2025년 오픈소스 개발자대회 기능명세서

가. 기본정보(팀명 : Blocker)

□ 프로젝트명

○ 블록체인 기반 자율주행 IoT 소프트웨어 업데이트 플랫폼

□ 프로젝트 개요

○ 자율주행 차량을 비롯한 전 세계 수십억 대의 디바이스는 항상 해커들의 주요 공격 대상이 됩니다. 특히 최근에는 공급망 공격(Supply Chain Attack) 을 통해 펌웨어·소프트웨어 업데이트 경로 자체가 악용되는 사례가 늘어나고 있습니다. 따라서 안전한 업데이트의 신뢰성 확보가 중요해졌습니다.

본 프로젝트는 공급망 공격, 위변조 공격 등 다양한 보안 위협에 대응하기 위해 안전하고 효율적인 무선 소프트웨어 업데이트 플랫폼을 오픈소스로 제공하여, 누구나 외부 공격으로부터 안전하게 업데이트를 배포하고 적용할 수 있도록 하는 것을 목표로 합니다. 자동차, 의료기기, 스마트 가전, 모바일 기기 등 다양한 임베디드 디바이스에 범용적으로 적용 가능하도록 설계되었으며, 기기별 권한 관리 시스템과 블록체인 기반 위변조 방지 기술을 통해 해킹으로부터 안전한 업데이트 환경을 제공합니다. 다른 개발자들이 손쉽게 기능을 확장하거나 다양한 기기에 통합할 수 있는 유연한 개발 환경을 지원합니다.

□ **Test 환경 정보(최적의 테스트 환경 구현을 위해 상세하게 작성 바람)**

※ 테스트에 반드시 필요한 사항 등 함께 기재

○ MacOS, Docker, PC Web(chrome), Node.js(v18.0 이상), npm(v9.0 이상), Git(v2.23 이상), Homebrew

○ 공통 환경 구성 (기존에 설치되어 있다면 생략 가능)

- macOS용 패키지 관리자 설치(터미널 환경)

```
/bin/bash -c "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

- Git 설치

```
brew install git
```

- Node.js & npm 설치

```
brew install node
```

- cask 설치

```
brew install cask
```

- Docker 설치

```
brew install -cask docker
```

- 구글 크롬(Chrome) 설치

```
brew install --cask google-chrome
```

○ 업데이트 파일 생성

- 아래의 명령어로 루트 디렉토리에 test_update_file.bin 파일 생성

```
cd
cat <<EOF > test_update_file.bin
Hello word
EOF
```

○ 프로젝트 실행 방법

- 프로젝트 실행 순서
  1. Blockchain Network
  2. Manufacturer Backend
  3. Device
  4. Manufacturer Frontend
  5. Device Frontend

- Blockchain Network 실행 방법
  1. 아래의 명령어로 루트 디렉토리로 이동 후, 레포지토리 clone

```
cd
git clone https://github.com/HSU-Blocker/Blocker_Blockchain_Network.git
```

  2. 아래의 명령어로 blockchain server 실행

```
cd Blocker_Blockchain_Network
cd blockchain-server
docker-compose up -d
```

3. 아래의 명령어로 registry service 배포

```
cd ../registry-service
docker-compose up --build -d

레지스트리 서비스 컨테이너 종료까지 대기
종료 확인 후 4번으로 이동
```

4. 아래의 명령어로 update service 배포

```
cd ../update_service
docker-compose up --build -d
```

- Manufacturer Backend 실행 방법

1. 아래의 명령어로 루트 디렉토리로 이동 후, 레포지토리 clone

```
cd
git clone https://github.com/HSU-Blocker/Blocker_Manufacturer_Backend.git
```

2. 아래의 명령어로 디렉토리 이동

```
cd Blocker_Manufacturer_Backend
```

3. **Blocker_Manufacturer_Backend** 프로젝트 루트 디렉토리에 .env 파일 생성 후 아래의 내용 입력 후 저장

```
BLOCKCHAIN_PROVIDER=http://host.docker.internal:8545
# 아래는 가나슈(Ganache) 기본 첫 번째 계정 (체크섬 주소로 수정)
BLOCKCHAIN_ACCOUNT=0xf39Fd6e51aad88F6F4ce6aB8827279cffFb92266
BLOCKCHAIN_PRIVATE_KEY=0xac0974bec39a17e36ba4a6b4d238ff944bacb478cbed5efcae
784d7bf4f2ff80
IPFS_API_URL=/dns/ipfs/tcp/5001/http
```

4. 아래의 명령어로 도커 이미지 빌드 후 실행

```
docker-compose up --build -d
```

- Device 실행 방법

1. 아래의 명령어로 루트 디렉토리로 이동 후, 레포지토리 clone

```
cd
git clone https://github.com/HSU-Blocker/Blocker_Device.git
```

2. 아래의 명령어로 디렉토리 이동

```
cd Blocker_Device
```

3. Blocker_Device 프로젝트 루트 디렉토리에 .env 파일 생성 후 아래의 내용 입력 후 저장

```
# 환경 변수 설정
FLASK_ENV=development

# 블록체인 설정
WEB3_PROVIDER=http:/host.docker.internal:8545
WEB3_WS_PROVIDER=ws://host.docker.internal:8545

# Ganache에서 제공하는 기본 개발 계정 중 하나를 사용
PRIVATE_KEY=0xac0974bec39a17e36ba4a6b4d238ff944bacb478cbed5efcae784d7bf4f2ff80
ACCOUNT_ADDRESS=0x70997970c51812dc3a010c7d01b50e0d17dc79c8
CONTRACT_ADDRESS=0x9fE46736679d2D9a65F0992F2272dE9f3c7fa6e0
CONTRACT_ABI_PATH=/blockchain/contract_address.json

# IPFS 설정
IPFS_API=/dns/host.docker.internal/tcp/5001/http # ipfshttpclient 연결용
IPFS_GATEWAY=http://host.docker.internal:8080    # 게이트웨이 다운로드용

# 포트 설정
DEVICE_API_PORT=5050
MANUFACTURER_API_PORT=5002

# IoT 디바이스 설정 예시
OWNER_ADDRESS=0x70997970C51812dc3A010C7d01b50e0d17dc79C8
OWNER_PRIVATE_KEY=0x59c6995e998f97a5a0044966f0945389dc9e86dae88c7a8412f4603b6b78690d
PUBLIC_KEY=0x038318535b54105d4a7aae60c08fc45f9687181b4fdfc625bd1a753fa7397fed75

# 제조사 URL
MANUFACTURER_API_URL=http://host.docker.internal:5002
```

4. 아래의 명령어로 도커 이미지 빌드 후 실행

```
docker-compose up --build -d
```

- Manufacturer Frontend 실행 방법

1. 아래의 명령어로 루트 디렉토리로 이동 후, 레포지토리 clone

```
cd
Git clone https://github.com/HSU-Blocker/Blocker_Manufacturer_Frontend.git
```

2. 아래 명령어로 디렉토리 이동

```
cd Blocker_Manufacturer_Frontend
```

3. Blocker_Manufacturer_Frontend 프로젝트 루트 디렉토리에 .env 파일 생성 후 아래의 내용 입력 후 저장

```
REACT_APP_API_BASE_URL=http://127.0.0.1:5002
```

4. 아래 명령어로 의존성 설치

```
npm install
```

5. 아래 명령어로 프론트엔드 서버 시작

```
npm start
```

6. PC(Web, Chrome) 환경에서 [http://localhost:3000] 링크로 제조사 프론트엔드 접속

- Device Frontend 실행 방법

1. 아래의 명령어로 루트 디렉토리로 이동 후, 레포지토리 clone

```
cd
git clone https://github.com/HSU-Blocker/Blocker_Device_Frontend.git
```

2. 아래의 명령어로 디렉토리 이동

```
cd Blocker_Device_Frontend
```

3. Blocker_Device_Frontend 프로젝트 루트 디렉토리에 .env 파일 생성 후 아래의 내용 입력 후 저장

```
VITE_API_URL=http://127.0.0.1:5050
```

4. 아래의 명령어로 의존성 설치

```
npm install
```

5. 아래의 명령어로 디바이스 프론트엔드 서버 시작

```
npm run dev
```

6. **PC(Web, Chrome)** 환경에서 **[http://localhost:5173]** 링크로 디바이스 프론트엔드 접속

나. 기능 리스트

대분류	중분류	소분류
제조사 기능 리스트		
HOME	Software Update Start! 아래 정육각형 버튼 : 클릭 시 화면 이동	-
	Update Monitoring : 클릭 시 화면 이동	-
	About us : 클릭 시 화면 이동	-
	Deploy 3D Animation	-

	: 클릭 시 화면 이동	
Software Update Start	Preparing Update File	upload 버튼 : 업데이트 파일 업로드
		← (왼쪽 화살표) 버튼 : 이전 화면으로 이동
	Update Settings	Update Version : 버전 입력
		Update Description : 업데이트 설정 입력
		Price : 이더리움 가격 입력
		Update Settings 아래 정육각형 버튼 : 다음 화면으로 이동
		← (왼쪽 화살표) 버튼 : 이전 화면으로 이동
	Access Control Policy Settings	Model Name : 모델명 설정
		Serial Number : 시리얼 넘버 설정
		Manufacture Date : 제조일자 설정
		Option Type : 옵션 선택 설정
		+ 버튼 : 데이터 추가
		Access Control Policy Settings 아래 정육각형 버튼 : 다음 화면으로 이동
		← (왼쪽 화살표) 버튼 : 이전 화면으로 이동
	Deploy Summary	Update File Info
		Update Details
		Policy Conditions

		Deploy 버튼 : 배포 시작
		자세한 설명 건너뛰기 체크박스
		← (왼쪽 화살표) 버튼 : 이전 화면으로 이동
Update Monitoring	업데이트 명	-
	IPFS 해시	-
	설명	-
	가격	-
	버전	-
	업데이트 삭제	-
Deploy 3D Animation	카메라 초기위치 이동 버튼	-
	업데이트 파일 업로드 버튼	-
	CP-ABE 암호문 업로드 버튼	-
	나가기 버튼	-
디바이스 기능 리스트		
Home	Connected / Disconnected : 블록체인 연결 여부	-
	Available Updates	업데이트 명 : 업데이트 명 출력
		업데이트 버전 : 업데이트 버전 출력
		가격 : 이더리움 가격 출력
		설치 버튼 : 업데이트 설치 실행
	Device Information	Device ID : 디바이스 ID 출력
		Model

		: 디바이스 모델명 출력
		Serial Number : 디바이스 시리얼 번호 출력
		Software Version : 소프트웨어 버전 출력
		Software Status : 소프트웨어 상태 출력
		Final Update : 최신 업데이트 일자 출력
	알림	New Update : 새 업데이트 알림
		Purchase / Installing... : 업데이트 진행 알림
		Complete! : 업데이트 완료 알림
History	업데이트 : 업데이트 명 출력	-
	버전 : 업데이트 버전 출력	-
	설명 : 업데이트 설명 출력	-
	상태 : 구매 완료/설치 완료/환불 완료/미설치 여부 출력	-
	가격 : 이더리움 가격 출력	-
	날짜 : 업데이트 날짜 출력	-
Help	왼쪽 하단 버튼 : 다음 화면으로 이동	-

다. 기능 시험 항목

대분류	중분류	소분류	시험항목	Result
제조사 기능 리스트				
HOME	Software Update Start! 문구 아래 정육각형 버튼 : 클릭 시 화면 이동	-	Preparing Update File 화면으로 이동	
	Update Monitoring : 클릭 시 화면 이동	-	Update Monitoring 화면으로 이동	
	About us : 클릭 시 화면 이동	-	About us 화면으로 이동	
	Deploy 3D Animation : 클릭 시 화면 이동	-	Deploy 3D Animation 화면으로 이동	
Software Update Start	Preparing Update File	upload 버튼 : 업데이트 파일 업로드	배포할 파일 업로드 [Comment] Test 환경정보 업데이트 파일 생성 항목에서 생성한 **test_update_file.bin** 파일 사용	

	Update Settings	Update Version : 버전 입력	[Comment] X.X.X 형식 사용	
		Update Description : 업데이트 설정 입력	[Comment] 필수 입력	
		Price : 이더리움 가격 입력	[Comment] 가스비 절약을 위해 0.1 사용	
		Update Settings 아래 정육각형 버튼 : 다음 화면으로 이동	Access Control Policy Settings 화면으로 이동	
	Access Control Policy Settings	Model Name : 모델명 설정	[Pre-condition] VS500 사용	
		Serial Number : 시리얼 넘버 설정	[Pre-condition] KMHEM42APXA75**** 사용	
		Manufacture Date : 제조일자 설정	[Pre-condition] 2015 사용	
		Option Type : 옵션 선택 설정	[Pre-condition] EXCLUSIVE PRESTIGE 두 개 모두 입력	
		+ 버튼 : 데이터 추가	입력 필드 생성됨	P
		Access Control Policy Settings 아래 정육각형 버튼 : 다음 화면으로 이동	Deploy Summary 화면으로 이동	P
	Deploy Summary	Update File Info	Preparing Update File에서 입력한 내용 출력	
		Update Details	Update Settings에서 입력한 내용 출력	

		Policy Conditions	Access Control Policy Settings에서 입력한 내용 출력	
		Deploy 버튼	클릭 시 배포 완료 후 시뮬레이션 화면으로 이동	
		자세한 설명 건너뛰기 체크박스	시뮬레이션 화면 건너뛰고 바로 배포 실행	
Update Monitoring	업데이트 명	-	업데이트 명 검색됨	
	IPFS 해시	-	IPFS의 CID 검색됨	
	설명	-	설명 검색됨	
	가격	-	가격 검색됨	
	버전	-	버전 검색됨	
	업데이트 삭제	-	리스트에서 해당 목록 삭제됨	
Deploy 3D Animation	카메라 초기위치 이동 버튼	-	초기 위치로 이동	
	업데이트 파일 업로드 버튼	-	파일 업로드 시뮬레이션 실행	
	CP-ABE 암호문 업로드 버튼	-	암호문 업로드 시뮬레이션 실행	
	나가기 버튼	-	메인 화면으로 이동	
디바이스 기능 리스트				
Home	Connected / Disconnected : 블록체인 연결 여부	-	블록체인 연결 상태에 따라 'Connected' 또는 'Disconnected' 출력	
	Available Updates	업데이트 명	업데이트 명 출력	
		업데이트 버전	업데이트 버전 출력	
		가격	이더리움 가격 출력	
		설치 버튼 : 업데이트 설치 실행	1. 버튼 클릭 시, 구매 완료&설치 완료 알림이 뜨고 Available	

			Updates에서 해당 업데이트 내역이 사라짐 2. client/updates 폴더에 설치 완료된 파일이 생성됨	
	Device Information	Device ID	디바이스 ID 출력	
		Model	디바이스 모델명 출력	
		Serial Number	디바이스 시리얼 번호 출력	
		Software Version	소프트웨어 버전 출력	
		Software Status	소프트웨어 상태 출력	
		Final Update	최신 업데이트 일자 출력	
	알림	New Update : 새 업데이트 알림	제조사에서 새로운 업데이트 배포 시, 새로운 업데이트 알림이 표시됨	
		Purchase / Installing... : 업데이트 진행 알림	구매/설치 진행 과정에 대한 알림(구매 중, 설치 중)이 표시됨 [Comment] 진행 과정이 끝나면 자동으로 삭제됨. 특히 Purchase 단계는 지속시간이 매우 짧으므로 사용자가 놓칠 수 있음	
		Complete! / Error : 업데이트 완료 알림	구매/설치에 대한 완료 또는 실패 알림이 표시됨	
History	업데이트	-	설치 시도한 업데이트 명 출력	
	버전	-	설치 시도한 업데이트 버전 출력	
	설명	-	설치 시도한 업데이트 설명 출력	

	상태 : 구매 완료+설치 완료/환불 완료+ 미설치 여부 출력	-	정상적으로 설치된 업데이트는 '구매 완료'와 '설치 완료' 표시되며, 설치 과정 중 오류가 있었던 업데이트는 '환불 완료'와 '미설치'로 표시됨.	
	가격 : 이더리움 가격 출력	-	-	
	날짜 : 업데이트 날짜 출력	-	-	
Help	왼쪽 하단 버튼	-	버튼 클릭 시, 다음 화면으로 넘어감 [Comment] '차량 뷰'에서 '새로운 업데이트를 시작하시겠습니까?' 과정에서는 화면 중앙의 '예' 버튼을 눌러야 다음 화면으로 넘어갈 수 있음.	