

캡스톤디자인

4SHARP 프로젝트



32150973 김재성

32154577 차무송

32154049 전상민

목차

1. 팀원 소개
2. 프로젝트 개요 및 요구사항
3. 설계 및 구현
4. 테스트 결과
5. 자체 평가 및 개선 방향
6. 데모 영상
7. Q&A

팀원 소개

Team Intro





김재성
소프트웨어학과
32150973
Frontend



차무송
소프트웨어학과
32154577
Frontend & Middleware



전상민
소프트웨어학과
32154049
Backend

4SHARP 팀

프로젝트 개요 및 요구사항

Overview and Requirements



프로젝트 개요



- 사진 속 인물을 인식하여 원하는 인물을 인페이팅 기술을 이용하여 자연스럽게 지워주는 어플리케이션
- 사용자가 임의로 그려서 사진에 지우고 싶은 부분을 인페이팅 기술을 이용하여 지워주는 어플리케이션

요구사항

1. 사용자는 촬영을 할 수 있습니다.
2. 사진 속 인물들을 찾아냅니다.
3. 사용자는 인물들을 선택하여 지울 수 있습니다.
4. 임의로 드로잉을 하여 지우고 싶은 부분을 지울 수 있습니다.
5. 최종 결과물을 라이브러리에 저장할 수 있다.

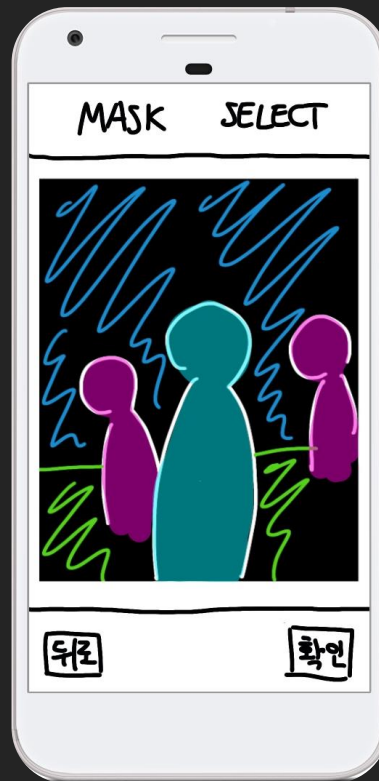
요구사항



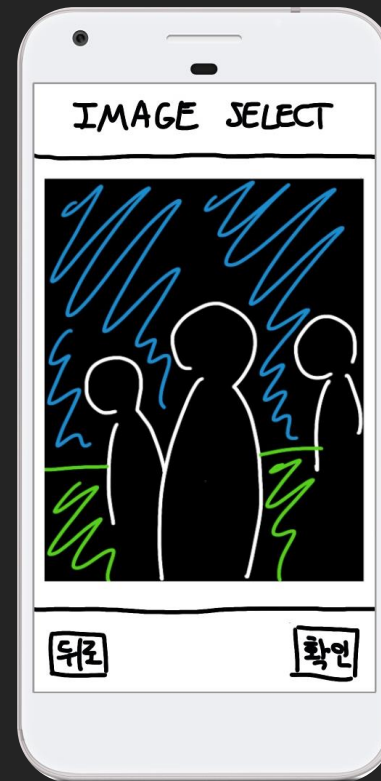
1



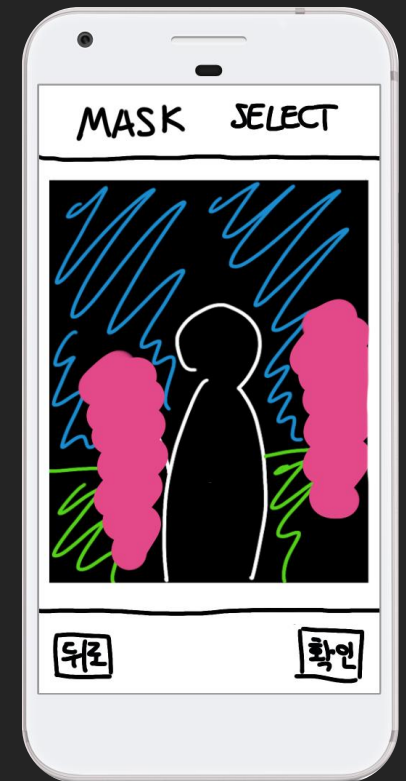
2



3



4



5

설계 및 구현

Design & Implementation

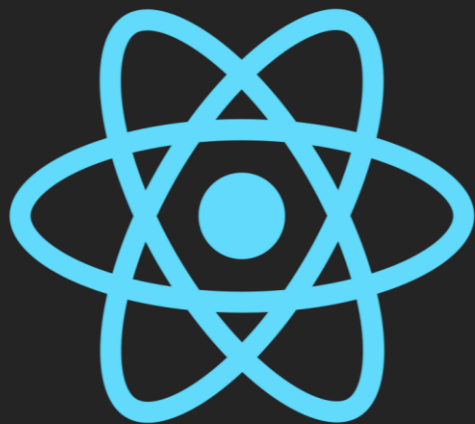


프론트 엔드

Frontend

○ 김재성

Frontend 구성



React Native

+



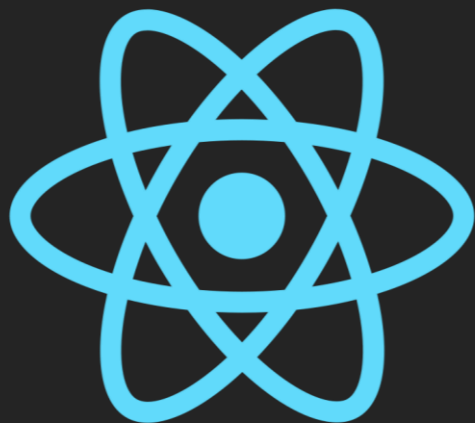
Expo

Frontend 구성

크로스 플랫폼

Android &
iOS 지원

기기 호환성



React Native

+



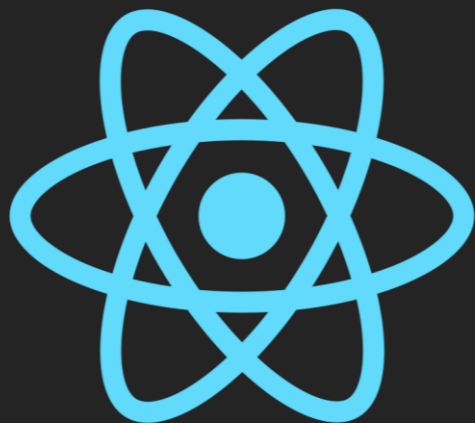
Expo

Frontend 구성

크로스 플랫폼

Android &
iOS 지원

기기 호환성



React Native

+

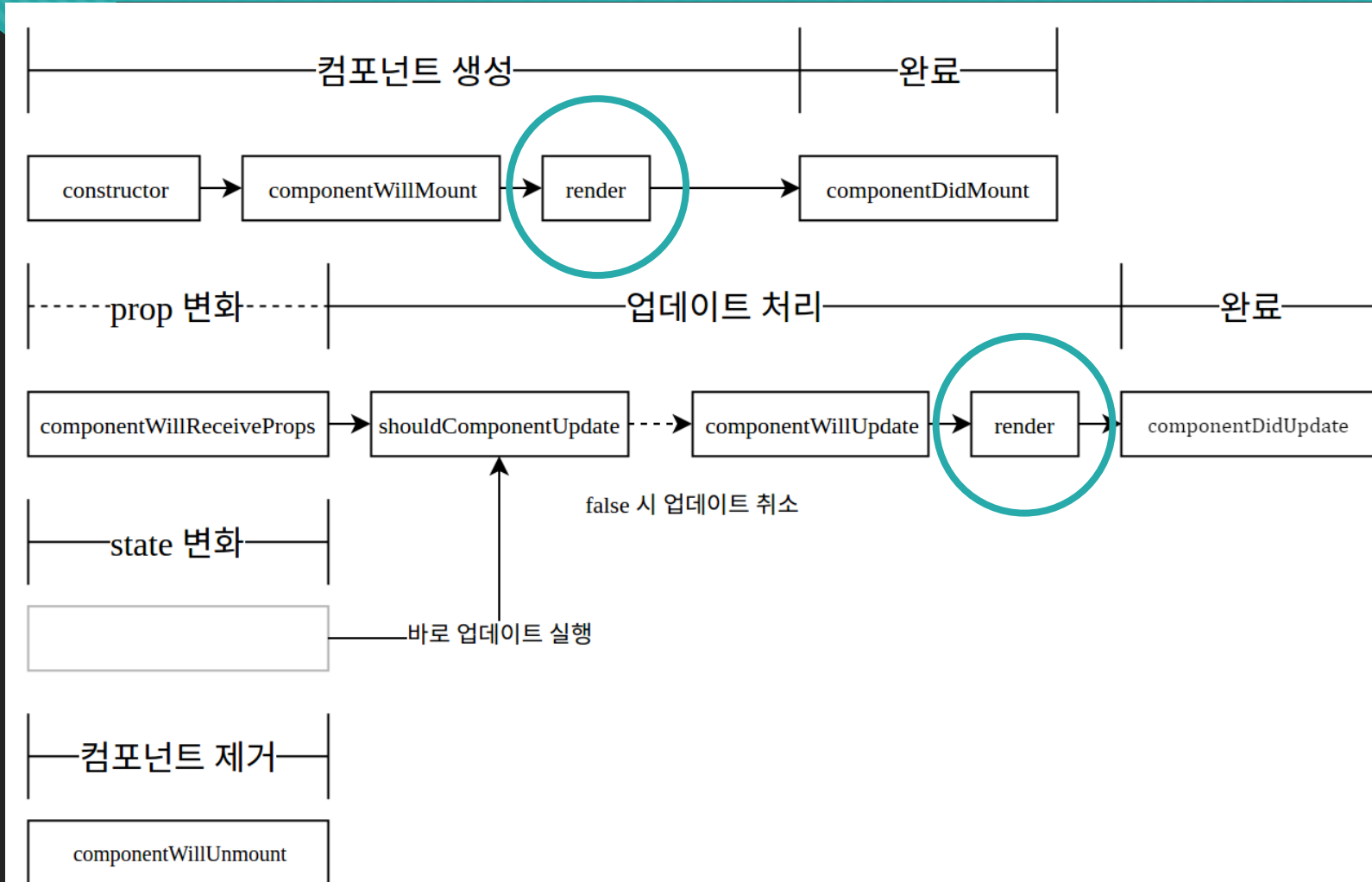


Expo

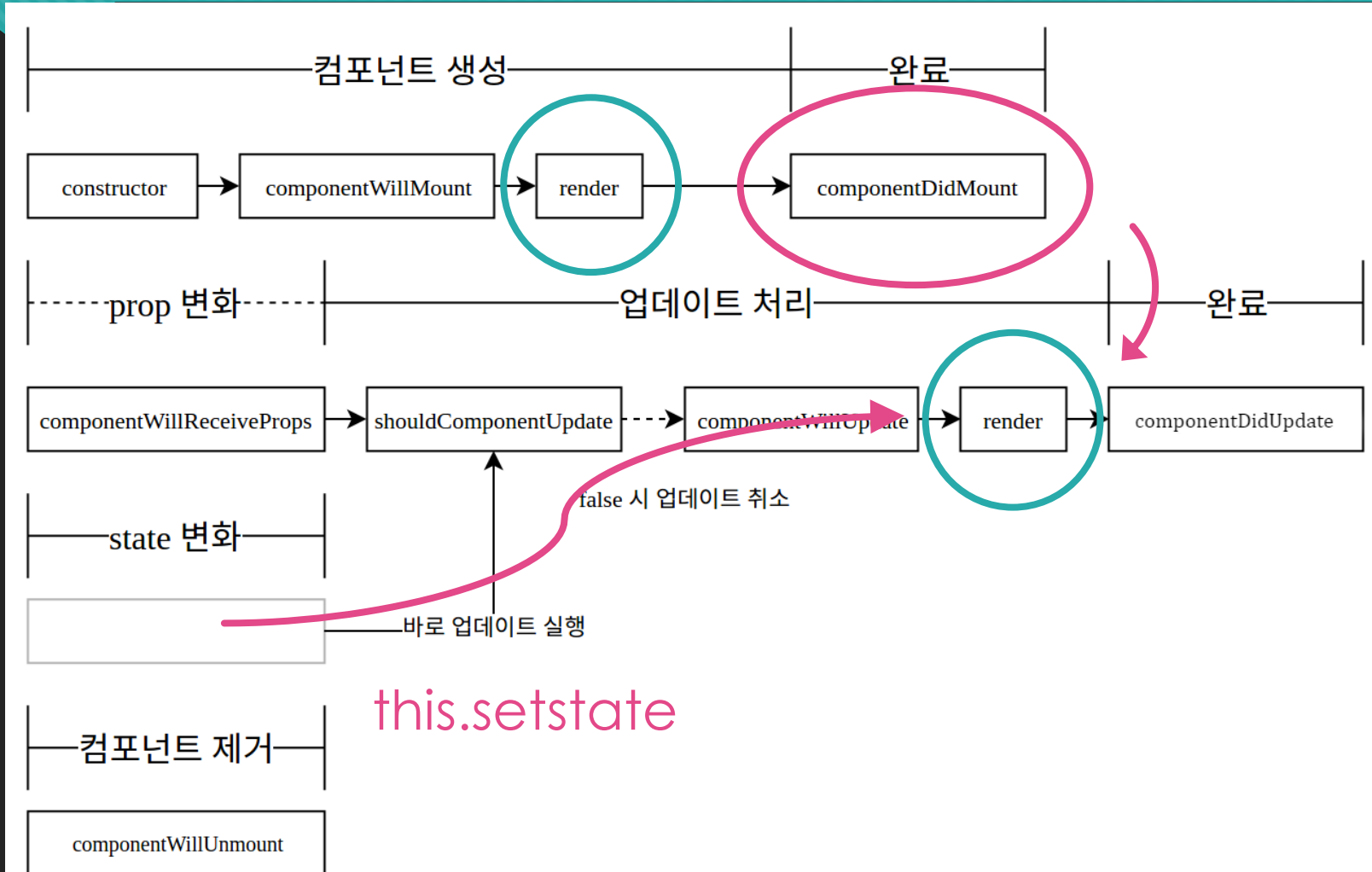
간단한 개발
(시간 절약)

배포 및
업데이트 용이성

React Component LifeCycle



React Component LifeCycle



React Component LifeCycle

컴포넌트 실행 시,
render()

```
render() {  
  if (this.state.camPerm && this.state.camRollPerm) {  
    return this.renderCamera();  
  } else if (this.state.showSelect && this.state.data) {  
    return this.renderSelect();  
  } else {  
    return this.renderNoPermission();  
  }  
}
```

< Home.js >

React Component LifeCycle

컴포넌트 실행 완료,
componentDidMount()

```
componentDidMount() {  
  this.setState({  
    sessionId: Constants.sessionId.slice(0, 8),  
  });  
  this.getPermissionAsync();  
  this.getPermissionRollAsync();  
}
```

카메라 변수

camPerm
camRollPerm

```
render() {  
  if (this.state.camPerm && this.state.camRollPerm) {  
    return this.renderCamera();  
  } else if (this.state.showselect && this.state.data) {  
    return this.renderSelect();  
  } else {  
    return this.renderNoPermission();  
  }  
}
```

< Home.js >

React Component LifeCycle

유저가 이미지 선택 시,
pickImage()

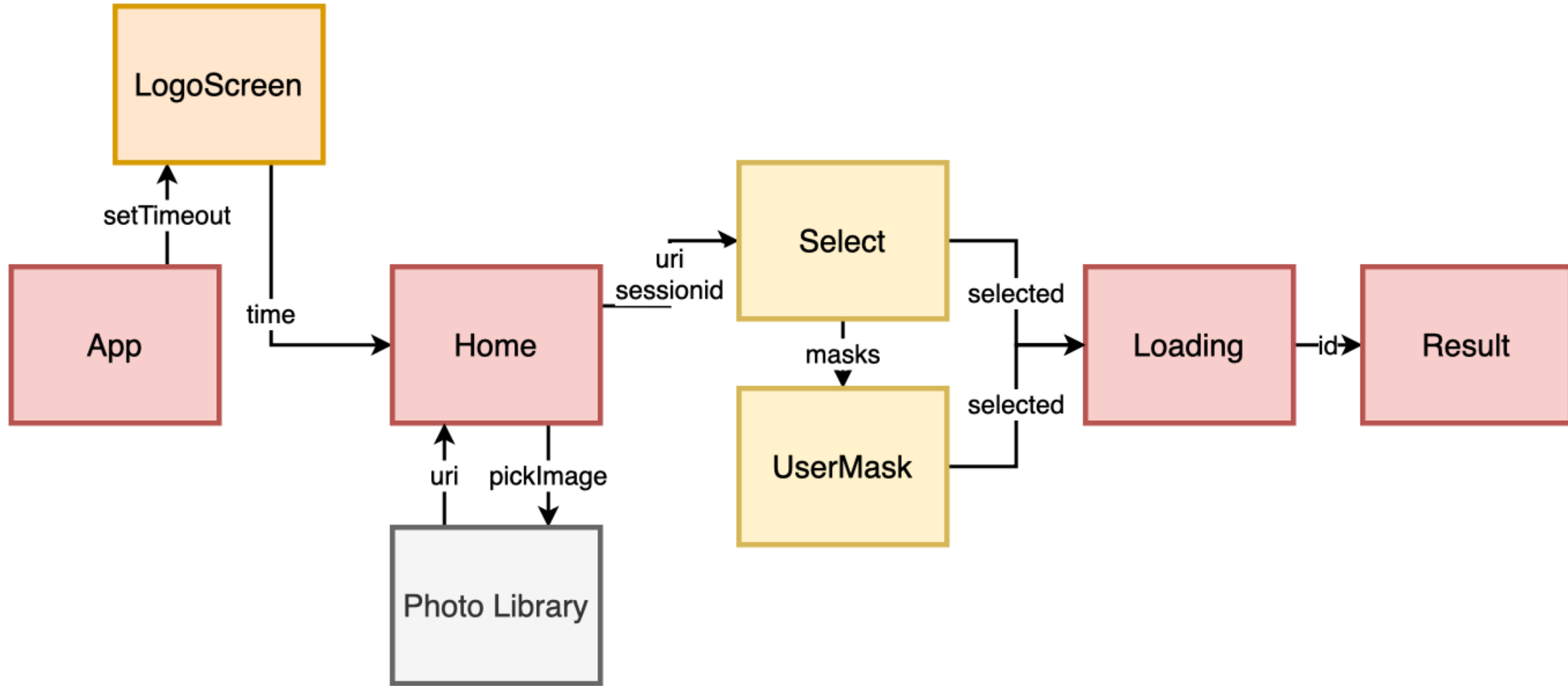
```
pickImage = async () => {  
  //album에서 사진 가져오는 메소드  
  this.setState({ galleryMode: true });  
  const { uri } = await ImagePicker.launchImageLibraryAsync({  
    mediaTypes: ImagePicker.MediaTypeOptions.Images,  
  });  
  this.setState({  
    showSelect: true,  
    data: uri,  
  });  
}
```

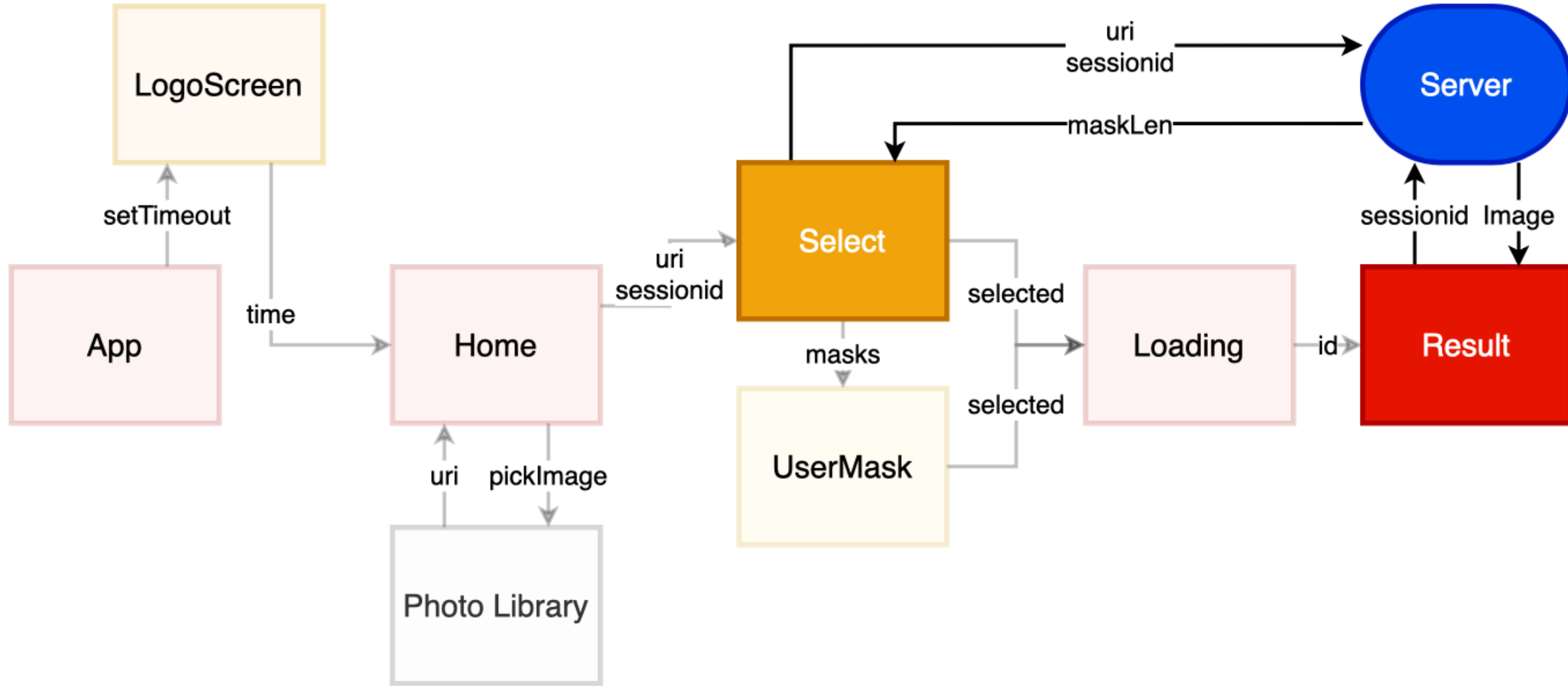
이미지 변수

showSelect
data

```
render() {  
  if (this.state.camPerm && this.state.camRollPerm) {  
    return this.renderCamera();  
  } else if (this.state.showSelect && this.state.data) {  
    return this.renderSelect();  
  } else {  
    return this.renderNoPermission();  
  }  
}
```

< Home.js >





미들웨어 Middleware

- 차무송

PHP 디렉토리 구조



waiting



고유 디렉토리



Image&mask



status.txt



Image
upload.php



Image
Upload
user.php



Inpainting.php



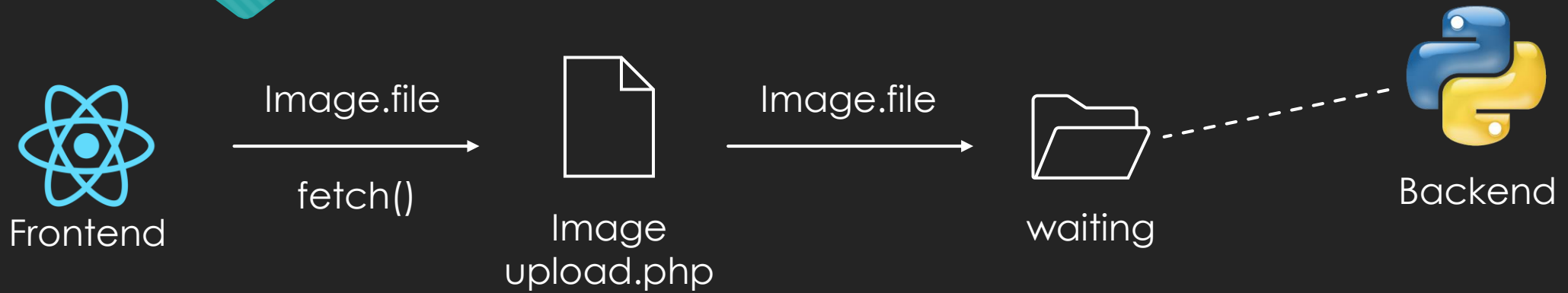
checking
mask
number.php



thanks.php

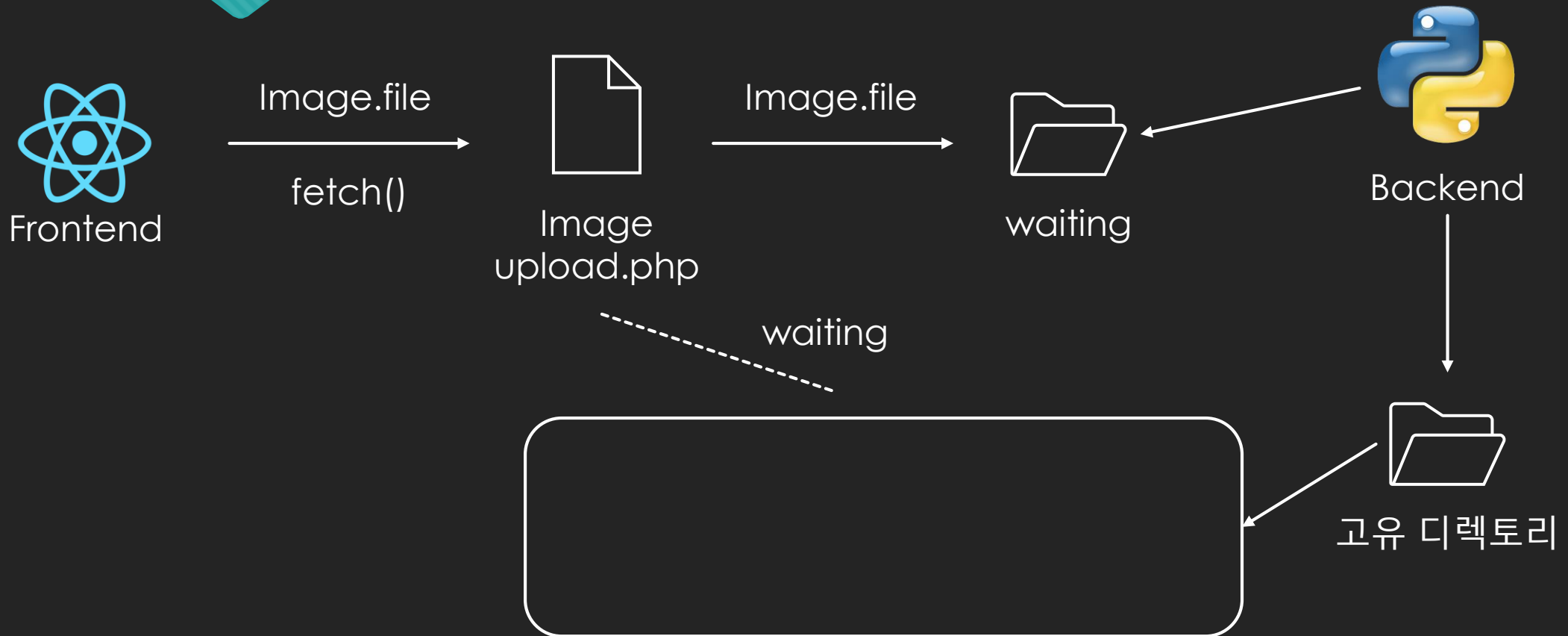
Image_upload.php

사용자의 이미지 파일을 업로드 후
마스크의 개수를 반환 시켜주는 PHP파일



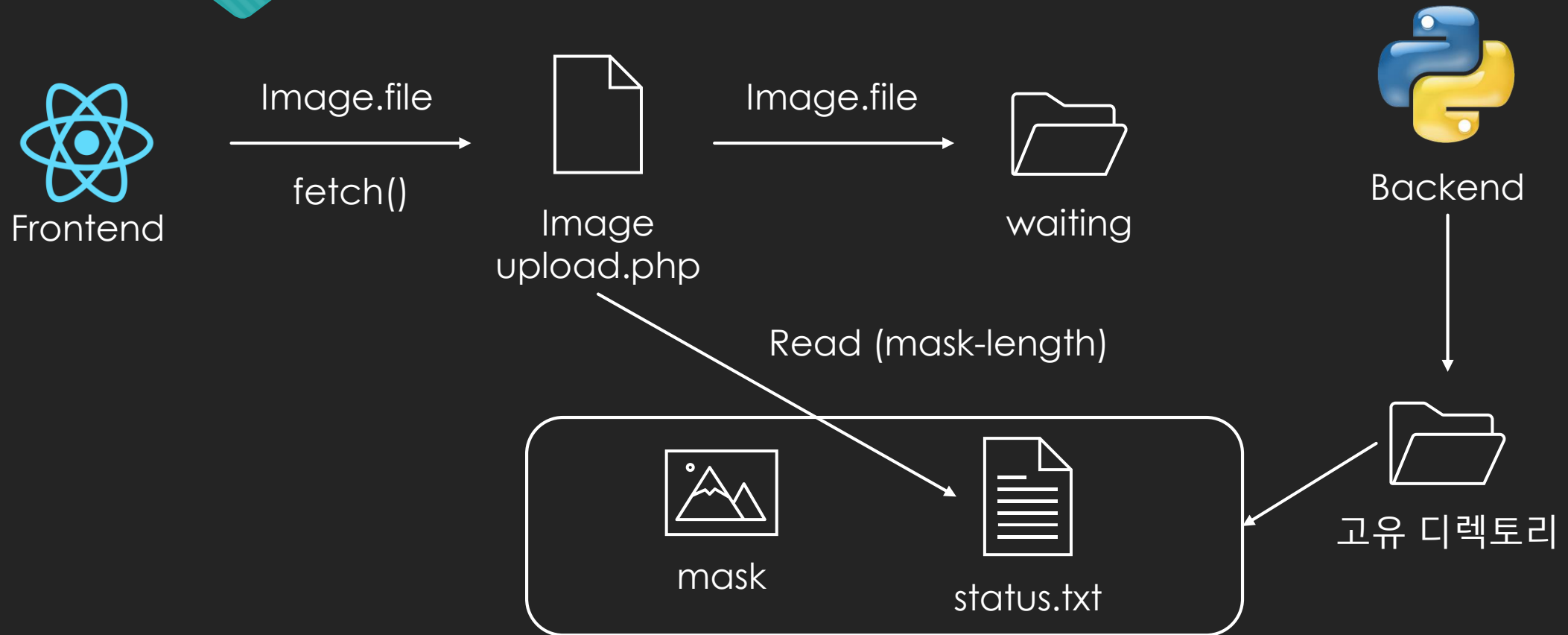
Image_upload.php

사용자의 이미지 파일을 업로드 후
마스크의 개수를 반환 시켜주는 PHP파일



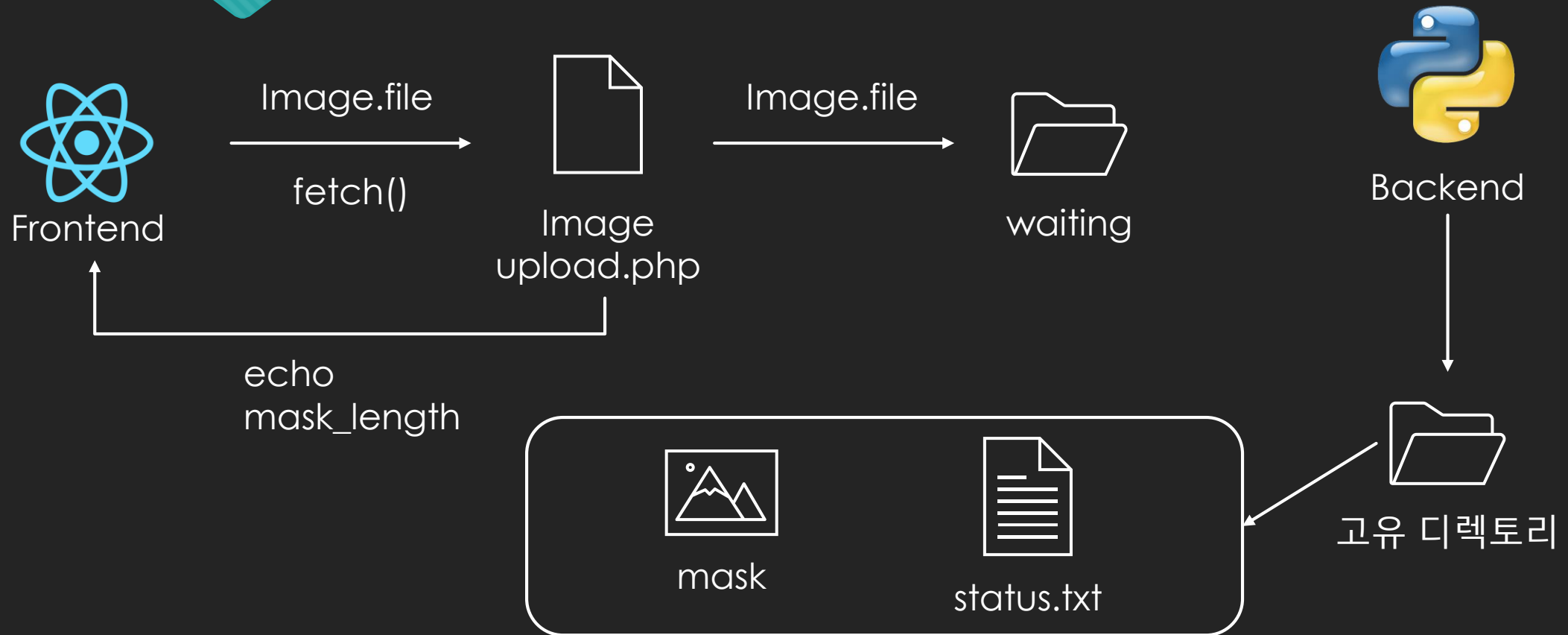
Image_upload.php

사용자의 이미지 파일을 업로드 후
마스크의 개수를 반환 시켜주는 PHP파일



Image_upload.php

사용자의 이미지 파일을 업로드 후
마스크의 개수를 반환 시켜주는 PHP파일



checking_ mask_number.php

사용자가 누른 좌표가 마스크 부분인지
x,y좌표의 RGB값을 비교하여 알려주는
php파일



Frontend

mask_length,
(x, y) 좌표

fetch()



checking
mask
number.php



mask(0)



mask(1)



mask(2)

.....



mask(3)



mask(n)



status.txt



고유 디렉토리

checking_ mask_number.php

사용자가 누른 좌표가 마스크 부분인지
x,y좌표의 RGB값을 비교하여 알려주는
php파일



Frontend

mask_length,
(x, y) 좌표

fetch()



checking
mask
number.php



mask(0)



mask(1)



mask(2)



mask(3)



mask(n)



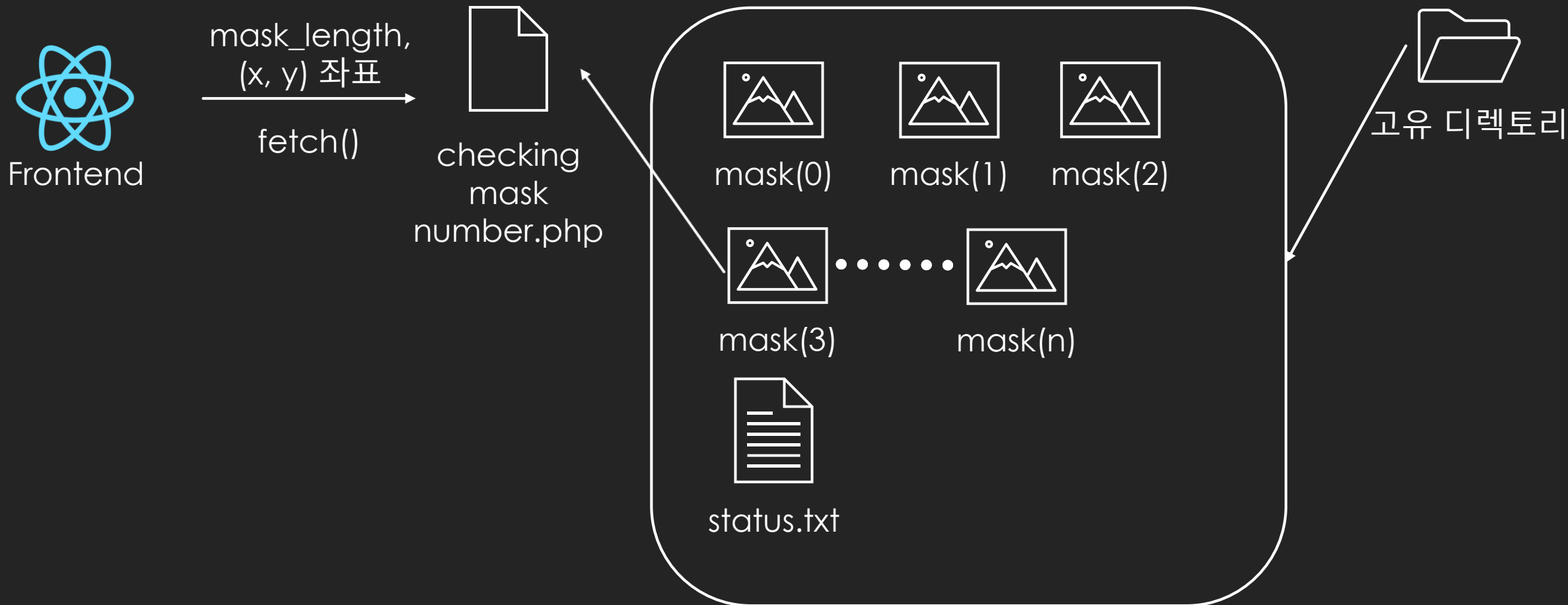
status.txt



고유 디렉토리

checking_ mask_number.php

사용자가 누른 좌표가 마스크 부분인지
x,y좌표의 RGB값을 비교하여 알려주는
php파일



checking_ mask_number.php

사용자가 누른 좌표가 마스크 부분인지
x,y좌표의 RGB값을 비교하여 알려주는
php파일



Frontend

echo "3"



checking
mask
number.php



mask(0)



mask(1)



mask(2)



mask(3)

.....



mask(n)



status.txt



고유 디렉토리

Image_upload_user.php

사용자가 생성한 임의의 마스크를
업로드하는 php파일



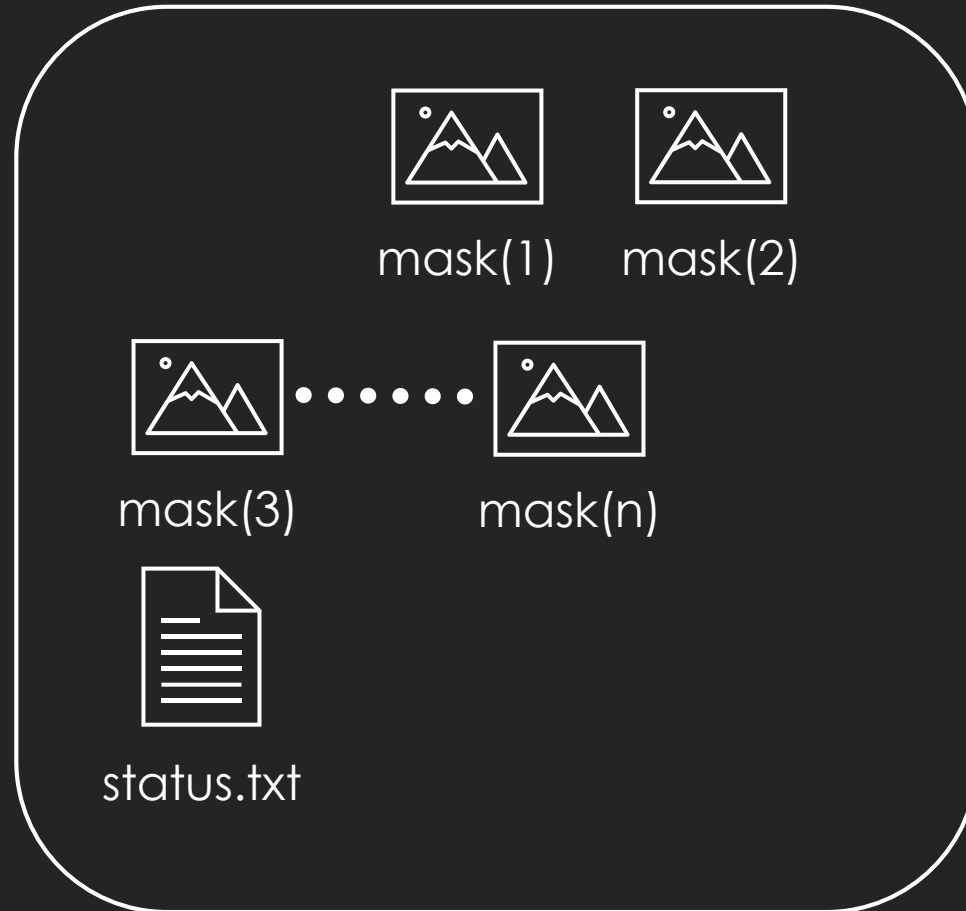
Frontend

mask.file

fetch()



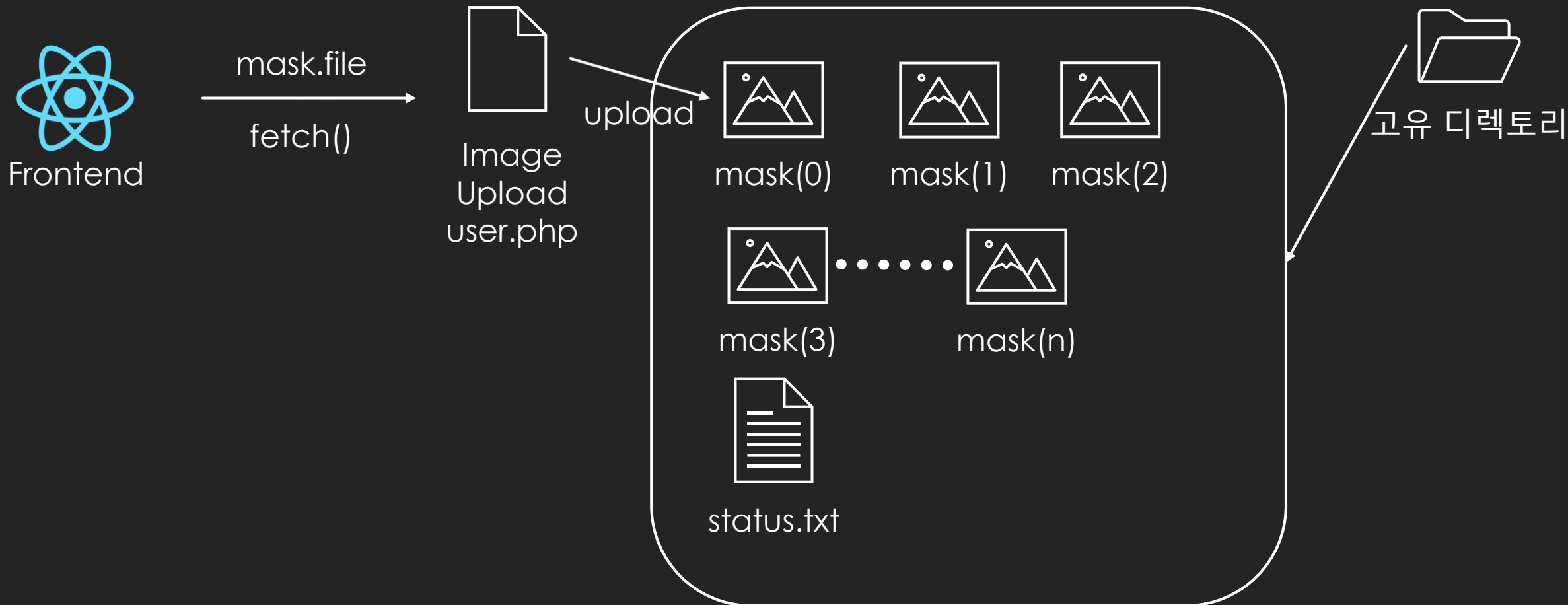
Image
Upload
user.php



고유 디렉토리

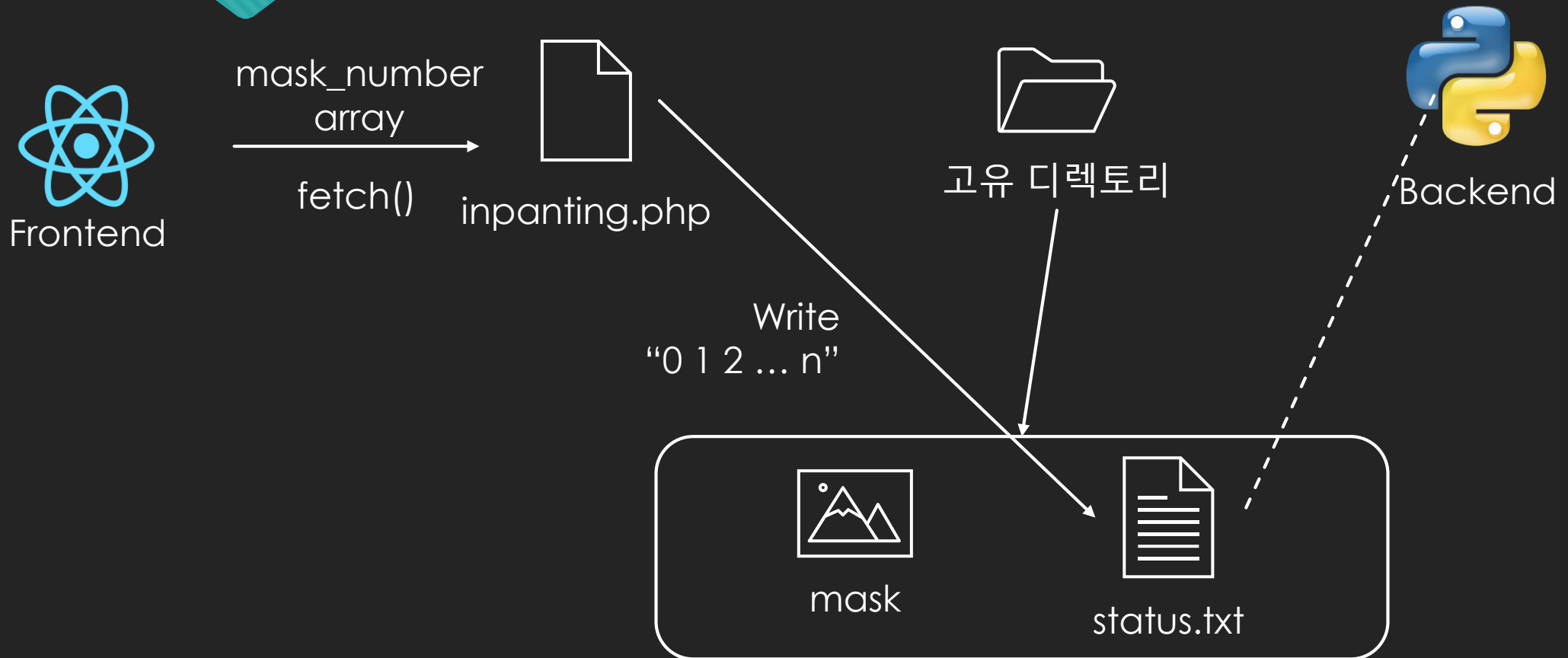
Image_upload_user.php

사용자가 생성한 임의의 마스크를
업로드하는 php파일



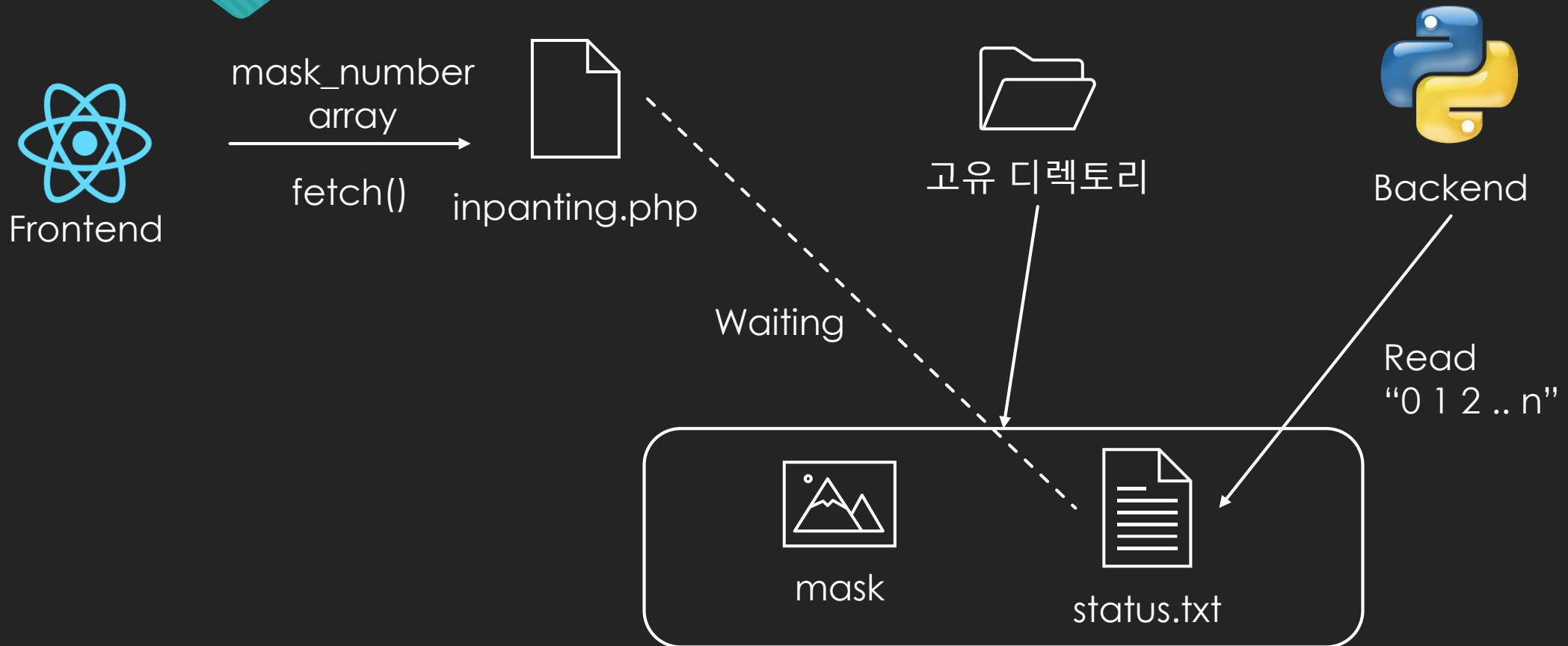
Inpainting.php

사용자 고른 마스크 번호를
Backend에게 전달 후 Inpainting 처리를
기다리는 php 파일



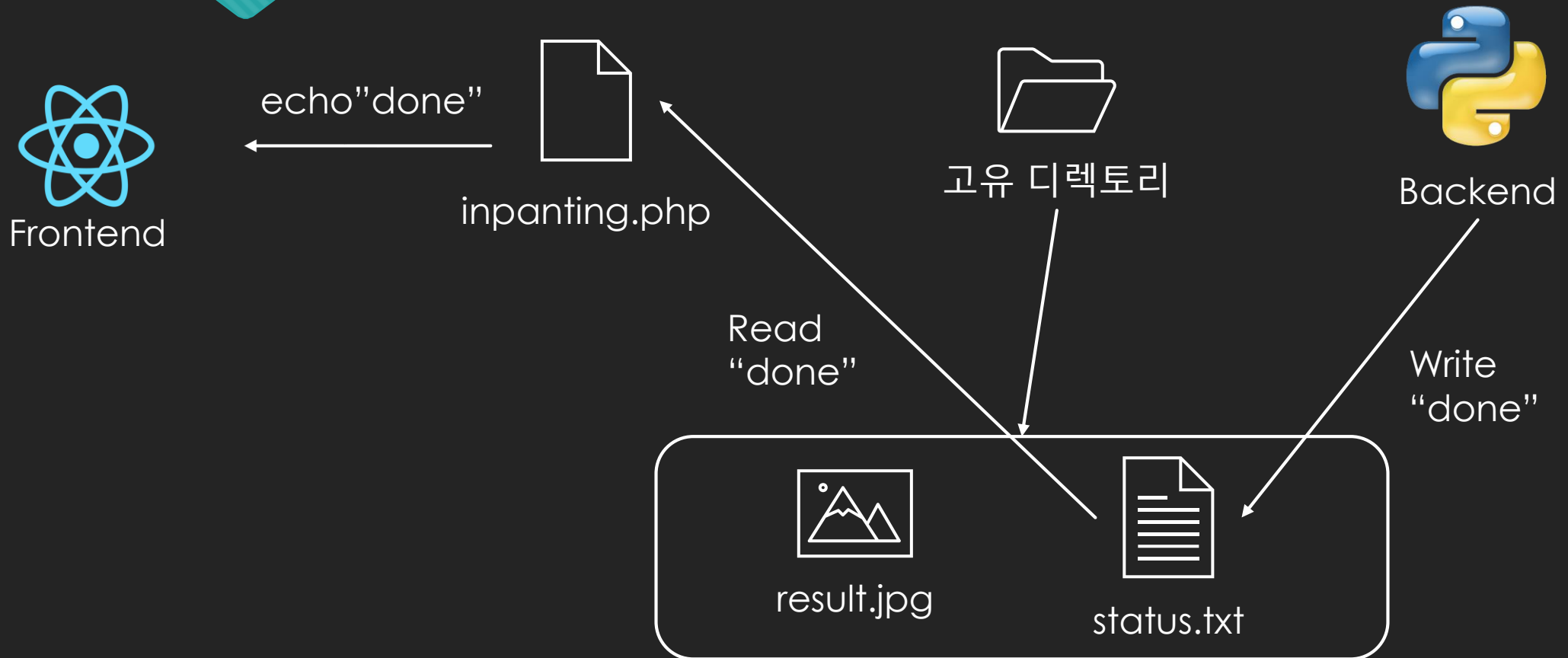
Inpainting.php

사용자 고른 마스크 번호를
Backend에게 전달 후 Inpainting 처리를
기다리는 php 파일



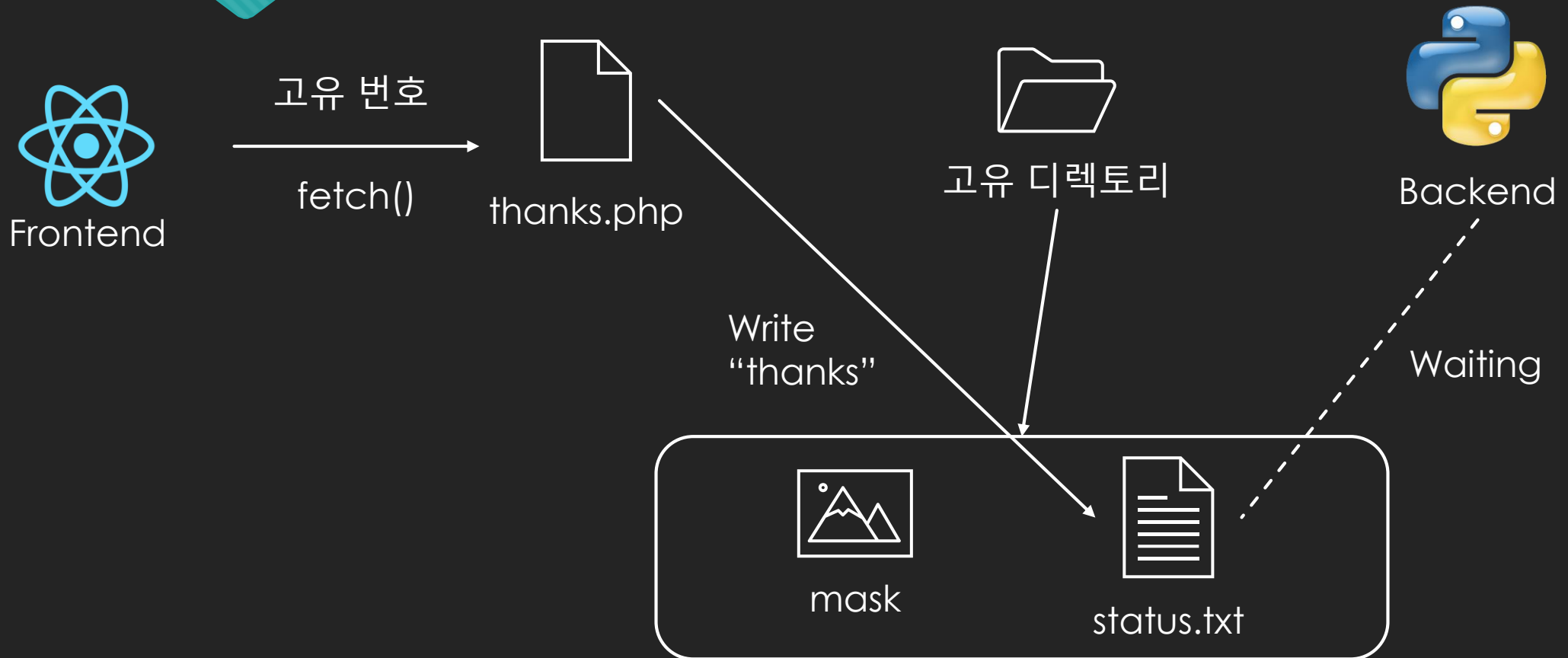
Inpainting.php

사용자 고른 마스크 번호를
Backend에게 전달 후 Inpainting 처리를
기다리는 php 파일



thanks.php

모든 작업이 끝났음을 Backend에게
알려주는 php 파일



thanks.php

모든 작업이 끝났음을 Backend에게
알려주는 php 파일



Frontend

고유 번호

fetch()



thanks.php



고유 디렉토리



Backend

Read
"done"



mask



status.txt

thanks.php

모든 작업이 끝났음을 Backend에게
알려주는 php 파일



waiting

Waiting



Backend



Image
upload.php



Image
Upload
user.php



Inpainting.php



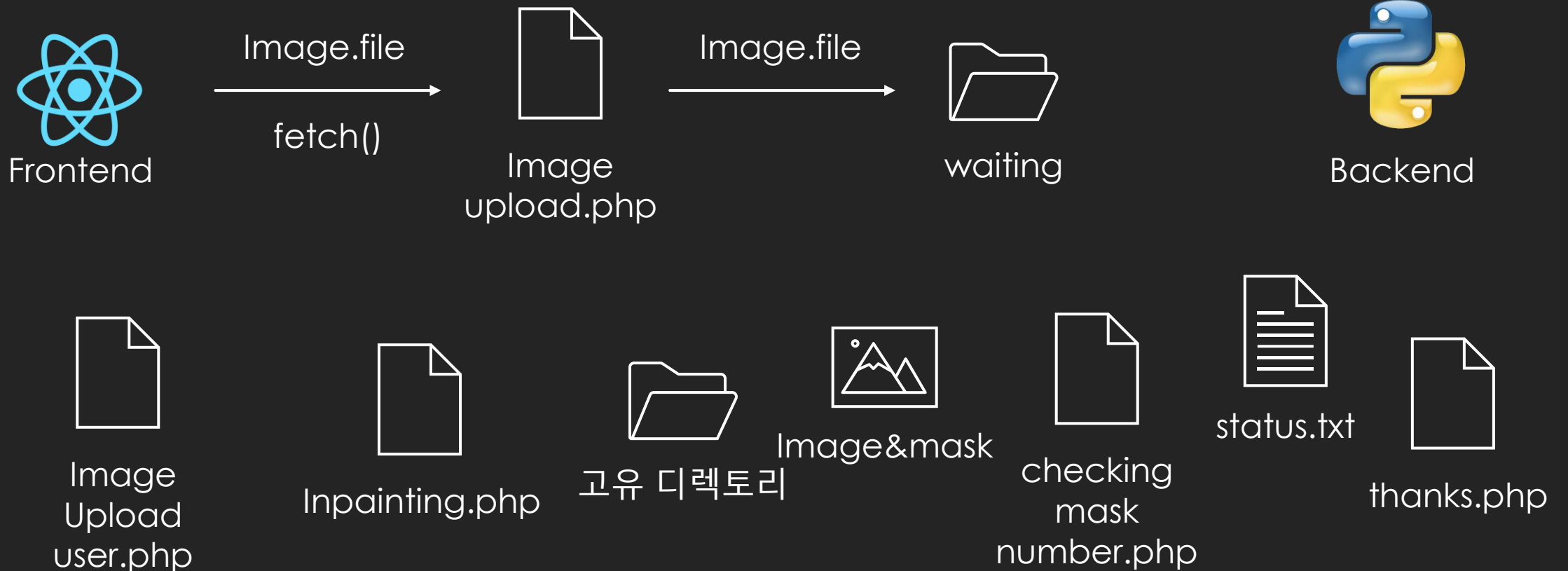
checking
mask
number.php



thanks.php

Image_upload.php

사용자의 이미지 파일을 업로드
시켜주는 PHP파일



백 엔드 Backend


○ 전상민

Machine Learning Models

MASK RCNN

Inpainting

Mask RCNN

 [matterport / Mask_RCNN](#) [github.com/matterport/Mask_RCNN/blob/master/setup.py](#)


<> Code ⓘ Issues 1.4k 🔗 Pull requests 97 ▶ Actions 📁 Projects 📖 Wiki 🛡 Security 📈 Insights





Jirka Borovec Borda
Computer Vision & Machine Learning &
Bio-Medical Imaging & Deep Learning
PRO
📍 Prague




Waleed waleedka
Deep learning, Computer Vision,
PyTorch, Tensorflow, Web development,
Python.
PRO
📍 Mountain View, CA

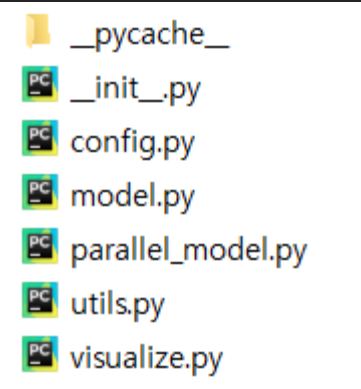
 [matterport / Mask_RCNN](#) 👁 Watch ▼ 591 ⭐ Star 18k 🍴 Fork 8.7k

 [jwyang / faster-rcnn.pytorch](#) 👁 Watch ▼ 98 ⭐ Star 5.7k 🍴 Fork 2k

 [endernewton / tf-faster-rcnn](#) 👁 Watch ▼ 118 ⭐ Star 3.5k 🍴 Fork 1.6k

 [rbgirshick / rcnn](#) 👁 Watch ▼ 185 ⭐ Star 2.1k 🍴 Fork 937

Mask RCNN



A file explorer window showing the contents of a project directory. The directory contains a folder named `__pycache__` and six Python files: `__init__.py`, `config.py`, `model.py`, `parallel_model.py`, `utils.py`, and `visualize.py`. Each file icon is a small green square with the letters 'PC' in white.

- `__pycache__`
- `__init__.py`
- `config.py`
- `model.py`
- `parallel_model.py`
- `utils.py`
- `visualize.py`

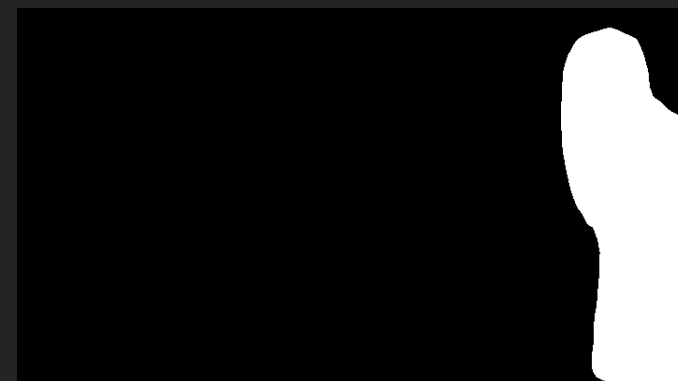
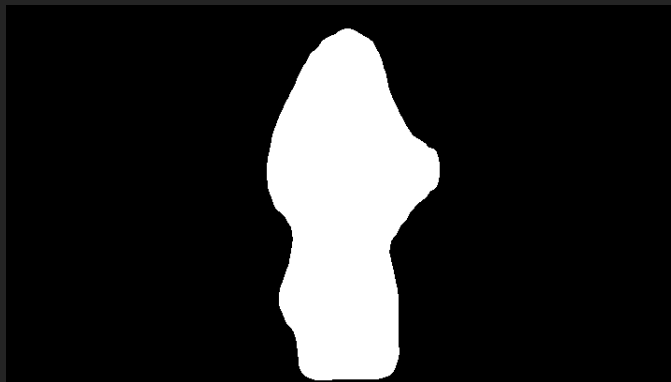
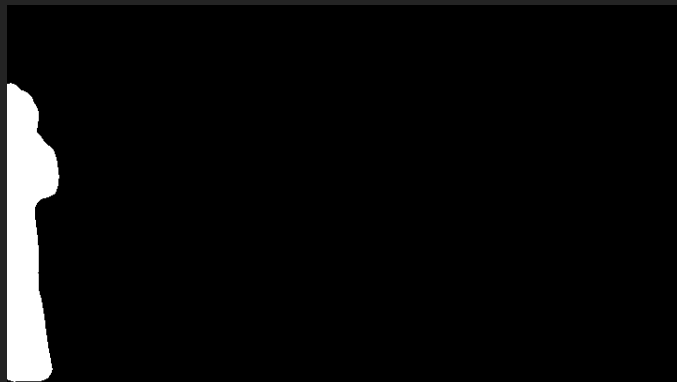
Mask RCNN



Mask RCNN



Mask RCNN



Inpainting




 [JiahuiYu](#) / **generative_inpainting**

 Watch ▾ **72**  Star **2.1k**  Fork **546**

About

DeepFill v1/v2 with Contextual Attention and Gated Convolution, CVPR 2018, and ICCV 2019 Oral



 [vt-vl-lab](#) / **3d-photo-inpainting**

 Watch ▾ **135**  Star **4.5k**  Fork **617**

About

[CVPR 2020] 3D Photography using Context-aware Layered Depth Inpainting



 [leehomyc](#) / **Faster-High-Res-Neural-Inpainting**

 Watch ▾ **50**  Star **1.2k**  Fork **216**

About

High-Resolution Image Inpainting using Multi-Scale Neural Patch Synthesis

 [lyndonzheng](#) / **Pluralistic-Inpainting**

 Watch ▾ **14**  Star **409**  Fork **101**

About






CVPR 2019: "Pluralistic Image Completion"

[image-completion](#) [inpainting](#)



[generative-adversarial-networks](#)

[multimodal](#)

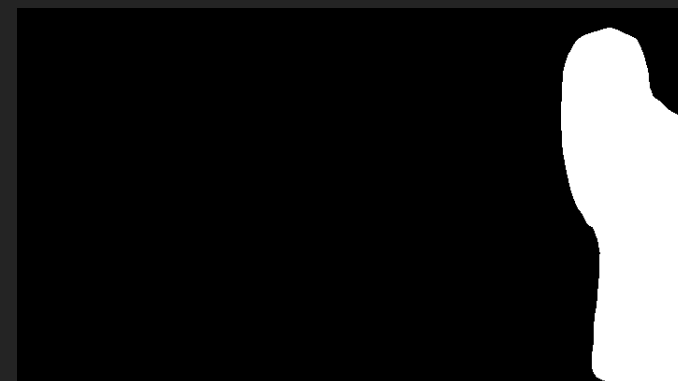
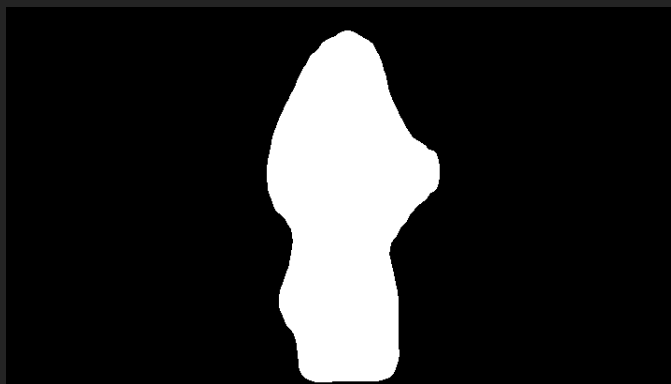
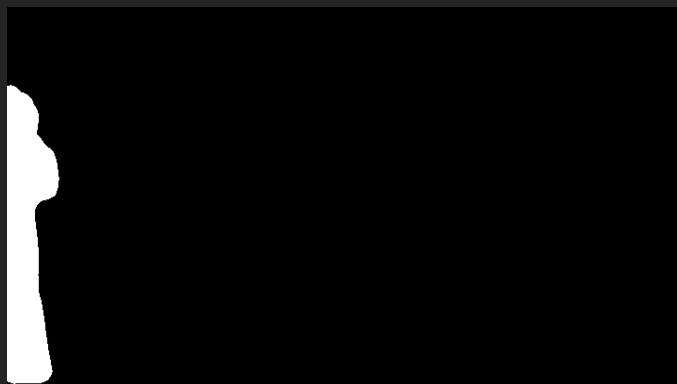
Inpainting

-  model_logs
-  inpaint.py
-  inpaint.yml
-  inpaint_model.py
-  inpaint_ops.py

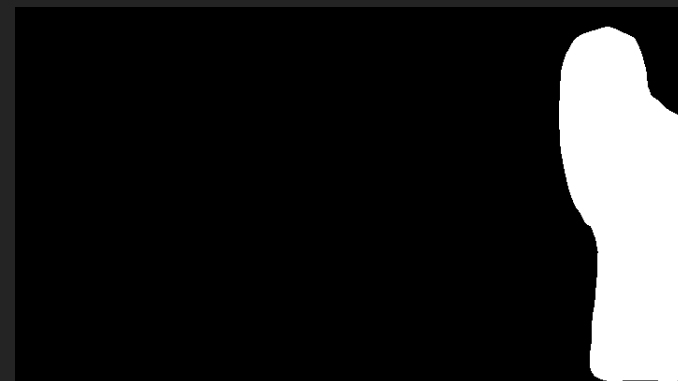
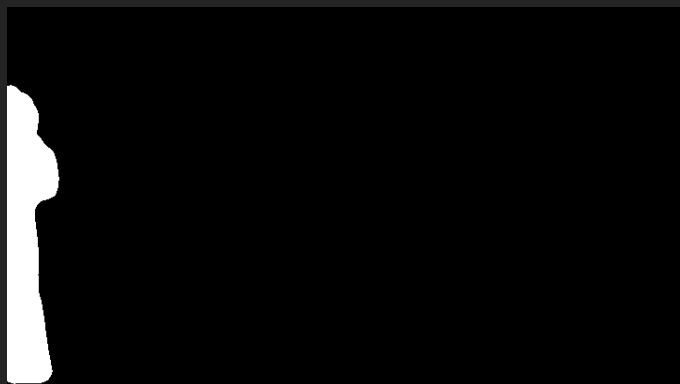
Inpainting

 release_celeba_hq_256
 release_places2_256

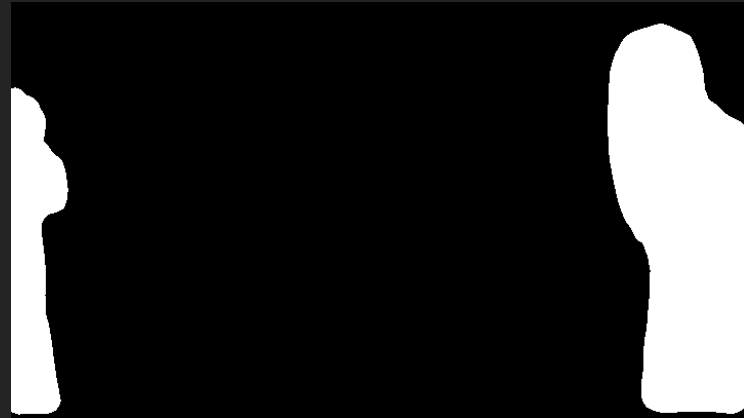
Inpainting



Inpainting



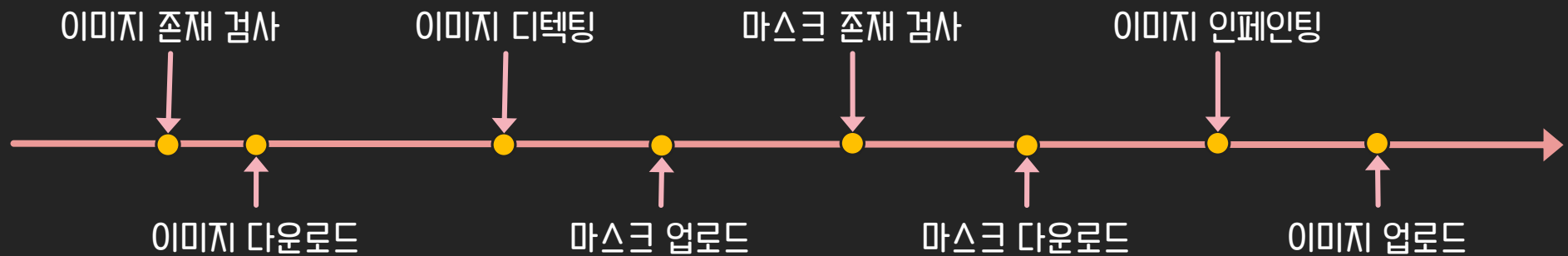
Inpainting



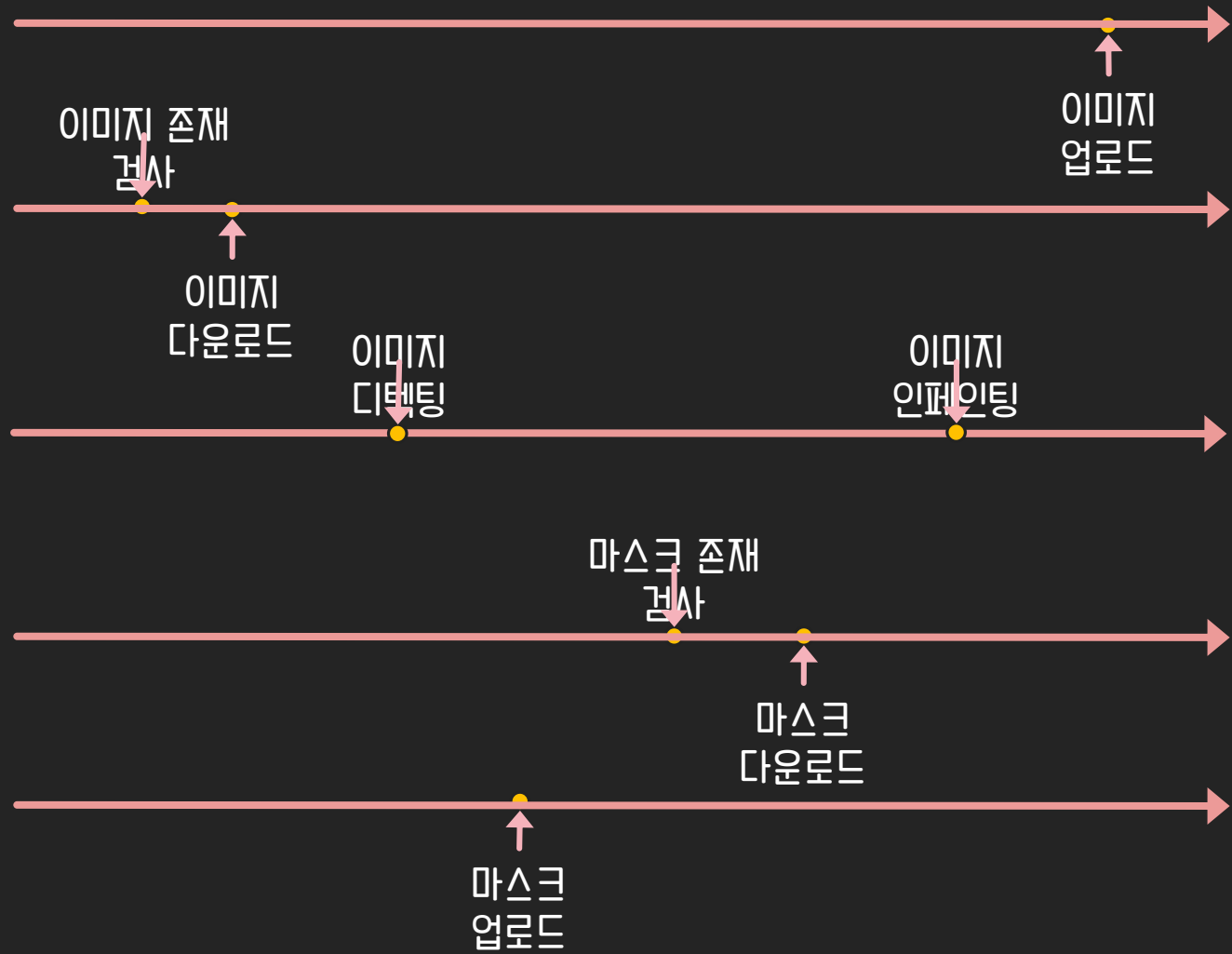
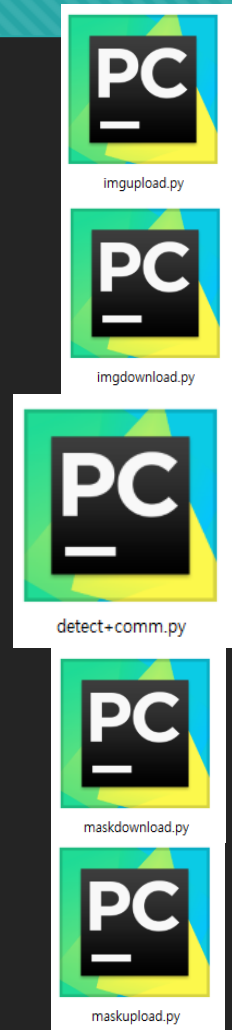
Inpainting



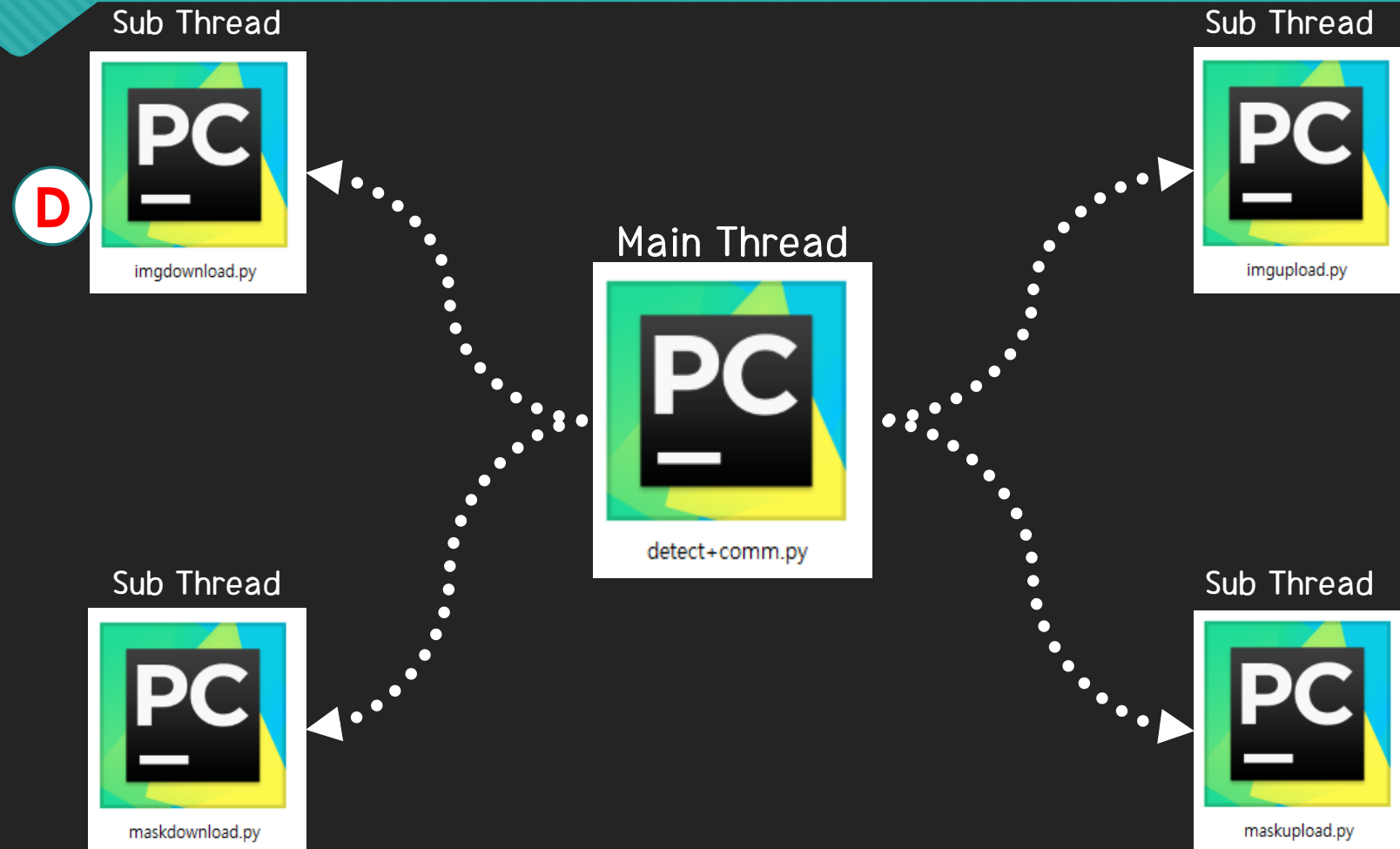
Background



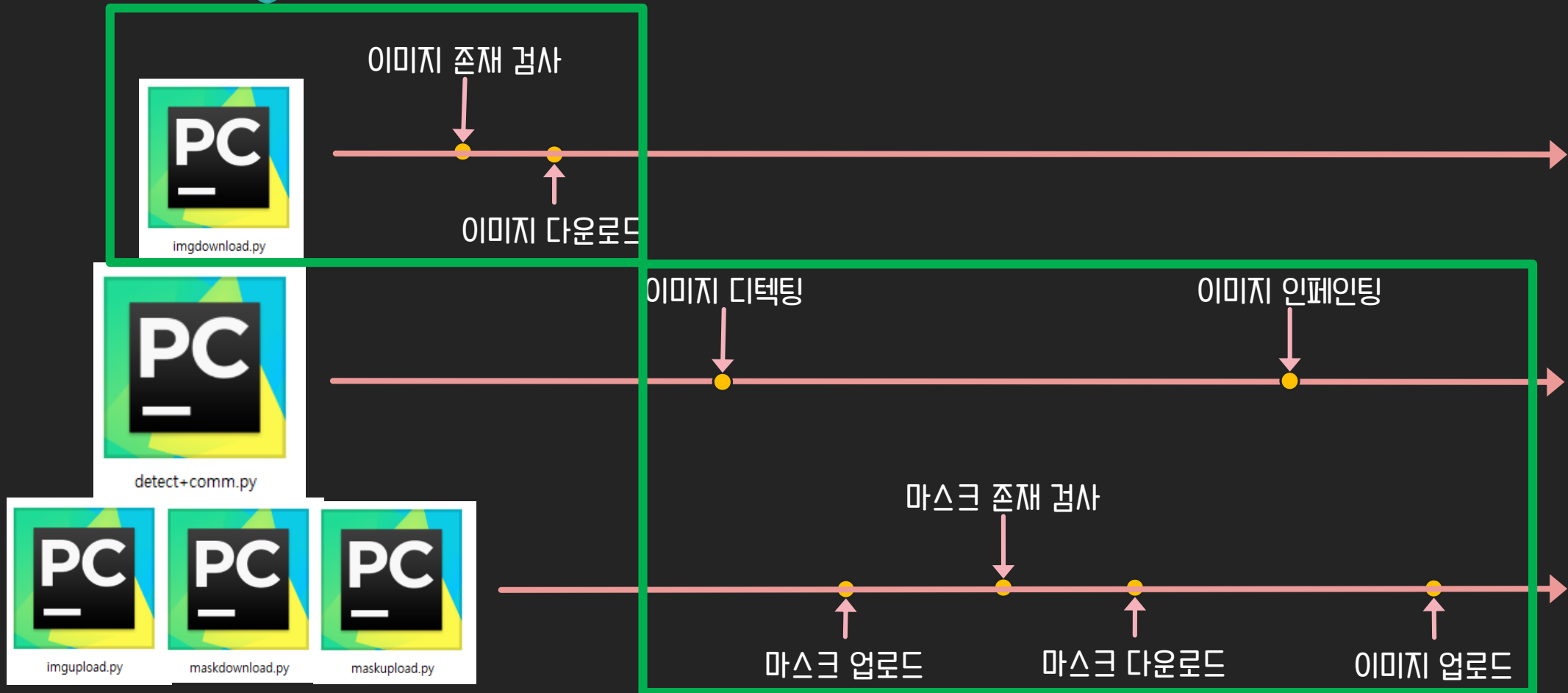
Background



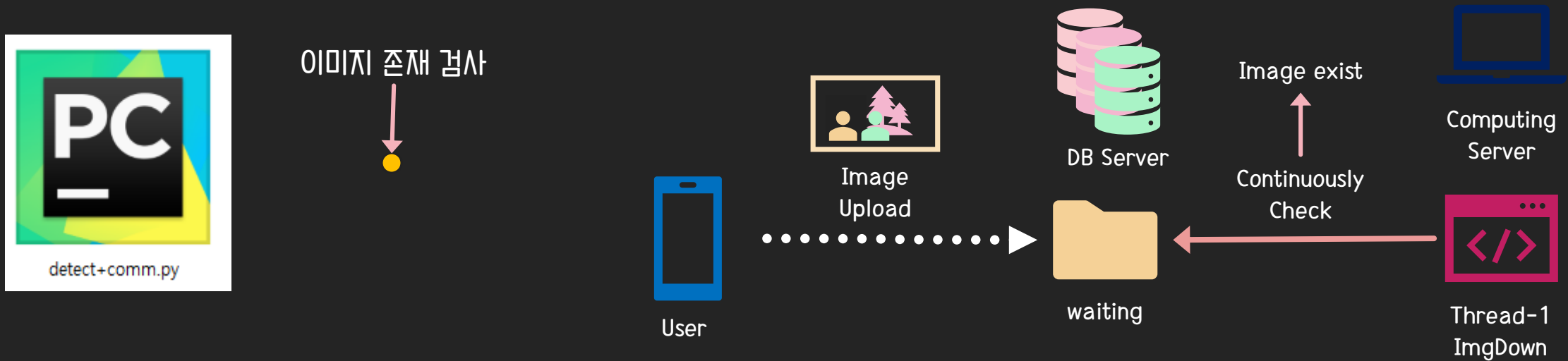
Background



Background



Background



Background



Image
Upload



waiting



Thread-1
ImgDown

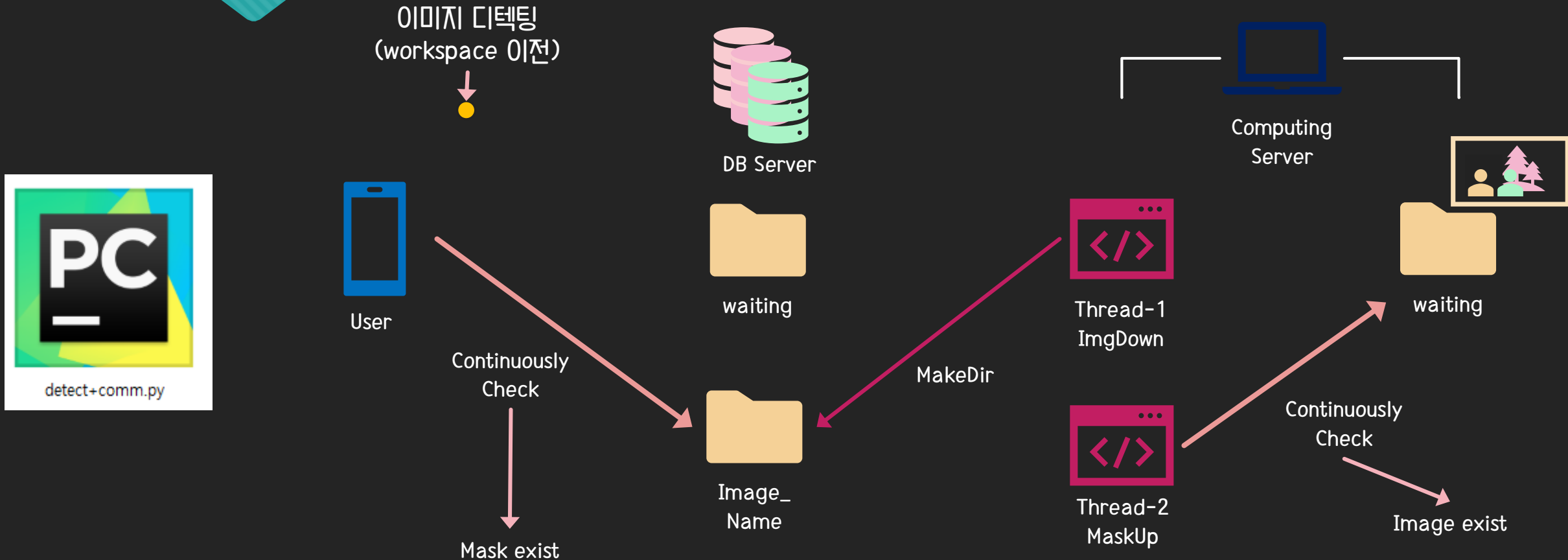


Computing
Server



waiting

Background

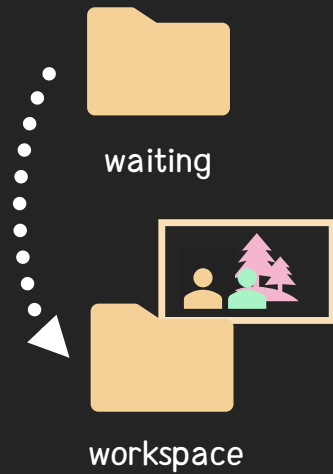
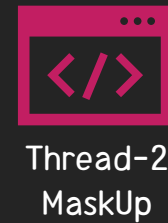
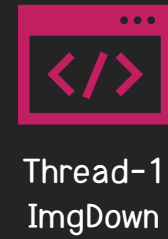
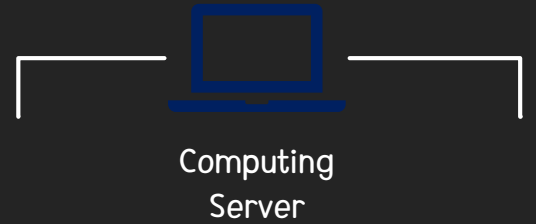
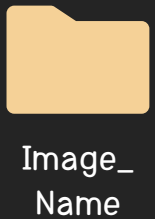
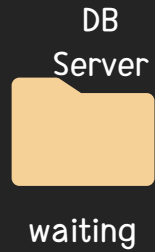


Background

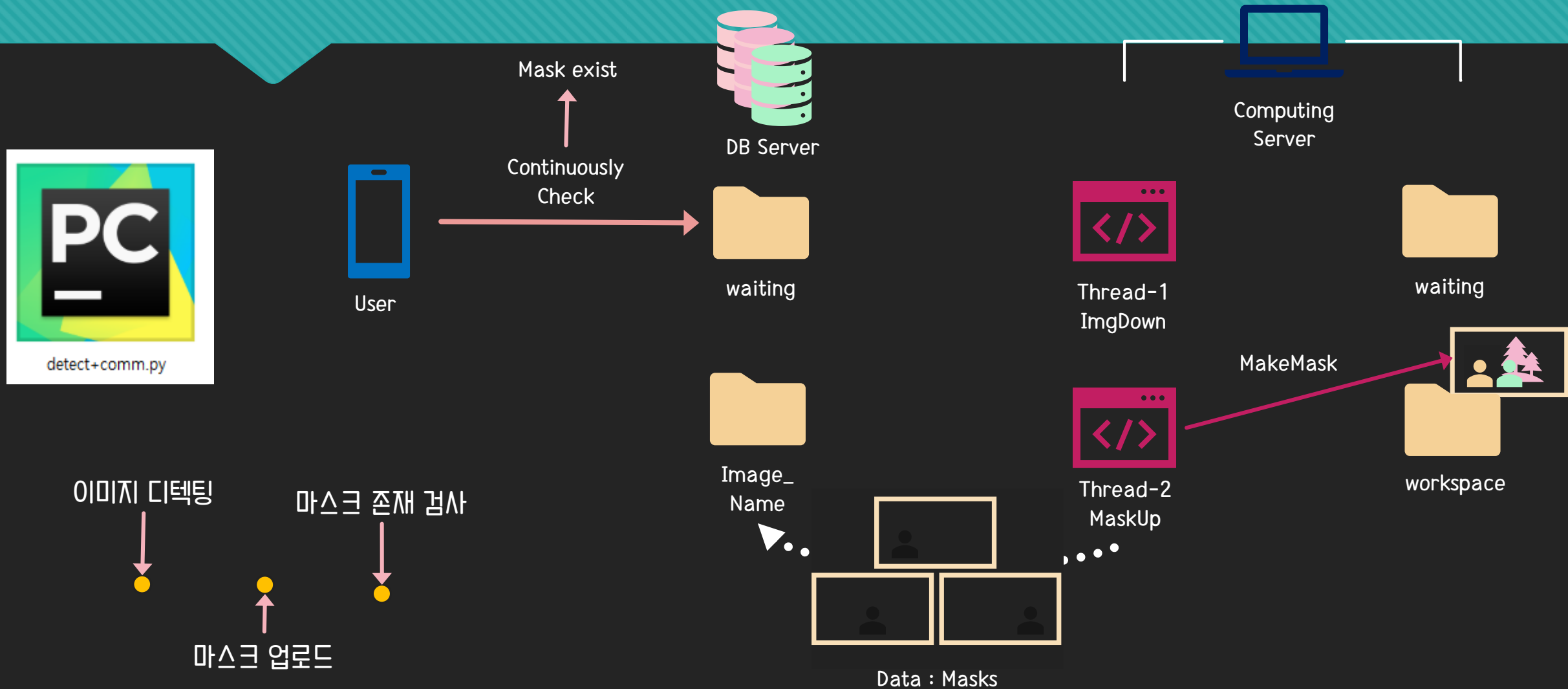
이미지 디텍팅
(workspace 이후)



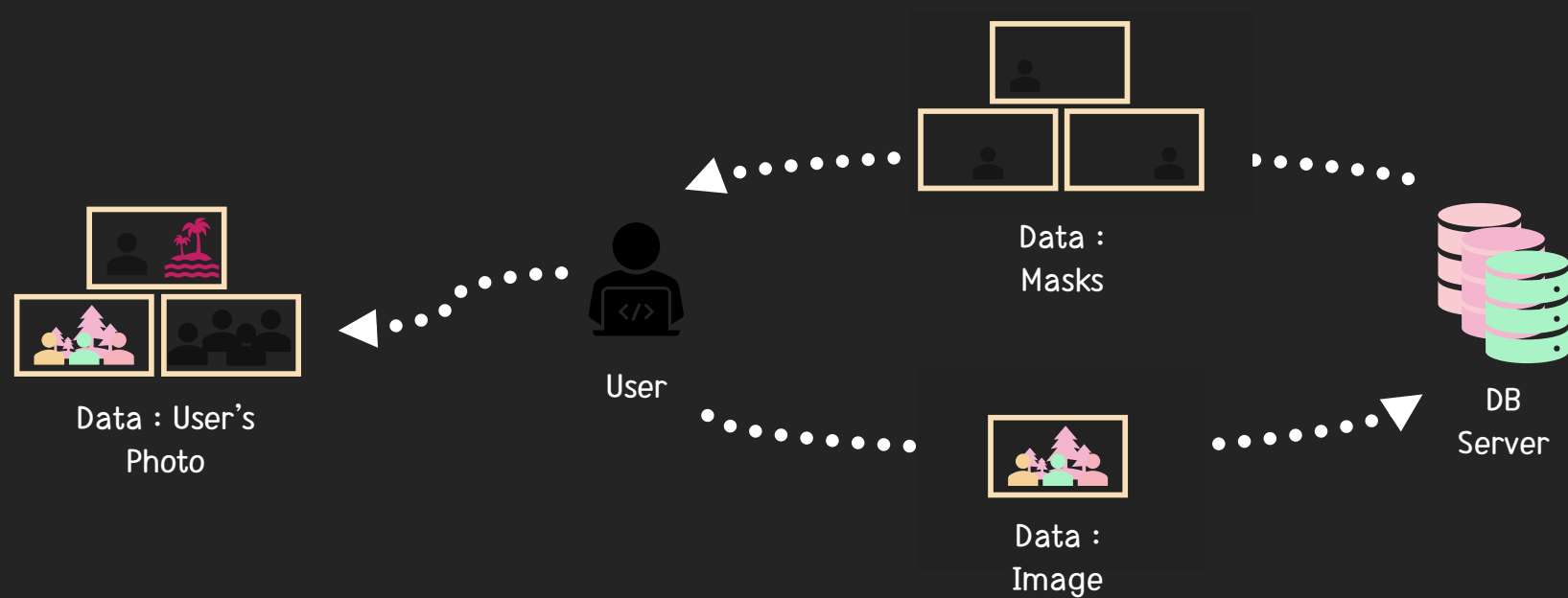
Mask exist
↑
Continuously
Check



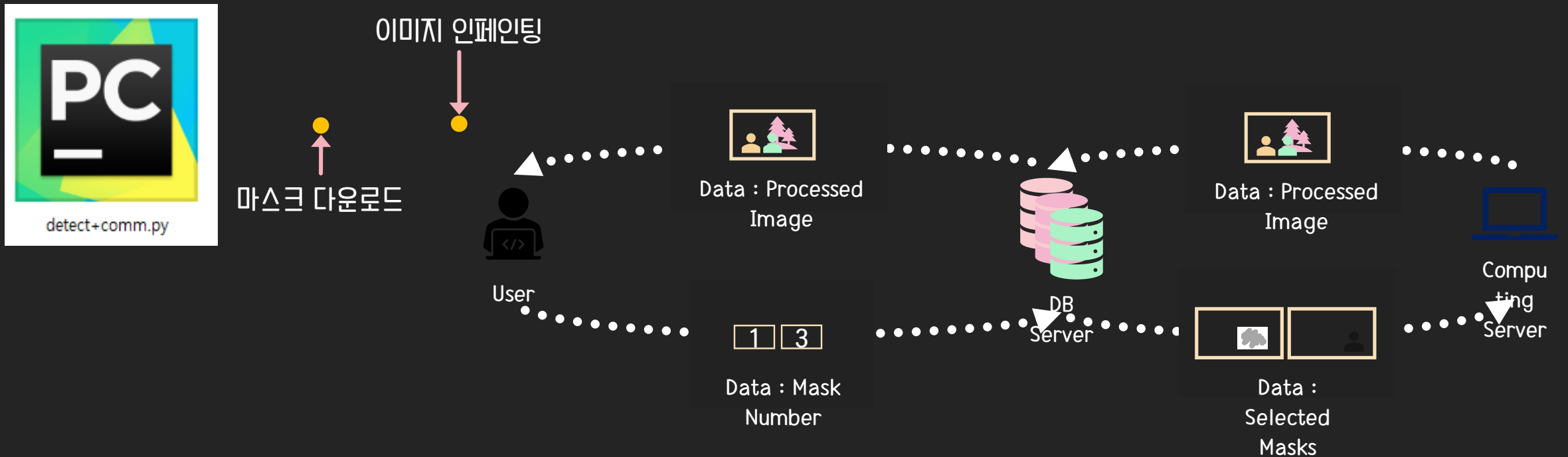
Background



Background



Background



테스트 결과

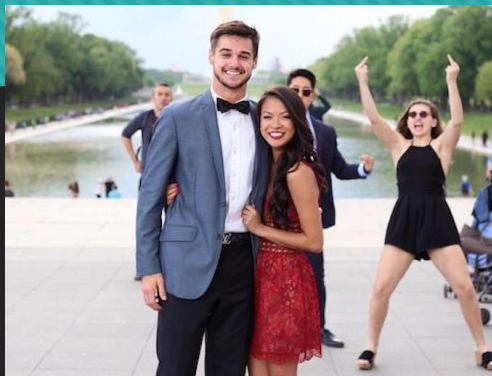
Test Results



테스트 결과



$512 * 288$



$680 * 512$

$2048 * 1487$



$1024 * 512$



테스트 결과



512 * 288

```
Processing 1 images
image                shape: (288
molded_images        shape: (1,
image_metas         shape: (1,
anchors              shape: (1,
detectP 29.913956초 걸렸습니다.
['0.png', '1.png', '2.png', '3.png',
----- /
```

```
Shape of image: (288, 512, 3)
Model loaded.
Image has been made
inpaintP 6.949903초 걸렸습니다.
```



테스트 결과



680 * 512

```
Processing 1 images
image                               shape
molded_images                       shape
image_metas                         shape
anchors                             shape
detectP 29.086348초 걸렸습니다.
['0.png', '1.png', '2.png', '3.png']
```

```
Shape of image: (512, 680, 3)
Model loaded.
Image has been made
inpaintP 13.467389초 걸렸습니다.
```



테스트 결과



1024*512

```
Processing 1 images
image                shap
molded_images        shap
image_metas         shap
anchors              shap
detectP 26.087975초 걸렸습니다.
['0.png', '1.png', '2.png', '3.png', '4.png', '5.png', '6.png', '7.png', '8.png', '9.png', '10.png', '11.png', '12.png', '13.png', '14.png', '15.png', '16.png', '17.png', '18.png', '19.png', '20.png', '21.png', '22.png', '23.png', '24.png', '25.png', '26.png', '27.png', '28.png', '29.png', '30.png', '31.png', '32.png', '33.png', '34.png', '35.png', '36.png', '37.png', '38.png', '39.png', '40.png', '41.png', '42.png', '43.png', '44.png', '45.png', '46.png', '47.png', '48.png', '49.png', '50.png', '51.png', '52.png', '53.png', '54.png', '55.png', '56.png', '57.png', '58.png', '59.png', '60.png', '61.png', '62.png', '63.png', '64.png', '65.png', '66.png', '67.png', '68.png', '69.png', '70.png', '71.png', '72.png', '73.png', '74.png', '75.png', '76.png', '77.png', '78.png', '79.png', '80.png', '81.png', '82.png', '83.png', '84.png', '85.png', '86.png', '87.png', '88.png', '89.png', '90.png', '91.png', '92.png', '93.png', '94.png', '95.png', '96.png', '97.png', '98.png', '99.png']
```



```
Shape of image: (512, 1024, 3)
Model loaded.
Image has been made
inpaintP 18.793956초 걸렸습니다.
```

테스트 결과

2048 * 1487



```
Processing 1 images
image                                     shap
molded_images                           shap
image metas                             shap
anchors                                 shap
detectP 29.547943초 걸렸습니다.
['0.png', '1.png', '2.png', '3.png', '4.png', '5.png', '6.png', '7.png', '8.png', '9.png']
```

Model loaded.

```
Process finished with exit code -1073740791 (0xC0000409)
```

테스트 결과

Detecting : Avg 30 sec

Inpainting : each different

자체 평가 및 개선방안

Self-Assessment &
Improvement

김재성 32150973 소프트웨어학과

처음으로 빌드 및 배포 단계까지 간 모바일 어플리케이션을 개발하면서, 프론트와 백 간의 소통을 위한 API에 생각보다 많은 노력이 들어간다는 것을 깨달았다고, 설계 단계에서 팀원들과 많은 토론을 거치면서, 데이터 흐름을 여러 번 수정하면서, 초반 설계가 전체적인 개발 기간에 많은 영향을 끼친다는 것을 느꼈습니다.

React를 처음 접하다 보니, 기본적으로 적극적으로 어필하는 개념들(hooks, redux, navigator 등)을 제대로 활용하지 못하여 “React스러운” 코드를 작성하지 못한 것 같아 아쉽습니다. JavaScript와 React.js 공부를 더 해서 Refactoring 과정으로 내부가 더 탄탄한 코드 작성이 필요하다고 생각합니다.

차무송 32154577 소프트웨어학과

처음으로 어플리케이션을 만들면서 안드로이드, iOS의 차이점도 알게 됐습니다. 간단한 API의 개발에도 어려움이 많았지만 팀원들과의 협업을 통하여 많이 발전할 수 있었습니다.

이번에는 급하게 만들려고 하다 보니 기본기가 많이 부족했던 것 같습니다. 다음부터는 천천히 기초를 확실히 다지고 개념을 다 잡고 개발에 임하고 싶습니다.

전상민 32154049 소프트웨어학과

머신 러닝 모델을 이용한 두번째 프로젝트입니다. 기존에는 동영상을 가지고 했다면 지금은 사진을 가지고 했는데, 알아갈수록 부족함을 느끼는 프로젝트였습니다. 아직 전반적으로 개발되지 않은 부분의 모델을 이용하다 보니 Performance가 우수하게 나오는 것도 아니었고 이를 실제 서비스 환경에 제공하려다 보니 여러 악조건이 많았는데, 그렇다고 개선할 수 있는 여지가 거의 없어 많이 발버둥 치고 노력한 프로그램입니다.

이번 프로젝트를 거치면서 머신 러닝에 대해 학습하는 과정을 거치지 않고 이미 설계된 코드와 학습을 마친 모델을 이용해 바로 사용한 것이 아쉬운 점으로 남습니다. 이 부분에 좀 더 많은 지식이 있어야 될 것 같고 다음 개발에는 직접 설계와 학습을 거친 모델을 사용하고 싶습니다.

데모 영상

Demonstration



[프로젝트 4#]



Q & A



감사합니다!