

# 숨겨진 인디게임을 찾아

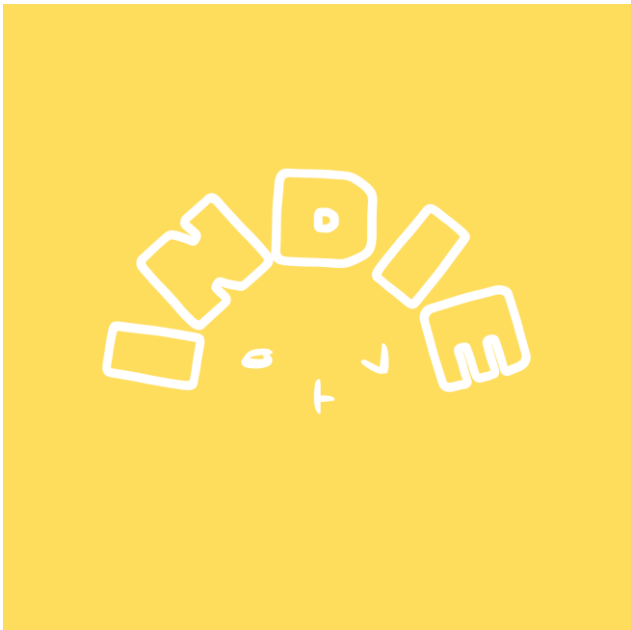
## IndieAn

RecSys-09

좋아요구독추천 | 김찬호 배성수 이지훈 정소빈 조원삼



# 목차



## 1. 팀 소개

## 2. 프로젝트 소개

0. 기획의도

1. 시연

2. 데이터 및 DB

3. 백엔드

4. 모델

5. 프론트엔드 및 Web API

6. CI/CD

7. 효과 및 한계점

## 3. 이후 프로젝트 관리

## 4. 협업방식 소개



# 팀 소개



김찬호

Data  
Modeling  
협업관리



배성수

Backend  
Database



이지훈

Backend  
CI/CD  
Data



정소빈

Frontend  
Web API

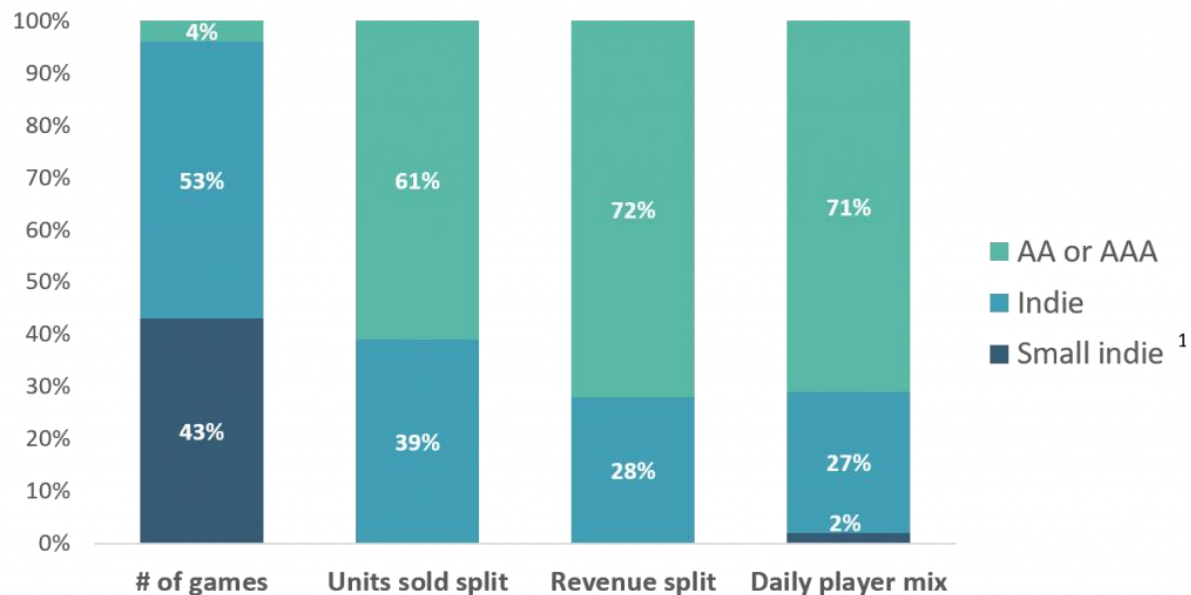


조원삼

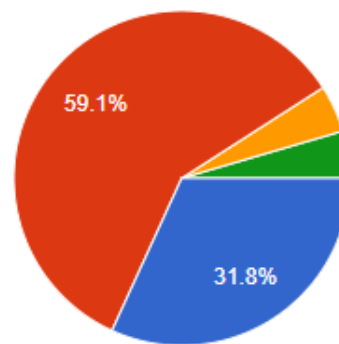
Modeling  
CI/CD  
발표



## Distribution of Steam Games Based on Publisher Type



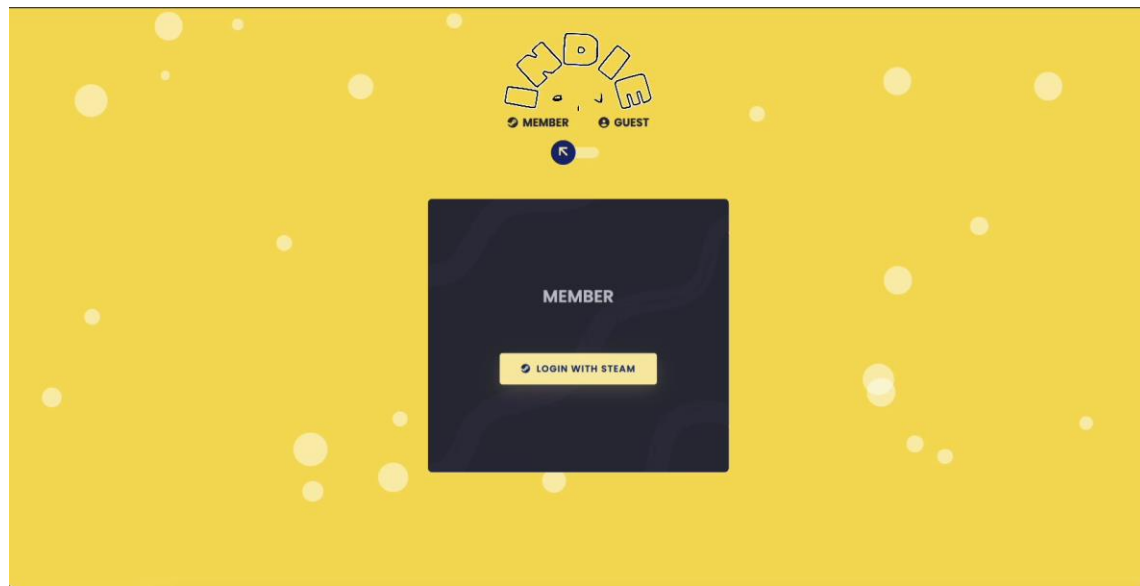
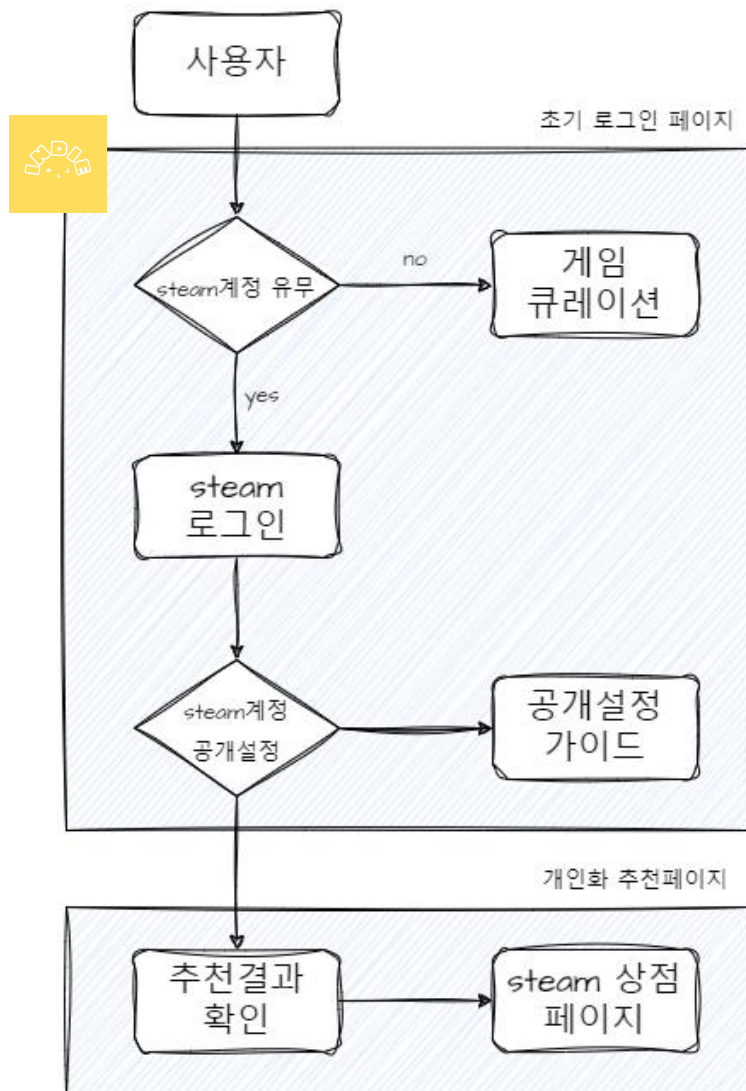
1 – Indie games with <300 units sold



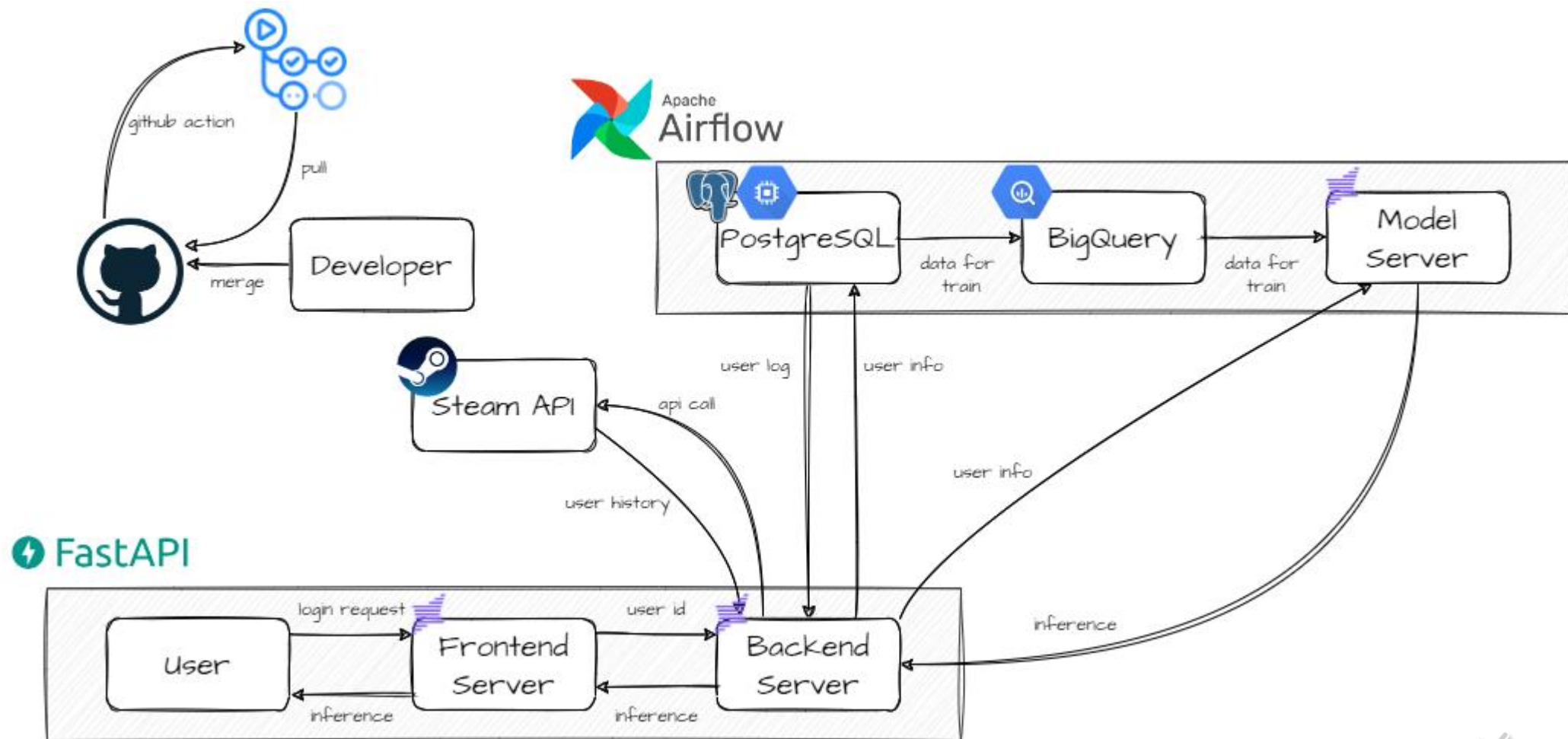
- 유명하고 많은 사람들이 플레이하는 아이템
- 유명도와 상관없이 나에게 잘 맞을 아이템
- 유명하고 나에게 잘 맞을 아이템
- 두 유형이 적절히 섞인 것, 3 : 7 정도면 좋을 것 같아요



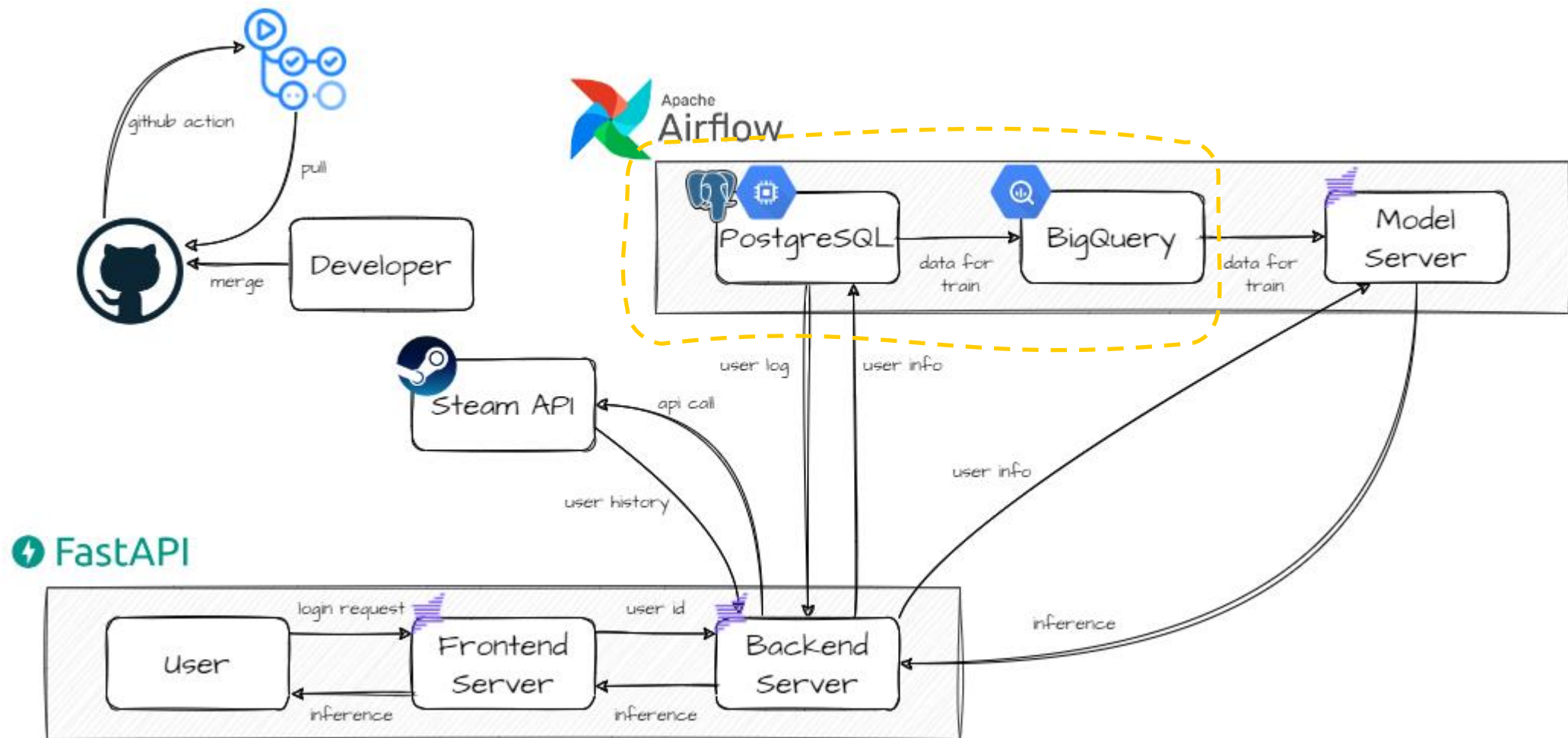
# 프로젝트 | 시연



# 프로젝트 | 구조도



# 프로젝트 | 데이터 및 DB



## 프로젝트 | 데이터 및 DB

데이터 분류	데이터 설명	수집 경로	활용	크기
유저-아이템 상호작용 데이터	Steam Video Game and Bundle Data UCSD에서 수집한 Steam 데이터셋 중 유저-아이템 상호작용 데이터	UCSD Steam Dataset	추천 시스템 모델 학습	5153209 × 5
게임 아이템 메타 데이터	2022 Steam Games Kaggle에서 수집한 2022.11 기준 게임 메타 데이터	Kaggle Steam Dataset	아이템 필터링, 큐레이션, 콘텐츠 기반 추천 시스템 모델	55691 × 23
유저 데이터	유저의 ID를 받아 Steam에서 API로 가져오는 데이터로 유저 프로필 데이터, 유저-아이템 상호작용 히스토리 데이터	Steam API	개인화 추천시스템 Inference	-





# 프로젝트 | 데이터 및 DB



## 데이터 레이크

유저-아이템 상호작용 데이터  
아이템 메타 데이터  
유저 데이터  
서비스 로그데이터

## VM 인스턴스

필터	속성	이름 또는 값 입력
<input type="checkbox"/>	상태	이름 ↑ 영역
<input type="checkbox"/>		<u>steamdb</u> asia-northeast3-a



## 데이터 웨어하우스

정제된 일부 유저-아이템 상호작용 데이터  
아이템 메타 데이터

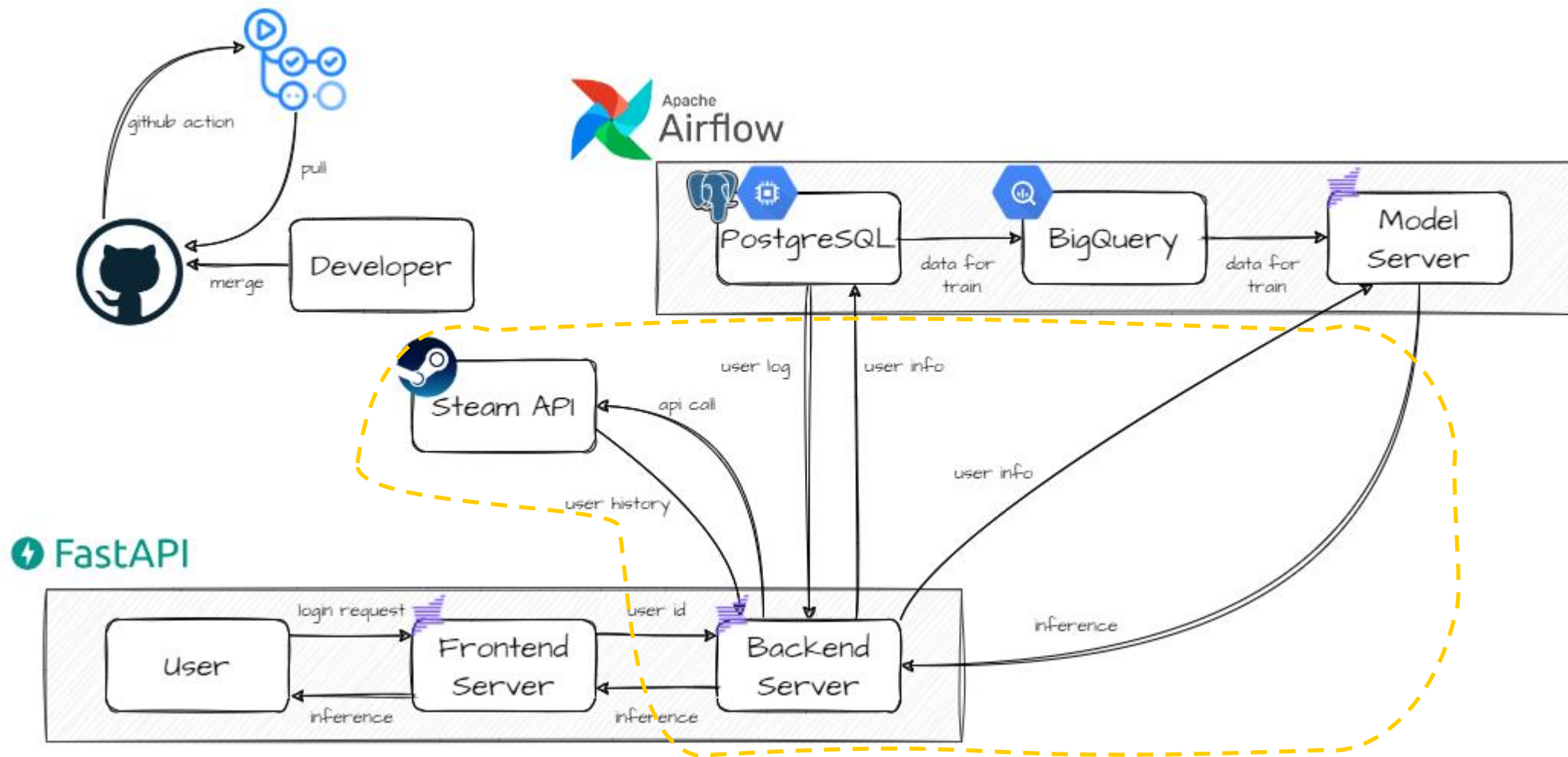
탐색기 + 데이터 추가

game

스키마 세부정보 미리보기 계보

행	App_ID	Name	Developer	Genre
1	1080790	Jack-in-A-Castle	null	null
2	1105150	Reverse Mel Rez/Ru	null	null

# 프로젝트 | 백엔드



# 프로젝트 | 백엔드

FastAPI 0.1.0 OAS3

/openapi.json

## default ^

GET	/user/{userid}	User By Id	▼
GET	/user/histories/{userid}	User History	▼
GET	/user/recommends/{userid}	User Recommend	▼
POST	/login/checkdb	User Login	▼
GET	/game/info/{gameid}	Game Info	▼
GET	/game/selection	Game Selection	▼
GET	/	Get Root	▼

## Game api

DB에서 게임의 정보를 끌어오는 API

## Login api

유저 ID 저장 및 히스토리 전달 API

## Inference api

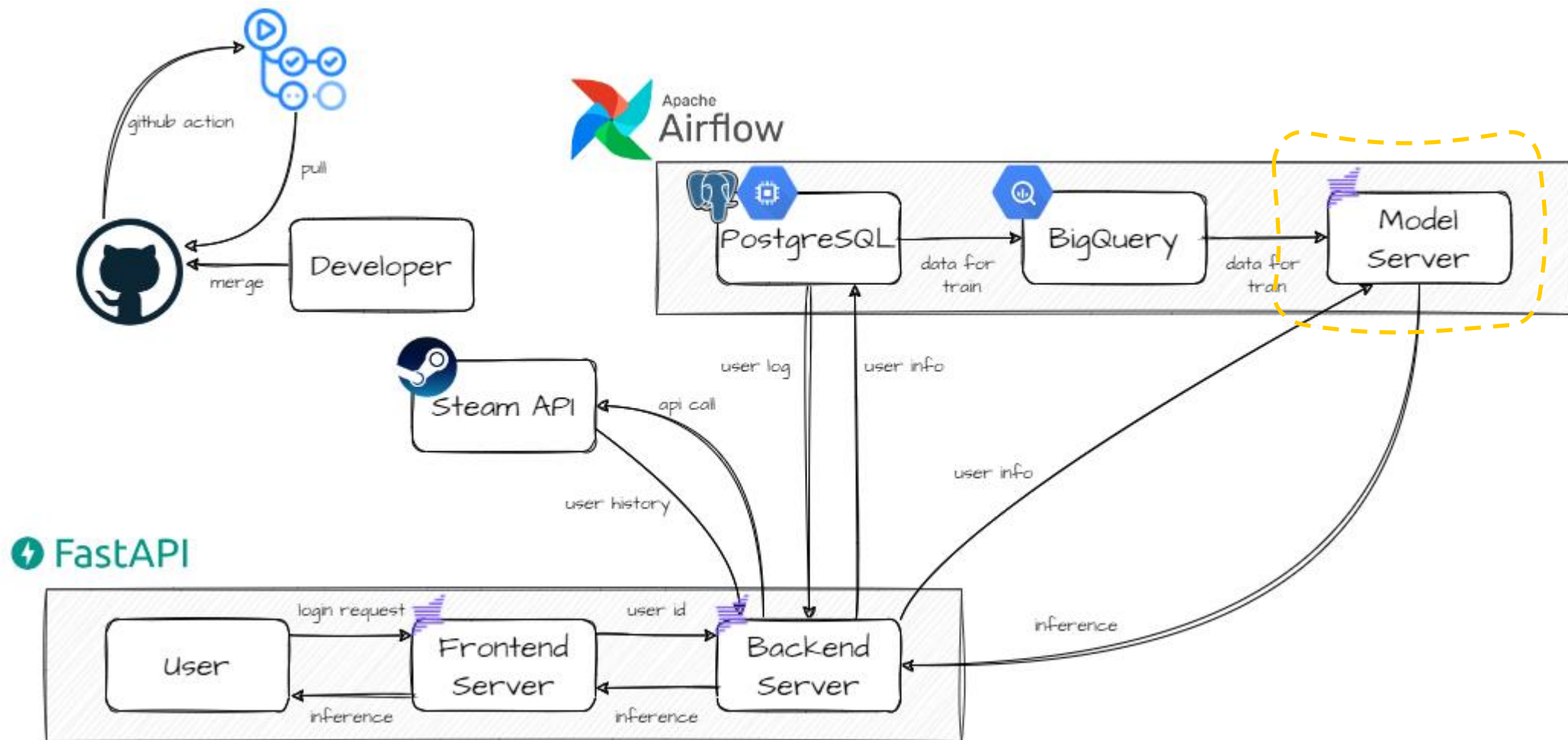
유저 Inference API

## User api

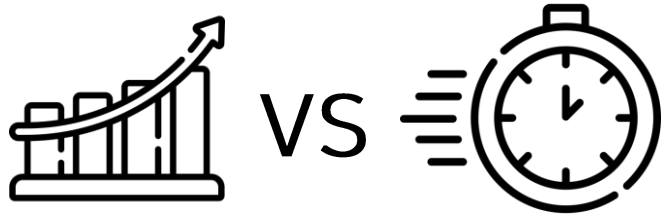
추천결과 Frontend 서버 전달 API



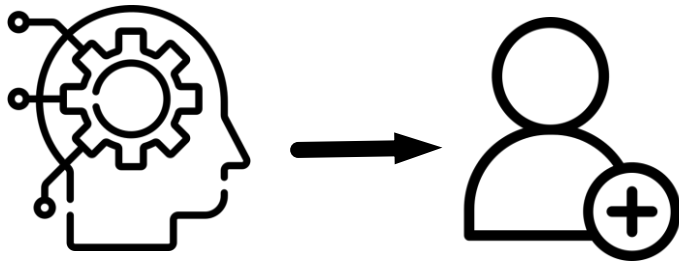
# 프로젝트 | 모델



## 프로젝트 | 모델



유저-아이템 상호작용 데이터를 이용한 협업 필터링 계열의 모델 탐색  
서비스 입장에서 성능과 Inference 속도의 타협 필요인식



새로운 아이템에 대한 Cold-start문제를 해결하기 위한 모델 탐색



### Embarrassingly Shallow Autoencoders for Sparse Data (EASE)

- 은닉층을 없애고, 희소데이터 문제와 Cold-start 문제를 비교적 해결할 수 있는 CF계열의 모델

$$\min_B ||X - XB||_F^2 + \lambda \cdot ||B||_F^2 \longrightarrow \text{Closed form solution의 목적함수}$$

**Input:** data Gram-matrix  $G := X^T X \in \mathbb{R}^{|I| \times |I|}$   $\longrightarrow$   $U^*$ 의 입력데이터를  $I^*$ 로 변환해 사용

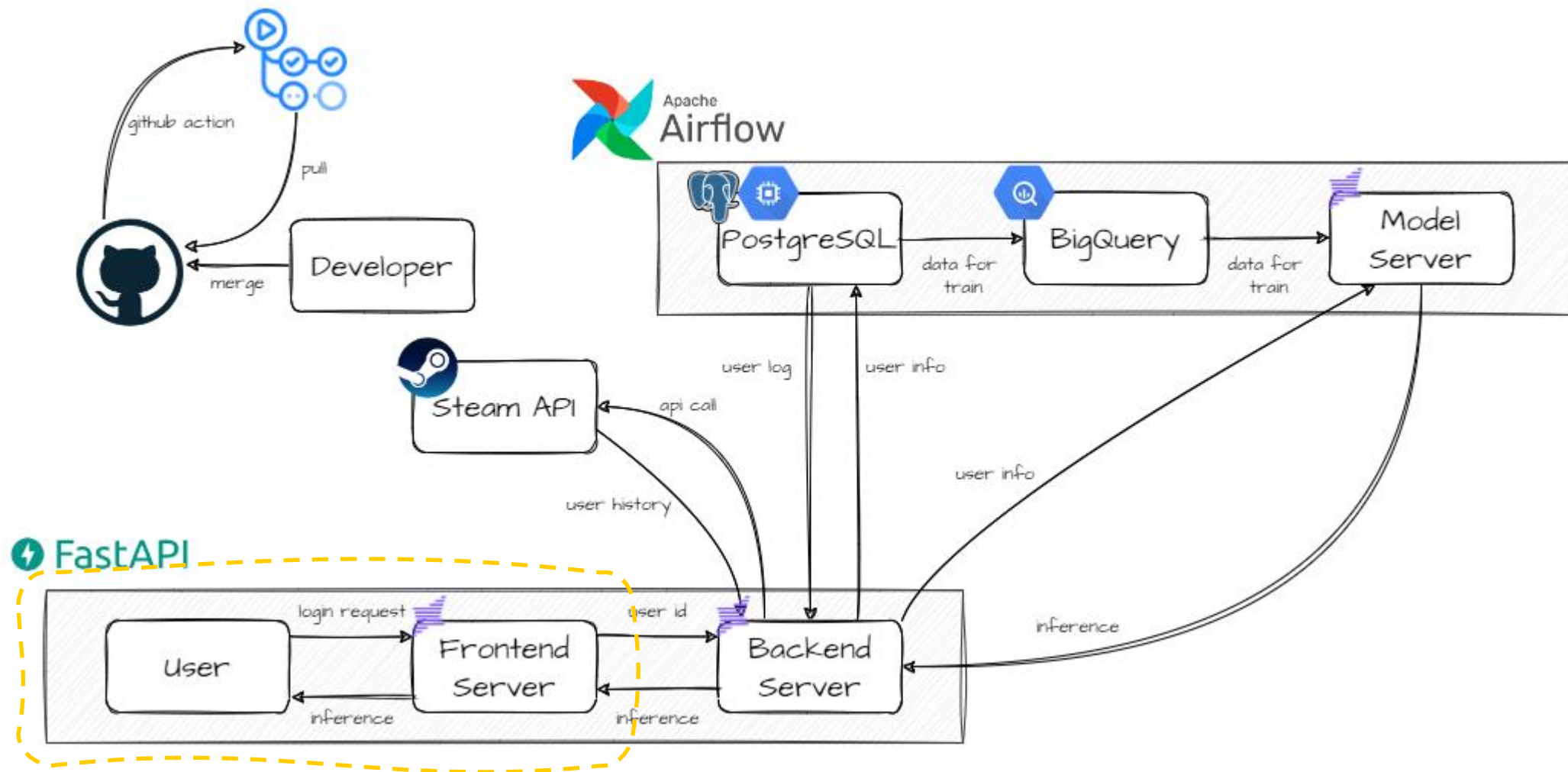


데이터 로드 포함, train&inference 과정 12-13초 이내 / pre-train시, 1초 이내

Precision@10 기준, 전체데이터 0.61 / 정제데이터 0.35의 성능

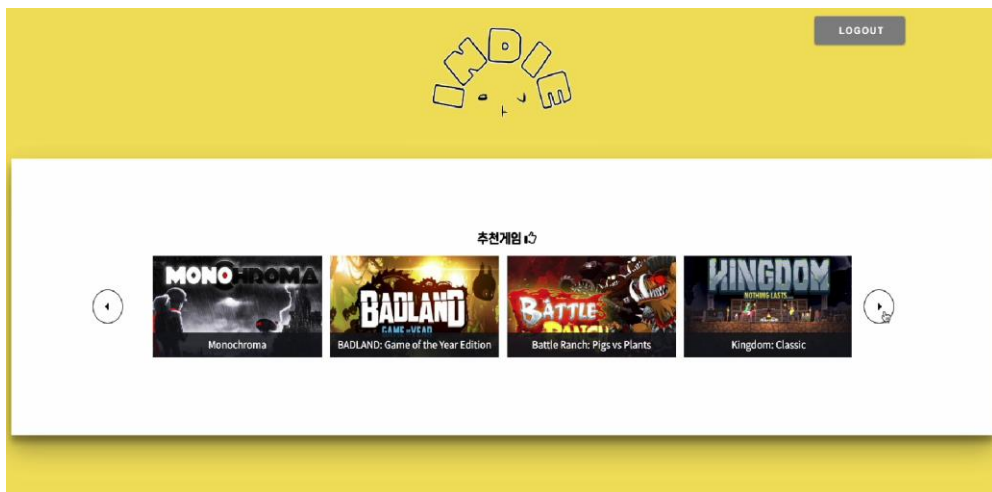
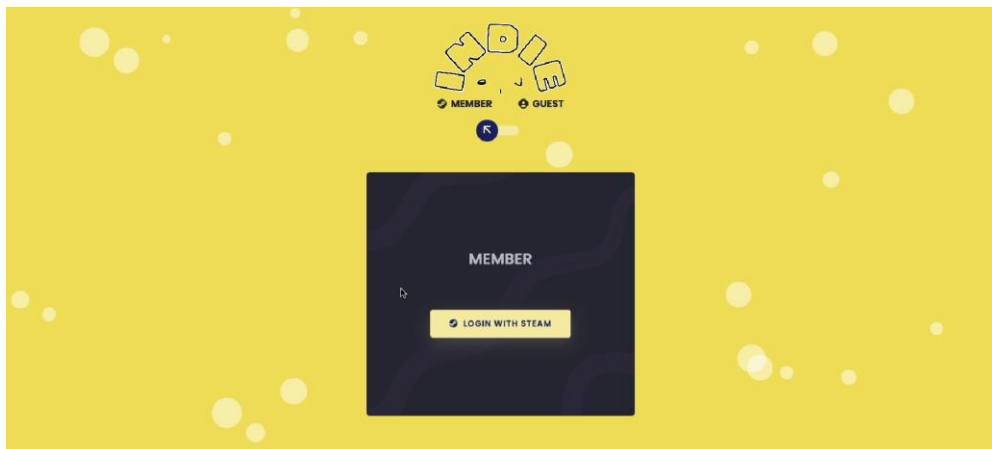


# 프로젝트 | 프론트엔드 및 Web API



# 프로젝트 | 프론트엔드 및 Web API

## 프론트엔드



## Steam Web API

### Steamworks Web API Reference

Steamworks Documentation > Steamworks Web API Reference

Welcome to the Steamworks Web API Reference!

Here you will find a complete listing of the currently supported Web API interfaces and methods.

To learn more about how to use the Steamworks Web API please see the [Web API Overview](#).

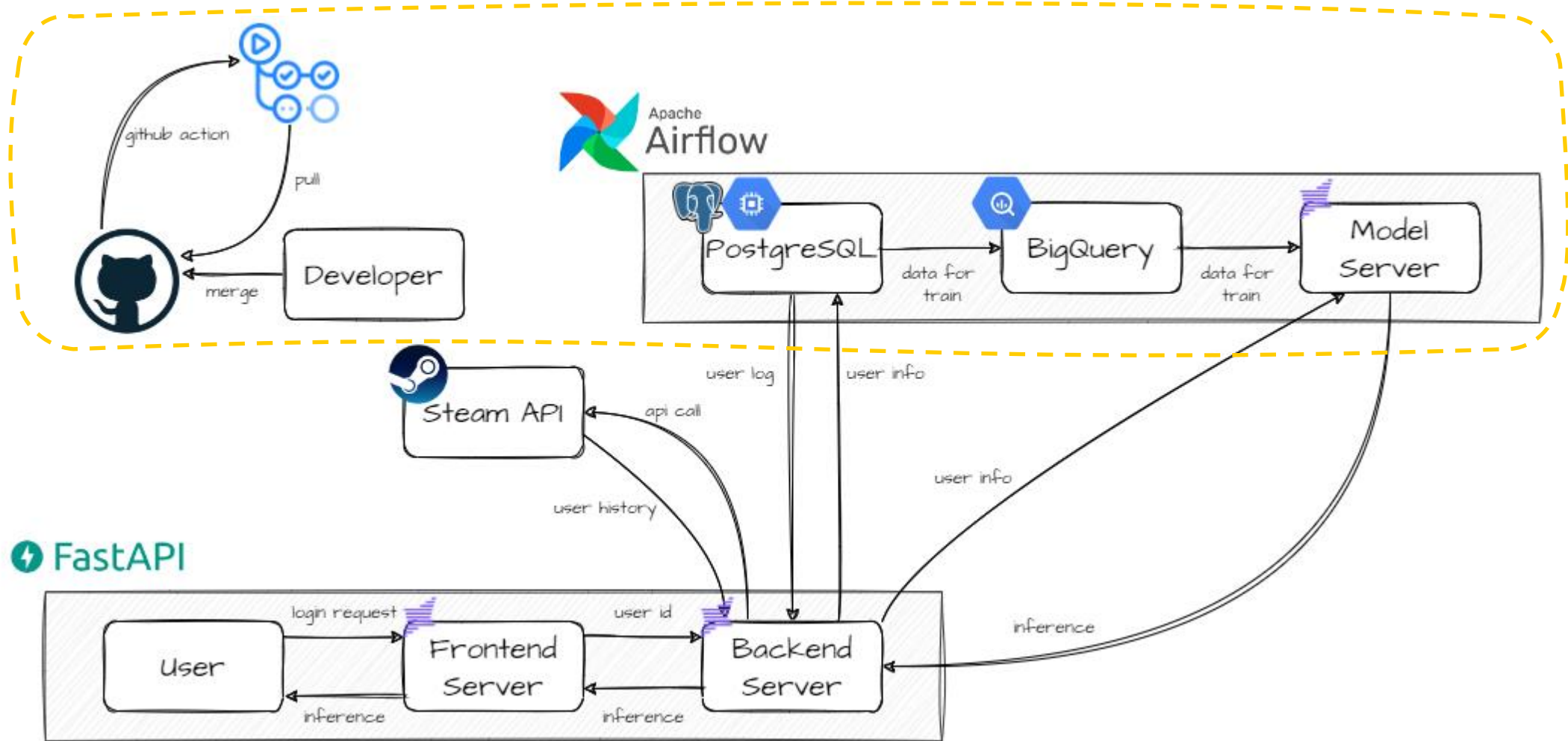
If you are using the client API in your application, then you can find documentation for that on the [Steamworks API Reference](#).

Interface	Description
<a href="#">IBroadcastService</a>	Provides access to Steam broadcasts.
<a href="#">ICheatReportingService</a>	This service allows your game to report cheats and cheaters to the <a href="#">VAC</a> system and provides the toolset behind the <a href="#">Game Bans</a> system.
<a href="#">ICloudService</a>	Steam Cloud API for use with cross-platform saves.
<a href="#">IEconMarketService</a>	Provides restricted access to the Steam Market for partners.
<a href="#">IEconService</a>	Additional Steam Economy methods that provide access to Steam Trading.
<a href="#">IGameInventory</a>	The primary interface to interact with the <a href="#">Steam Economy</a> .
<a href="#">IGameNotificationsService</a>	This is the interface used to interact with <a href="#">Game Notifications</a> .
<a href="#">IGameServersService</a>	Methods to improve the administration of Steam Game Servers.

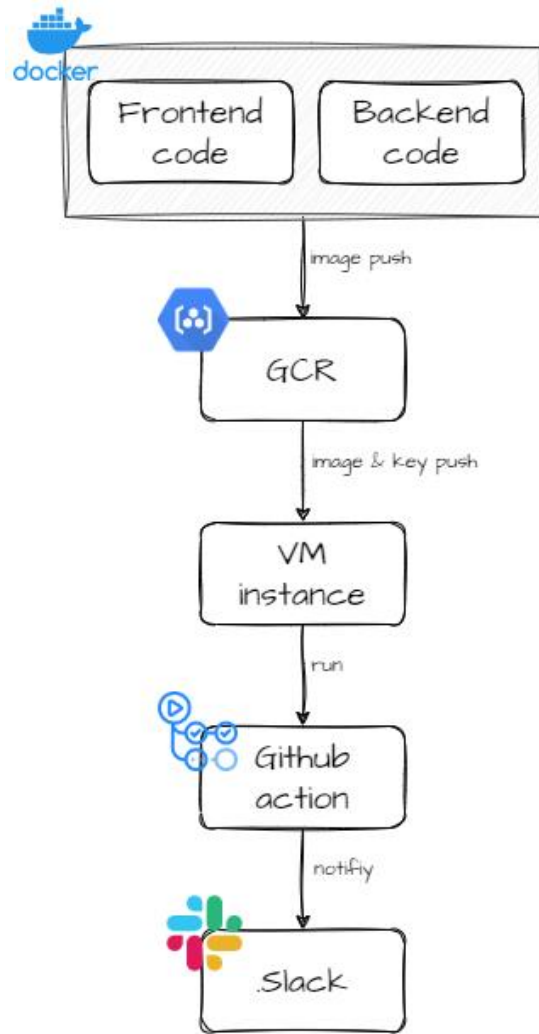




# 프로젝트 | CICD



# 프로젝트 | CICD ~ Github action



← Build and Deploy to Google Compute Engine

✓ [Test] #2

Summary

Jobs

- ✓ Setup, Build, Publish, and Deploy

Run details

- Usage
- Workflow file

Triggered via push 1 minute ago

Status: Success

Total duration: 1m 48s

Artifacts: -

JI-Hoon pushed 4a468e5 main

deploy\_docker.yml

on: push

✓ Setup, Build, Publish, an... 1m 36s

CI-CD 오후 2:42

✓ Succeeded GitHub Actions

8398a7@action-slack

repo

JI-Hoon/final-project-level3-recsys-09

commit

65579044

message

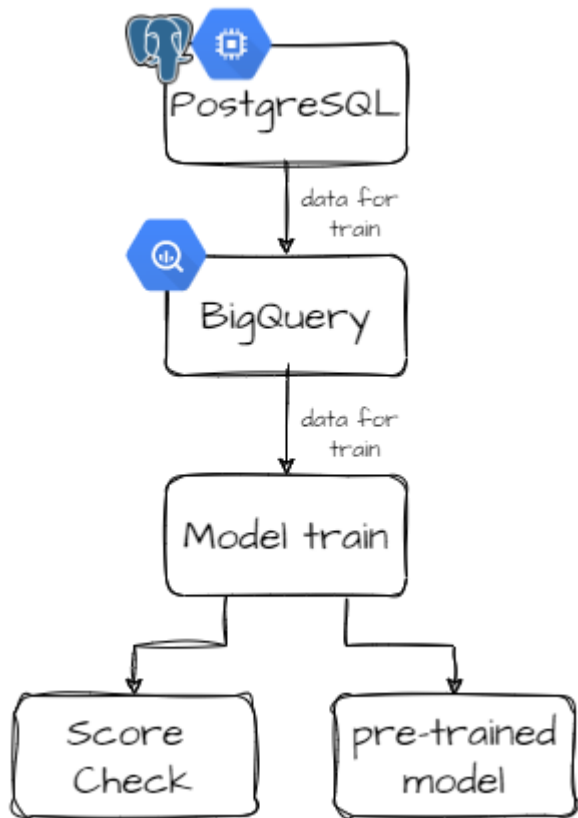
[Test] Slack Action

author

JI-Hoon<legg0724@naver.com>



# 프로젝트 | CICD ~ Airflow



☒ DAG: model\_train model\_train

☐ Grid

☒ Graph

☐ Calendar

☐ Task Duration

☐ Task Tries

☐ Landir



2023-02-07T07:00:01+09:00

Runs

25



Run

scheduled\_\_2023-02-06T2

PythonOperator

get\_newdata

check\_score

## 프로젝트 | 기대효과 및 한계점



### 기대효과

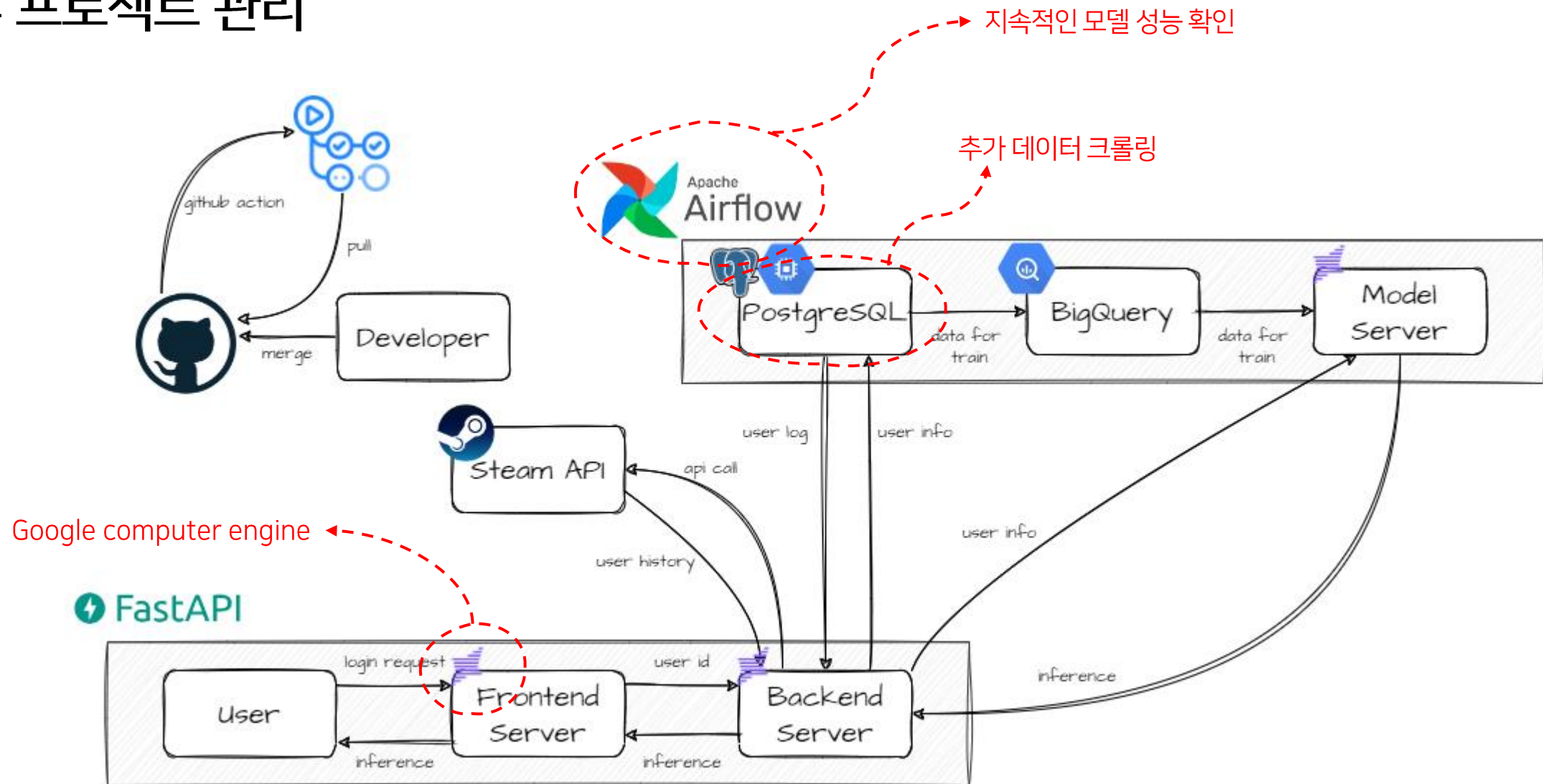
AAA급 게임만 알던 유저에게 적합한 인디게임 추천  
인디 게임사 및 전반적인 게임 시장 활성화

### 한계점

최신화 되지 않은 유저-데이터 상호작용 데이터  
참신함 혹은 의외성을 갖기 힘든 CF기반의 모델



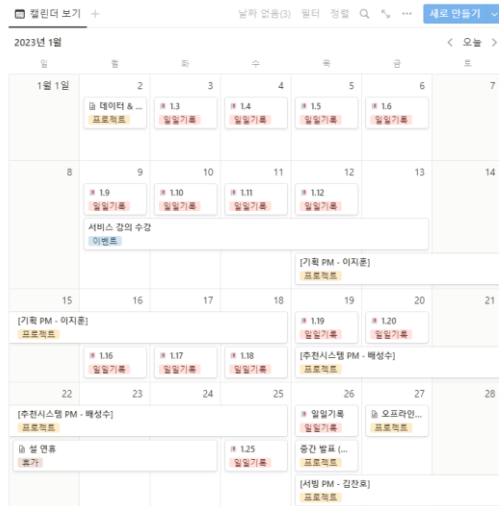
# 이후 프로젝트 관리



# 협업방식

## Notion

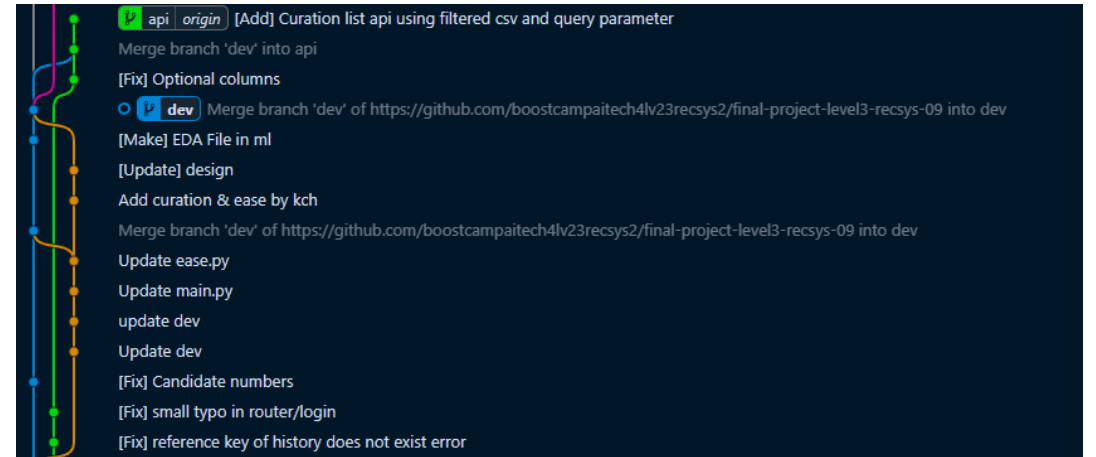
### 프로젝트 타임라인



### Todo / In Progress / Done

태그	상태	Aa 이름	내용
PM	Done	Dataset 파악	
PM	In progress	CI/CD	
PM	In progress	Data EDA	
PM	In progress	기획 초안	
Data	Not started	크롤링 활용 방법 찾기	유저 데이터, 최신 데이터 가져
Data	Not started	DB 구축	Back-End 팀과 협업하여 DB 구
Modeling	Done	간단한 모델로 특징 생성 for 프로토타입	간단 모델 구현/ Inference 속5
Modeling	Done	recholo로 모델 테스트	계열 별 모델 테스트(성능, 속5
Modeling	In progress	추천 시스템 클로우 설정	후보군 생성, 개인화 추천, 필드
Modeling	Not started	콜드스타트 문제 해결방안 모색	유저/아이템 콜드스타트 해결
Modeling	Not started	인퍼런스 속도 개선	실시간 서비스에 가장 중요한
Modeling	Not started	[post] 결과 필터링 알고리즘 탐색	
Modeling	Not started	[pre] 후보군 생성 알고리즘 탐색	
Modeling	Not started	모델 성능 평가 방법 설정	모델 성능 평가 방법 정하기. e
Back-End	Done	이미지 다운 및 처리 설계	이미지 저장 및 처리 방식
Back-End	In progress	DB 설계 및 구축	DBMS, Cloud Service 어떤걸로
Back-End	In progress	API 개발	필요 API를 나열하고 개발한다
Back-End	In progress	Inference 서버 구축	

## Github



- 백엔드 및 프론트 > api

- 모델 > model

- Api와 model 브랜치 테스트 > dev

- Dev test 이후 > main



---

# 감사합니다.

