

동남권 지역산업밀착형 PKNU 실전문제연구단

2020. 04. 17. Fri

X-Corps 최종 점검 발표회

연구팀명	LACUC
연구주제구분	지정주제2 인공지능 및 블록체인 기반 스마트시티
연구과제명	IoT 센서를 이용한 블록체인 기반 식품 유통망 관리 시스템 연구
지도교수	IT융합응용공학과 신상욱 교수
참여기업	(주)스마트엠투엠



동남권 지역산업 밀착형
PKNU 실전문제연구단

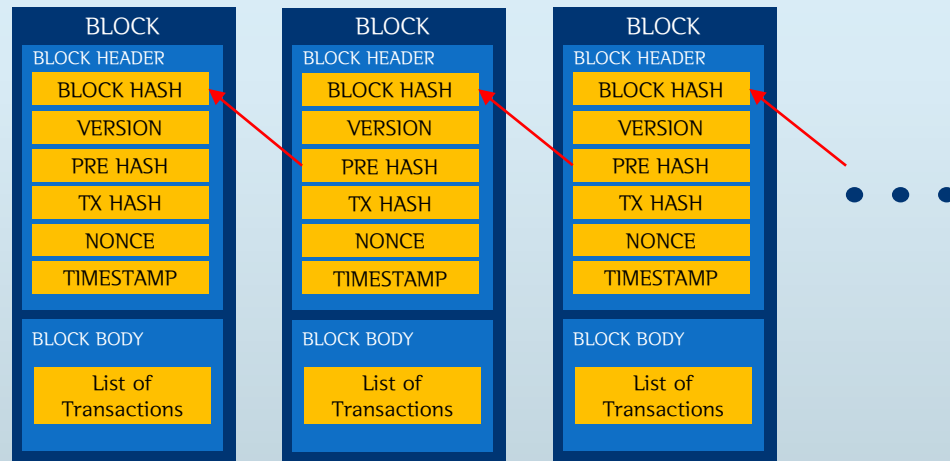
Contents

1. 연구개요
2. 연구목적 및 필요성
3. 연구내용
4. 연구과제 활용방안 및 기대효과
5. 최종결론 및 성과물의 내용
6. 건의사항

연구개요

◆블록체인

- 분산 기록 데이터베이스
- 트랜잭션을 모아 블록으로 만들고 체인처럼 연결
- 블록체인의 기능
 - 투명성 : 생성된 모든 트랜잭션과 블록들은 투명하게 공개
 - 무결성 : 네트워크의 모든 노드들이 동일한 데이터를 보유하고 있기 때문에 위변조 불가
 - 추적성 : 블록체인은 비가역적(삭제와 변경 불가, 오직 추가만 가능)



- 기밀 데이터, 프라이버시는 어떻게?

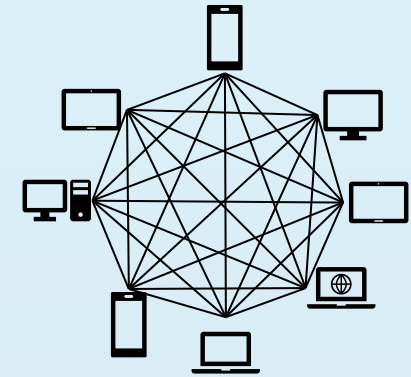
연구개요

◆ 블록체인

• 블록체인의 종류

➤ Public 블록체인

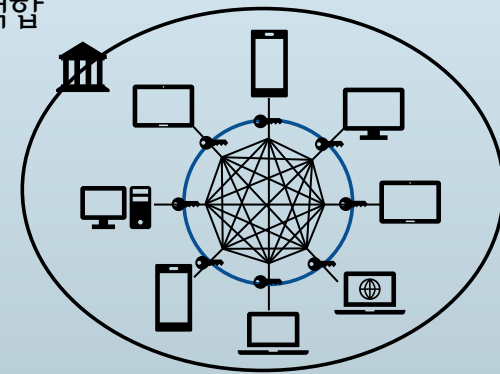
- ❖ 투명성과 공개성 보장 → **프라이버시 보장 x**
- ❖ TPS(Transaction per second)가 낮음
- ❖ TTP(Trusted Third Party:신뢰기관)이 없음
- ❖ 예) 비트코인(Bitcoin), 이더리움(Ethereum)



Public Blockchain

➤ Private/Consortium 블록체인

- ❖ **특정 노드**들에게만 접근 권한이 부여되는 블록체인
- ❖ 멤버십 서비스를 두고 신원이 보장된 노드들만 블록체인 네트워크에 접근 허용
- ❖ 트랜잭션과 블록의 내용이 공개되어서는 안되는 응용분야에 적합
- ❖ TPS 높음
- ❖ 단점 : 참여자의 권한 부여를 위한 어쩔 수 없는 TTP 존재
- ❖ 예) Hyperledger, Nexledger



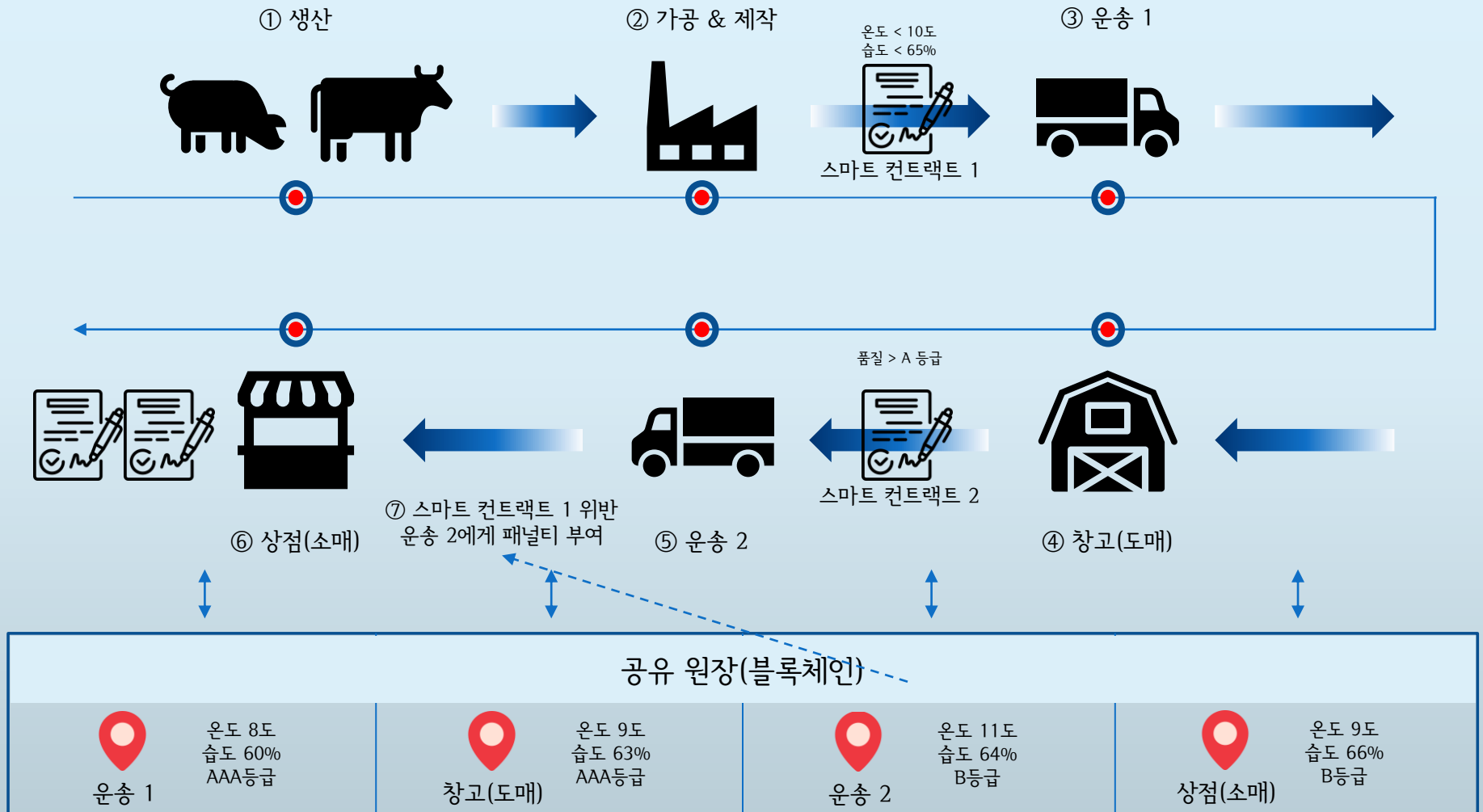
Private Blockchain

◆ Hyperledger Fabric

- Linux Foundation에서 진행하는 비즈니스 모델 블록체인 오픈 소스 프로젝트
- 허가형 프라이빗 블록체인(Permissioned Private Blockchain)
 - MSP(Membership Service Provider)라는 인증 관리 시스템에 등록된 사용자만이 블록체인에 참여 가능
- 비즈니스 목적에 알맞은 형태로 블록체인 플랫폼 구축 가능
- 하이퍼레저 패브릭 특징
 - 프라이버시와 기밀성
 - ❖ 같은 채널에 소속된 조직끼리만 트랜잭션 확인 가능
 - 작업 구간별 병렬 처리
 - ❖ 작업을 분리하여 처리함으로써 노드의 부하 감소, 시스템 성능 향상
 - 체인코드
 - ❖ 일반적인 블록체인에서의 스마트 컨트랙트
 - 모듈화된 디자인
 - ❖ 참여자들이 원하는 형태로 선택해서 블록체인을 운영할 수 있도록 모듈화된 디자인 지원
 - ❖ 개발의 유연성 제공

연구개요

◆그래서 우리는



연구개요

◆그래서 우리는



연구목적 및 필요성

◆ 2017년 8월 국내 살충제 달걀 파동

- 1,239곳 농장 중 49곳의 농장에서 오염된 달걀 발견
- 이후 정부의 노력 : 달걀 GP 센터 유통 의무화, 이력 추적제, 달걀의 산란일자 표기 의무화, 난각 코드 도입
 - 난각 코드의 형식이 통일되어 있지 않음
 - 생산 농가의 추적과 유통 경로 파악 어려움
 - 달걀 GP 센터 유통 의무화는 3년 이상의 시간이 소요

◆ 미국의 시금치, 로메인 상추 유통 사고

- 진상 규명에 2주 이상 시간 소요

◆ 중국의 식품 안전성 문제 대두

- 중국에서 돼지에게 약품이나 중금속이 섞인 사료를 먹여 단기간에 출하

◆ 까다로운 참치 유통 과정

- 생산지에서부터 원산지 표기가 엄격히 관리되지 않음 → 다른 생선들이 고급 참치로 둔갑해 남품

◆ 과거의 식료품 공급 과정

- 전혀 디지털화되어 있지 않고 대부분 종이 문서로 적혀 있는 경우가 많음
- 디지털화되어 있다 하더라도 다양한 시스템에서 상호작용 불가능
- 상품의 종류와 수가 무수히 많음 → **상품 관리의 복잡성, 유통 과정의 불투명성**

연구목적 및 필요성

◆ 블록체인을 적용한다면?

- 블록체인 기술의 탈중앙화, 보안성, 투명성이라는 특징이 유통과정에 적용되면 **상품의 신뢰와 소비자의 안전을 보장**할 수 있고 **비용 절감과 효율성 향상** 등의 긍정적인 효과를 얻을 수 있음
- 유통과정에서 발생하는 데이터들을 블록체인에 기록해두면 **제품의 원산지, 배송 과정, 보관 상태를 비롯한 정보들을 추적**할 수 있음
→ **제품의 안전 신뢰도 향상**
- 블록체인 기술이 진품 여부 확인, 유통 이력 추적과 상호 모니터링에 활용되면 **식품을 안전하게 유통**하는데 도움
- 중개 수수료를 지불해야하는 거대 플랫폼 없이도 거래가 가능
- 블록체인 기록을 통해 이력을 추적하면
 - 현재의 소유권 상태와 거래 내역을 실시간 확인 가능
 - 관리 비용이나 검증 수수료를 내지 않고도 사기 방지 가능
 - 허위 매물 등록이나 거래가 원천적으로 불가능
 - 인증 과정이 간소화되고, 소유권 이전 비용의 절감도 가능

◆ 해결해야 할 문제점

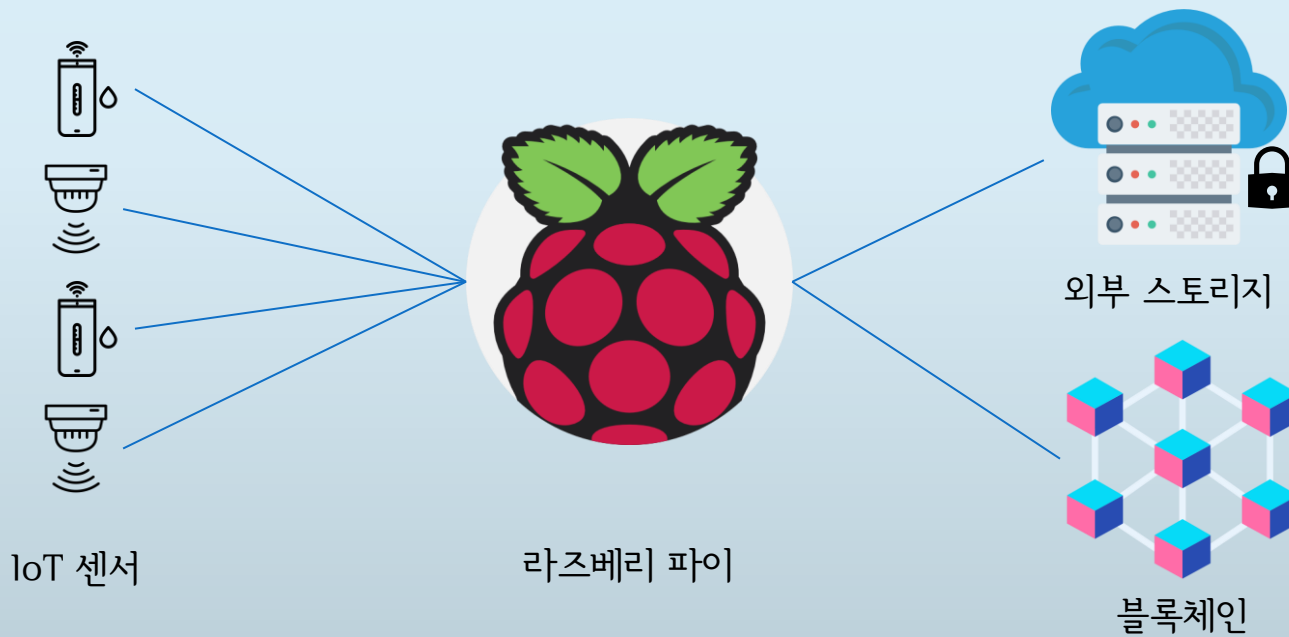
- **‘최초 1마일’** 문제 : 유통 과정에는 블록체인에 기록이 제대로 남더라도 블록체인과 사람이 접촉하는 지점에서는 허점이 생길 수 있음
- **IoT 기술과의 연계를 통해 해결 가능**

연구내용

◆ IoT 센서 데이터 처리

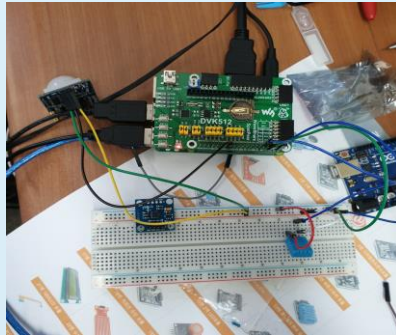
- 가정

- 생산지, 가공지, 소비지에 각종 IoT 센서 설치
- 운송 시 컨테이너 및 차량에 IoT 센서 설치
- IoT 센서를 자동으로 읽고 전송 가능한 네트워크 장비 구축

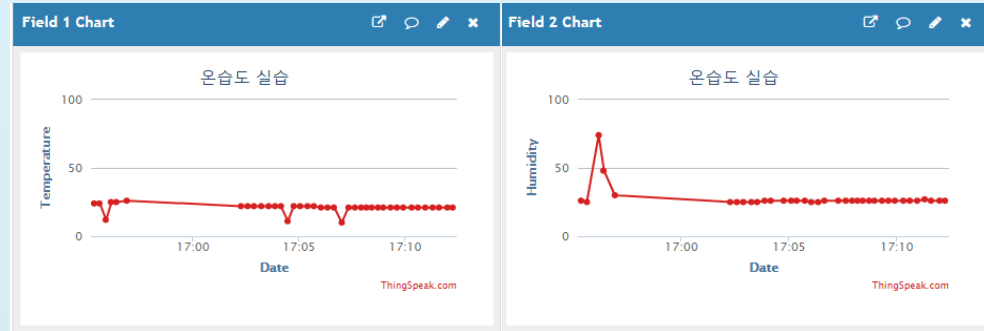


연구내용

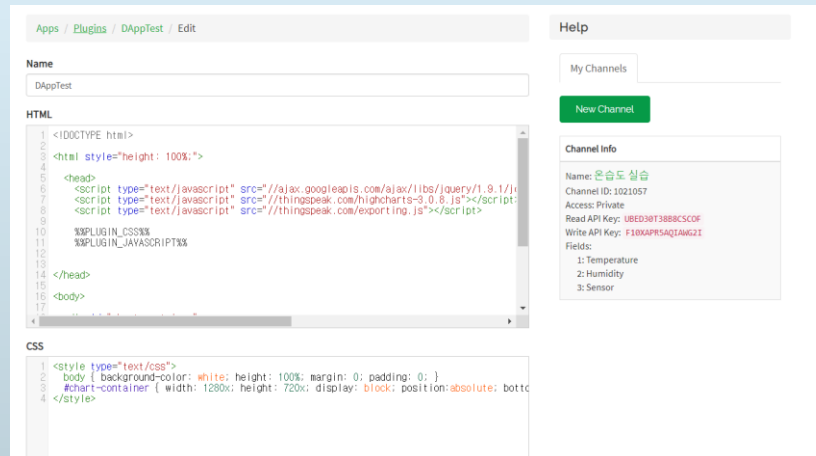
◆ IoT 센서 데이터 처리



1. IoT 센서와 라즈베리파이



2. IoT 센서로부터 읽어온 온도와 습도 데이터



3. IoT 데이터를 블록체인과 외부 스토리지에 저장

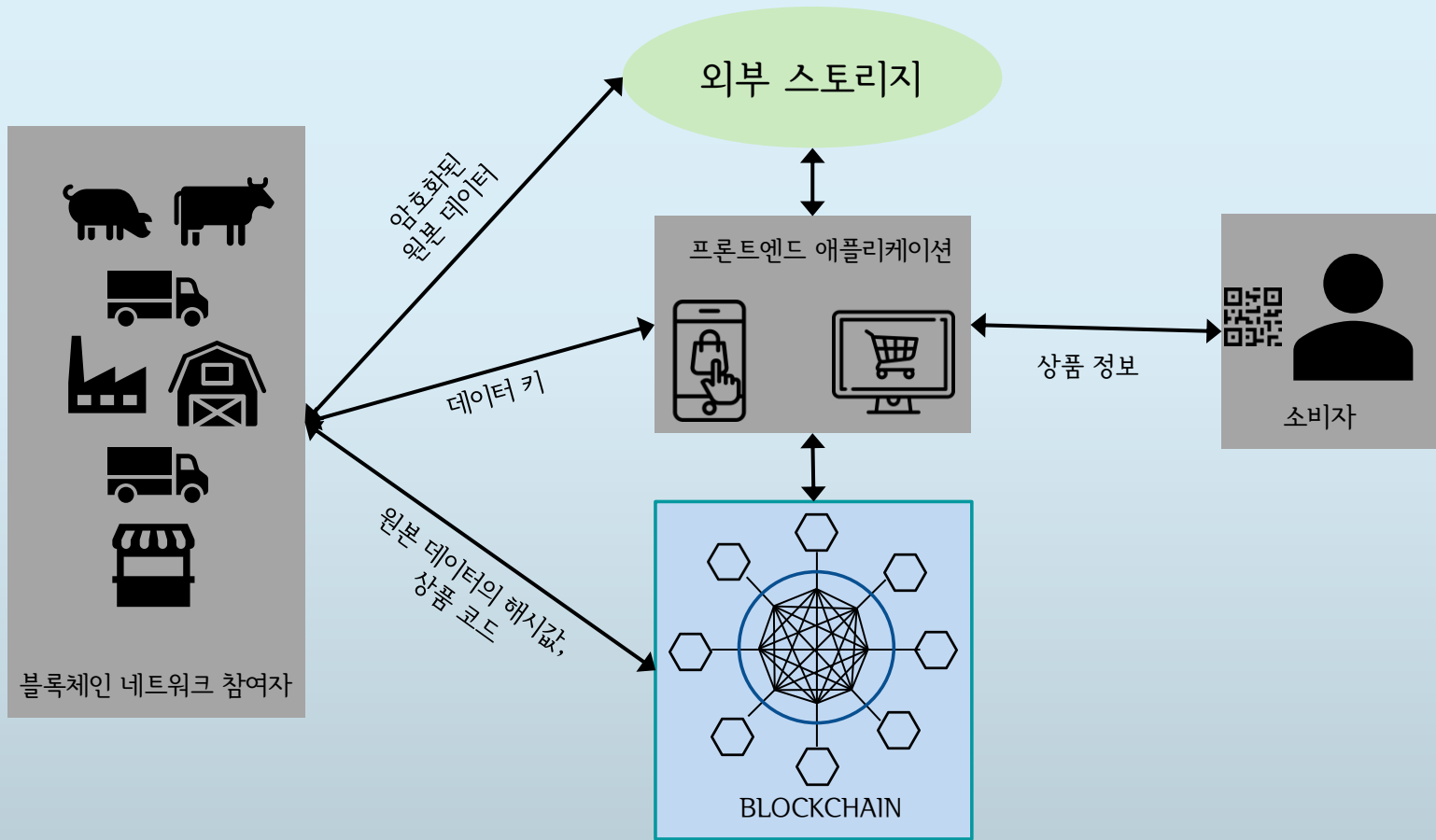
◆ IoT 센서 데이터 처리

- 생산지
 - 생산날짜, 생산지, 관리자 정보(이름, 주소, 번호), 판매가격, 수량, 상품정보(도축기간, 등급, 상품코드)
- 가공지
 - 구입날짜, 판매날짜, 가공에 걸린 시간, 구입 가격, 판매 가격, 관리자 정보, 구입당시 상품정보, 판매당시 상품정보
- 운송
 - 출발시 상품정보, 도착후 상품정보, 도착지, 운송시간(출발, 도착 시간), 운송 중 온도 및 습도 등 센서 데이터, 운전자 정보
- 판매지
 - 구매정보, 담당자 정보, 판매지 정보, 상품정보
 - QR코드 생성

담당자 정보를 소비자가 알아야 할까?

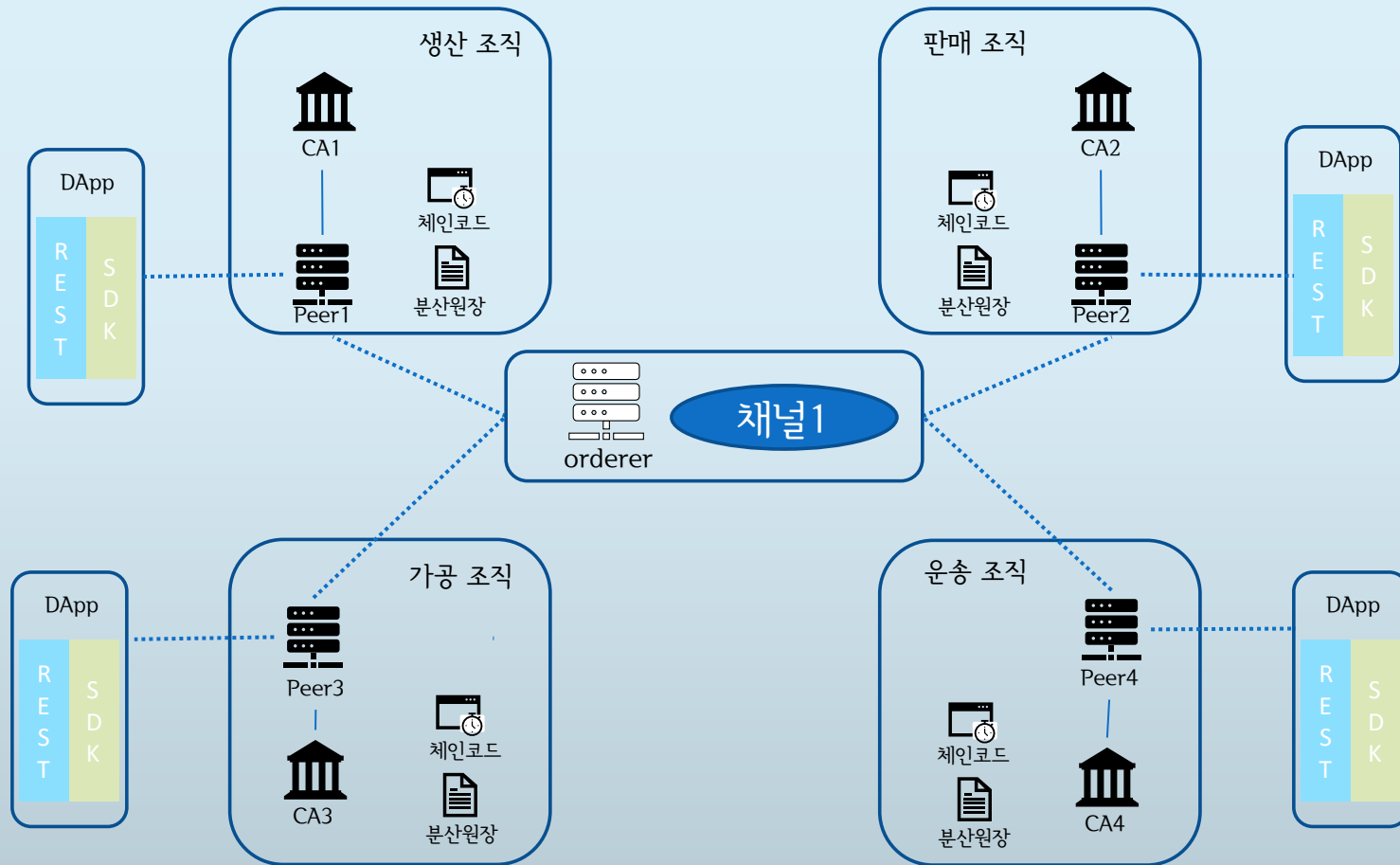
연구내용

- ◆ 블록체인 네트워크 구축
 - 전체 네트워크 구조



연구내용

- ◆블록체인 네트워크 구축
 - 블록체인 네트워크 구조



◆ 프론트엔드 애플리케이션 구축

• 웹 페이지 구현

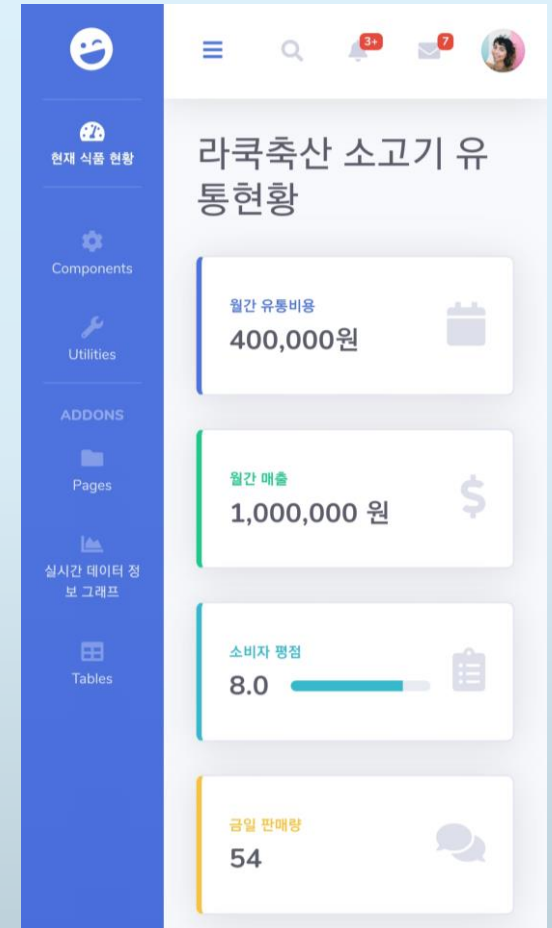
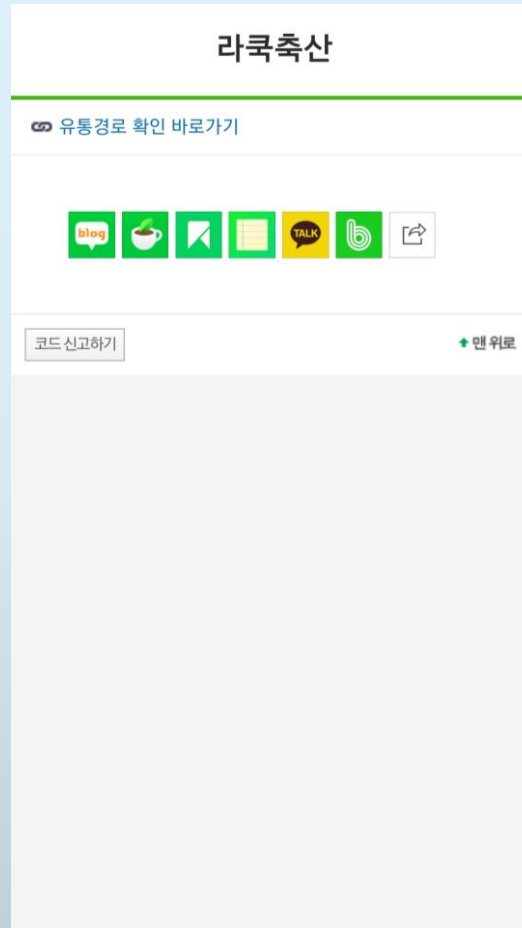
- 소비자가 상점에서 상품의 QR 코드를 통해 상품 정보를 확인할 수 있는 웹페이지
- 소비자는 블록체인 네트워크 참여자가 아니기 때문에 모든 블록체인 데이터를 확인할 수 없음
- 하이퍼레저 패브릭 체인코드에 의해 하드코딩된 정보들만 소비자에게 제공
- 따라서 유통에 문제가 생겼을 때 사용되는 담당자의 개인정보는 보호됨
- Vscode 1.44
- 센서 데이터 저장 서버 : ThingSpeak 클라우드 서버
- 웹페이지 서버 : netify
- 개발 언어 : HTML 5.0, CSS 2.1, Javascript 1.5

연구과제 활용방안 및 기대효과

- ◆ 효율적이고 안전한 식품 유통 가능
- ◆ 소비자들도 유통 과정을 확인할 수 있으므로 안심하고 제품 구매 가능
- ◆ 유통 시 발생하는 제품 손상이나 사고에 대한 책임 추적 가능
- ◆ IoT 센싱 데이터를 이용한 자동화된 트랜잭션 처리로 인한 무결성 및 신뢰성 향상
- ◆ 돼지고기 시나리오 뿐 아닌 다른 식품에도 적용 가능
- ◆ 블록체인 특구로 지정된 부산광역시 지역 산업 수용에 부응
 - 수산물 유통
 - 항만 물류 시스템
 - 관광 산업

최종결론 및 성과물의 내용

◆개발 웹페이지



최종결론 및 성과물의 내용

◆개발 웹페이지



최종결론 및 성과물의 내용

◆하이퍼레저 패브릭 네트워크

```
bbaing@ubuntu:~/go/src/fabric-samples/first-network$ ./byfn.sh generate
Generating certs and genesis block for channel 'mychannel' with CLI timeout of '10' seconds and CLI delay of '3' seconds
Continue? [Y/n] Y
proceeding ...
/home/bbaing/go/src/fabric-samples/first-network/./bin/cryptogen

#####
##### Generate certificates using cryptogen tool #####
#####
+ cryptogen generate --config=./crypto-config.yaml
org1.example.com
+ res=0
+ set +x

Generate COO files for Org1 and Org2
/home/bbaing/go/src/fabric-samples/first-network/./bin/configtxgen
#####
##### Generating Orderer Genesis block #####
#####
CONSENSUS_TYPE=solo
[ 'solo == solo' ]
+ configtxgen -profile TwoOrgsOrdererGenesis -channelID byfn-sys-channel -outputBlock ./channel-artifacts/genesis.block
2020-03-10 21:39:21.790 PDT [common.tools.configtxgen] main -> INFO 001 Loading configuration
2020-03-10 21:39:21.836 PDT [common.tools.configtxgen.localconfig] completeInitialization -> INFO 002 orderer type: solo
2020-03-10 21:39:21.836 PDT [common.tools.configtxgen.localconfig] Load -> INFO 003 Loaded configuration: /home/bbaing/go/src/fabric-samples/first-network/configtx.yaml
2020-03-10 21:39:21.882 PDT [common.tools.configtxgen.localconfig] completeInitialization -> INFO 004 orderer type: solo
2020-03-10 21:39:21.882 PDT [common.tools.configtxgen.localconfig] LoadTopLevel -> INFO 005 Loaded configuration: /home/bbaing/go/src/fabric-samples/first-network/configtx.yaml
2020-03-10 21:39:21.883 PDT [common.tools.configtxgen] doOutputBlock -> INFO 006 Generating genesis block
2020-03-10 21:39:21.884 PDT [common.tools.configtxgen] doOutputBlock -> INFO 007 Writing genesis block
+ res=0
+ set +x

#####
##### Generating channel configuration transaction 'channel.tx' #####
#####
+ configtxgen -profile TwoOrgsChannel -outputCreateChannelTx ./channel-artifacts/channel.tx -channelID mychannel
2020-03-10 21:39:21.906 PDT [common.tools.configtxgen] main -> INFO 001 Loading configuration
2020-03-10 21:39:21.957 PDT [common.tools.configtxgen.localconfig] Load -> INFO 002 Loaded configuration: /home/bbaing/go/src/fabric-samples/first-network/configtx.yaml
2020-03-10 21:39:22.003 PDT [common.tools.configtxgen.localconfig] completeInitialization -> INFO 003 orderer type: solo
2020-03-10 21:39:22.003 PDT [common.tools.configtxgen.localconfig] LoadTopLevel -> INFO 004 Loaded configuration: /home/bbaing/go/src/fabric-samples/first-network/configtx.yaml
2020-03-10 21:39:22.003 PDT [common.tools.configtxgen] doOutputChannelCreateTx -> INFO 005 Generating new channel configtx
2020-03-10 21:39:22.005 PDT [common.tools.configtxgen] doOutputChannelCreateTx -> INFO 006 Writing new channel tx
+ res=0
+ set +x

#####
##### Generating anchor peer update for Org1MSP #####
#####
+ configtxgen -profile TwoOrgsChannel -outputAnchorPeersUpdate ./channel-artifacts/Org1MSPanchors.tx -channelID mychannel -asOrg Org1MSP
2020-03-10 21:39:22.030 PDT [common.tools.configtxgen] main -> INFO 001 Loading configuration
2020-03-10 21:39:22.076 PDT [common.tools.configtxgen.localconfig] Load -> INFO 002 Loaded configuration: /home/bbaing/go/src/fabric-samples/first-network/configtx.yaml
2020-03-10 21:39:22.122 PDT [common.tools.configtxgen.localconfig] completeInitialization -> INFO 003 orderer type: solo
2020-03-10 21:39:22.122 PDT [common.tools.configtxgen.localconfig] LoadTopLevel -> INFO 004 Loaded configuration: /home/bbaing/go/src/fabric-samples/first-network/configtx.yaml
2020-03-10 21:39:22.122 PDT [common.tools.configtxgen] doOutputAnchorPeersUpdate -> INFO 005 Generating anchor peer update
2020-03-10 21:39:22.122 PDT [common.tools.configtxgen] doOutputAnchorPeersUpdate -> INFO 006 Writing anchor peer update
+ res=0
+ set +x

#####
##### Generating anchor peer update for Org2MSP #####
#####
+ configtxgen -profile TwoOrgsChannel -outputAnchorPeersUpdate ./channel-artifacts/Org2MSPanchors.tx -channelID mychannel -asOrg Org2MSP
2020-03-10 21:39:22.153 PDT [common.tools.configtxgen] main -> INFO 001 Loading configuration
2020-03-10 21:39:22.240 PDT [common.tools.configtxgen.localconfig] Load -> INFO 002 Loaded configuration: /home/bbaing/go/src/fabric-samples/first-network/configtx.yaml
2020-03-10 21:39:22.240 PDT [common.tools.configtxgen.localconfig] completeInitialization -> INFO 003 orderer type: solo
2020-03-10 21:39:22.240 PDT [common.tools.configtxgen.localconfig] LoadTopLevel -> INFO 004 Loaded configuration: /home/bbaing/go/src/fabric-samples/first-network/configtx.yaml
2020-03-10 21:39:22.240 PDT [common.tools.configtxgen] doOutputAnchorPeersUpdate -> INFO 005 Generating anchor peer update
2020-03-10 21:39:22.240 PDT [common.tools.configtxgen] doOutputAnchorPeersUpdate -> INFO 006 Writing anchor peer update
+ res=0
+ set +x
```

최종결론 및 성과물의 내용

◆하이퍼레저 패브릭 네트워크

```
hyperledger-fabric@capstone2:~/src$ docker ps
CONTAINER ID        IMAGE                               COMMAND                  CREATED       STATUS       PORTS
c5494a72edde        hyperledger/fabric-tools          "/bin/bash"             4 seconds ago Up 2 seconds 0.0.0.0:20051->20051/tcp, 0.0.0.0:20053->20053/tcp
5e80d263a7dc        hyperledger/fabric-peer:1.4.3    "peer node start"       7 seconds ago Up 5 seconds 0.0.0.0:12051->12051/tcp, 0.0.0.0:12053->12053/tcp
e15137448430        hyperledger/fabric-peer:1.4.3    "peer node start"       7 seconds ago Up 5 seconds 0.0.0.0:22051->22051/tcp, 0.0.0.0:22053->22053/tcp
8bcbd393d807        hyperledger/fabric-peer:1.4.3    "peer node start"       7 seconds ago Up 4 seconds 0.0.0.0:23051->23051/tcp, 0.0.0.0:23053->23053/tcp
b55c729ffb58        hyperledger/fabric-peer:1.4.3    "peer node start"       7 seconds ago Up 5 seconds 0.0.0.0:21051->21051/tcp, 0.0.0.0:21053->21053/tcp
4ad8cd036510        hyperledger/fabric-peer:1.4.3    "peer node start"       7 seconds ago Up 5 seconds 0.0.0.0:13051->13051/tcp, 0.0.0.0:13053->13053/tcp
1c8feb019b2        hyperledger/fabric-peer:1.4.3    "peer node start"       7 seconds ago Up 5 seconds 0.0.0.0:24051->24051/tcp, 0.0.0.0:24053->24053/tcp
4cee6d04942c        hyperledger/fabric-peer:1.4.3    "peer node start"       7 seconds ago Up 5 seconds 0.0.0.0:24051->24051/tcp, 0.0.0.0:24053->24053/tcp
```

```
hyperledger-fabric@capstone2:~/src$ docker ps
CONTAINER ID        IMAGE                               COMMAND                  CREATED       STATUS       PORTS
c5494a72edde        hyperledger/fabric-tools          "/bin/bash"             4 seconds ago Up 2 seconds 0.0.0.0:20051->20051/tcp, 0.0.0.0:20053->20053/tcp
5e80d263a7dc        hyperledger/fabric-peer:1.4.3    "peer node start"       7 seconds ago Up 5 seconds 0.0.0.0:12051->12051/tcp, 0.0.0.0:12053->12053/tcp
e15137448430        hyperledger/fabric-peer:1.4.3    "peer node start"       7 seconds ago Up 5 seconds 0.0.0.0:22051->22051/tcp, 0.0.0.0:22053->22053/tcp
8bcbd393d807        hyperledger/fabric-peer:1.4.3    "peer node start"       7 seconds ago Up 4 seconds 0.0.0.0:23051->23051/tcp, 0.0.0.0:23053->23053/tcp
b55c729ffb58        hyperledger/fabric-peer:1.4.3    "peer node start"       7 seconds ago Up 5 seconds 0.0.0.0:21051->21051/tcp, 0.0.0.0:21053->21053/tcp
4ad8cd036510        hyperledger/fabric-peer:1.4.3    "peer node start"       7 seconds ago Up 5 seconds 0.0.0.0:13051->13051/tcp, 0.0.0.0:13053->13053/tcp
1c8feb019b2        hyperledger/fabric-peer:1.4.3    "peer node start"       7 seconds ago Up 5 seconds 0.0.0.0:24051->24051/tcp, 0.0.0.0:24053->24053/tcp
4cee6d04942c        hyperledger/fabric-peer:1.4.3    "peer node start"       7 seconds ago Up 5 seconds 0.0.0.0:24051->24051/tcp, 0.0.0.0:24053->24053/tcp
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
c5494a72edde	hyperledger/fabric-tools	"/bin/bash"	4 seconds ago	Up 2 seconds	0.0.0.0:20051->20051/tcp, 0.0.0.0:20053->20053/tcp	Capstone_design_cli
5e80d263a7dc	hyperledger/fabric-peer:1.4.3	"peer node start"	7 seconds ago	Up 5 seconds	0.0.0.0:12051->12051/tcp, 0.0.0.0:12053->12053/tcp	peer0.Sales.chain.com
e15137448430	hyperledger/fabric-peer:1.4.3	"peer node start"	7 seconds ago	Up 5 seconds	0.0.0.0:22051->22051/tcp, 0.0.0.0:22053->22053/tcp	peer0.People.chain.com
8bcbd393d807	hyperledger/fabric-peer:1.4.3	"peer node start"	7 seconds ago	Up 4 seconds	0.0.0.0:23051->23051/tcp, 0.0.0.0:23053->23053/tcp	peer2.Sales.chain.com
b55c729ffb58	hyperledger/fabric-peer:1.4.3	"peer node start"	7 seconds ago	Up 5 seconds	0.0.0.0:21051->21051/tcp, 0.0.0.0:21053->21053/tcp	peer0.Ship.chain.com
4ad8cd036510	hyperledger/fabric-peer:1.4.3	"peer node start"	7 seconds ago	Up 5 seconds	0.0.0.0:13051->13051/tcp, 0.0.0.0:13053->13053/tcp	peer1.Sales.chain.com
1c8feb019b2	hyperledger/fabric-peer:1.4.3	"peer node start"	7 seconds ago	Up 5 seconds	0.0.0.0:24051->24051/tcp, 0.0.0.0:24053->24053/tcp	peer1.People.chain.com
4cee6d04942c	hyperledger/fabric-peer:1.4.3	"peer node start"	7 seconds ago	Up 5 seconds	0.0.0.0:24051->24051/tcp, 0.0.0.0:24053->24053/tcp	peer1.Ship.chain.com

최종결론 및 성과물의 내용

◆ 결론

- 현 식품 유통망은 비효율적이며 문제점이 많다.
 - 책임 추적의 문제
 - 식품에 대한 안전성 문제
 - 유통 과정에 대한 신뢰성 문제
- IoT 센서와 블록체인을 이용하여 이를 해결한다.
 - IoT 센서 데이터를 자동으로 입력해 “최초 1마일” 문제 해결
 - 블록체인을 이용해 위에서 언급한 문제 해결

◆ 향후 연구

- 돼지고기 시나리오를 넘어 더 많은 식품 적용
- 소비자 뿐만 아니라 네트워크 참여자들도 확인할 수 있는 애플리케이션 개발
- 더 강력한 프라이버시 보호를 위한 암호화 저장 기법 연구
- 가정으로 두었던 무선 네트워크 환경에서의 구현

Thank you for listening

Q&A

holine0622@pukyong.ac.kr



동남권 지역산업 밀착형
PKNU 실전문제연구단