

나도 오픈소스 하나쯤 내놔볼까

김영재 / LINE 기술임원

이 세션에 적합한 분

오픈소스에 호기심 있으신 분

소프트웨어 아키텍처에 관심 있으신 분

근성, 인내, 꾸준함, 시행착오 스토리 좋아하시는 분

오픈소스가 뭐지?

원래는 동작하는 프로그램 코드만 오픈소스 였다

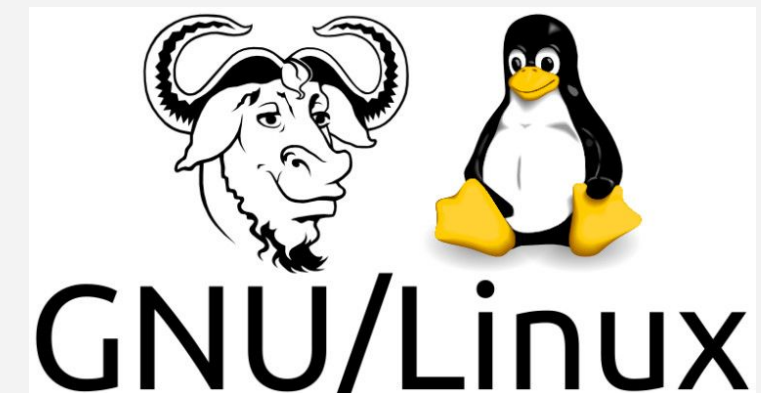
요즘은 “GitHub에 올릴 수 있는 내용”이면 대체로 오픈소스로 의미가 확대되었음

- 예 - 도움말, 즐겨찾기(Awesome) 모음, 버저닝 규칙, 개인 블로그, ...

대단한 철학으로 시작하기도 하지만, **배포방식 중 하나로** 생각하는 편이 좋다

그럼에도 지켜야 하는 본질 몇가지

라이선스, 보안, 건전한 커뮤니케이션



기술 명세서의 사례 - HeadVer

버저닝 규칙을 GitHub에 릴리즈

- <https://github.com/line/headver>

SemVer 호환

네이버/LINE 70여개 프로젝트에서 사용 중

오늘 릴리즈했으면 4.2441.60

The screenshot shows the GitHub repository for 'line/headver'. The repository is public and has 260 stars, 13 watchers, and 6 forks. The main branch is 'main'. The repository contains a README, a Code of Conduct, and an Apache-2.0 license. The README is titled 'HeadVer Specification' and shows the current version as 'version: 3.1924.59'. The version is broken down into three parts: 'head' (manual), 'yearweek' (automatic), and 'build' (automatic). The 'head' part is '3', 'yearweek' is '1924', and 'build' is '59'. The README also includes an introduction to the HeadVer versioning system and a list of contributors.

line / headver

Code Issues Pull requests Actions Security Insights Settings

headver Public

main Branches Tags

Go to file Add file Code

About

SemVer compatible version specification that has {head}.{yearweek}.{build} system.

convention semver versioning specification version working-semantics

Readme Apache-2.0 license Code of conduct Activity Custom properties 260 stars 13 watching 6 forks Report repository

Contributors 4

YoungjaeKim YoungjaeKim

lundiff PARK DongHa

shinshun-kuniyoshi Shinshun Kuniyoshi

thobestluck Lena

HeadVer Specification

version: 3.1924.59

head manual yearweek automatic build automatic

<https://github.com/line/headver>

It says: the 3rd public release, at around the 24th week of 2019, from build 59.

Introduction

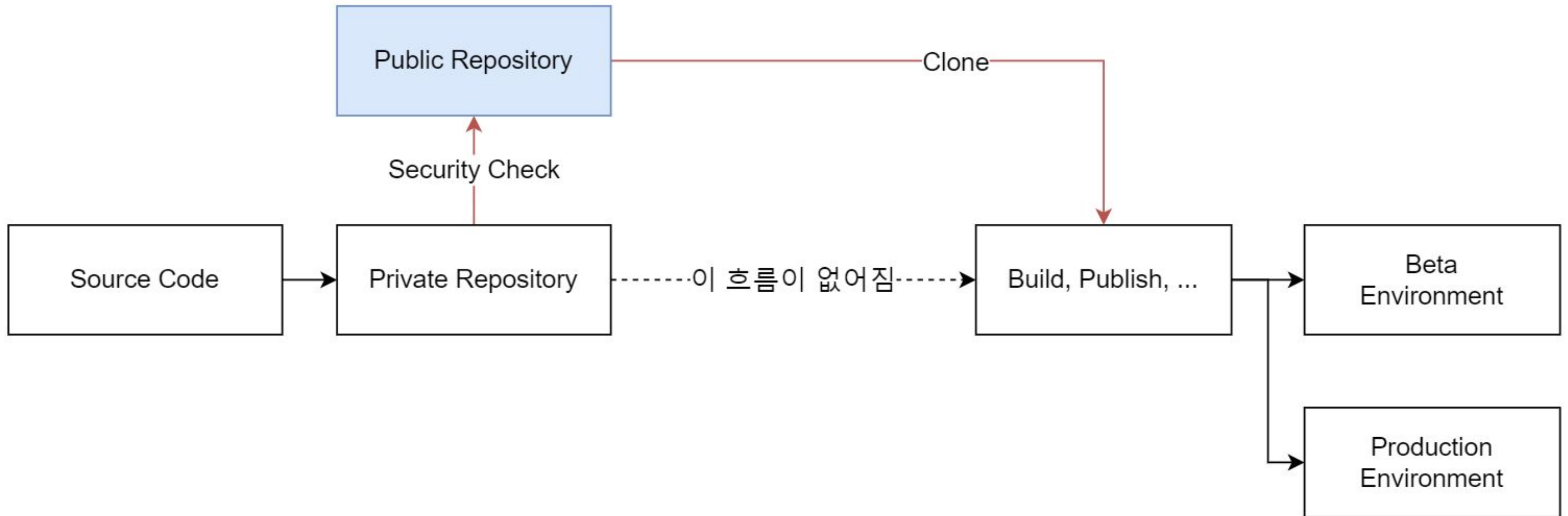
a version specification that has {head}.{yearweek}.{build} system.

The name 'headver' stands for "only head number is enough to maintain!" because it only allows to set the first number manually, and rest numbers are automatic. It simplifies x.y.z versioning and guarantees incremental version stamping on top of various versioning scripts introduced in this repository. HeadVer versioning system is battle-tested on the 40+ app/web services.

- {build} - manual Zero-based number.
- {yearweek} - automatic 2-digit for year and 2-digit for week number.
- {build} - automatic incremental number from a build server.

오픈소스로 배포할 때의 장점

오픈소스 배포



01

아키텍처

비즈니스 로직의 의존성을 낮출 수 있다

예: 우리 회사에서만 사용하는 인프라를 추상화
(특정 SMS API → SMS Sender Interface)

시작부터 보안에 좀 더 신경쓴다

예: secret key를 커밋하지 않는다

02

라이선스의 자유도

사용권리의 법무체크 부담을 낮출 수 있다

예: 관계가 복잡한 회사에서 다른 법인이 개발한
소프트웨어를 쓰는 경우

03

지속가능성

팀 내의 코드가 아닌, 회사내 모두의 코드

회사내 모두의 코드가 아닌, 회사 이름의 코드

비즈니스와 팀에만 공헌하는 것이 아니라,
유용한 기술에 공감하는 사람들과 함께 만든다

5년동안

5번 시도

5명이 만드는 중

ABC User Feedback 사례

<https://github.com/line/abc-user-feedback>

ABC User Feedback

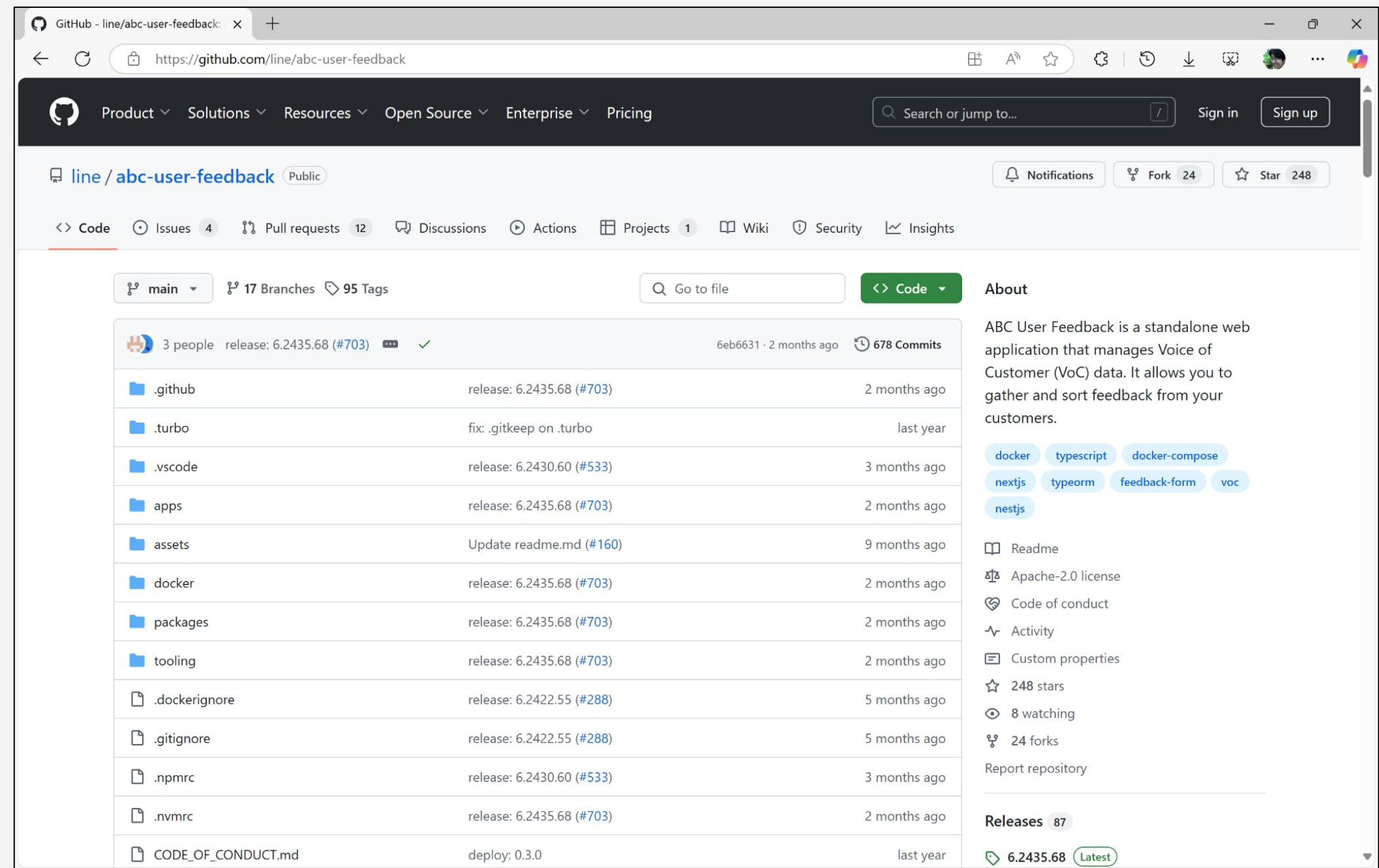


ABC User Feedback

사용자 의견을 수집하는 도구

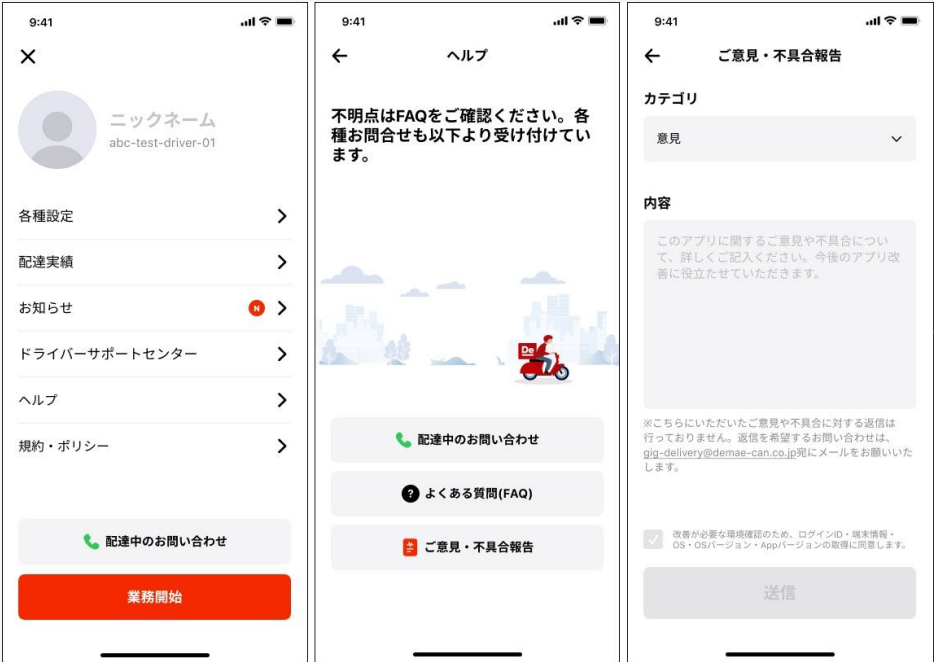
React+Node.js WebApp

<https://github.com/line/abc-user-feedback>



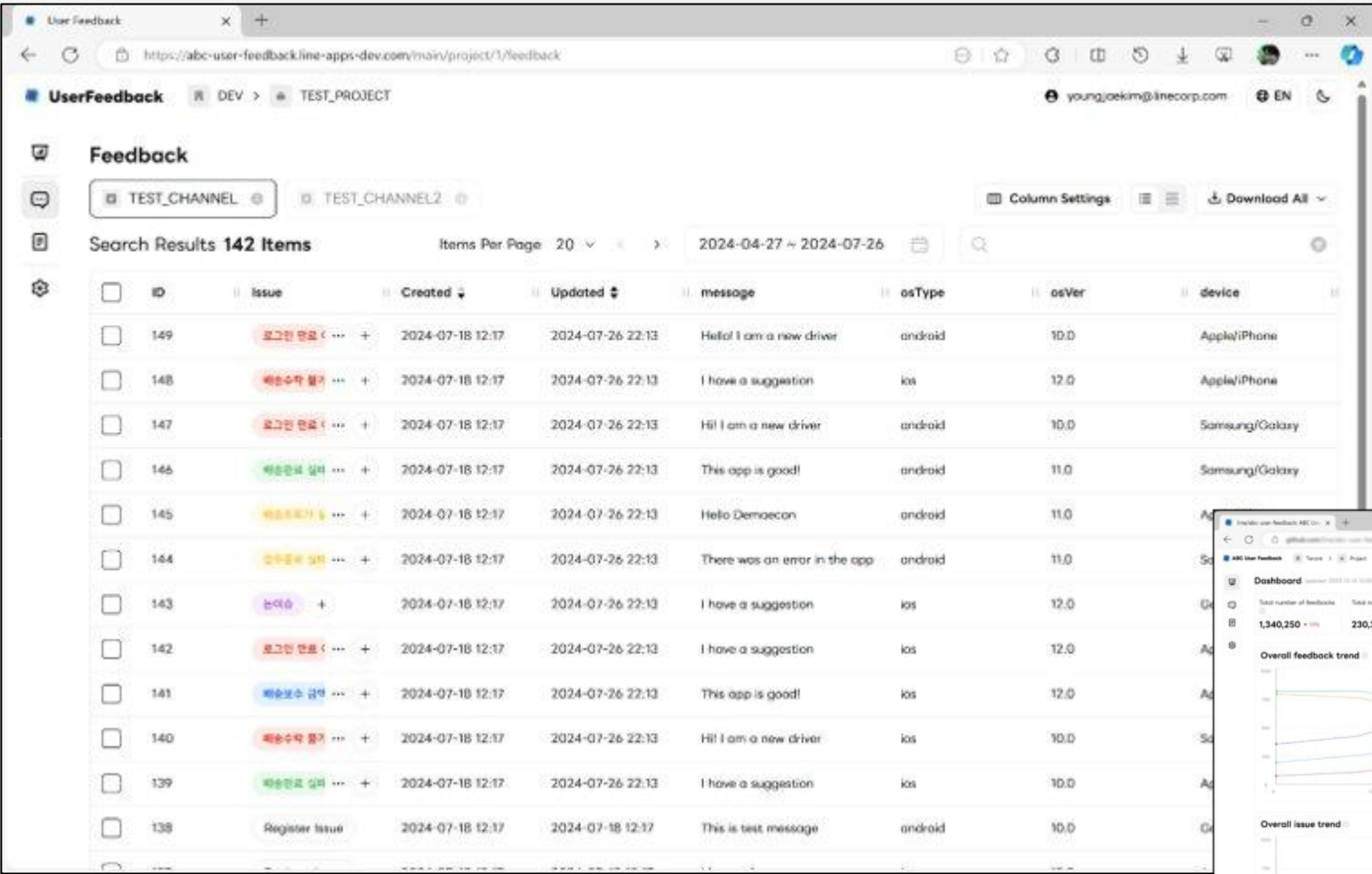
<https://github.com/line/abc-user-feedback>

ABC User Feedback

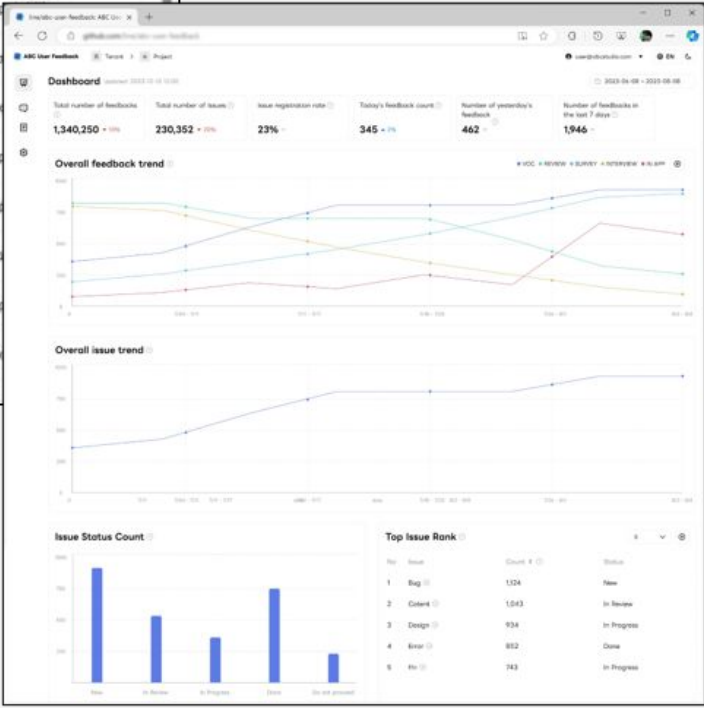


Client

API

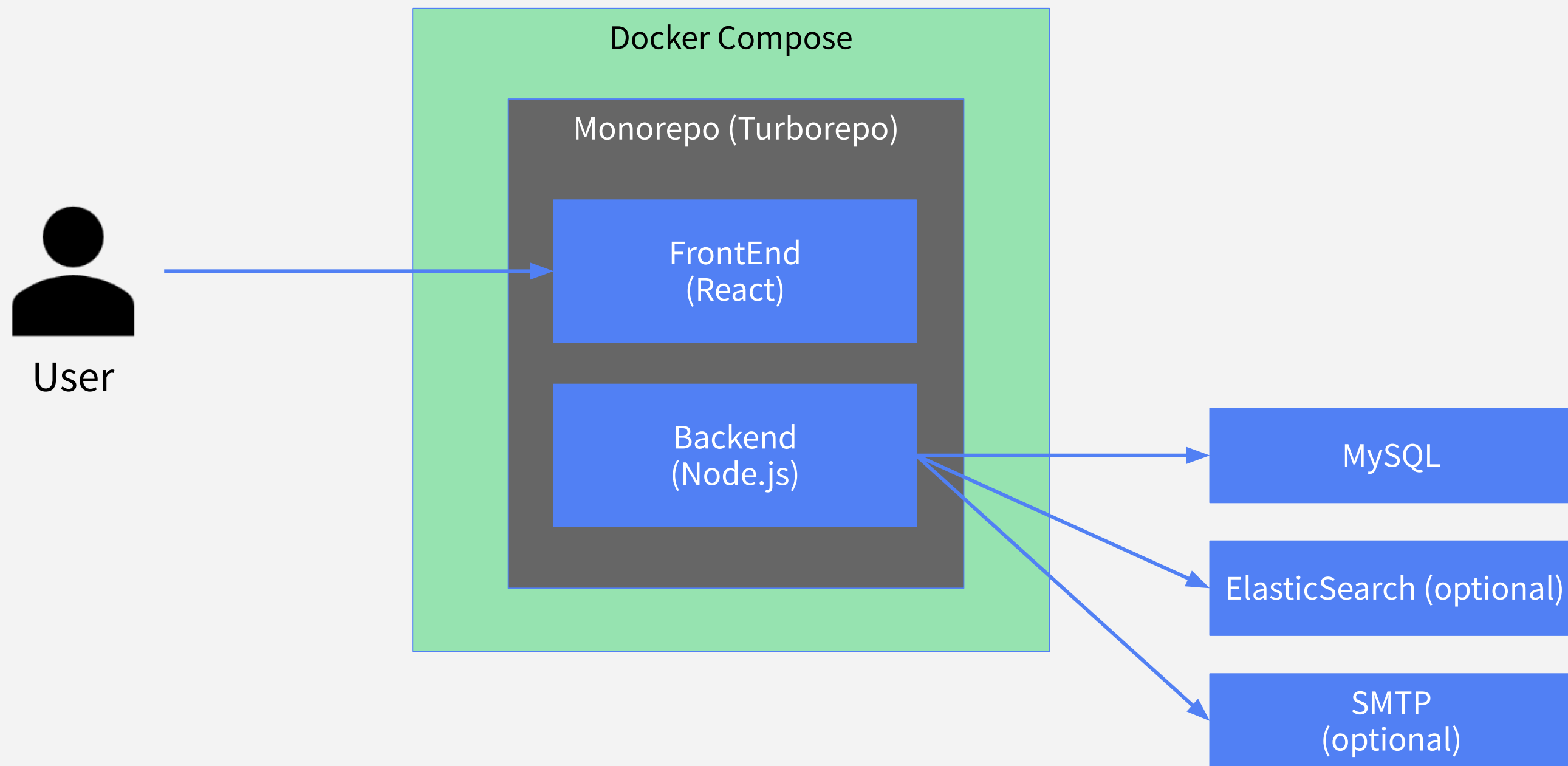


Admin



Dashboard

대략적인 구조



첫번째 시도

2019년

스타트업 때 사용자 의견 수집에 민감했다

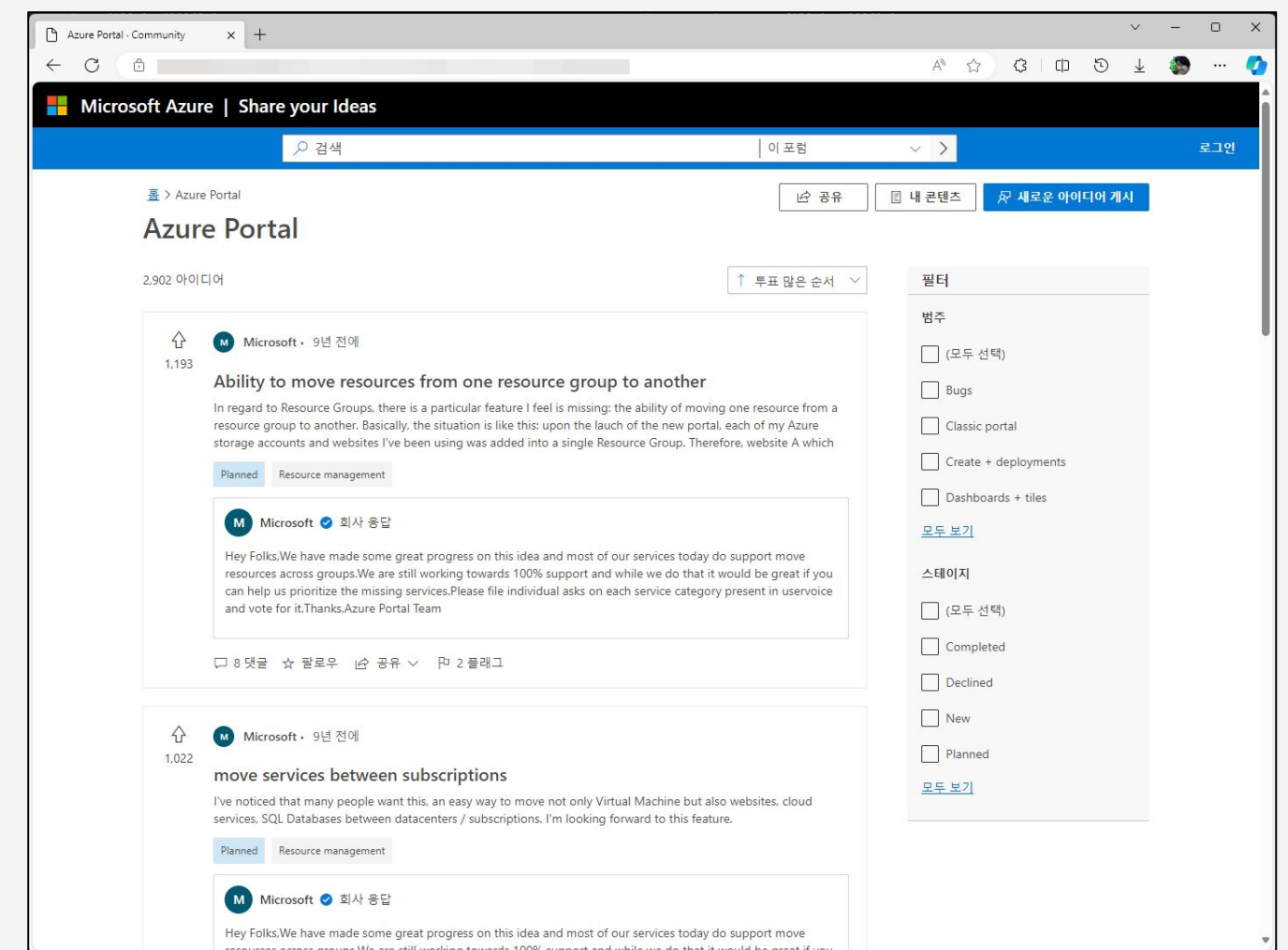


신고받은 버그를 수정하고

사용자 의견으로 기능을 만들고

No.1 회사가 되고

성공적으로 빅테크에 엑싯하고...



Microsoft Azure (uservoice.com)

“이거 진짜 중요한데 왜 없지”

사용자 의견 수집 SaaS는 많다

하지만, 개인 정보에 민감한 회사에서 SaaS를 도입하는건 쉽지 않다
가격도 부쩍 올랐다 (1000명 의견에 월 899달러!)

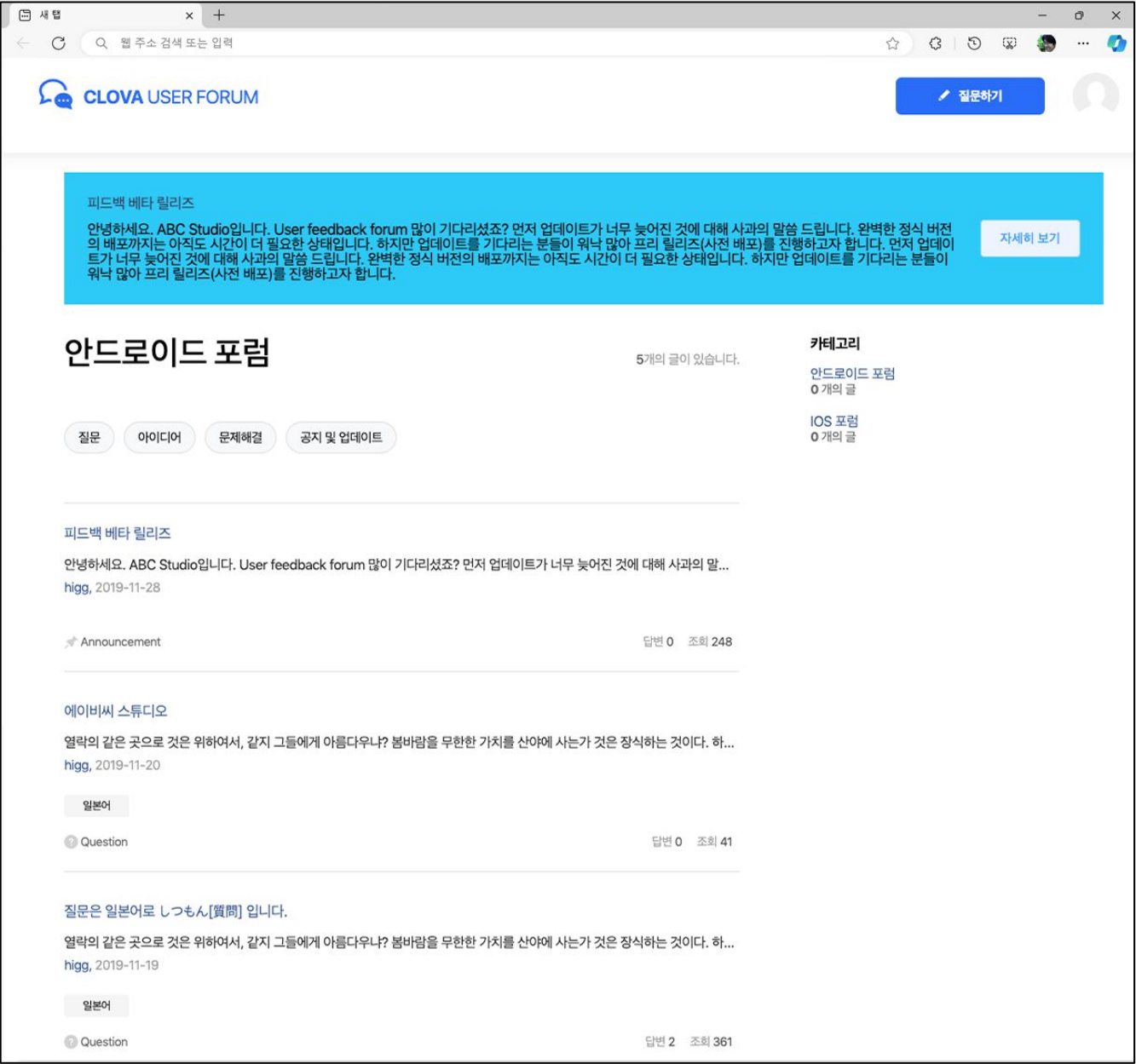
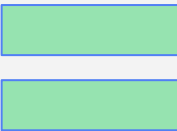
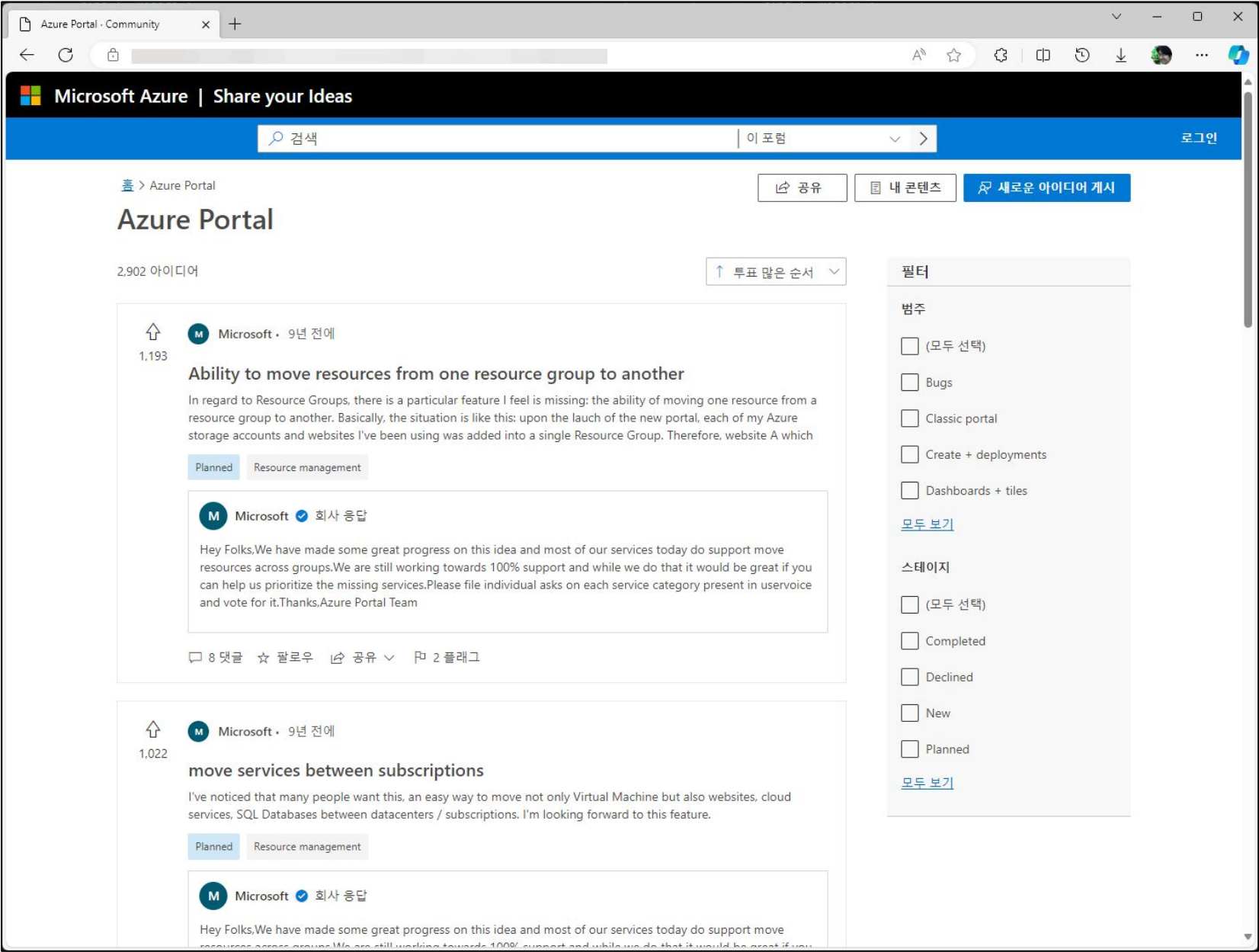
기능을 들여다보면...

- 기본적으로는 게시판 형태 (글쓰기, 댓글)
- upvote
- 이슈 상태
- 그 외에 태그, 정렬 등



까짓거 강 만들까

Clone 프로젝트처럼, 거의 유사하게 만들었음



그런데 동료들도 공감하지 못함 - PoC 실패

“혼자 하는건 안말리는데, 그게 왜 필요하죠”

“vote에 따라가느라 우리 서비스를 못만들어요”

“방치하는게 더 욱먹어요”

프로덕트 메이커의 자세 - 동료들 설득 못하면 유저도 설득 못한다

두번째 시도

2020년

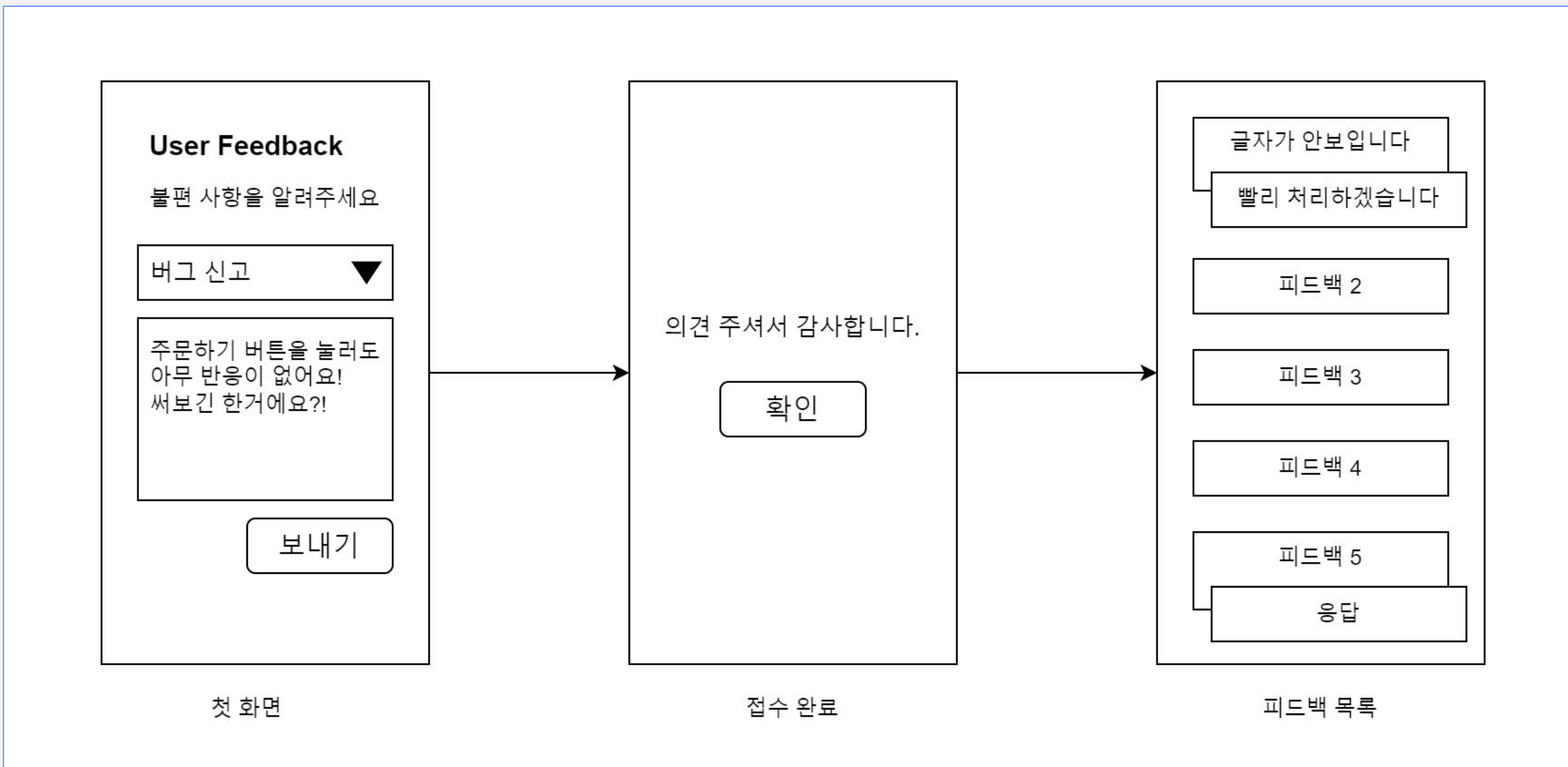
MVP라도 제대로 만들어야지

“혼자 하는건 안말리겠는데, 그게 왜 필요하죠” → FrontEnd 동료 1명 설득과 협업

“vote에 따라가느라 우리 서비스를 못만들어요” → upvote 기능 없앴

“사용자 의견을 방치하는게 더 욕먹어요” → 공개 포럼 없앴

대략 이런 모습



빛도 못보고 운영에서 막힘 TT

“운영할 자신이 없어요”

“비용 대비 효과를 모르겠어요”

피드백 관리는 어디서 하는게 맞을까?

- CS - 앵무새 같은 답변만 달릴 것 같다
- 운영 - 다른 일도 바쁘다
- 사업 - 저희는 숫자만 봅니다
- 프로덕트 - 제가요? 왜요? ㅎ



“나의 시야는 좁았구나..”

세번째 시도

2021년

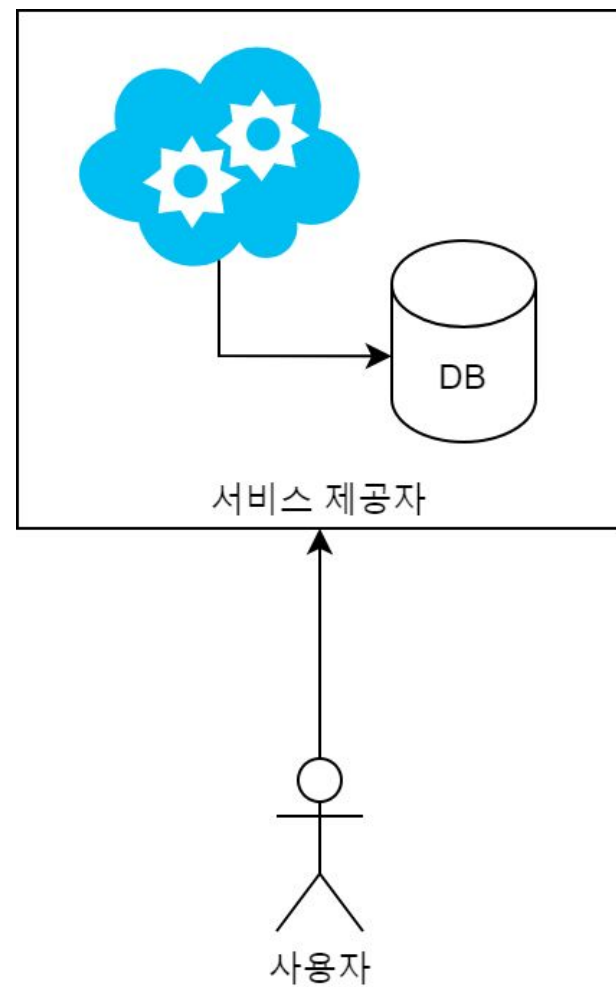
오픈소스로 릴리즈하자

우리 팀은 못써도 다른 누군가는 쓰겠지...

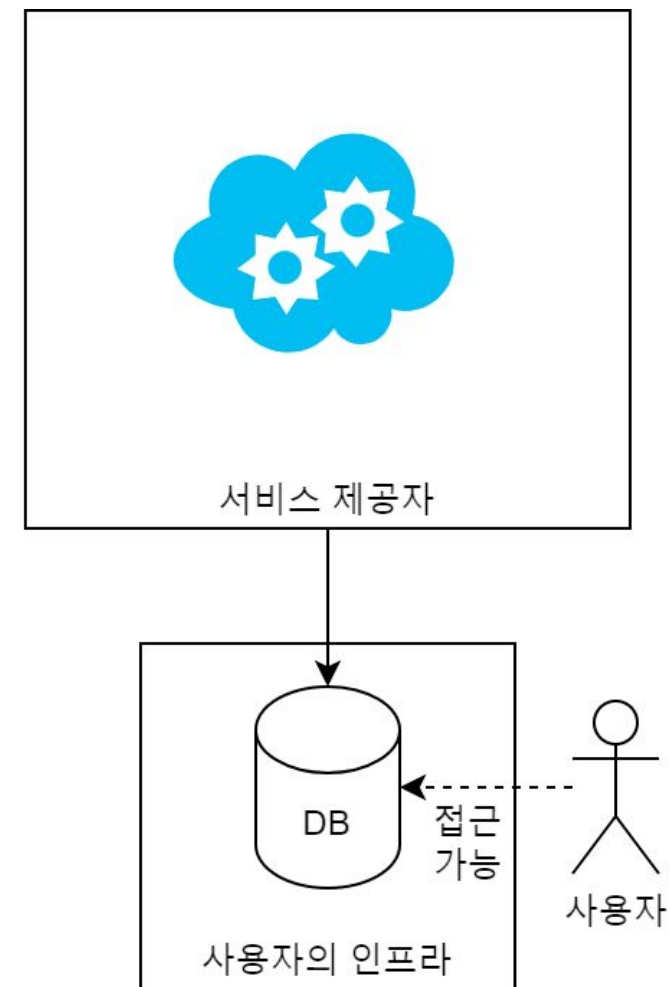
중요 설계 방침

1. 쉬운 설치 (Docker, npx, Web Admin) → 기술사용자 친화적
2. 데이터 레이어 분리 (Your Data is Yours) → 개인정보 문제 해소

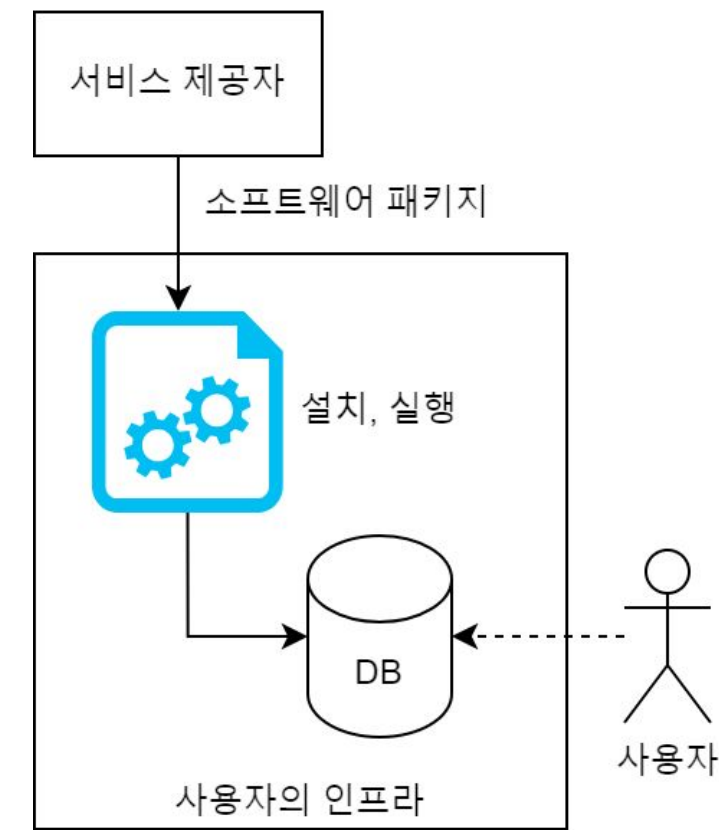
어플리케이션과 데이터베이스의 구성 패턴



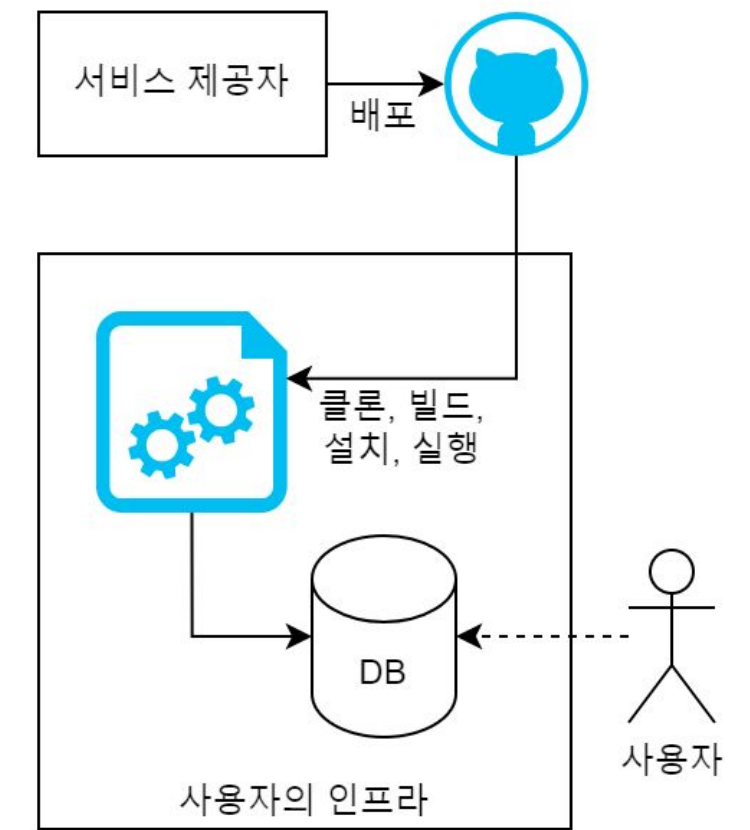
1. 일반적인 SaaS



2. DB 선택 가능

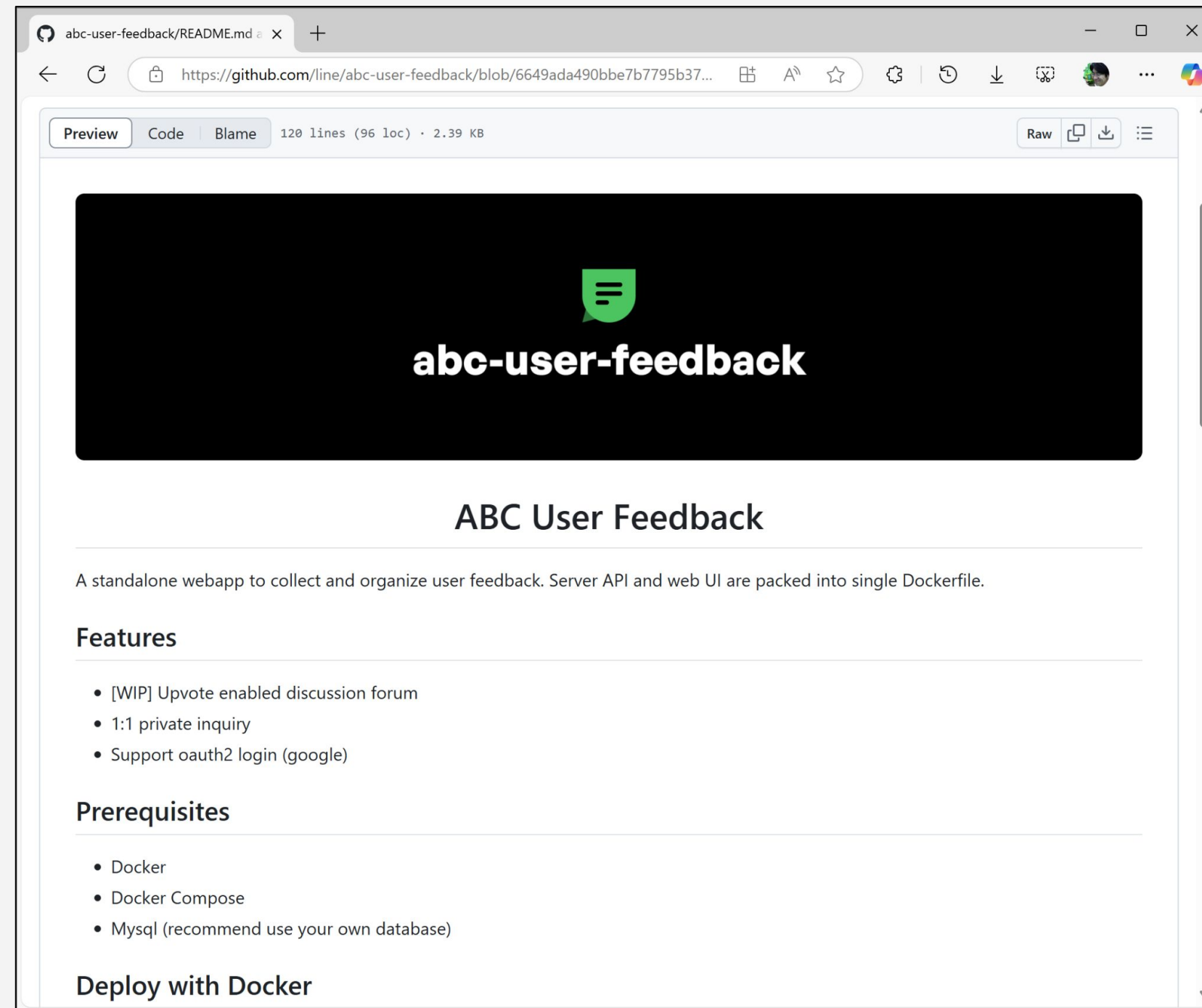


3. 설치형 소프트웨어



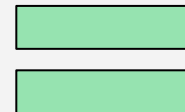
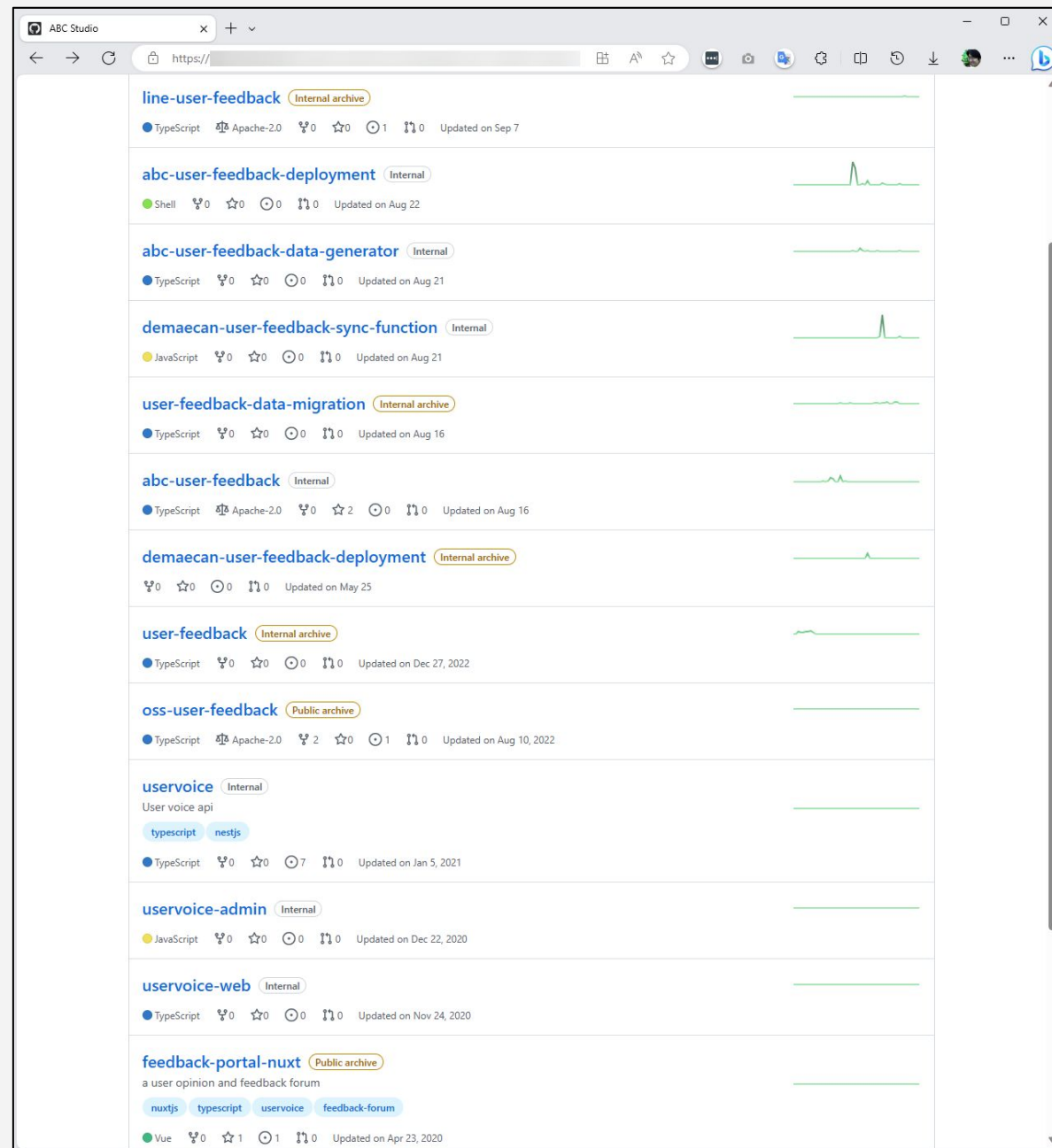
4. 오픈소스

ABC User Feedback 이름으로 LINE GitHub 공개



README도 공들여서...

만들다 부수기만 10여번...



질کم대다 Archive 처리 수두룩...



현실세계의 빈집만 가득한 꼴

네번째 시도

2022년

드디어 프로덕트 하나에 적용했다

어느덧 상장회사의 경영진이 됨

첫 프로젝트는 Delivery 영역의 쇄신

- 배달원의 만족도를 KPI 중 하나로 설정 (그 외에 배달시간, 건당 단가 등)
- KPI에 따라 필요한 전략으로 ABC User Feedback 적용
- “이것은 그저 Apache 라이선스의 소프트웨어일 뿐”

9:41

← ご意見・不具合報告

カテゴリ

意見

内容

このアプリに関するご意見や不具合について、詳しくご記入ください。今後のアプリ改善に役立させていただきます。

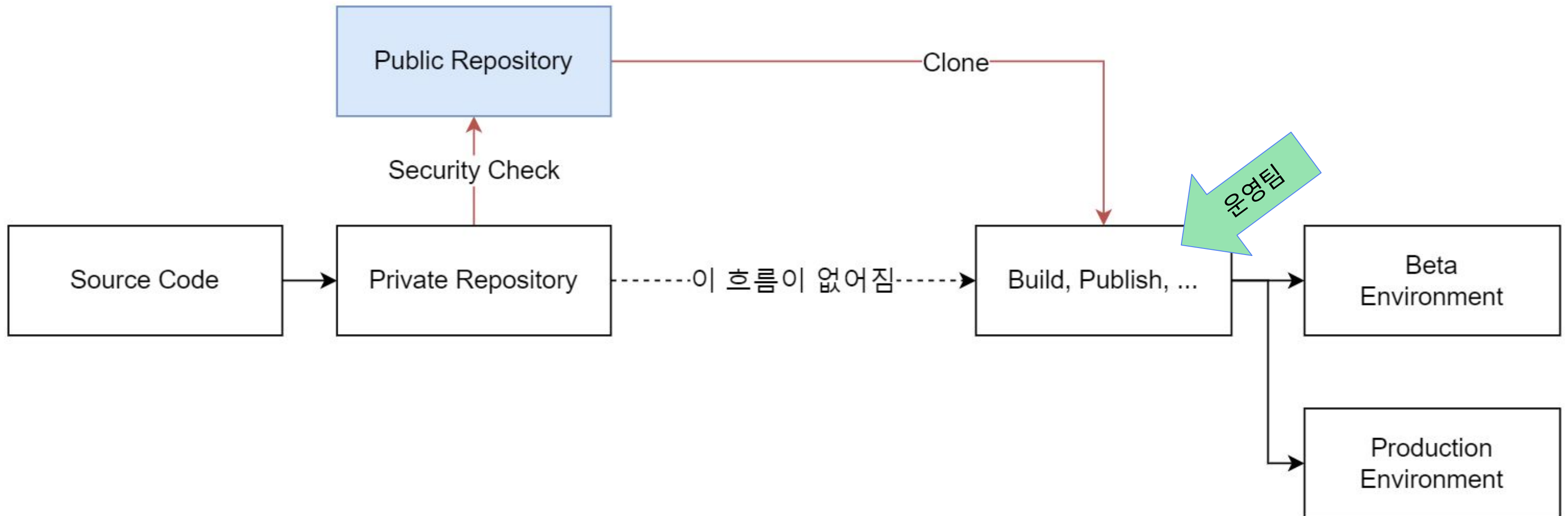
※こちらにいただいたご意見や不具合に対する返信は行っておりません。返信を希望するお問い合わせは、gig-delivery@demae-can.co.jp宛にメールをお願いいたします。

☒ 改善が必要な環境確認のため、ログインID・端末情報・OS・OSバージョン・Appバージョンの取得に同意します。

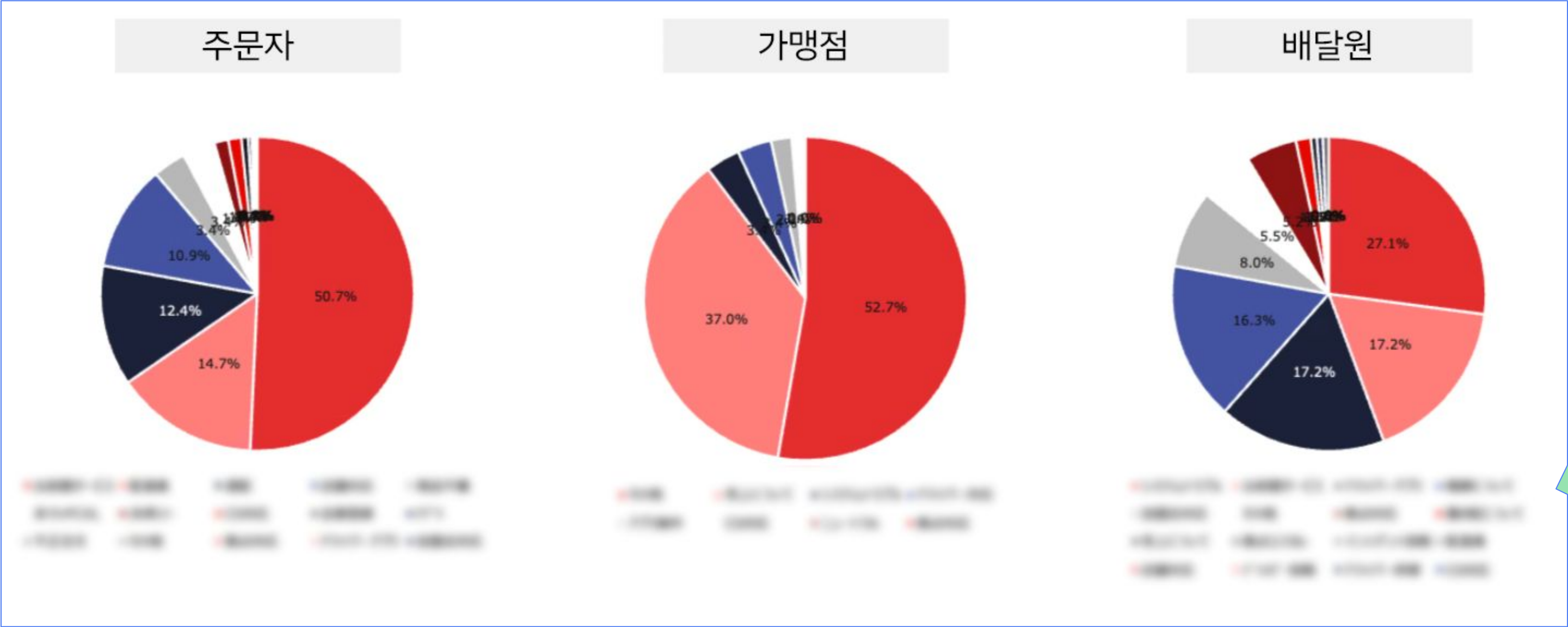
送信

고작 필드 두 개 뿐이지만 감격...

오픈소스 배포



효과가 슬슬 보이기 시작



중요

유저 피드백을 충실히 반영할 수록
불만의 카테고리가 다양해진다

다섯번째 시도

2023년

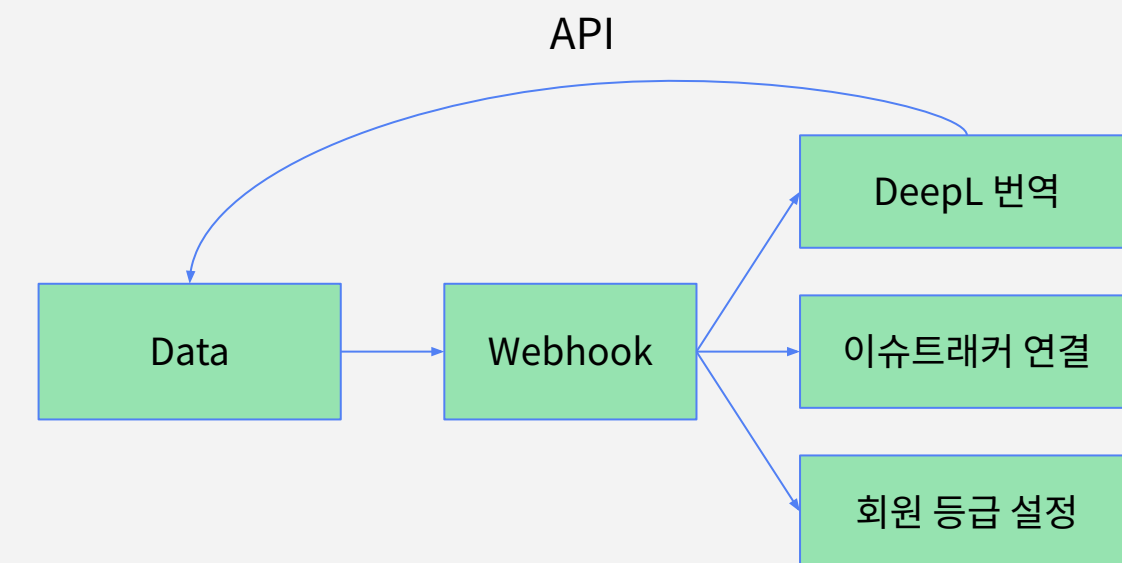
다시 바닥부터 새로 만들기

작고 귀여운 게시판이었을 뿐, 여러 프로덕트로 확장하려니 한계가 발생

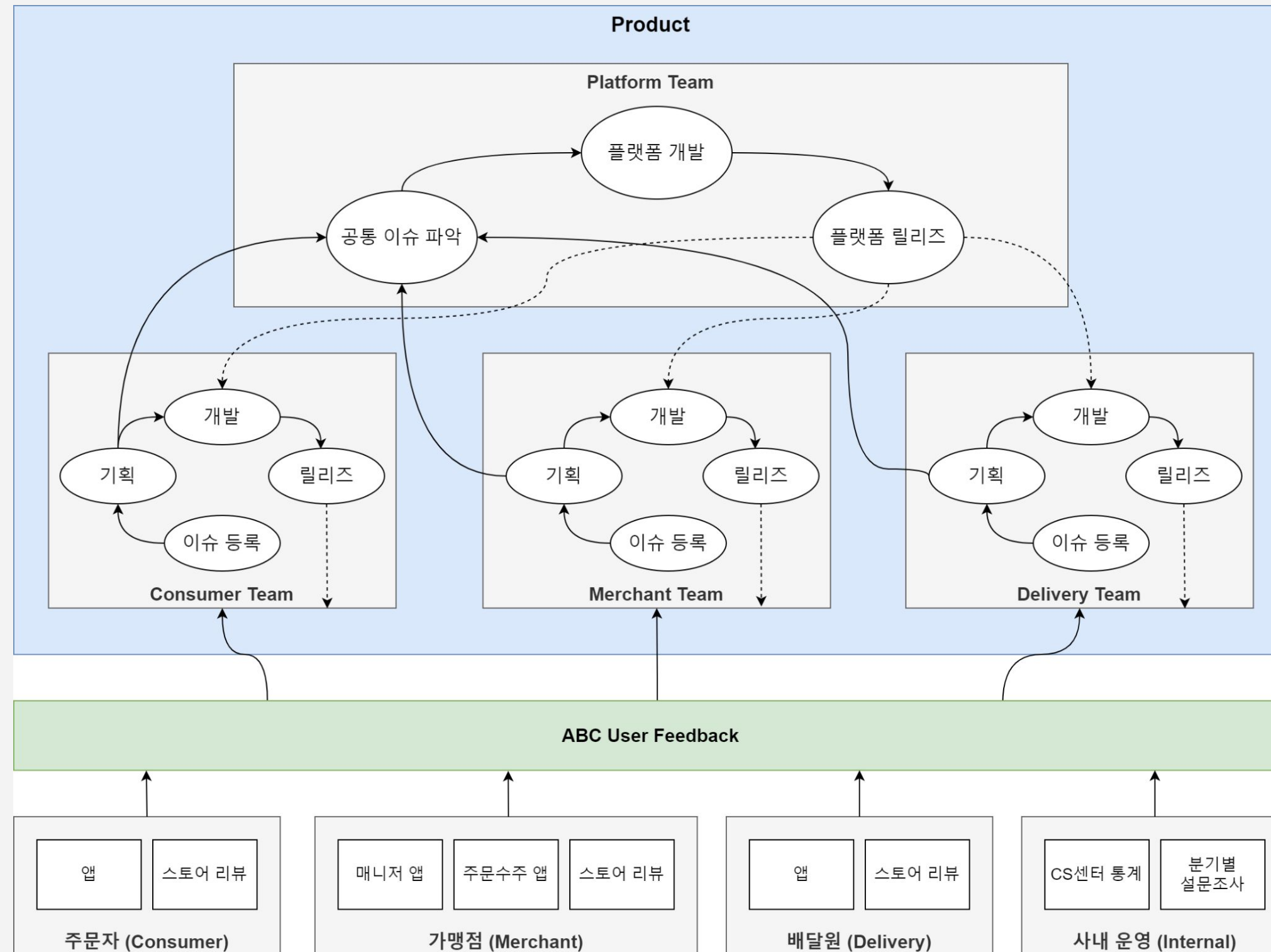
데이터 마이그레이션 스크립트가 필요할 정도로 과감하게 재구성

- ‘프로젝트’ 단위로 생성
- 검색 강화를 위해 Elasticsearch를 옵션으로 제공
- API, Webhook, 다국어 지원

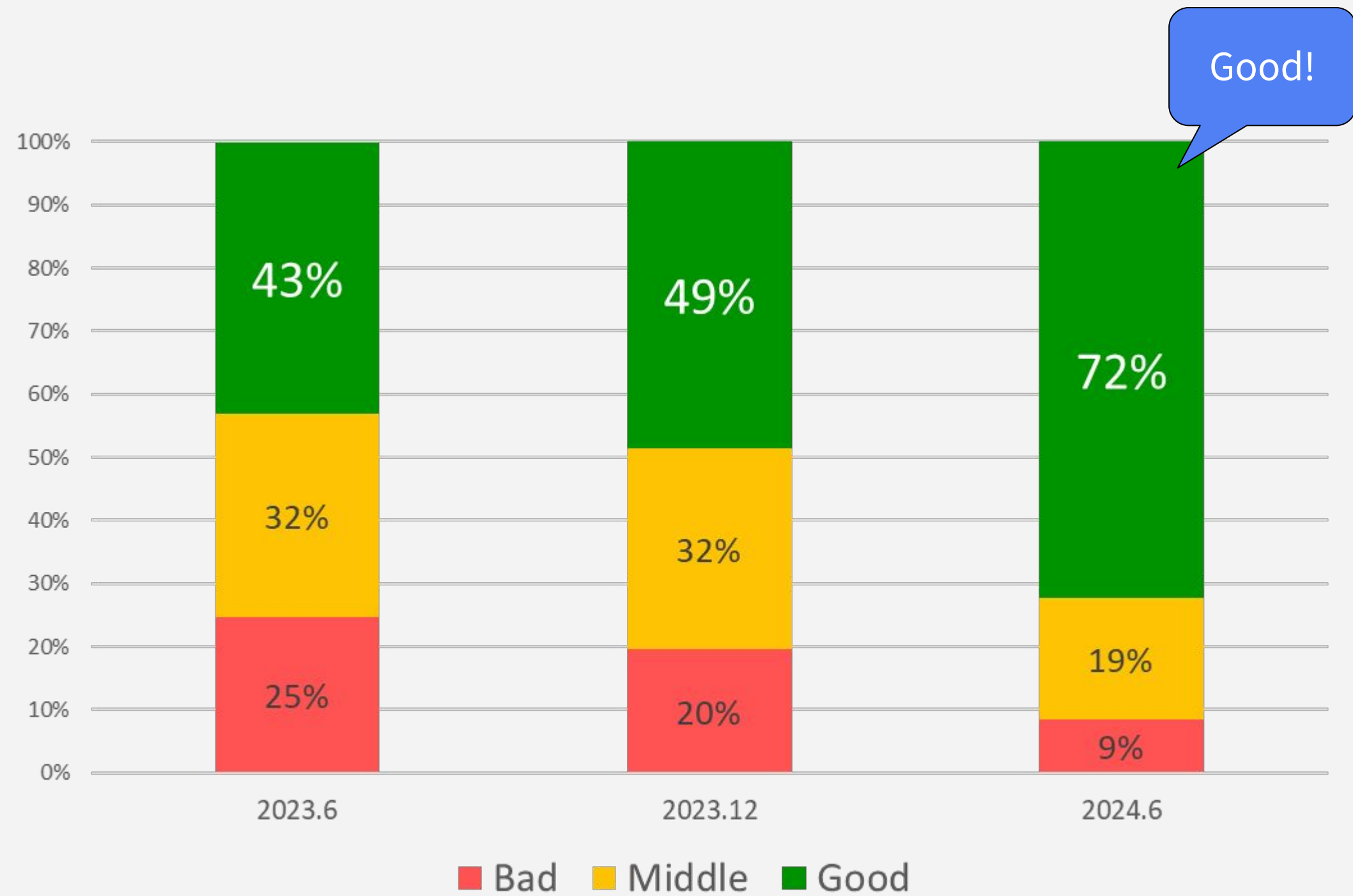
5명의 TF 팀으로 재구성 - 팀장, 기획 1, FE 1, Design 1, BE 1



오픈소스 하나로 사용자 의견에 민감한 회사가 되었다



결과로 효과를 증명



언젠가부터 이곳저곳에서 사용하기 시작

귀하신 회사께서 누추한 곳에 어찌...



+ 스타트업 10여곳



따봉에 눈길이 가는게 인지상정 ㅋ

오픈소스로 해결했나?

01

개인 정보 취급

기술사용자가 DB를 직접 관리하므로 해결

서비스 내부의 개인정보와 동일한 취급성 확보

02

회사의 의존성 제거

Apache 2.0 라이선스 채택으로 관계사 모두
지적재산권 법률검토 없이 사용 가능

필드 편집 기능으로 비즈니스 의존성 제거

하나의 회사에서 여러 프로젝트 적용 가능

03

확장성

API, Webhook 제공으로 연동성 강화

SSO, OAuth 지원 (엔터프라이즈급 기능)

외부 PR 8건

- 설치 경험 강화 (SMTP option, pnpm)
- API, UI 등의 사용성
- 문서화 보충
- 독일어 문법 수정

배울 점 요약

1년차 - 내가 만들고 싶은 소프트웨어에 **공감**하지 않는 사람은 많다

2년차 - MVP를 만들어도 **운영**이 따르지 않으면 어렵다

3년차 - 오픈소스로 릴리즈할 때 **쉬운 설치 경험**이 중요하다

4년차 - 릴리즈한게 끝이 아니라, **실전에 적용**해보는게 중요하다

5년차 - 소프트웨어 하나가 **의미를 만들기까지 5년은 걸리더라**

나도 해볼까?

오픈소스를 시작할 때
가장 중요한 것 → 마음가짐

첫째, 아무도 나의 코드에 관심 없습니다

GitHub 수억개의 Repo에서 나의 작고 귀여운 코드에 관심 가질 사람은 없다.

부끄러워하지 말고 공개하세요.



제가 3개월간 짠 코드를 공개합니다!

성숙한(?) 모습



별거 없다.
그저 배포방식 중 하나일 뿐.

Git Push 딸깍

둘째, 따봉★은 아무 의미 없습니다

한 때 즐겨찾기 모아놓은 “Awesome 시리즈”가 유행했음

- Awesome Android, Awesome Kotlin, Awesome LLM, ...

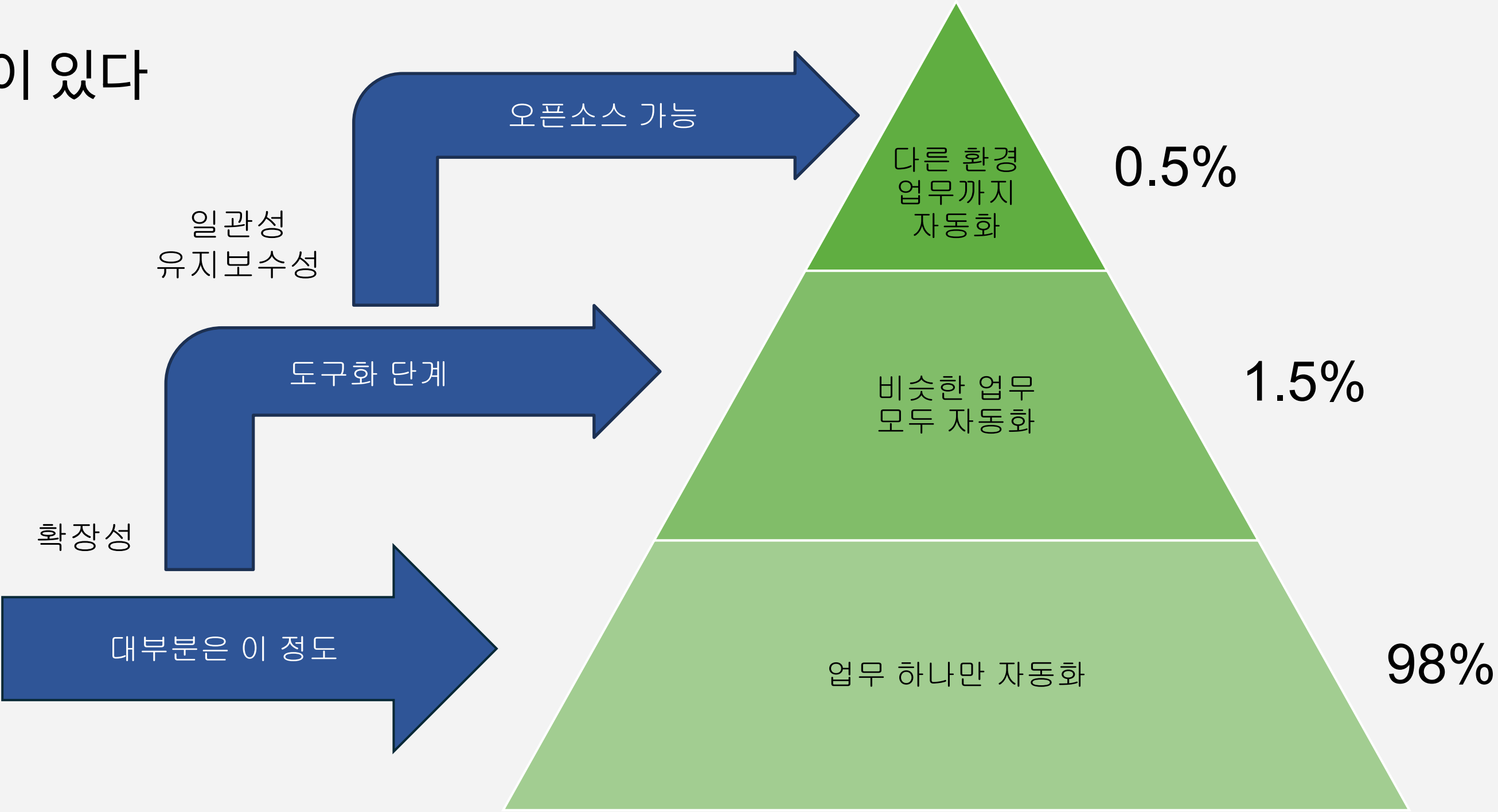
외부 반응에 집착하면 소프트웨어의 깊이를 잃어버린다.

결국 힘있는 오픈소스는, 규모가 어떻든 README에서 선언한 기능을 **충실히 실현**하면서도
기술 사용자가 직접 설치해서 **사용하기 쉽고, 꾸준히 가꾸는** 소프트웨어다.

가볍게 시작하기

자동화 하나부터 시도해보세요

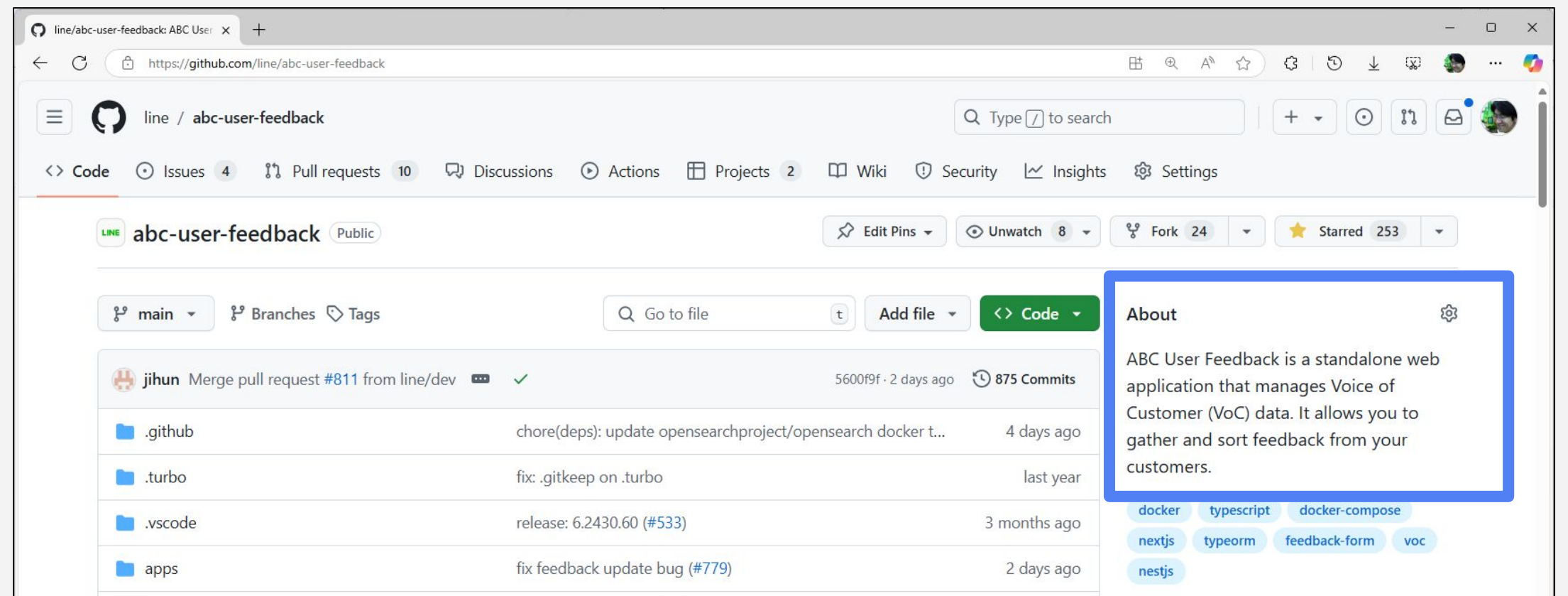
자동화도 나름의 수준이 있다



이상한 이름 짓지 말고 목적을 명확하게

기능이 곧 이름, 이름이 곧 기능!

명확한 세계관 설정



- 데스크톱 소프트웨어의 예: Microsoft PowerToys is a set of utilities for power users to tune and streamline their Windows experience for greater productivity.
- SDK의 예: Flutter is Google's SDK for crafting beautiful, fast user experiences for mobile, web, and desktop from a single codebase.
- 데이터베이스의 예: OrientDB is an Open Source Multi-Model NoSQL DBMS with the support of Native Graphs, Documents, Full-Text search, Reactivity, Geo-Spatial and Object Oriented concepts.

좋은 훈련

- 로직 없이 설정파일 부터 작성해보세요

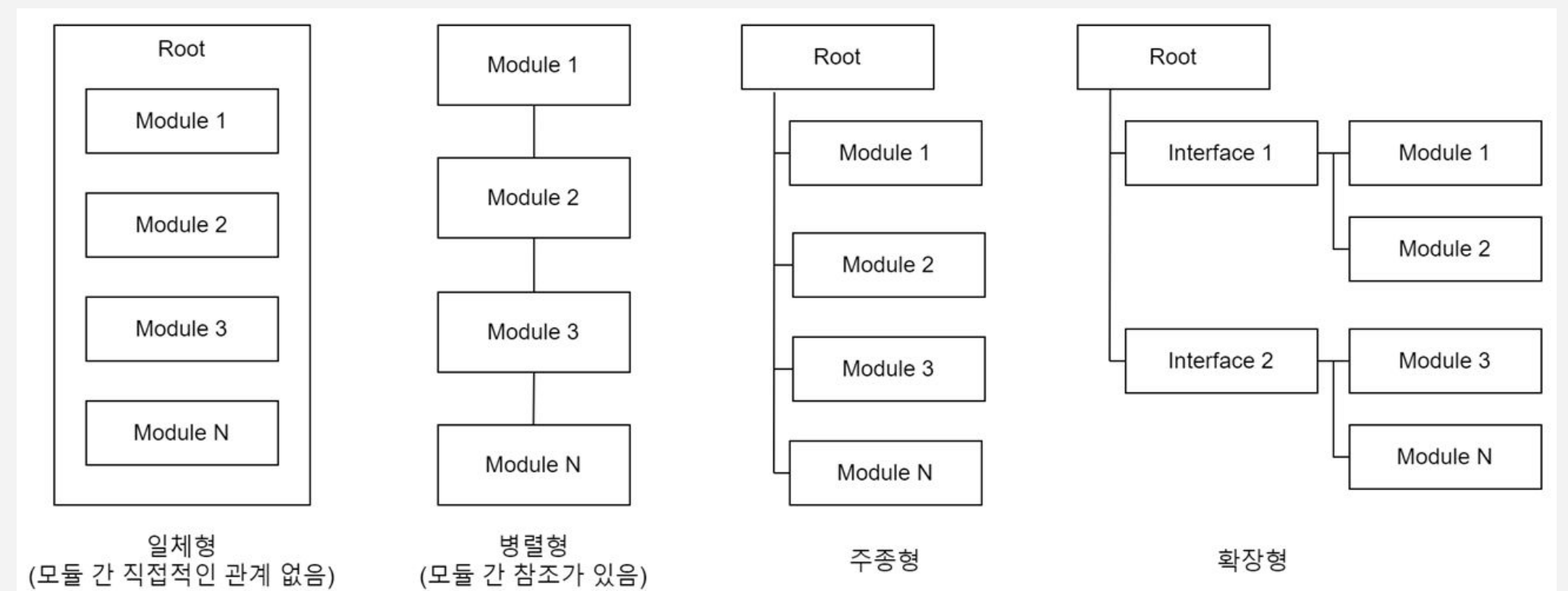
```
github-pr-webhook-to-slack-notification:  
  github:  
    url: https://sample.github.com  
    username: sample-name  
    authtoken: sample-key  
  slack:  
    url: sample-webhook.slack.com  
    authtoken: sample-slack-key  
  template:  
    default: sample-template.txt
```

어느 자동화 스크립트의 설정파일 예시

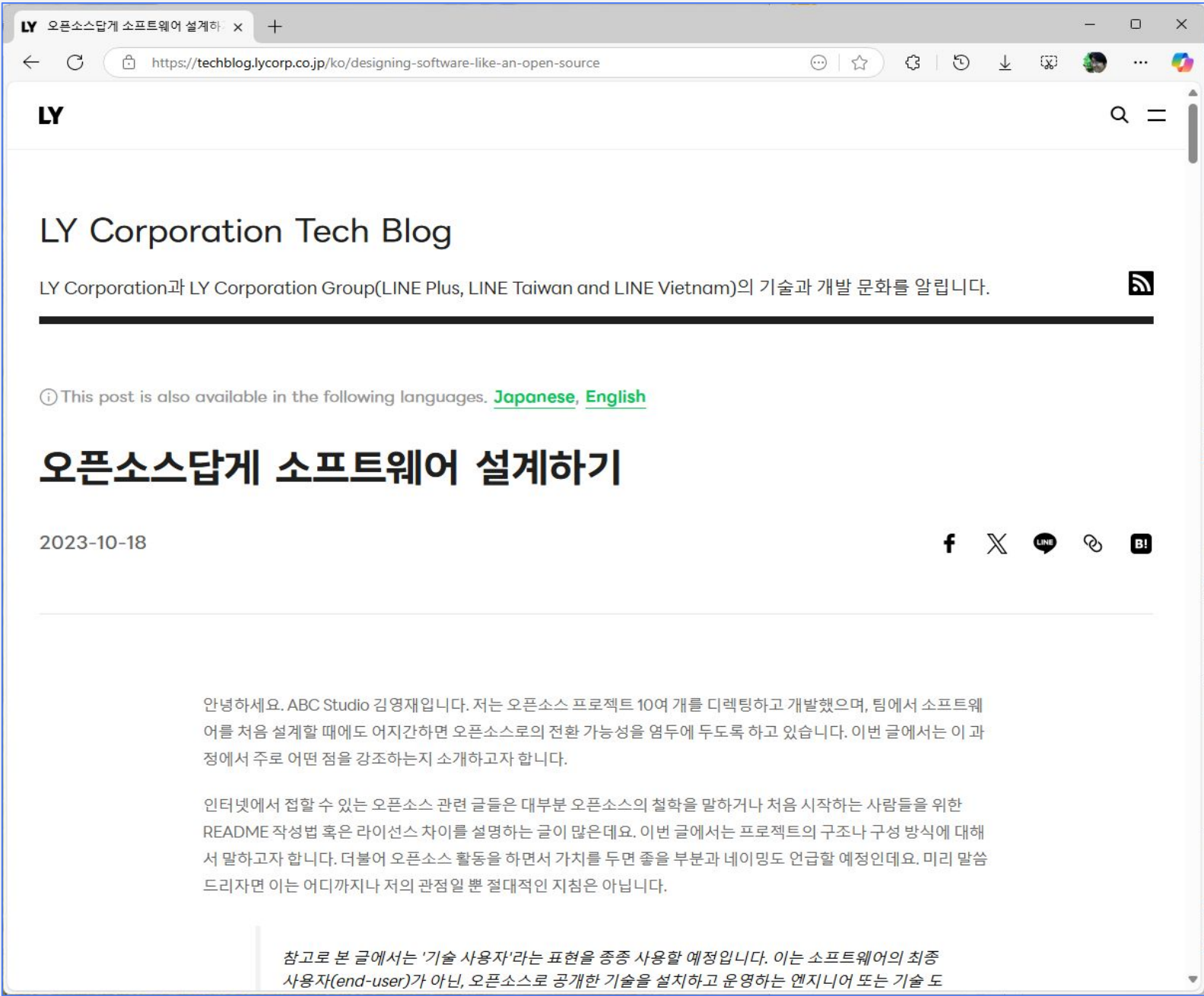
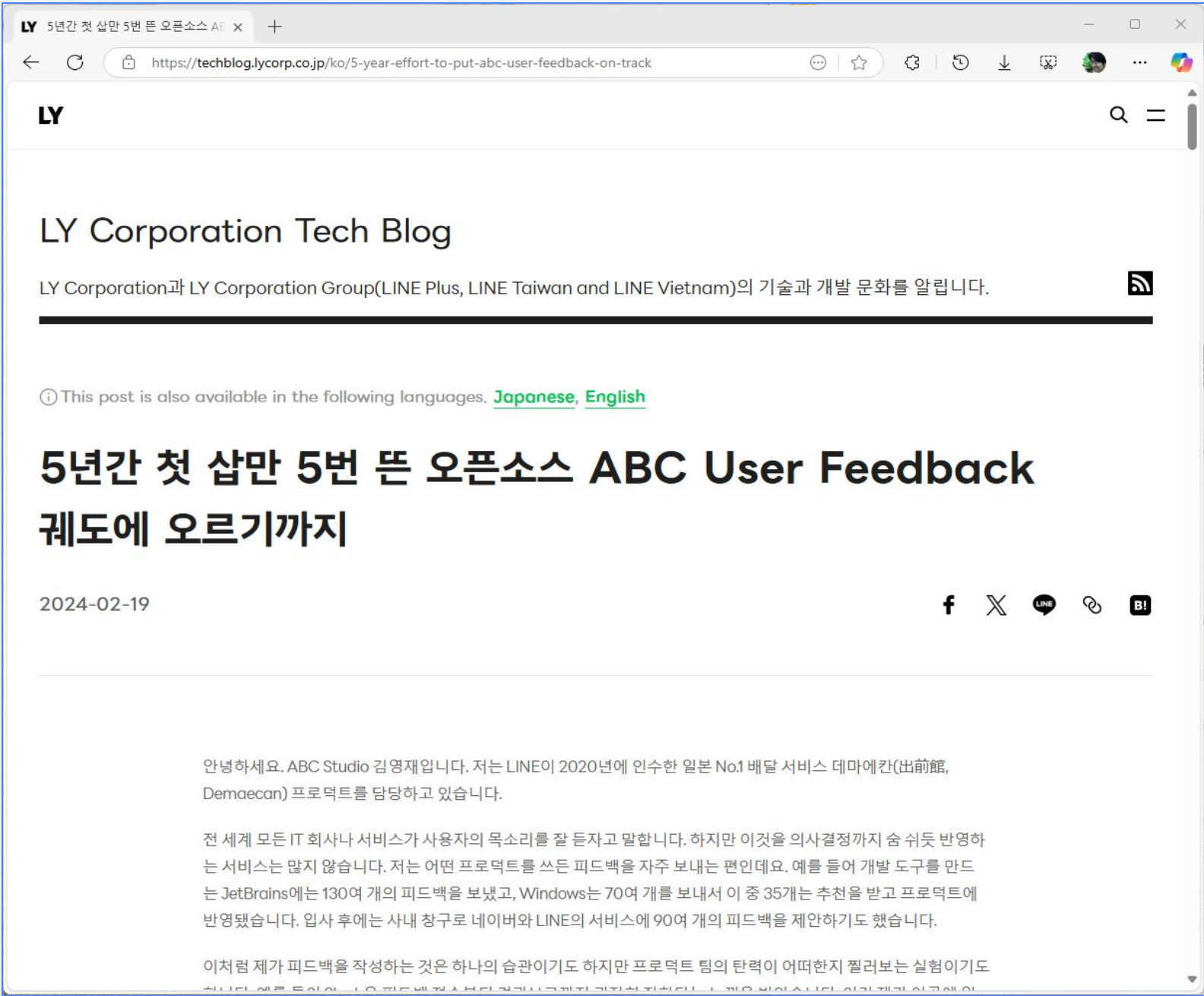
소프트웨어 유형과 구조 - 보다 간단하게 만들려면

내가 만드는게 어떤 유형과 구조가 될지 미리 생각하자

- 유틸리티 라이브러리
- 데이터 의존 라이브러리
- API/CLI 애플리케이션
- 최종 사용자(End User)가 있는 애플리케이션



오늘 발표를 더 깊게 알고 싶으면 - LY 기술블로그



<https://techblog.lycorp.co.jp/ko/5-year-effort-to-put-abc-user-feedback-on-track>

<https://techblog.lycorp.co.jp/ko/designing-software-like-an-open-source>

지치지 않고 시도하는 방법을 알고 싶으면

인프콘 2024

- 성장하지 않아도 괜찮습니다
- <https://www.youtube.com/watch?v=uWDRw6LzeCM&t=853s>

성장하지 않아도 괜찮습니다 | 인프콘 2024

인프콘 2024

INFCON 2024

지금 하는걸 더 많이 하자 - 일반화

- 기본 컨셉 - 단 한번이라도, 통료가 나처럼 일하게 하자
- 나의 스타일이나 관점을 상대가 그대로 하도록 하려면?
지식을 깎고 깎아서 일반화 할 필요가 있다
- 쉬운 방법 - 회사 블로그에 글쓰기
개인 SNS/블로그 - X
회사 블로그 - O **중요**

그냥 일하는 1인

어쩌서인지 더 많이 일하는 1인

시간

인프콘 2024

인프콘 inflearn - 30 / 52

Thank You



김영재

LINE / 기술임원 (Executive Officer of Tech)

youngjaekim@linecorp.com / youngjae.kim@gmail.com