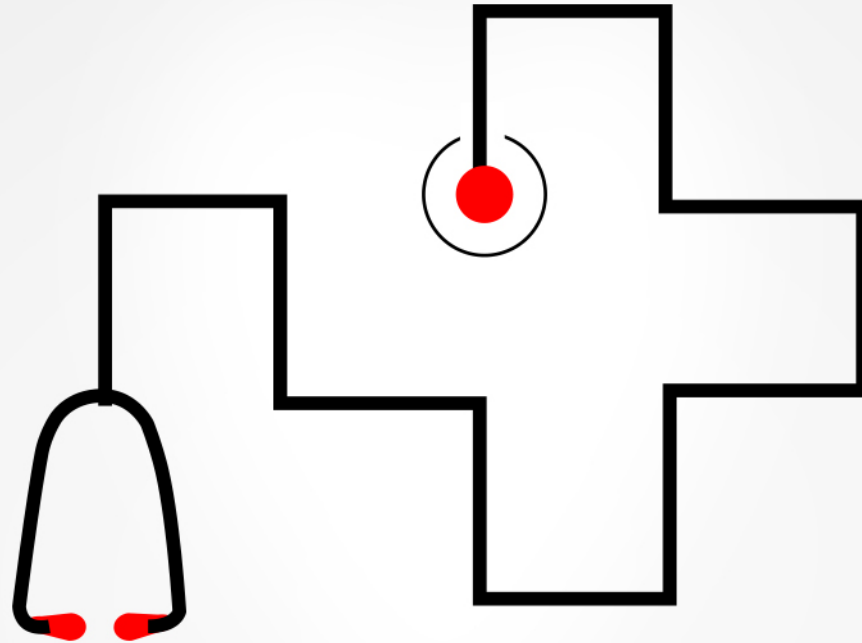




지도교수:
하영국 교수님



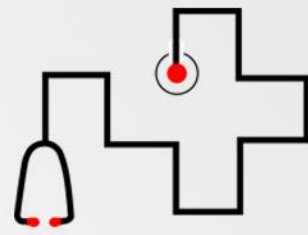
졸업 프로젝트2 T7

2nd Implementation

컴퓨터공학과

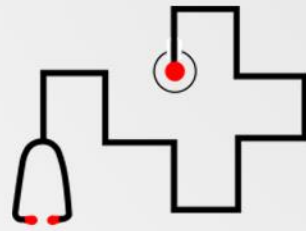
201511272 양재민, 201411295 이상훈, 201511295 조범석

0. 목차



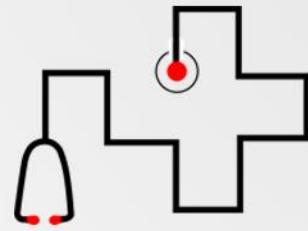
1. 프로젝트 제안 배경
2. 프로젝트 소개
3. SW, HW 구성
4. 요구사항 분석
5. SW 분석 (한계점 포함)
6. System Test - Pass/Fail
7. 추적성 분석

1. 프로젝트 제안 배경



1. 현재 의료 이미지 데이터들은 의료진들만 접근 가능하고, 병원이 달라질 경우 의료진들은 다른 병원의 정보를 확인할 수 없다. 이에 웹 서비스로 의료 이미지 데이터 통합 시스템을 구성한다.
2. 의료 이미지 데이터 관리를 블록체인으로 관리해 신뢰성과 보안성을 높여주고 환자들이 자신의 의료 이미지를 조회할 수 있도록 한다. 즉 데이터 주권을 확립한다. 또한 원본 이미지 데이터에서 용종의 위치를 찾아내 예측해주는 기능을 딥러닝으로 개발해 의료진들이 쉽게 용종을 파악하고 분석할 수 있게 한다

2. 프로젝트 소개

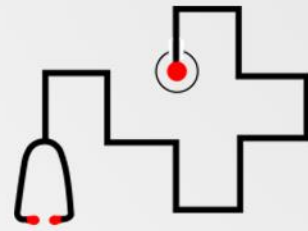


1. 프로젝트 명: MediChA.I.n = Medical + BlockChain + A.I.

2. Description

1. 총 세 가지의 SW로 구성된다. 의료 이미지 데이터와 이력 관리는 블록체인 (하이퍼레저 패브릭, IPFS), 의료 이미지 데이터 학습 및 예측은 딥러닝, 데이터 업로드 및 사용자 관리 및 접근은 웹 서비스로 각 기능을 구성한다.
2. 초기에는 공공 의료 이미지 데이터로 학습을 한다. 원본 이미지, 용종 데이터가 담긴 이미지 두 이미지로 학습을 진행하며 분산 환경에서 학습을 진행 후 중앙 서버에서 학습 모델을 합치는 방식을 사용한다. 이를 통해 데이터가 한 곳에 몰리는 것을 방지하고, 보안성을 높이고 과부하가 발생할 가능성을 줄인다.
3. 이미지들은 전부 IPFS 에서 관리되고 IPFS를 통해 이미지를 조회할 수 있는 해시들은 블록체인에서 관리된다. 이를 통해 데이터의 위, 변조를 방지하고 신뢰성을 높인다. 또한 환자들이 쉽게 자신의 데이터를 확인할 수 있게 한다.

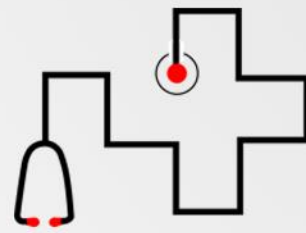
2. 프로젝트 소개



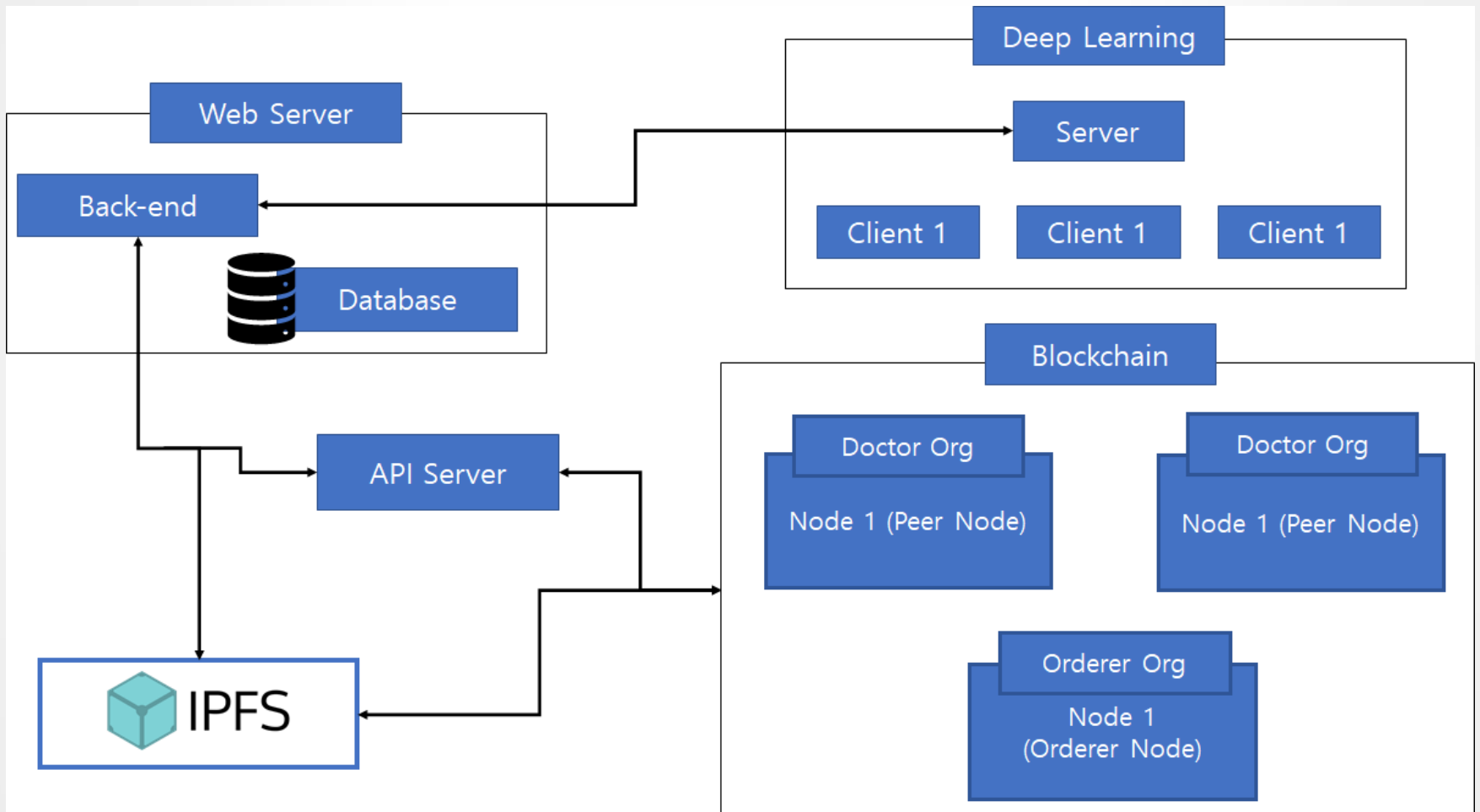
3. 차별성

1. 이상한 이미지 데이터 (상이한 신체 부위 또는 의료 데이터가 아닌 경우) 가 들어올 경우 딥러닝으로 거르거나, 경고 메시지를 보내줄 수 있다.
2. 학습을 분산화해서 하기 때문에 데이터의 중앙화를 막는다.
3. 이미지들은 전부 IPFS 에서 관리되고 IPFS를 통해 이미지를 조회할 수 있는 해시들은 블록체인에서 관리된다. 이를 통해 데이터의 위, 변조를 방지하고 신뢰성을 높인다. 또한 환자들이 쉽게 자신의 데이터를 확인할 수 있게 한다.
4. 초기 학습이 완료된 이후 랜덤으로 블록을 지정해 모델을 검증, 추가 학습한다.

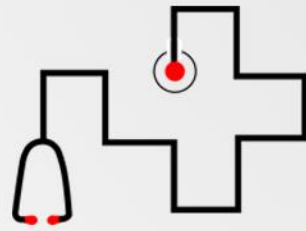
3. SW, HW 구성



1. 전체 시스템 아키텍처



3. SW, HW 구성



2. SW 구성

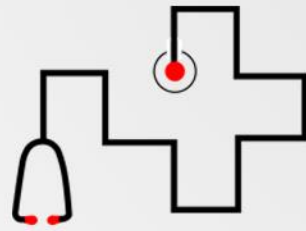
1. 의료 데이터를 보관할 수 있는 블록체인

- Name: Hyperledger Fabric
- Version: 2.2
- 역할: 프라이빗 블록체인 네트워크 구성 (의사 조직, 환자 조직)
- Source: Github (<https://github.com/hyperledger/fabric>)

2. 블록체인 관리자 페이지

- Name: Hyperledger Explorer
- Version: 1.1.3
- 역할: 블록체인 네트워크 상태 확인 가능
- Source: Github (<https://github.com/hyperledger/blockchain-explore>)

3. SW, HW 구성



2. SW 구성

3. 이미지 저장소

- Name: IPFS
- Version: 0.7
- 역할: 이미지를 분산 저장한다.

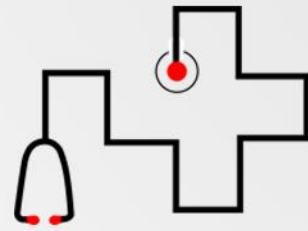
4. 딥러닝 시스템

- Name: Tensorflow
- Version: 2.0 이상
- 역할: 딥러닝 모델을 생성할 때 사용한다.

5. 웹 서비스

- Name: Python Flask / HTML, Javascript, CSS
- 역할: 웹 서비스를 제공한다.

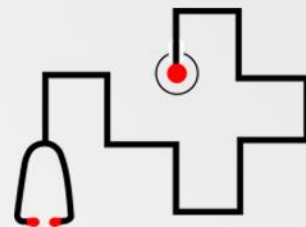
3. SW, HW 구성



3. HW 구성

1. Web Service 구동을 위한 AWS 서버
 - AWS Cloud 이용, HW 비용 절감을 시도한다. 필요한 포트는 1가지, 의사 또는 환자가 접근하기 위함이다.
2. Blockchain Network + Deep Learning Client 를 위한 3개의 서버 컴퓨터
 - HW 자원 부족을 해결하기 위해 하나의 컴퓨터에서 Docker 기반으로 3개의 서버 컴퓨터를 생성해 블록체인 네트워크를 돌린다. Window 위 WSL 을 사용해 Ubuntu 처럼 사용하고 각 포트들은 서로 통신할 수 있다.
3. Deep Learning 을 위한 서버 컴퓨터
 - 마찬가지로 Docker 기반으로 운영되며 서버와 클라이언트는 모델을 주고받으면서 서버에서 최종 모델로 만들어 웹 서버에 전송한다.

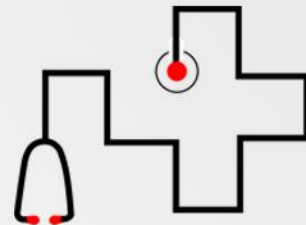
4. 요구사항 분석



1. 기능적 요구사항

Term	Description	Operation
1.1 Sign In	로그인하여 서비스를 이용한다.	signIn()
1.2 Sign Out	로그인이 되어있는 상태에서 로그아웃을 한다.	signOut()
1.3 Sign Up	회원 가입한다. 의료진은 의사 번호 입력 과정을 더 거친다.	signUp()
1.4 Upload Image	이미지를 업로드 한다. 이 행동은 의료진만 가능하다.	uploadImage()
1.5 Detect Strange	이상한 이미지가 들어왔는지 감지한다.	detectStrange()
1.6 Get Info	의료 정보를 요청해 가져온다.	getInfo()
1.7 Authorize	로그인할 때의 회원 정보를 바탕으로 시스템 접근 권한을 얻는다.	getAuthorize()
2.1 Check Authorized Data	의료진 권한을 가진 사용자가 환자 고유 주소 (환자의 주민등록 번호 해시화한 형태) 를 바탕으로 환자의 개인 데이터를 조회한다.	checkAuthorizedData()
2.2 Check Self Data	환자인 사용자가 환자 고유 주소를 이용하여 자신의 데이터를 조회한다.	checkMyData()
2.3 Upload Data	의료진 사용자가 환자 고유 주소를 이용하여 환자의 데이터를 업로드 한다.	uploadData()
2.4 Upload To IPFS	IPFS에 이미지를 업로드 한 뒤 이미지에 해당하는 해시 아이디를 받는다.	uploadToIPFS()

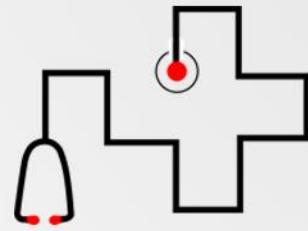
4. 요구사항 분석



1. 기능적 요구사항

Term	Description	Operation
3.1 Send Model	딥러닝 서버에서 각 노드에게 모델을 전송한다.	sendModel()
3.2 Learning Model	각 노드별로 모델을 학습한다.	Fitting()
3.3 Send Model from Node to Server	각 노드에서 정해진 블록들을 모두 학습하면 모델을 전송한다.	sendToServer()
3.4 Modify Model	딥러닝 서버에서 각 노드에서 보내온 모델들을 종합한다.	modifyModel()
3.5 Validate Model	딥러닝 서버에서 종합한 모델을 검증한다.	validateModel()
3.6 Check Loss	서버에서 각 노드에서 나온 validation loss를 이전의 validation loss와 비교하여 master 모델을 교체할지 결정하고 특정 횟수 만큼 개선이 이루어지지 않으면 training을 중지한다.	checkLoss()
3.7 Fetch Model	딥러닝 서버에서 웹서버에 분석 모델을 전송한다.	fetchModel()

4. 요구사항 분석

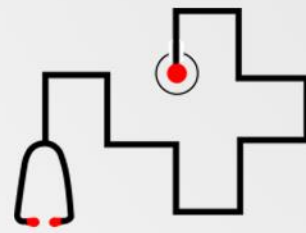


2. 비기능적 요구사항

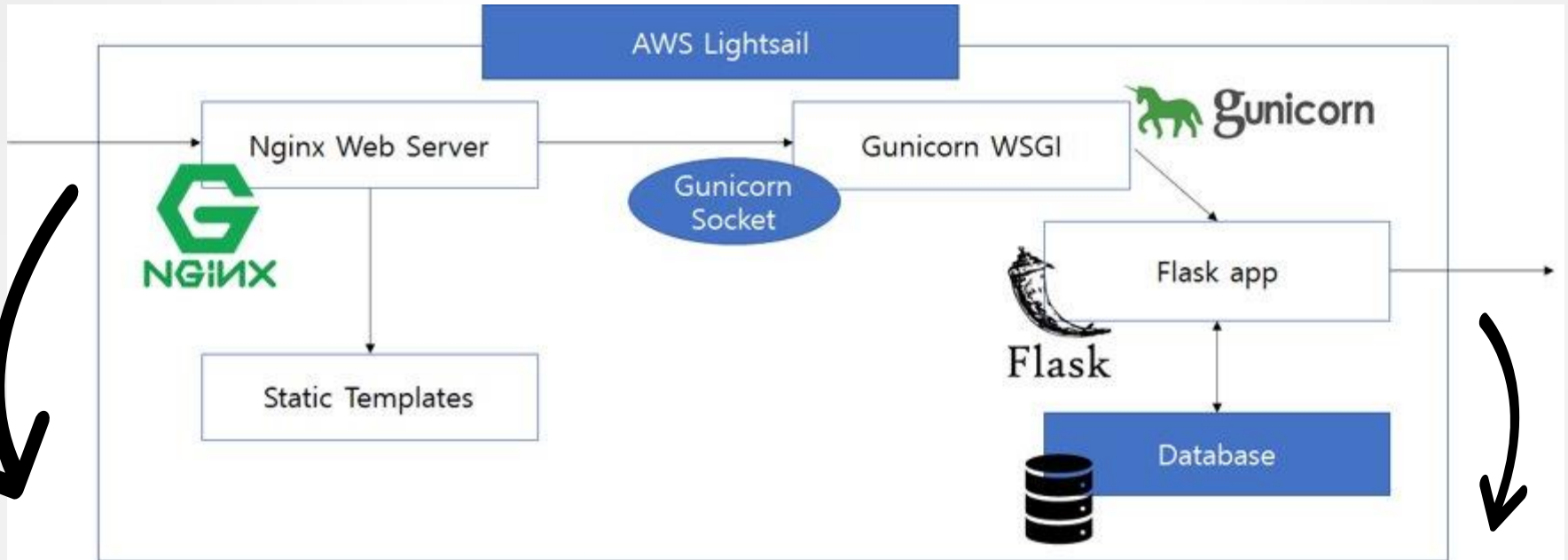
Description

1. 의료 이미지 업로드가 30초 내로 빠르게 되어야한다.
2. 딥러닝 예측이 1분 안에 이미지 생성까지 빠르게 되어야 한다.
3. 사용자는 자신의 개인 정보를 안전하게 유지할 수 있어야한다.
4. 의료 정보 조회가 1분 내로 완료된다.
5. 하이퍼레저 패브릭을 활용해 의료 이미지 데이터 링크와 개인 데이터로 블록 체인을 구성할 수 있어야 한다.
6. 학습할 수 있는 환경이 구성되어야 한다.

5. SW 분석



1) Web System 인터페이스



1. Request to Web

Sign In

Sign out

Sign Up

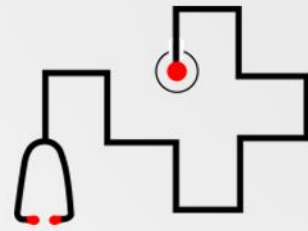
REST API (Blockchain)

GET /doctor/get-patient-image

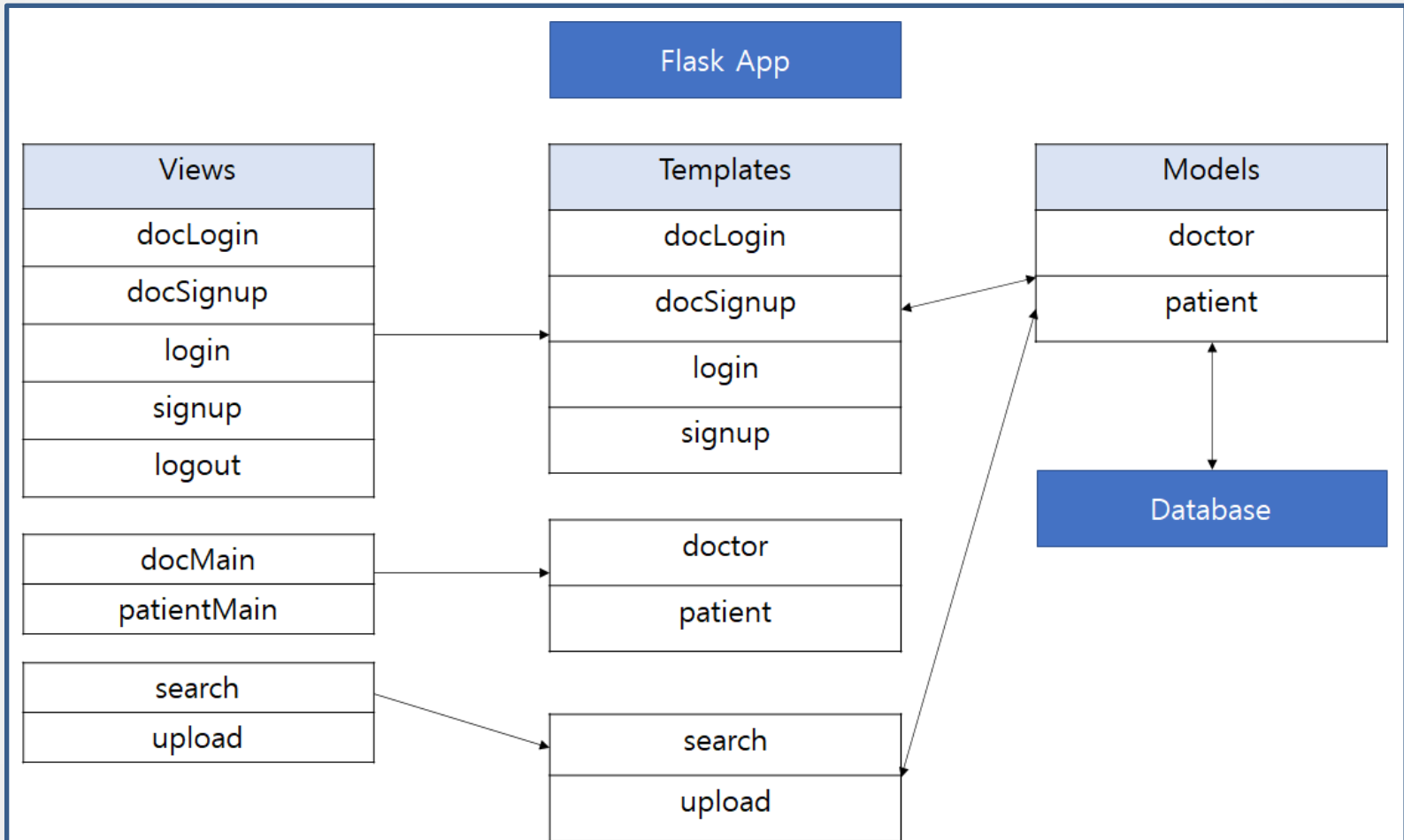
GET /patient/get-my-img

POST /doctor/upload-patient-img

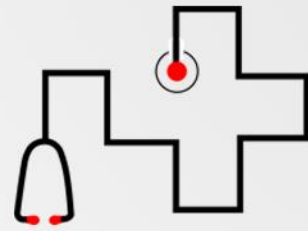
5. SW 분석



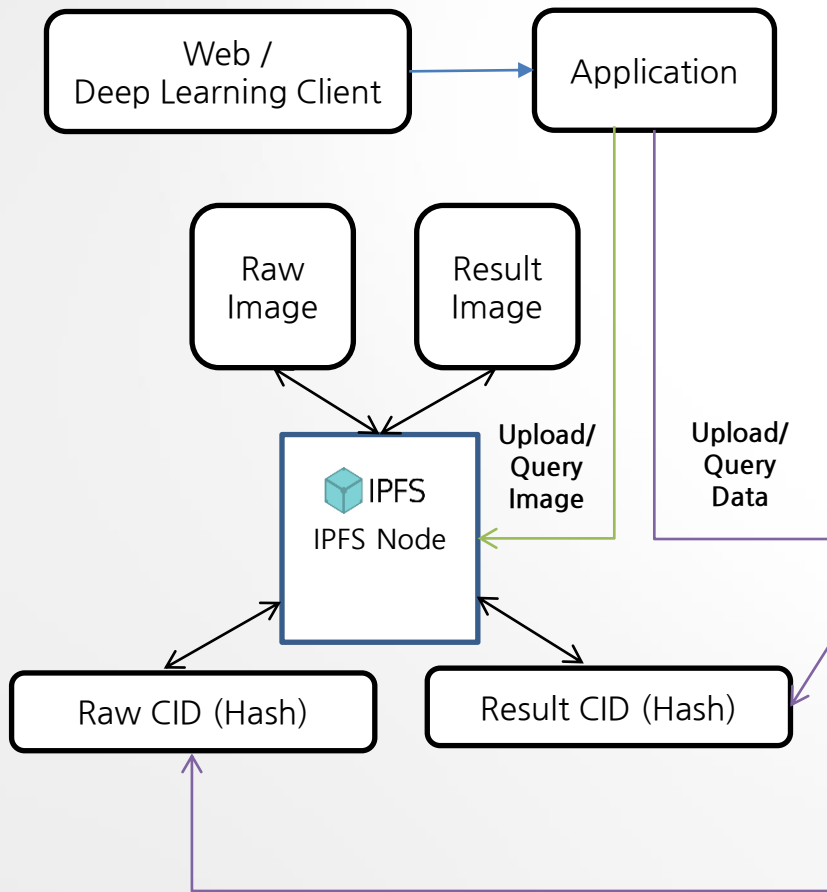
1) Web System 상세 디자인



5. SW 분석

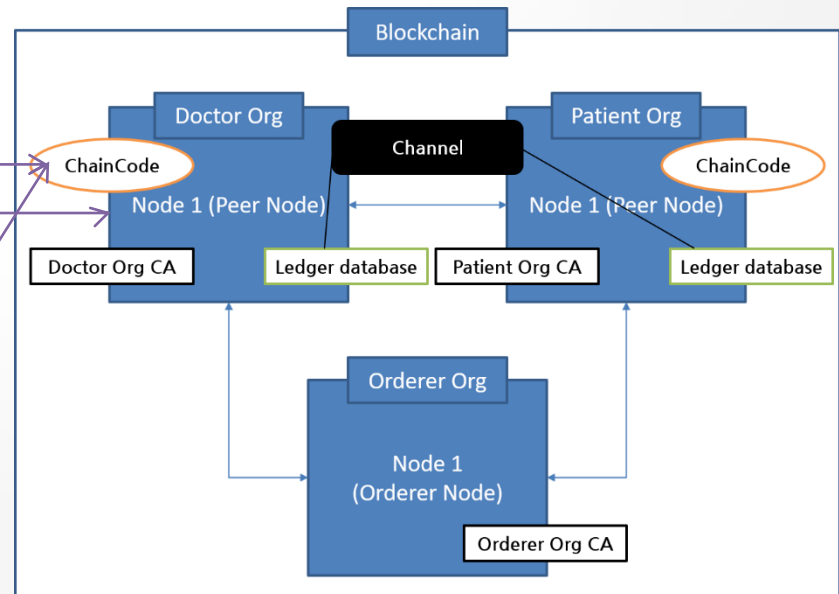


2) Blockchain & IPFS 인터페이스

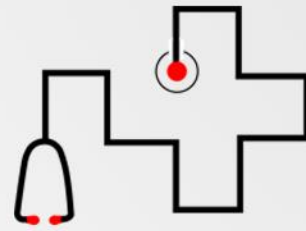


REST API (Web)
GET /doctor/get-patient-img
GET /patient/get-my-img
POST /doctor/upload-patient-img

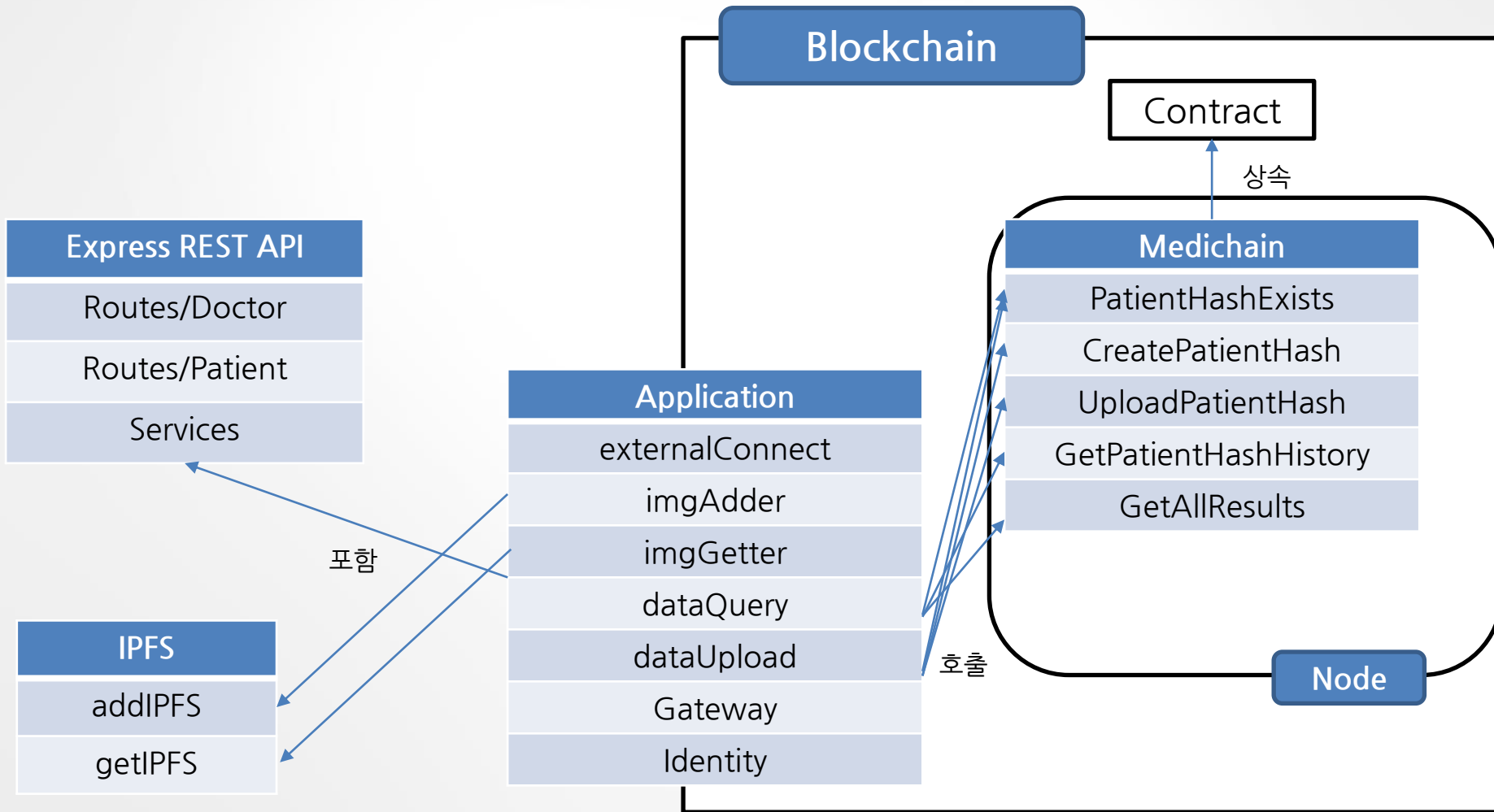
Application (Deep Learning)
Query Traing 1, 2, 3's image



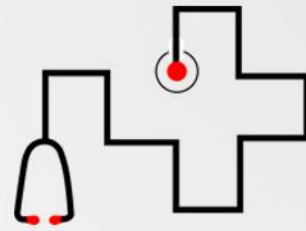
5. SW 분석



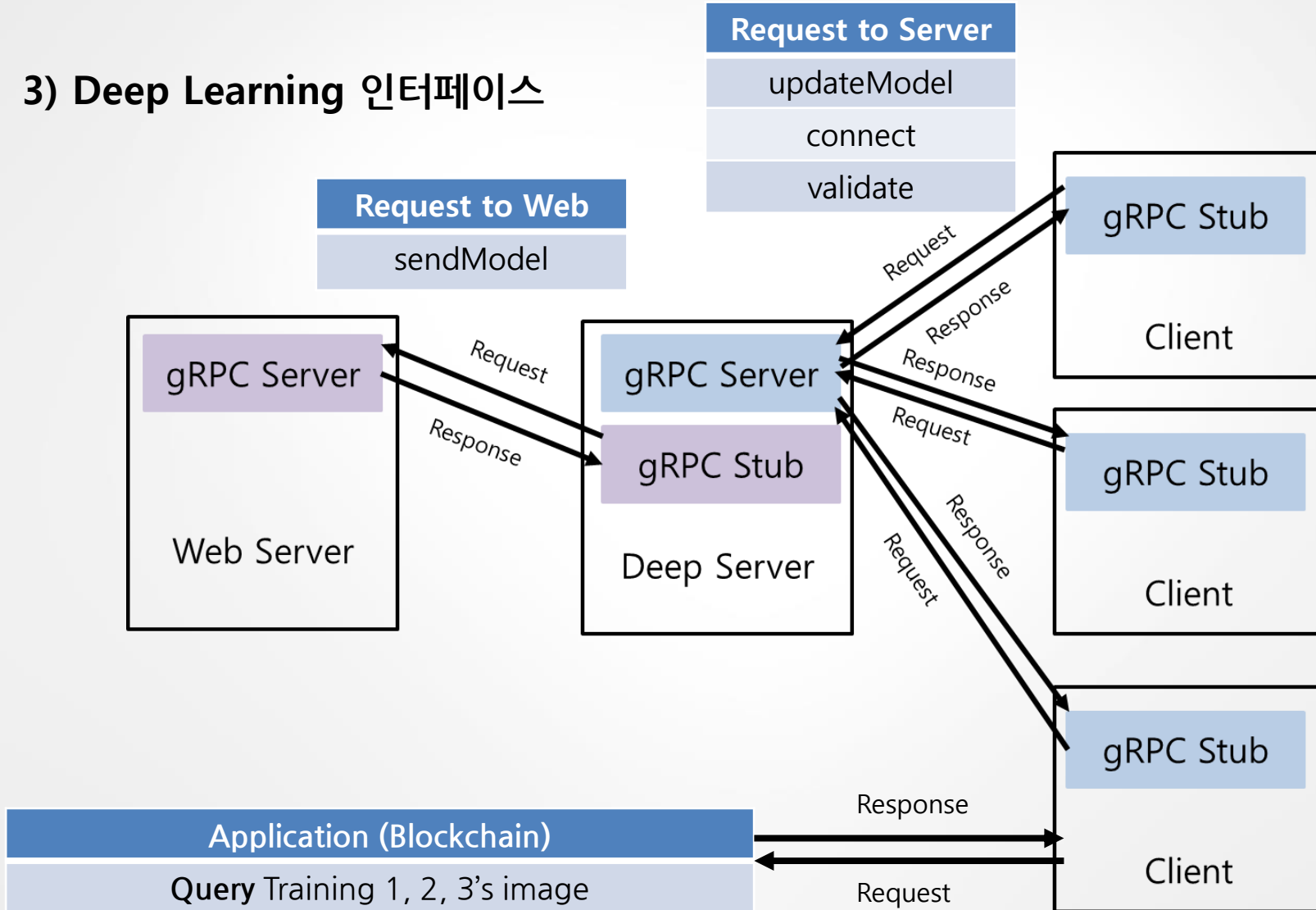
2) Blockchain & IPFS 상세 디자인



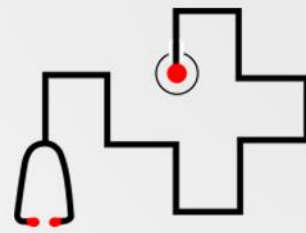
5. SW 분석



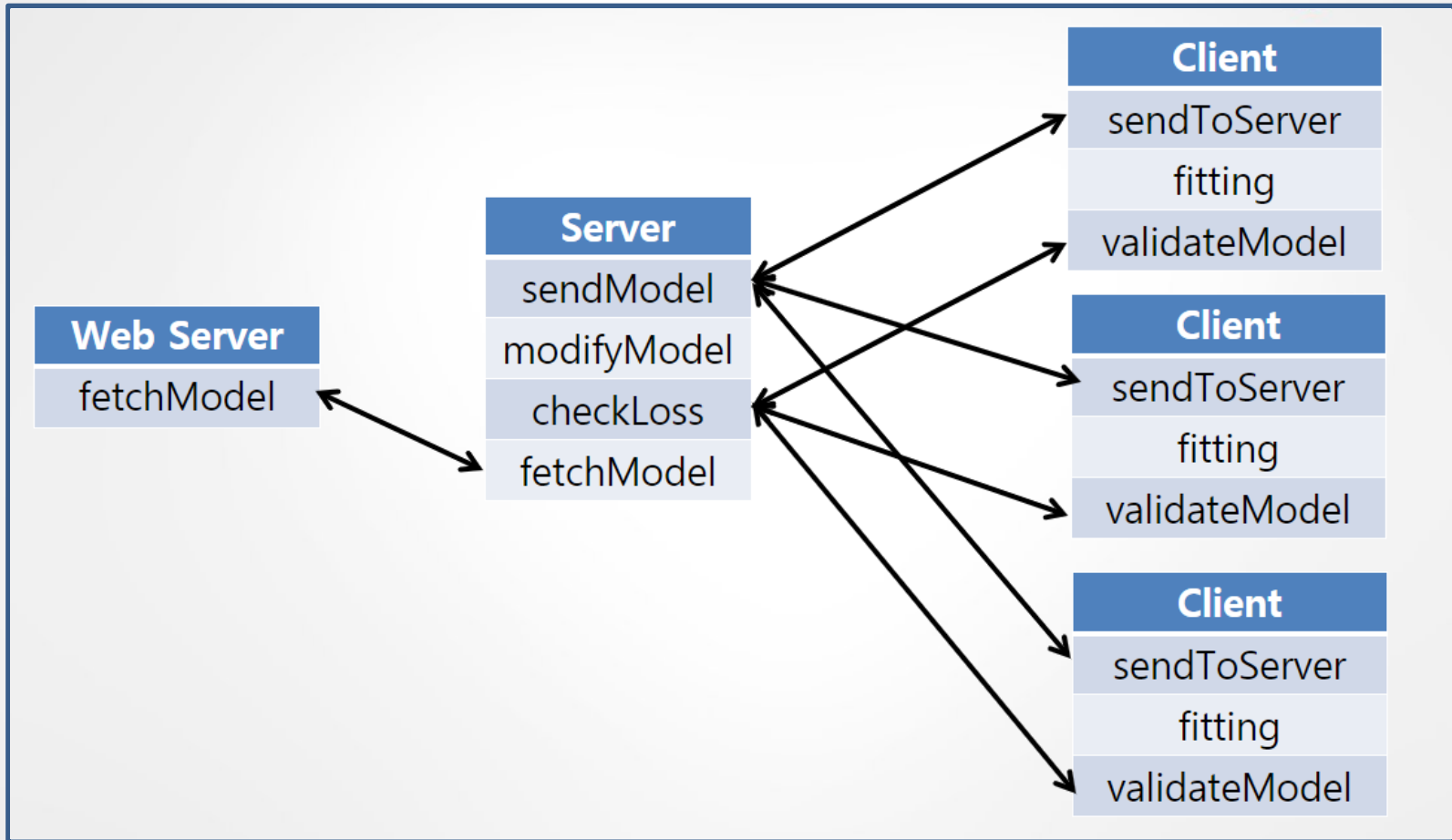
3) Deep Learning 인터페이스



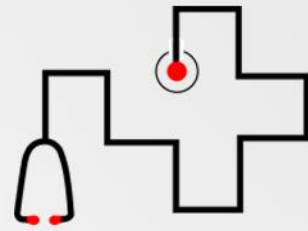
5. SW 분석



3) Deep Learning 상세 디자인



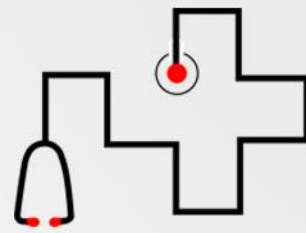
6. System Test



1. Web System

No.	상세 설명	입력	기대 결과	Pass/Fail
1.1	시스템 사용 중에 사용자의 정보가 유지되는지 테스트한다.	로그인 하여 시스템을 사용한다.	서비스 이용 중에 고유의 회원 주소가 변하지 않는다.	Pass
1.2	시스템에 정상적으로 회원가입이 가능한지 테스트 한다.	회원가입을 한다.	가입 시 입력한 신분에 맞게 회원 정보가 저장된다.	Pass
1.3	시스템에 로그인이 가능한지 테스트한다.	로그인을 한다.	저장 되어있는 회원 정보에 따라 시스템 접근 권한을 얻는다.	Pass

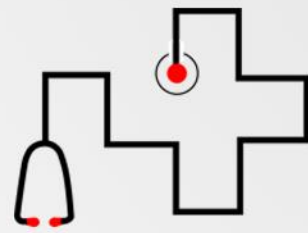
6. System Test



2. Blockchain

No.	상세 설명	입력	기대 결과	Pass/Fail
2.1	의료진은 환자 고유 ID를 입력해 환자의 이미지 데이터를 가져온다.	환자의 고유 ID (주민번호 해시화) 입력	환자의 원본 이미지, 예측 이미지 등 환자의 블록체인 데이터가 반환된다.	Pass
2.2	환자는 자신의 고유 ID를 입력해 자신의 이미지 데이터를 가져온다.	자신의 고유 ID (주민번호 해시화) 입력 (추가 인증과정)	자신의 원본 이미지, 예측 이미지 등 블록체인 데이터가 반환된다.	Pass
2.3	의료진은 환자의 고유 ID와 함께 환자의 데이터를 블록체인에 저장한다.	환자의 고유 ID (주민번호 해시화), IPFS CID (Raw Image, Prediction Image)	블록체인에 환자의 데이터가 기록된다.	Pass
2.4	원본 이미지와 예측 이미지를 IPFS를 통해 저장한다.	원본 이미지와 예측 이미지를 입력한다.	IPFS에 접근 가능한 CID(해시 문자열) 2개가 반환된다.	Pass
2.5	CID를 통해 IPFS에서 이미지를 가져온다.	두 개의 CID를 입력한다.	원본 이미지와 예측 이미지가 파일 형태로 반환된다.	Pass

6. System Test



3. Deep learning

No.	상세 설명	입력	기대 결과	Pass/Fail
3.1	Response 하기 전 Server의 Model과 Response 후 Client의 Model을 비교	Master model과 averaging 한 결과를 전송	Client에 모델이 전송된다.	Pass
3.2	Client에서 Epoch이 끝났을 때, 학습 전 Model과 학습 후 Model을 비교	학습용 데이터, hyperparameter	Loss에 진전이 있으면 Model에 변화가 있고 없으면 변화가 없다.	Pass
3.3	전송 후 노드 클라이언트와 서버의 Model을 비교	학습 완료 후 학습한 Model	Server에 Client와 동일한 Model을 전송받는다.	Pass
3.4	서버에서는 각 노드에서 보내온 Model들의 업데이트 내용들을 사용하여 새로운 Master model을 업데이트 하는지 테스트	노드 클라이언트에서 전송한 Model	Master model과 Client에서 보내온 Model이 Averaging 된다.	Pass
3.5	Validation data로 학습이 잘 되었는지 테스트	Validation data	Validation data로 검증한 loss와 accuracy가 출력된다.	Pass
3.6	이전의 Loss를 저장하고 있는지, 지금의 Loss와 비교가 되는지, 이전 Model을 저장하고 있는지 테스트	이전 loss, 현재 loss, 이전 Model	서버에서 Master 모델을 교체할지 결정하고 training을 중지할지 결정	Pass
3.7	전송 후 딥러닝 서버와 웹서버의 모델을 비교	최종 Model	웹서버에 딥러닝 서버에서 전송한 모델이 저장된다.	Pass

7. Traceability Analysis

