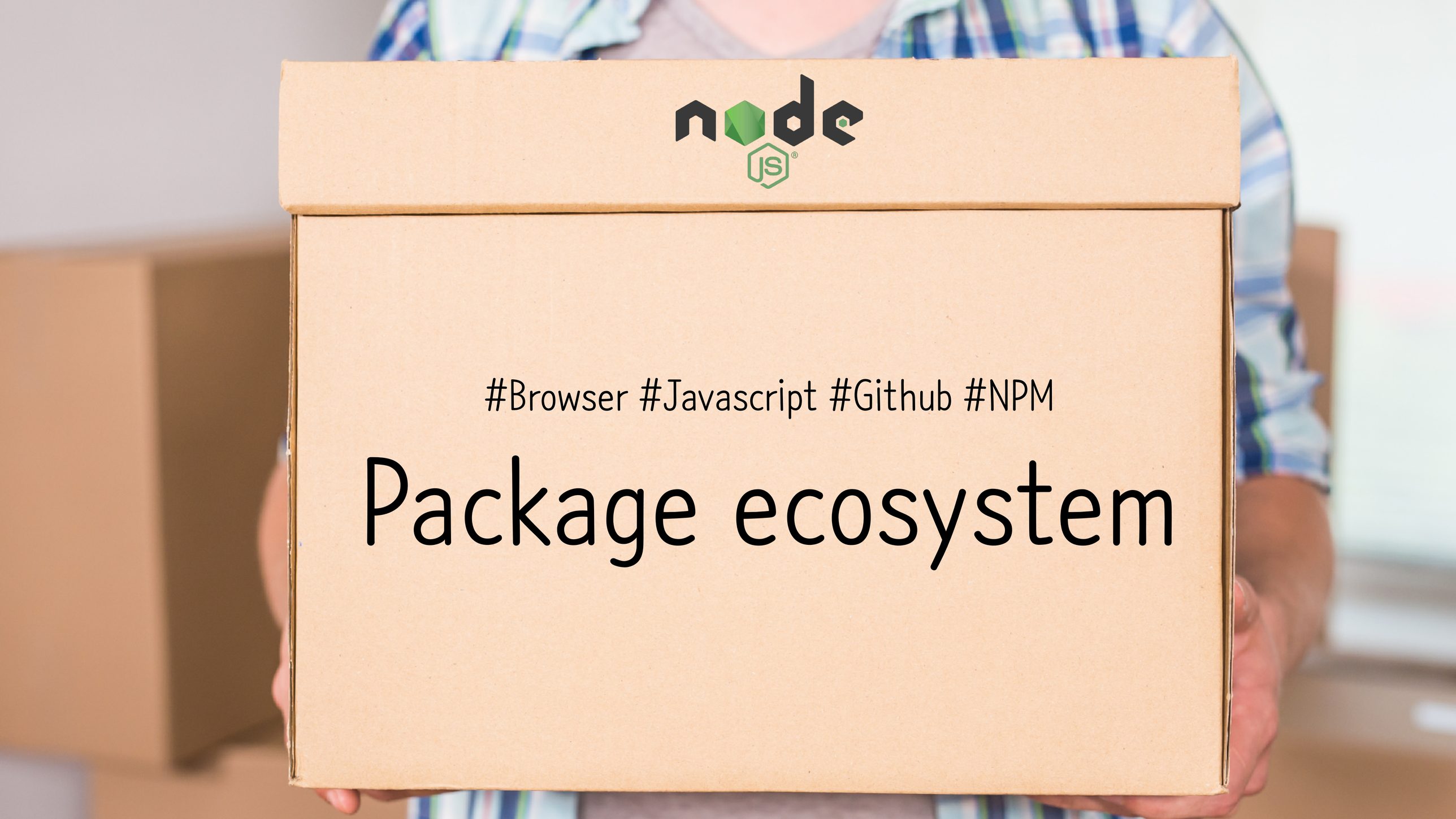
Javascript

Browser

NPM

github

event-driven, non-blocking I/O


node
JS
#Browser #Javascript #Github #NPM
Package ecosystem

Node ecosystem

By the numbers

Packages

585,647

Downloads · Last Day

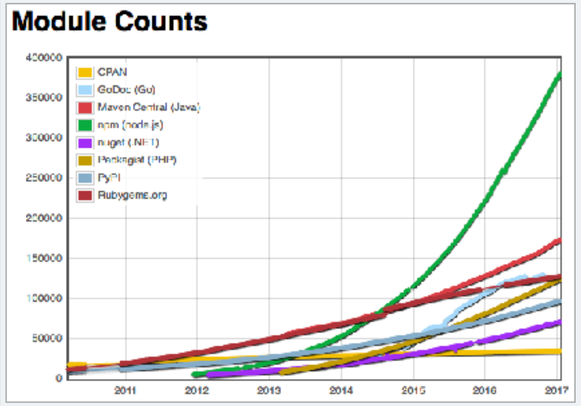
184,548,673

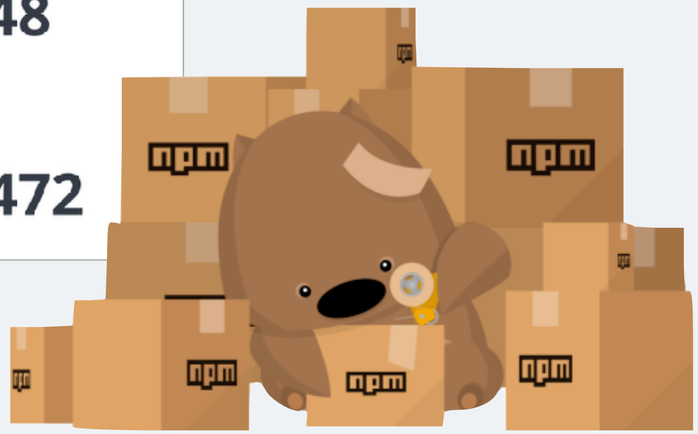
Downloads · Last Week

3,253,087,048

Downloads · Last Month

13,813,532,472





2017 .11. 05 기준

My Project

Folder = Module

```
require('../../../../models/email');
```

- admin
  - client
    - App
      - App.js
      - components
        - Footer
          - index.js
          - test
        - Navigation
          - Mobile
            - index.js
            - ListItem.js
            - SectionItem.js
            - test
              - ListItem.test.js
              - SectionItem.test.js
          - Primary
            - index.js
            - NavItem.js
            - test
          - Secondary
      - elemental
      - index.js
      - parsers
      - sagas
      - screens
      - shared
      - store.js
      - test
      - constants.js
      - packages.js
      - README.md
    - Signin
    - theme.js
  - public
  - server
  - build.js
  - CONTRIBUTING.md
- docs
- fields
  - components
  - explorer
  - mixins
  - README.md
  - types
  - utils
- greenkeeper-prs
- HISTORY.md
- index.js
- lib
- LICENSE
- netlify.toml
- package.json
- README.md
- server
  - bindBodyParser.js
  - bindErrorHandlers.js
  - bindIPRestrictions.js
  - bindLessMiddleware.js

Project repo v1.0.0

Module1 repo  v1.0.0

Module2 repo  v1.2.0

… 만약 20개 이상의 모듈을 관리한다면?

Reactive-Extensions / RxJS

Watch 586 | Star 17,766 | Fork 2,079

rx-core-binding	Add support for passive ev
rx-core-testing	Add support for passive ev
rx-core	Fix typo (Diposable -> Disp
rx-lite-aggregates-compat	Add support for passive ev
rx-lite-aggregates	Add support for passive ev
rx-lite-async-compat	Add support for passive ev
rx-lite-async	Add support for passive ev
rx-lite-backpressure-compat	Add support for passive ev
rx-lite-backpressure	Add support for passive ev
rx-lite-coincidence-compat	Add support for passive ev
rx-lite-coincidence	Add support for passive ev

babel / babel

Watch 730 | Star 23,409 | Fork 2,191

babel-cli	v7.0.0-b
babel-code-frame	v7.0.0-b
babel-core	v7.0.0-b
babel-generator	v7.0.0-b
babel-helper-annotate-as-pure	v7.0.0-b
babel-helper-bindify-decorators	v7.0.0-b
babel-helper-builder-binary-assi...	v7.0.0-b
babel-helper-builder-react-jsx	v7.0.0-b
babel-helper-call-delegate	v7.0.0-b

Monorepo

Single Repository, Many Modules

https://www.youtube.com/watch?v=PvabBs_utr8&feature=youtu.be&t=16m24s

Workspace repo

Module1 – v1.0.0

Module2 – v1.2.0

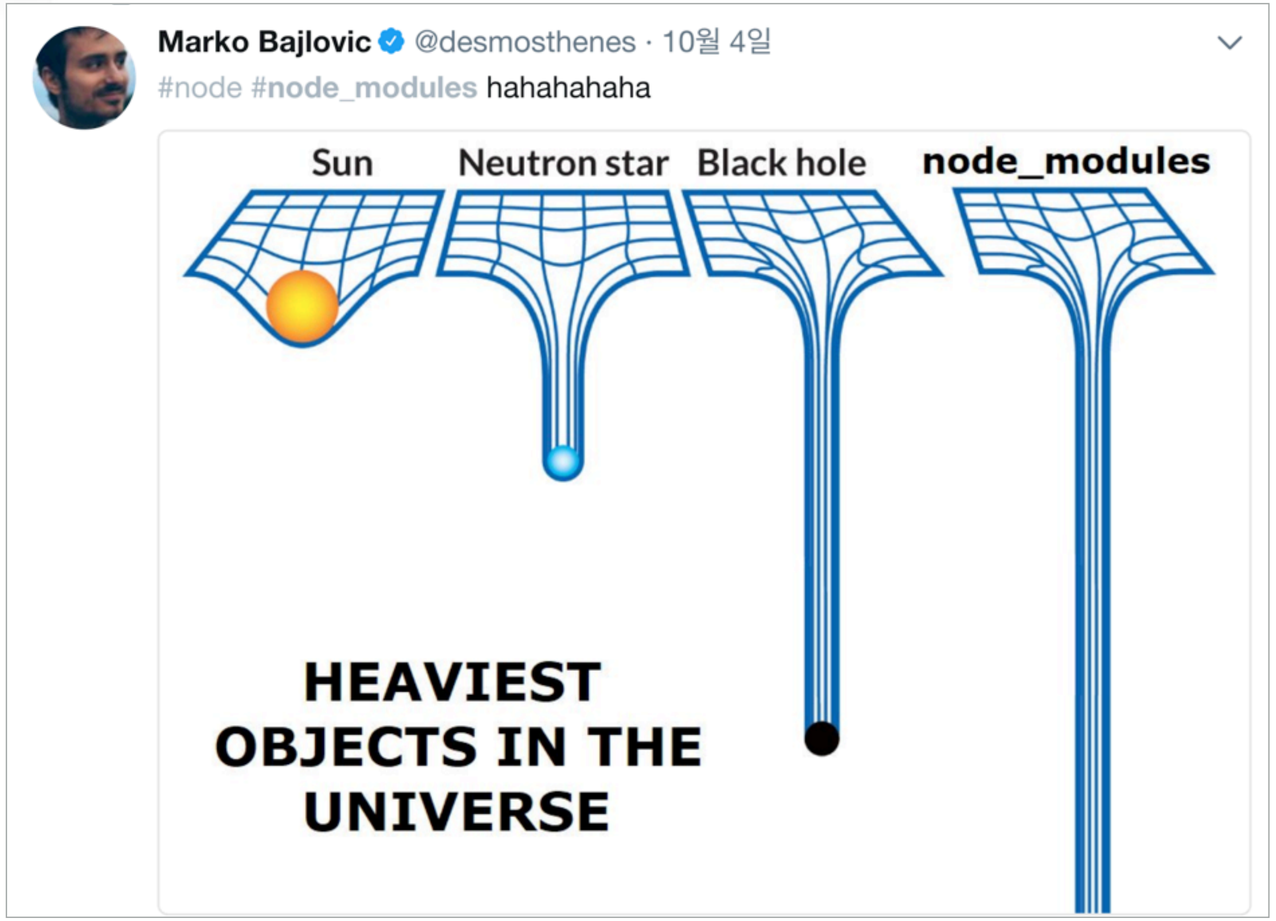
…

Module n …

awesome-monorepo

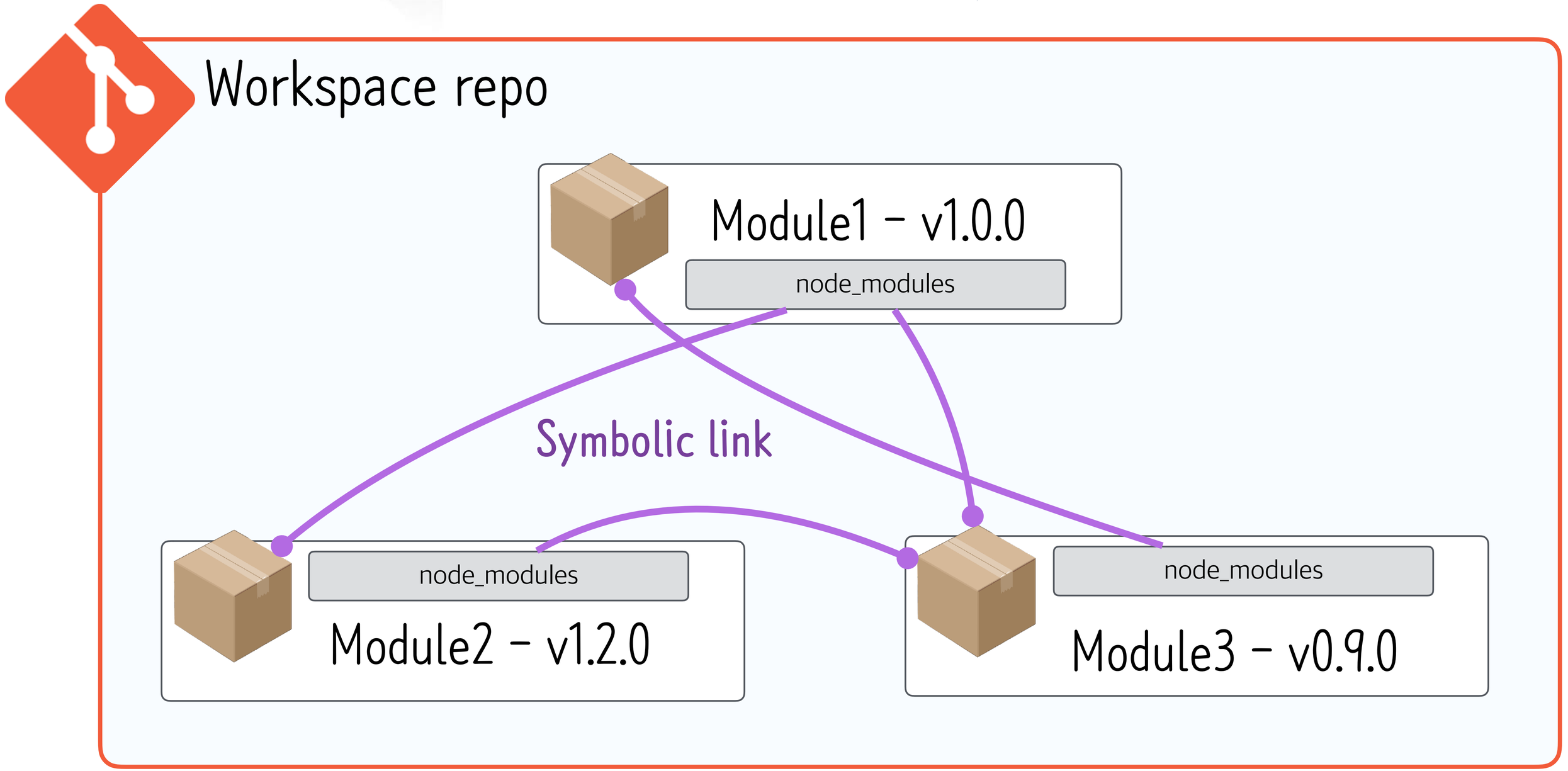
RxJs: https://github.com/Reactive-Extensions/RxJS/tree/master/modules
Angular https://github.com/angular/angular/tree/master/modules
Babel https://github.com/babel/babel/tree/master/packages
cycle.js: https://github.com/cyclejs/cyclejs
ngrx: https://github.com/ngrx/platform
Meteor: https://github.com/meteor/meteor/tree/devel/packages
ember.js: https://github.com/emberjs/ember.js/tree/master/packages
pouchdb: https://github.com/pouchdb/pouchdb/tree/master/packages/node_modules

Node.js 환경에서 monorepo를 사용하려면



node_modules + symbolic link

Node.js 환경에서 monorepo를 구축

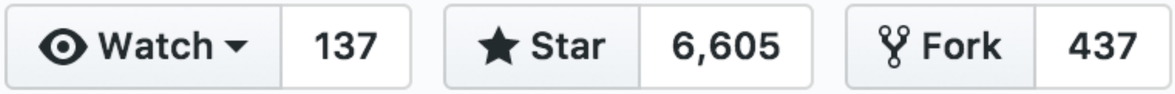


https://www.flickr.com/photos/lyre/5635626550

Monorepo 관리 모듈

Lerna 레르나

https://github.com/lerna/lerna

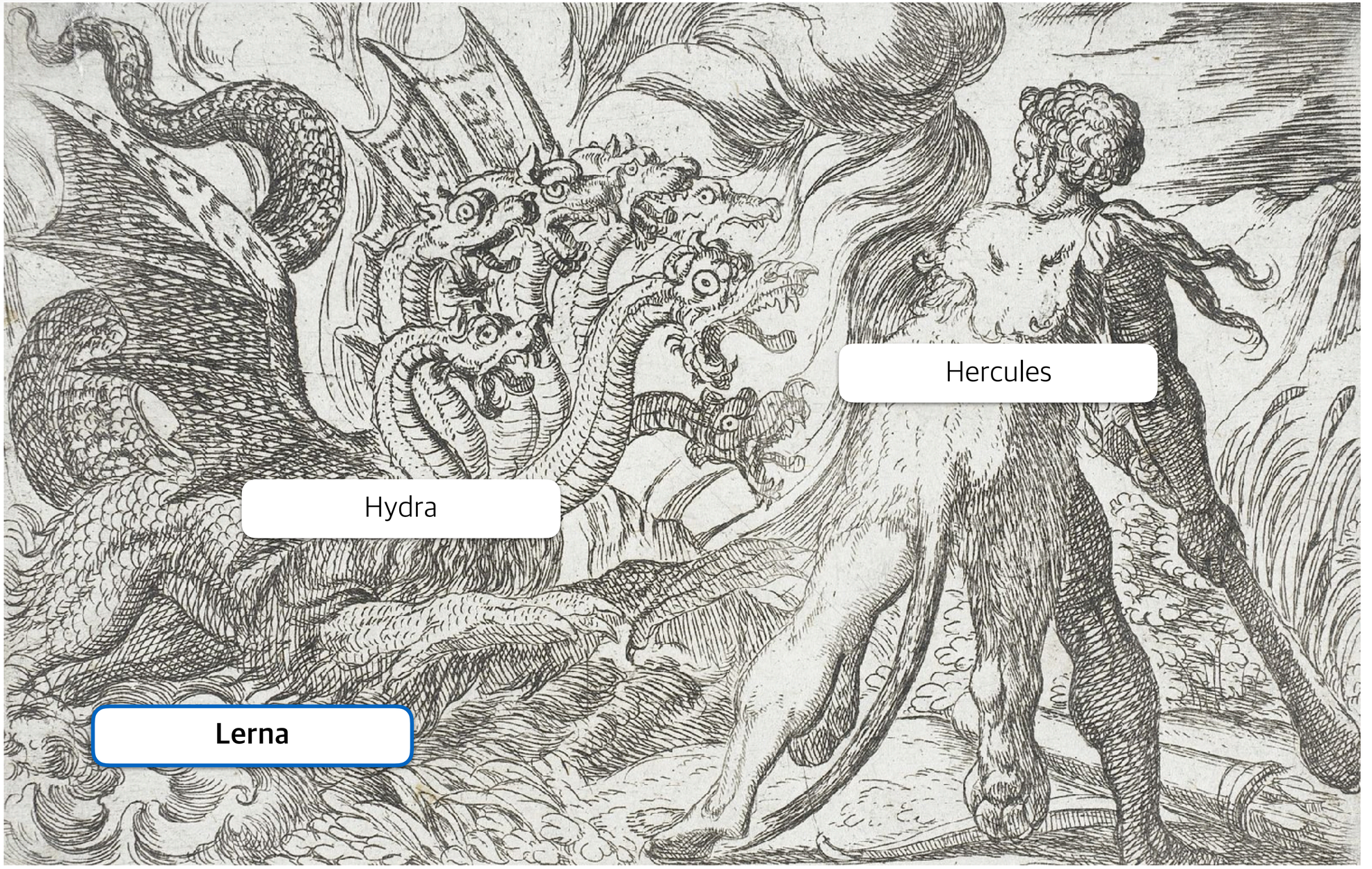


Monorepo with Lerna

레르나를 이용한 모노레포 구축

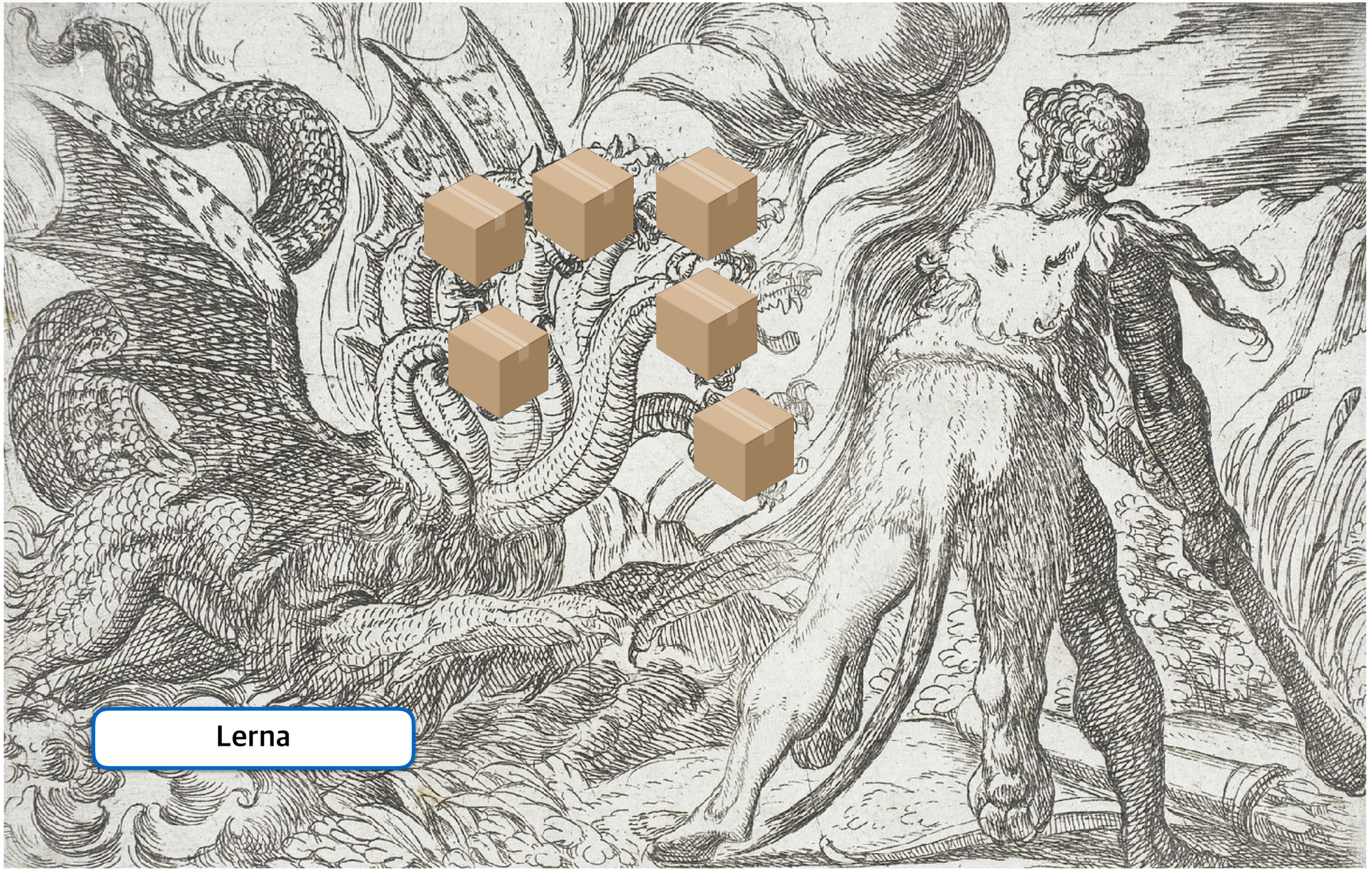
Hercules and the Hydra of Lerna

https://commons.wikimedia.org/wiki/File:Hercules_and_the_Hydra_of_Lerna_LACMA_65.37.9.jpg



Hercules and the Hydra of Lerna

https://commons.wikimedia.org/wiki/File:Hercules_and_the_Hydra_of_Lerna_LACMA_65.37.9.jpg



Hercules and the Hydra of Lerna

https://commons.wikimedia.org/wiki/File:Hercules_and_the_Hydra_of_Lerna_LACMA_65.37.9.jpg

$ npm install -g lerna

$ lerna init

lerna.json 생성

```
{
  "lerna": "2.4.0",
  "packages": [
    "services/*",
    "apis/*",
    "packages/*"
  ],
  "version": "independent",
  "npmClient": "npm"
}
```

```
$ lerna bootstrap

$ lerna publish

$ lerna exec ls

$ lerna exec --scope module1 ls

$ lerna clean
```

https://github.com/lerna/lerna

$ lerna bootstrap

각각의 모듈 폴더에서 npm install + **모듈간의 symbolic link 연결**

Workspace repo

Module1 - v1.0.0

```
packages/module1$ npm install
```

Module2 - v1.2.0

```
packages/module2$ npm install
```

Module3 - v1.0.3

```
packages/module3$ npm install
```

$ lerna exec -- ls -al

각각의 모듈 폴더에서 cmd 실행

Workspace repo

Module1 - v1.0.0

```
packages/module1$ ls -al
```

Module2 - v1.2.0

```
packages/module2$ ls -al
```

Module3 - v1.0.3

```
packages/module3$ ls -al
```

```
$ lerna exec --scope module1 --  ls -al
```

Module1 모듈 폴더에서만 cmd 실행

Workspace repo

Module1 - v1.0.0

```
packages/module1$ ls -al
```

Module2 - v1.2.0

```
-
```

Module3 - v1.0.3

```
-
```

$ lerna exec --ignore module3 -- ls -al

Module3만 무시하고 모듈 폴더에서 cmd 실행

Workspace repo

Module1 - v1.0.0

```
packages/module1$ ls -al
```

Module2 - v1.2.0

```
packages/module1$ ls -al
```

Module3 - v1.0.3

```
-
```

$ lerna clean

각각의 모듈 폴더의 node_modules 폴더 삭제

Workspace repo

Module1 - v1.0.0

```
packages/module1$ rm -rf node_modules
```

Module2 - v1.2.0

```
packages/module2$ rm -rf node_modules
```

Module3 - v1.0.3

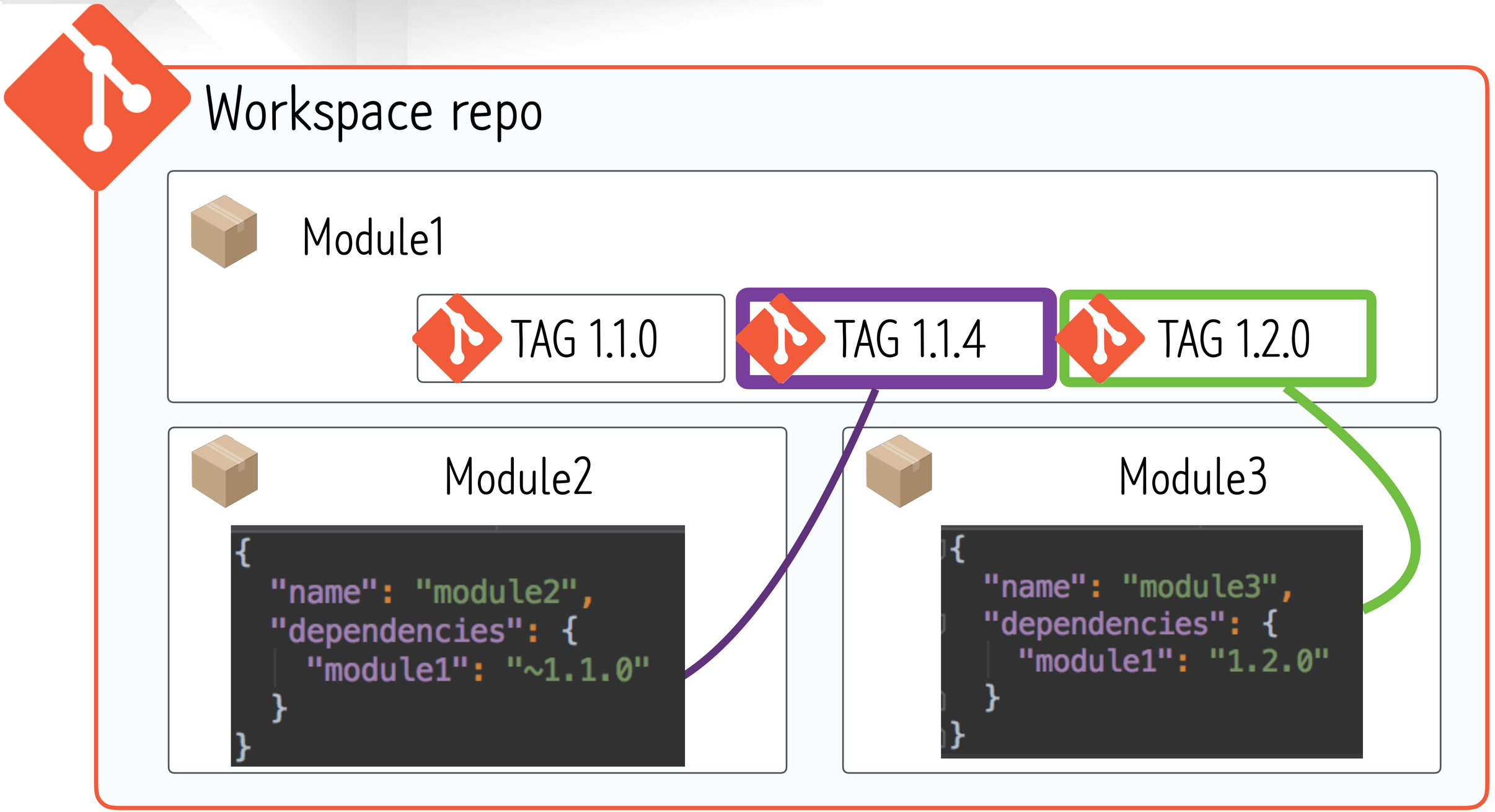
```
packages/module3$ rm -rf node_modules
```

$ lerna publish

각각의 모듈 폴더에서 git tag를 달거나, npm publish

Workspace repo

Module1 - v1.0.0	npm publish	git tag
Module2 - v1.2.0	npm publish	git tag
Module3 - v1.0.3	npm publish	git tag



Git tag publish된 모듈은 package.json **semver**으로 버전 관리가 가능합니다.

Monorepo + lerna

- **모듈 간의 느슨한 결합**
  - 독립된 lint, build, deploy
  - 모듈별 개별적인 버전 관리
  - 부담 없는 개선과 리팩토링 가능

- **한 저장소에서 이슈를 관리**
  - 하나의 레포에서 작업내역 추적 가능

- **빠르고 효율적인 개발 환경 구축**
  - 한 번의 클론으로 n개 이상의 모듈 개발 환경 구축

https://github.com/babel/babel/blob/master/doc/design/monorepo.md


조금 더 실무 경험을 포함해서...

에멘탈 프로젝트 환경

TypeScript

클라이언트

WEB(Angular4)

MOBILE(IONIC3)

서버

NodeJs(Hapi)

에멘탈 모노레포 구성

$ lerna ls

```
~/emmental/em-server(develop-firmOrg*) » lerna ls
lerna info version 2.4.0
lerna info versioning independent
antman-server      v1.0.0 (private)
bznav-server       v1.0.0 (private)
em-xray            v1.0.0 (private)
antman-api         v1.0.0 (private)
antman-worker      v1.0.0 (private)
bznav-acc          v1.0.0 (private)
bznav-cs           v1.0.0 (private)
bznav-firm         v1.0.0 (private)
bznav-hr           v1.0.0 (private)
bznav-org          v1.0.0 (private)
em-resource        v1.0.0 (private)
em-sso             v1.0.0 (private)
em-status          v1.0.0 (private)
@emmental/em-aws   v1.0.0 (private)
@emmental/em-core  v1.0.0 (private)
@emmental/em-sdk   v1.0.1
```

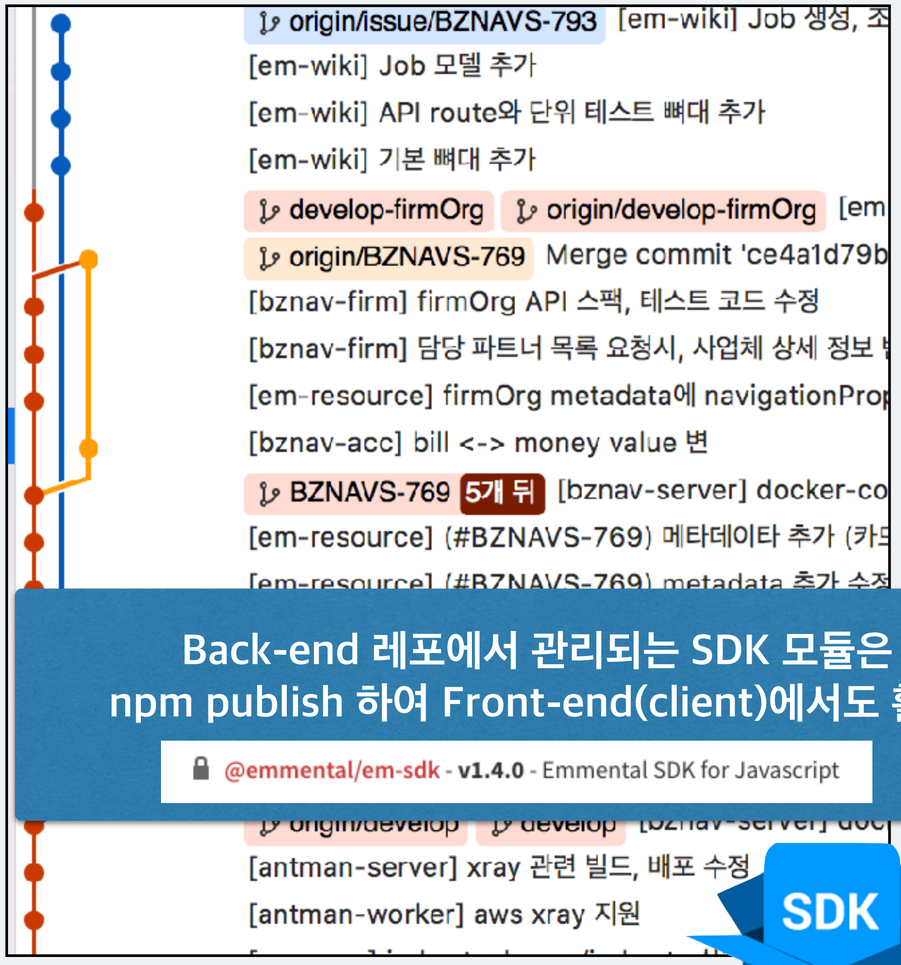
Back-end (19개 모듈)

```
~/emmental/em-web(develop-bznav*) » lerna ls
lerna info version 2.4.0
em-angular                  v1.0.0 (private)
@emmental/em-actions        v1.0.0
@emmental/em-breeze         v1.0.0
em-core                     v1.0.0
@emmental/em-effects        v1.0.0
@emmental/em-message        v1.0.0
@emmental/em-page           v1.0.0
@emmental/em-ui-scss        v1.0.0
@emmental/em-ui             v1.0.0
@emmental/em-util           v1.0.0
@emmental/wijmo       v5.20172.334
```

Front-end (11개 모듈)

에멘탈 레포지터리 구성

커밋 메시지 앞에 [모듈명]을 적어서 구분



Back-end 레포에서 관리되는 SDK 모듈은
npm publish 하여 Front-end(client)에서도 활용

@emmental/em-sdk - **v1.4.0** - Emmental SDK for Javascript

Back-end(server, worker)

Front-end(html, angular4)

```
$ lerna exec --  tsc -p .
```

각각의 모듈 폴더의 tsconfig.json을 기준으로 타입스크립트 컴파일

꿀팁!! 개발시 개별로 컴파일을 하기 때문에 컴파일 속도 구조적으로 향상 !!

Workspace repo

Module1 - v1.0.0

```
packages/module1$ tsc -p .
```

Module2 - v1.0.0

```
packages/module2$ tsc -p .
```

Module3 - v1.0.0

```
packages/module3$ tsc -p .
```

꿀팁!!

package.json에 스크립트를 정의해서 사용하면 편합니다. (특히 postinstall)

$ npm install

```
{
  "name": "em-server",
  "version": "1.0.0",
  "private": true,
  "scripts": {
    "clean": "rm -rf node_modules && lerna clean",
    "init": "npm install --global --silent typescript@2.5.3 mocha@3.5.3 lerna@2.4 && npm install --ignore-scr
    "tsc": "lerna exec --ignore @emmental/em-sdk --ignore \"@(em-cli|em-xray|em-build)\" -- tsc -p .",
    "start": "lerna exec npm start --scope bznav-server",
    "postinstall": "lerna bootstrap && npm run tsc",
  },
  "author": "Emmental, Inc.",
```

Monorepo for independence

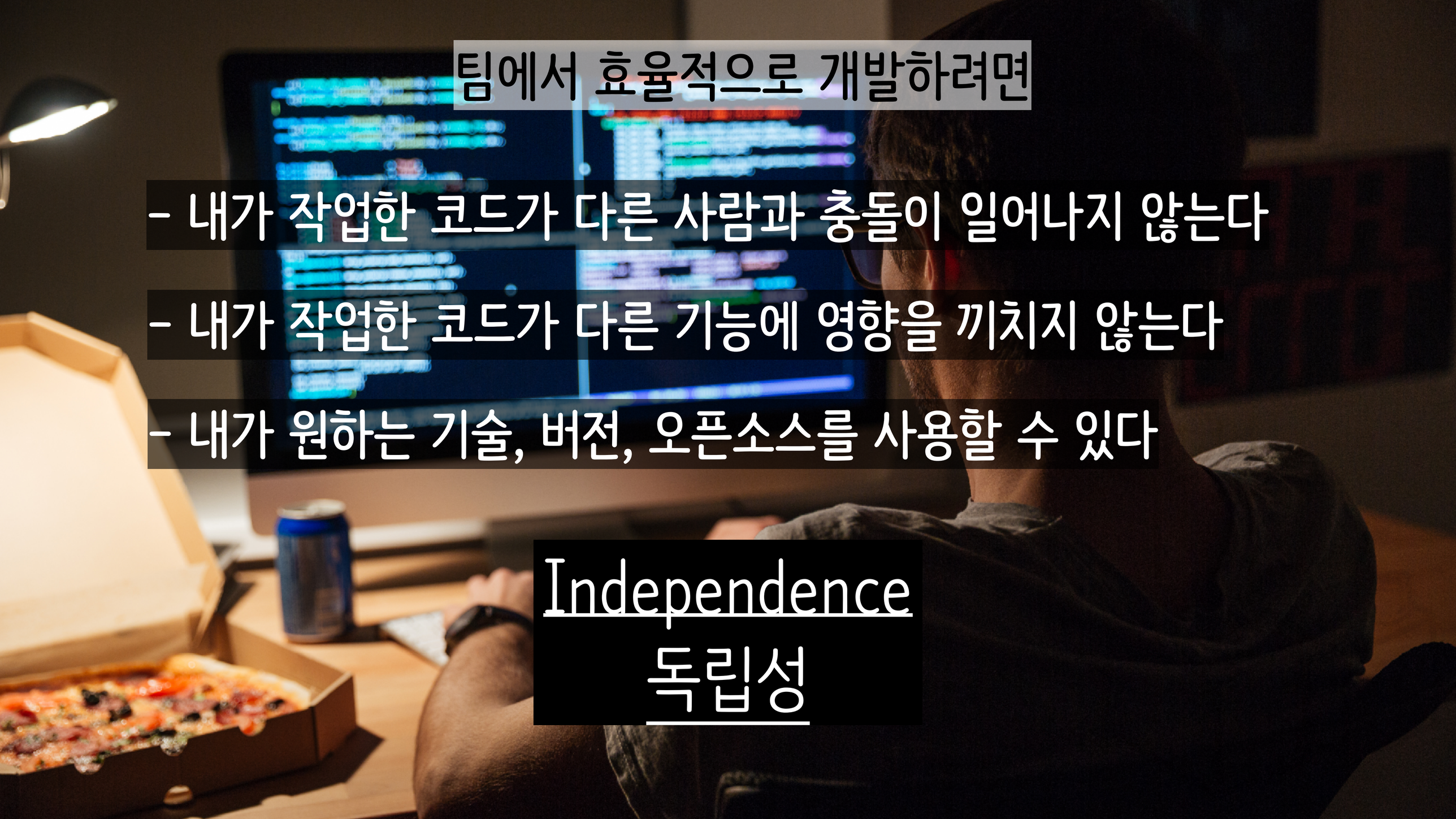
독립성을 위한 모노레포

Monorepo 사내 개발문화 고민

독립성 있는 모듈로 주인의식을 높이고자 하는 욕심이 담겨있는 영상

이모콘 2017 F/W - 프로젝트를 잘게! 잘게!

https://www.youtube.com/watch?v=4czTSeTpoQo



팀에서 효율적으로 개발하려면

- 내가 작업한 코드가 다른 사람과 충돌이 일어나지 않는다
- 내가 작업한 코드가 다른 기능에 영향을 끼치지 않는다
- 내가 원하는 기술, 버전, 오픈소스를 사용할 수 있다

Independence
독립성

Independent Module

Independent Build
Independent Test
Independent Deploy
Independent Owner

모듈로 나누어 개발하게 되면 폴더 구조보다 강한 독립성을 가질 수 있습니다

Microservices, Serverless 최상의 궁합!

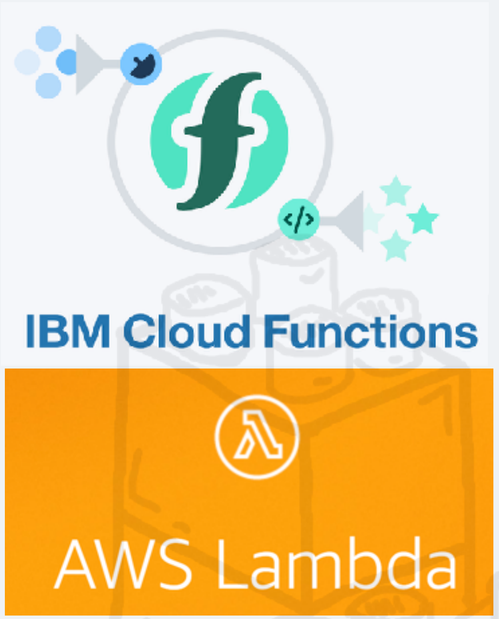
Independent Service

서비스(daemon)의 한계

- 연결을 위한 프로토콜이 필요함
- Core, util 재사용의 어려움

Microservices, serveless 아키텍처에 궁합

- 모듈 조합으로 빠른 서비스 제작과 배포



모듈간 느슨한 결합

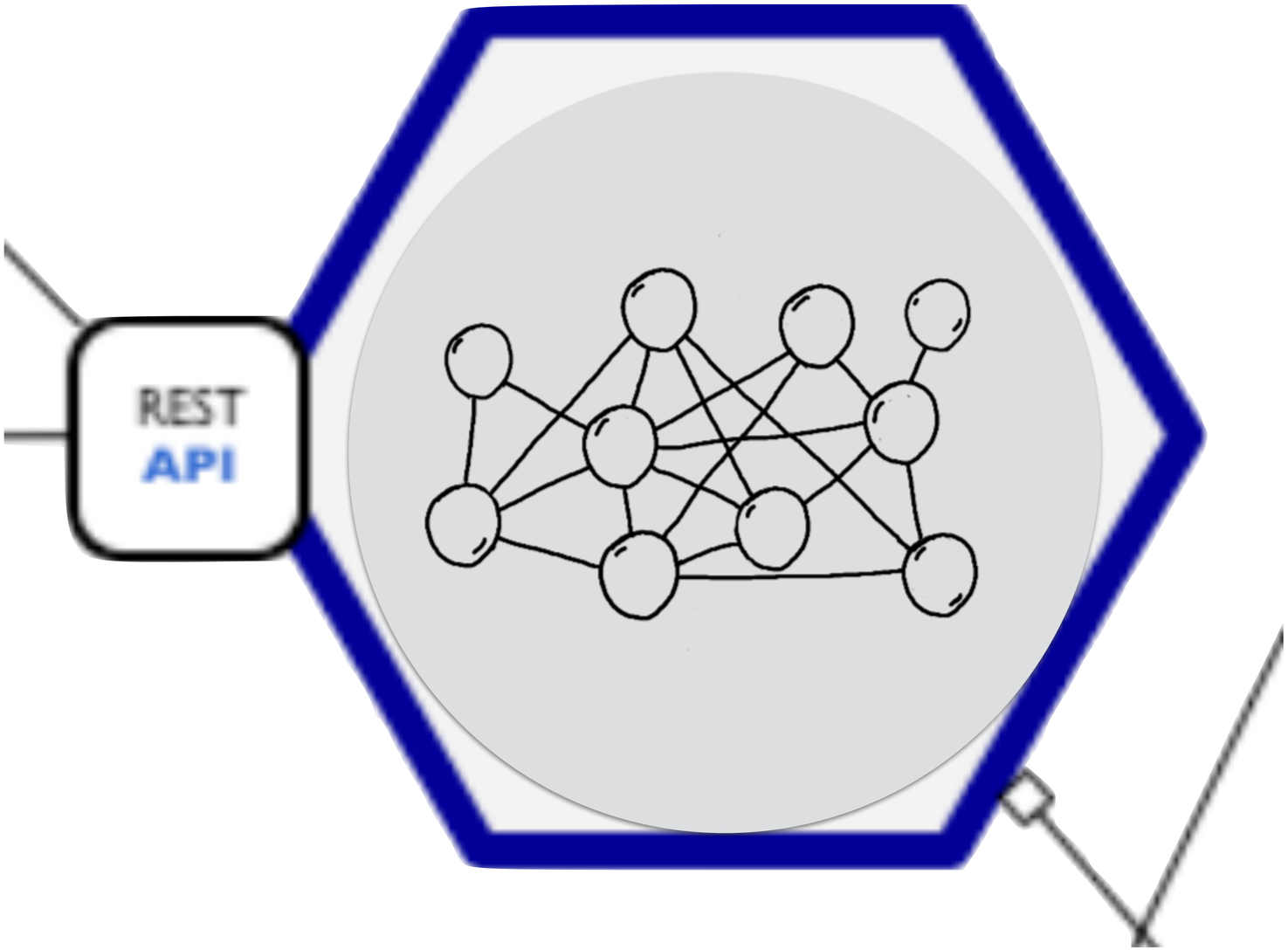
새로운 시도에 적합

- 독립된 lint, build, deploy = 부담없는 리팩토링
- 같은 기능을 하지만 새로운 기술을 이용한 모듈을 만들어 바꿔치기
- 레거시(legacy)에 대한 근본적인 해결 방법

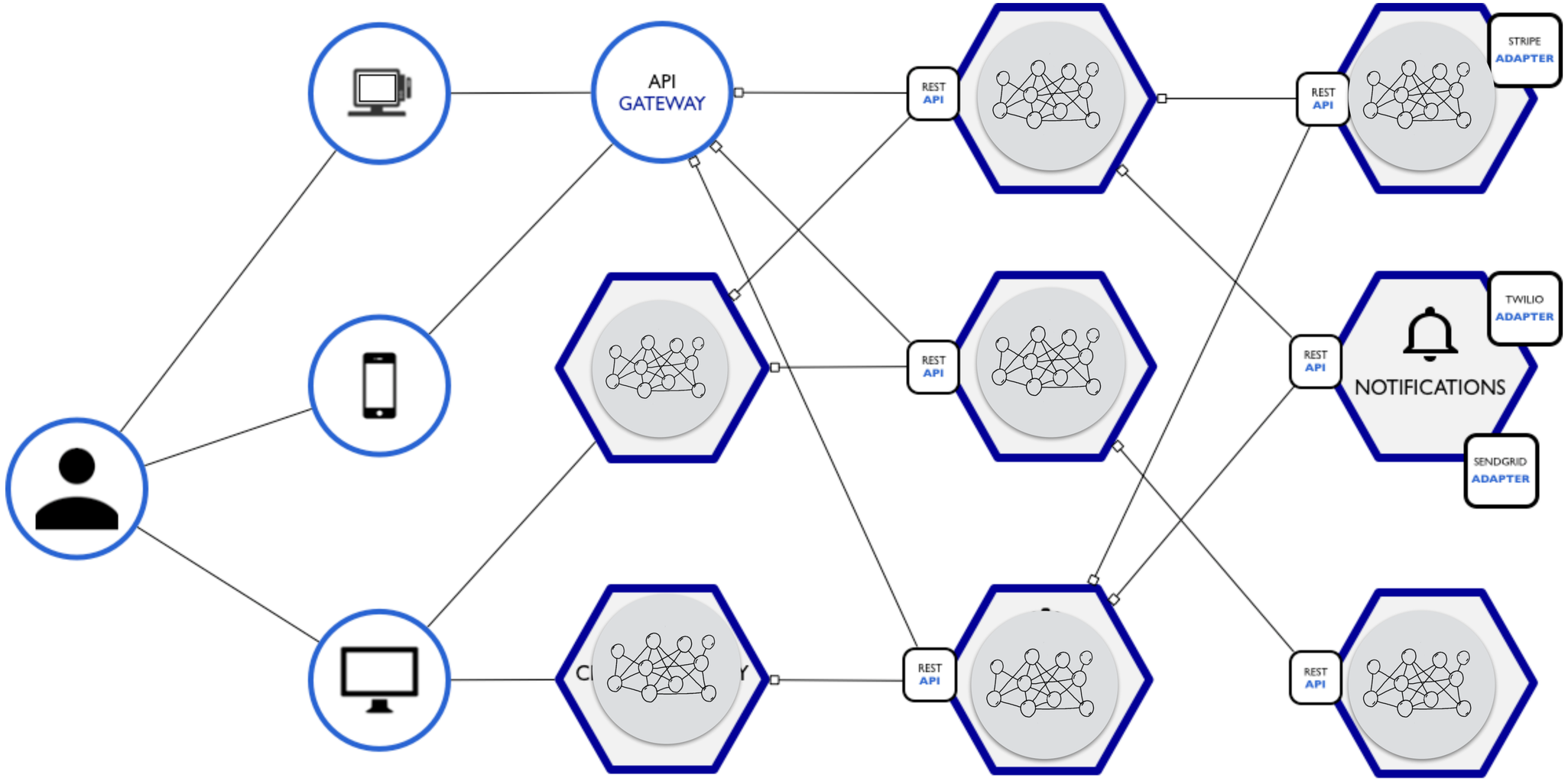
각각의 모듈로 구성된 서비스

새로운 모듈

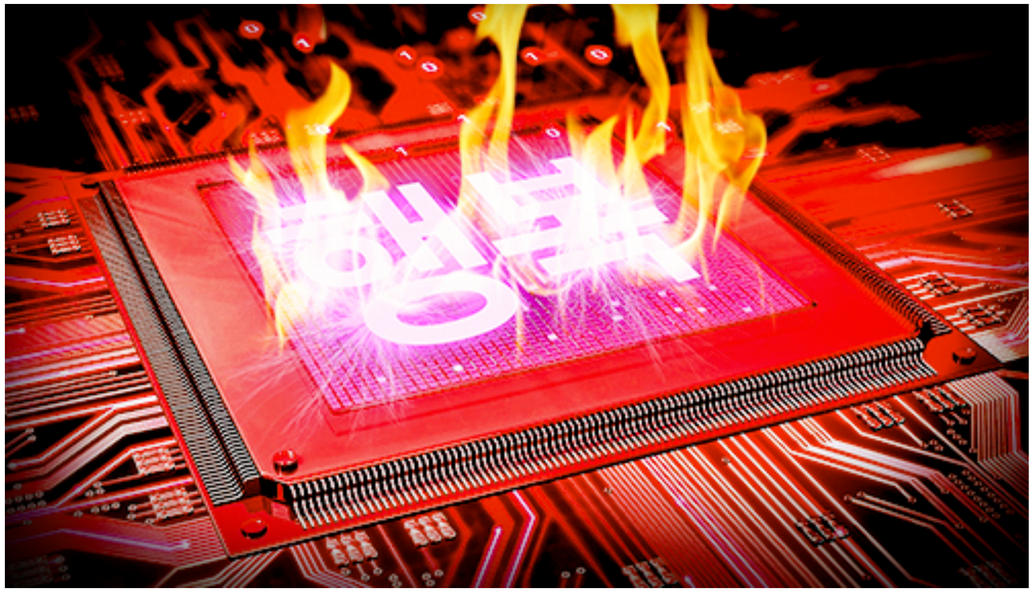
에멘탈 단일 서비스



에멘탈 마이크로서비스


API
GATEWAY
REST
API
REST
API
STRIPE
ADAPTER
REST
API
REST
API
TWILIO
ADAPTER
NOTIFICATIONS
SENDGRID
ADAPTER
REST
API
REST
API

모듈의 완성도가 서비스의 완성도로 이어지기를...



완성도 높은 모듈

완성도 높은 서비스

완성도 높은 마이크로서비스

Monorepo 실무 장점

1. 모듈 중심의 프로젝트 구조
각각의 모듈은 강한 독립성

2. 모듈 담당자의 책임감에 따른 코드 품질 향상

3. 마이크로서비스, 서비리스 아키텍처에 적합한 구조

Monorepo 실무 단점

1. Symbolic link가 고려되지 않은 모듈, IDE에서 예외 발생

2. 모노레포 이해, 러닝커브

Monorepo 도입을 고민해야 하는 환경

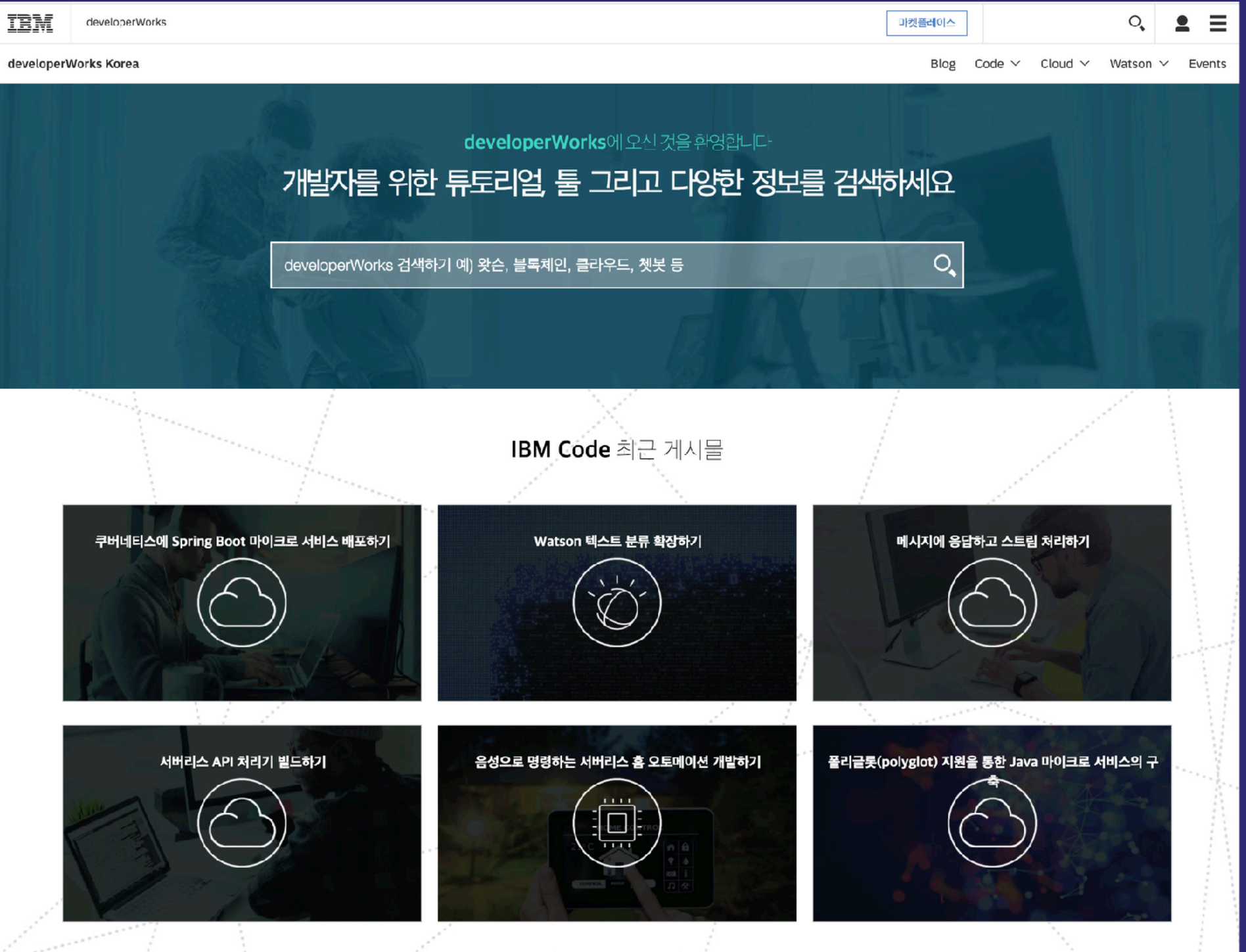
관리하는 모듈이 6개 이상이다

Node 프로젝트가 점점 커지고 있다

Monorepo 도입이후 장점 한마디

강제 모듈화로...

조금 더 (구조화|조직화) 된 프로젝트 구성



play.node(); 2017 with IBM

IBM

개발자라면 지금 방문하세요!

https://developer.ibm.com/kr/

emmental

Node로 코딩하기 좋은 회사, job@emmental.co.kr

Thanks

Speaker
han@hansh.kr