목차

1 과제 배경 및 목표 2

1.1 배경 및 필요성 2

1.2 과제 목표 4

1.3 국내외 기술 및 시장 현황 5

2 구현 내용 8

2.3 시나리오 8

2.4 개발환경 / 기술스택 12

3. 세부 사항 13

3.1 블록체인 네트워크 13

3.2 EVM 14

3.3 스마트 컨트랙트 15

3.4 IPFS 16

3.5 현실적 제약사항 17

4 개발 일정 및 역할분담 17

4.1 개발일정 17

4.2 역할분담 18

1 과제 배경 및 목표

1.1 배경 및 필요성

소비자들은 구매하려는 수산물의 유통 과정과 원산지를 투명하게 확인하기 어려워 수산물 구매 시 원산지 표시에 의존할 수밖에 없다.

하지만 해양수산부 조사에 따르면 지난 4년간 원산지 미 표시 및 표시방법 위반 사례가 4936건, 거짓표시는 1700건으로 집계되어 그 신뢰성을 의심받고 있다.

"수산물 원산지 허위 표시 4년간 19% 증가...158억원 규모"

[그림 1] 원산지 허위 표시 규모

(단위 : 건)

위반현황				
순위	미표시 및 표시방법 위반		거짓표시	
	생산지 (비중 %)	건수	생산지 (비중 %)	건수
1	국내산 (56.2%)	2,772	중국 (39.8%)	401
2	중국 (19.9%)	983	일본 (15.8%)	159
3	러시아 (6.4%)	314	원양산 (7.2%)	73
4	일본 (5.6%)	274	러시아 (6.8%)	68
5	베트남 (1.8%)	88	국내산 (3.4%)	34

자료: 해양수산부, 어기구의원실 재구성

[그림 2] 원산지 표시 위반 현황

이 같은 원산지 표기를 위반한 허위매물 단속은 쉽지 않은 실정이다. 거래 증빙자료를 보관할 의무가 없기 때문이다. 한국해양수산개발원 현안보고서에 따르면 수산물은 농산물과 축산물에 비해 유통과정이 불투명해 거래 증빙 자료 발급 의무의 부과가 어려운 것으로 파악됐다.

최근 일본의 후쿠시마 원전 오염수 방류 조치에 대응하고자 내린 후쿠시마 산 수산물 수입 금지 조치를 우회하려 원산지를 위변조한 사례도 발생했던 만큼 원산지와 유통과정의 투명성은 식품 안전성과도 직결되는 사안이다.

[단독] 일본 대신 수입산...'국적 세탁' 수산물, 후쿠시마 사고 이후 최대

[그림 3] 식품 안전성 위협 사례

소비자의 선택권과 식품 안전성이 보장하기 위해 투명한 유통 과정을 확인할 플랫폼이 필요하다.

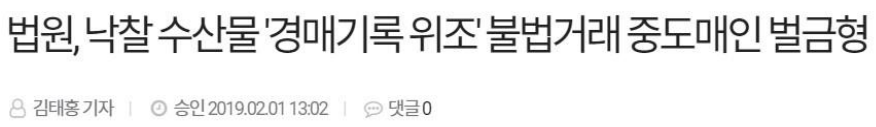
불투명한 요소는 유통 과정 뿐만이 아니다.

현재 수산물은 주로 생산자 -> 산지위판장 -> 소비지 도매시장 -> 도매상 -> 소매상을 거쳐 소비자로 유통된다.

위 과정 중 경매가 이루어지는데 산지 위판장에서 1차 상장 경매, 소비지 도매시장에 와서 2차 상장 경매를 거쳐야 한다.

기존 경매 내역들은 수기로 작성되기에 위변조가 용이하고 불투명한 거래가 이뤄질 위험이 크다.

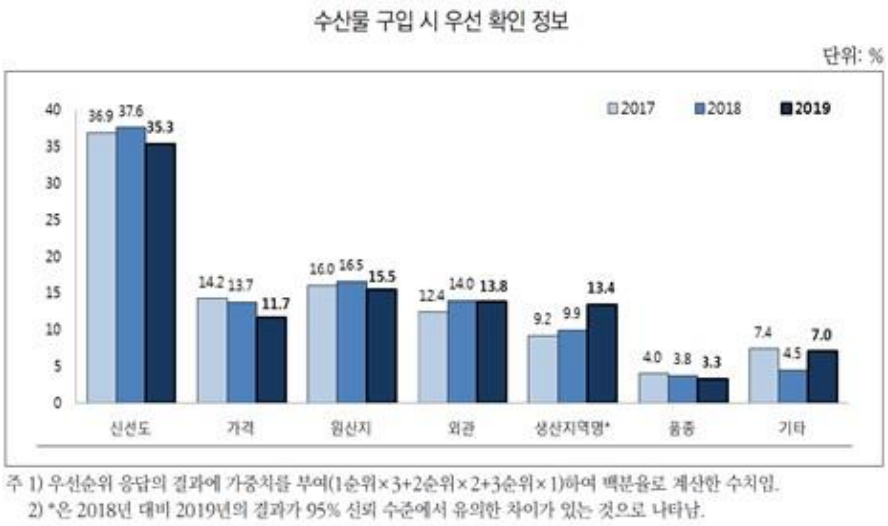
이를 해결하기 위해선 경매의 투명성을 높여 경매 기록의 위변조를 방지해야 한다.



[그림 4] 경매 위변조 사례

수산물 시장에서 유통과정과 경매내역의 투명성을 확보해 원산지와 가격 정보의 신뢰성을 높이는 것만큼 중요한 요소가 신선도 유지이다.

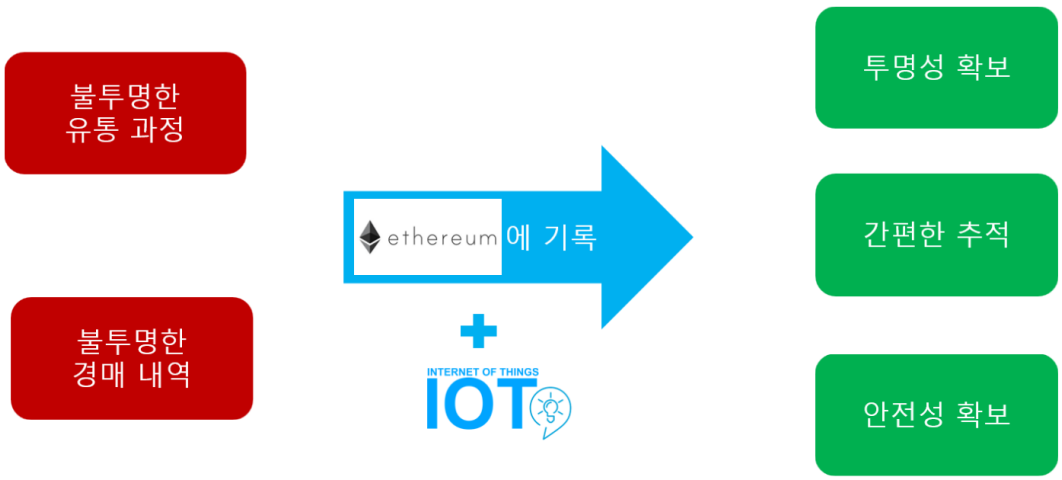
해양수산부의 조사에 따르면, 수산물을 구입할 때 소비자들이 중요하게 생각하는 요소는 1위 신선도(32.3%), 2위 원산지(16.2%), 3위 가격(15.2%)으로 나타났다.



[그림 5] 수산물 구입 시 우선 고려 요소

소비자가 수산물을 구입할 때 신선도를 가장 중요하게 생각하는 만큼 소비자 입장에서 수산물이 유통과정을 거칠 때 신선도를 확인하기 위한 환경 정보를 제공해야 할 필요성이 있다.

1.2 과제 목표



[그림 6] 요약도

본 졸업과제는 이더리움 기반 수산물 유통 플랫폼 개발을 목표로 한다.

플랫폼에서 발생한 거래 내역과 유통 과정을 원장에 기록하여 관리함으로써 정보의 투명성을 확보한다.

경매의 입찰과정은 우선적으로 내부 DB에 기록, 낙찰 이후엔 거래가 진행된 입찰 내역을 모두 블록체인 원장에 기록하는 방식이다. 경매 과정을 원장에 기록함으로써 경매 기록의 위변조와 같은 불법적인 거래의 추적을 용이하게 한다.

소비자가 유통과정 중 신선도 유지 여부를 확인할 수 있도록 IoT를 활용해 수집한 배송 환경(온/습도) 또한 원장에 기록한다.
또한, 거래가 이루어진 수산물의 거래 및 유통 과정을 손쉽게 확인 가능하도록 QR코드를 발급할 예정이다. 소비자가 수산물을 구매하기 전에 QR코드로 앞선 거래 및 유통과정 모두 확인 가능하다.

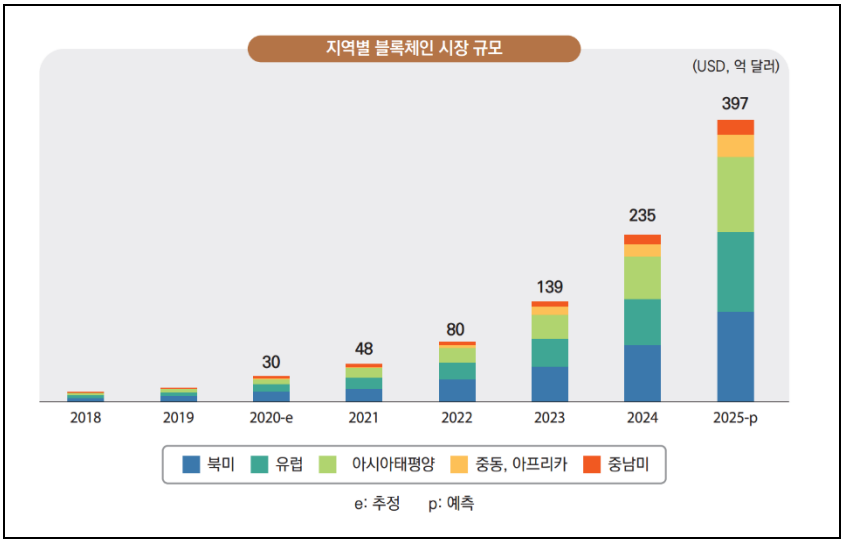
1.3 국내외 기술 및 시장 현황

A. 블록체인 시장현황

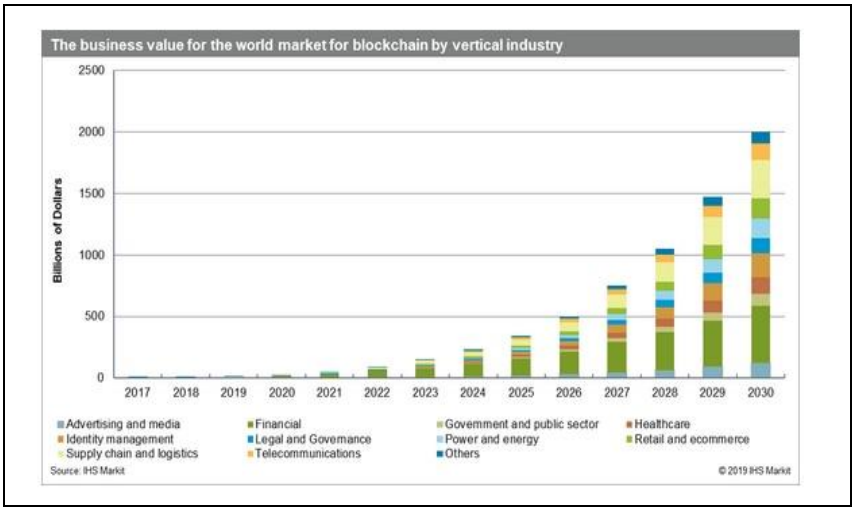
블록체인은 분산 컴퓨팅 원장관리 기술로 4차 산업 혁명의 주요 요소로 각광받는 기술 중 하나이다.

시장 또한 성장하고 있으며 글로벌 시장조사 기업인 마켓앤마켓은 세계 블록체인 시장 규모가 연평균성장률 67.3%로 성장하여 2020년 30억달러에서 2025년 397억달러에 이를 것으로 전망하였다.

투명성과 위변조 방지로 인한 신뢰성을 강점으로 금융, 헬스케어, 물류, 통신, 예술 등 사회 전반에 적용하려는 시도가 늘고 있다.



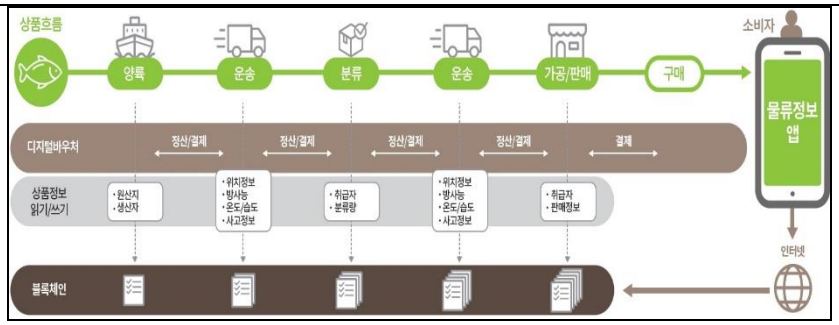
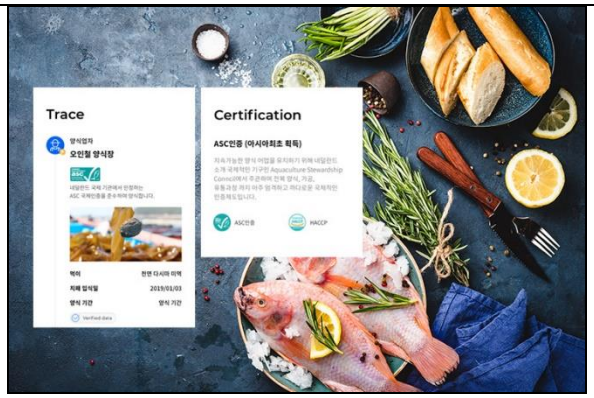
[그림 7] 세계 블록체인 시장 규모



[그림 8] 분야별 블록체인 시장 규모 및 전망

B. 국내외 기술동향

사례	기술내용
부산시 블록체인 기반 스마트 해양물류 플랫폼	

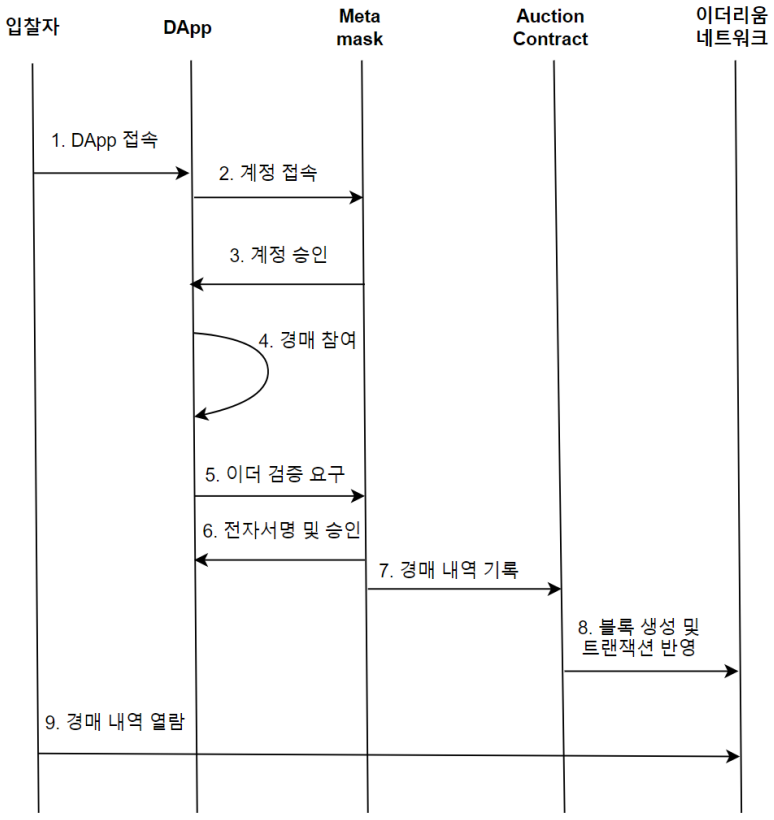
	 수산물 유통 과정 정보를 소비자에게 제공하는 플랫폼 물류 과정과 콜드체인을 블록체인에 기록하여 관리하지만 거래기능은 포함하지 않음 시범 사업이며 현재까진 상용화 X
샵블리	SHOP BLY 농업데이터 플랫폼 블록체인을 통한 이력추적 서비스 제공
삼성 SDS Cello Trust	 원자재의 수출입, 통관 등 운송 데이터를 원장에 기록 IoT 기기를 부착해 유통 상태 정보 증명 QR코드를 통해 소비자용 유통이력을 확인하는 모바일 화면 제공 자체 물류 서비스 플랫폼 Cello과 연계 수산물, 와인 등 적용

IBM Food Trust	IBM Food Trust™ 식품추적 블록체인 플랫폼 식품 거래내역을 원장에 기록 식품의 생산 및 유통과정을 2초안에 추적 클라우드 플랫폼으로 운영 월마트, 네슬레 등 거대 유통 및 식품기업에서 사용 Hyperledger fabric을 기반으로 개발

2 구현 내용

2.1 시나리오

A. 경매



위 시나리오에선 자체 개발한 스마트 컨트랙트를 사용하며 'Auction Contract'로 지칭한다.

MetaMask는 입찰자가 DApp 브라우저에서 MetaMask에 계정 접속을 하면 본인 인증을 통해 계정 승인을 한다.

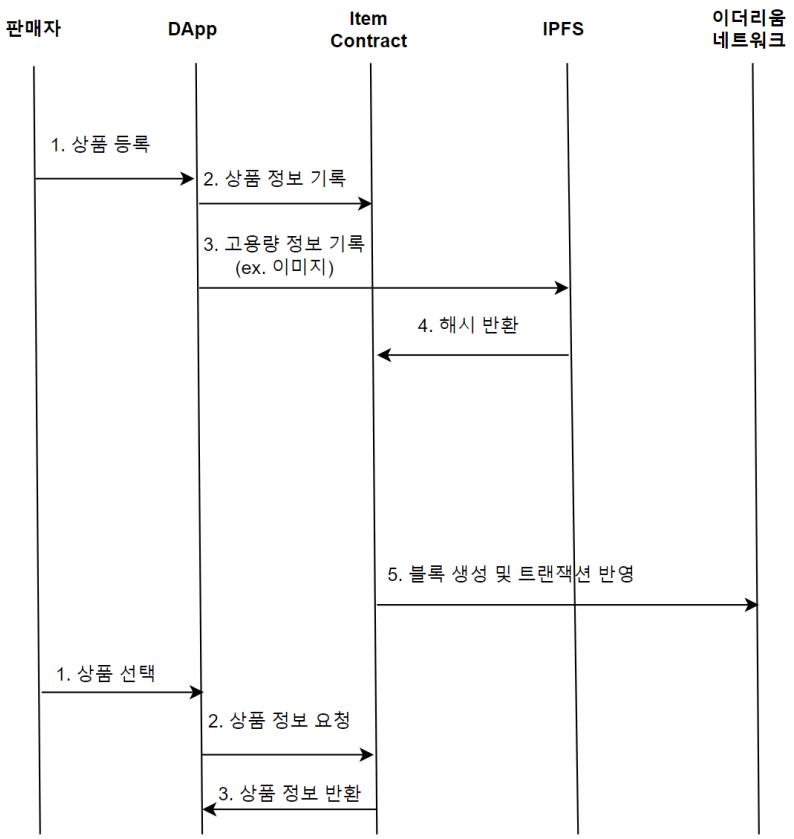
입찰자는 승인된 계정을 통해 DApp에서 경매에 참여한다.

경매에 참여할 경우 필요한 이더를 MetaMask를 통해 확인하고 검증을 요구한다.

MetaMask는 검증된 이더에 전자서명과 승인을 통해 구매 확정을 진행하게 되며 경매에 참여한 구매 정보 및 경매 내역을 Auction Contract 내 상태 변수에 저장하는 트랜잭션을 발생시킨다.

발생한 트랜잭션을 병합해 블록을 생성 후 이더리움 네트워크에 반영한다.

B. 상품 등록 및 조회



위 시나리오에선 자체 개발한 스마트 컨트랙트를 사용하며 'Item Contract'로 지칭한다.

- 상품 등록

  판매자가 DApp에 상품을 등록 시 입력한 상품의 정보를 Item Contract 내 상태

변수 배열에 저장하는 트랜잭션을 발생시킨다.

용량이 큰 정보는 실제 정보 대신 IPFS에 기록하고 반환받은 해시를 저장한다.

발생한 트랜잭션을 병합해 블록을 생성 후 이더리움 네트워크에 반영한다.

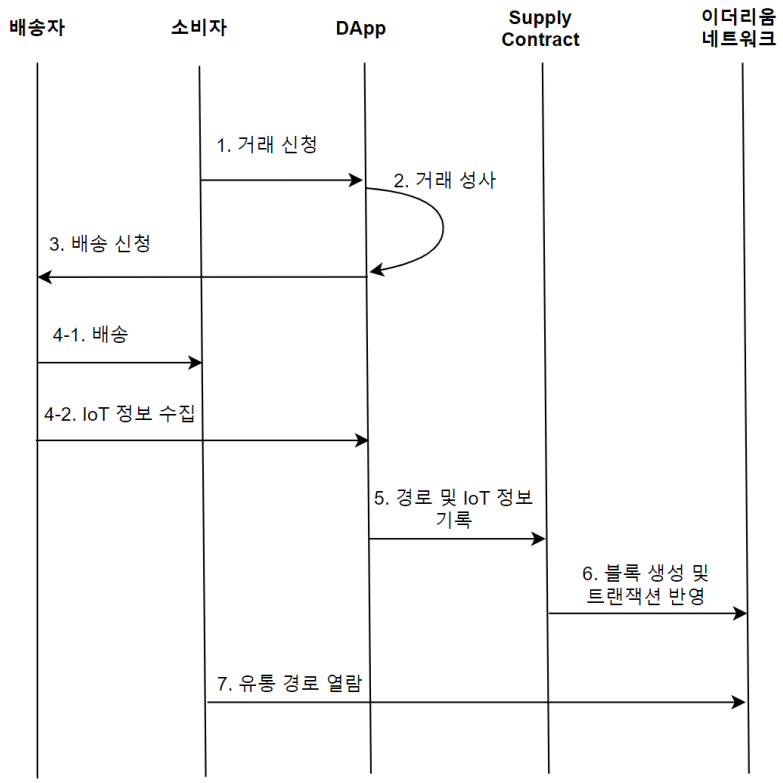
- 상품 조회

  소비자가 DApp에 등록된 상품을 선택하면 Item Contract에게 상품 정보를 요청하는 트랜잭션을 발생시킨다.

  Item Contract 상태 변수 배열에 저장된 정보를 리턴해 DApp에 전달하는 트랜잭션을 발생시킨다.

  발생한 트랜잭션을 병합해 블록을 생성 후 이더리움 네트워크에 반영한다.

C. 유통



위 시나리오에선 자체 개발한 스마트 컨트랙트를 사용하며 "Supply Contract"로 지칭한다.

거래가 성사되었을 경우 DApp은 배송자에게 배송을 신청한다.

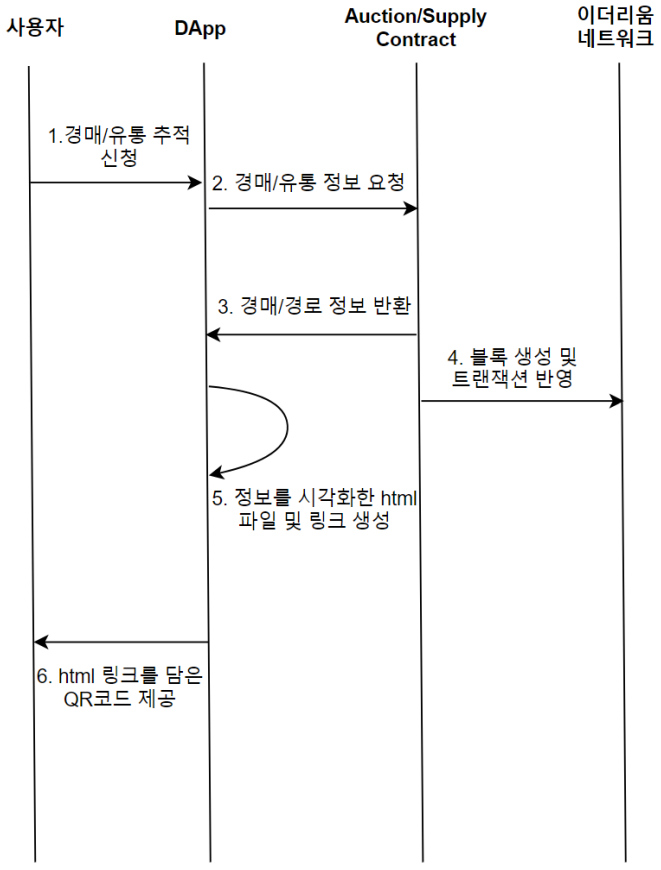
배송 중 IoT 센서로 수집한 온/습도 정보를 DApp DB에 저장한다.

배송이 완료되면 DApp에서 배송 경로(발송/도착지) 및 온/습도 정보를 Supply Contract 상태변수 배열에 기록하는 트랜잭션을 발생시킨다.

발생한 트랜잭션을 병합해 블록을 생성한 후 이더리움 네트워크에 반영한다.

모든 사용자는 QR코드를 거쳐 상품의 유통 경로를 열람할 수 있다.

D. QR코드 생성



앞서 유통에서 언급한 Auction or Supply Contract를 활용한다

.

사용자(발급자)가 DApp으로 경매/유통 정보를 요청하는 트랜잭션을 생성한다.

Supply Contract의 로컬 변수에 저장된 경매/유통 정보를 반환하는 트랜잭션을 생성한다.

발생한 트랜잭션을 병합해 블록을 생성한 후 이더리움 네트워크에 반영한다.

DApp에서 수신한 정보를 시각화한 html 페이지를 생성하고 링크를 담은 QR 코드를 발급한다.

해당 링크는 누구든지 접근 가능하고 QR코드 스캔을 지원하는 장비(ex. 스마트폰)로 코드를 스캔하면 경매/유통 추적 페이지의 링크로 이동한다.

2.2 개발환경 / 기술스택

A. 개발 도구 / 소스 / 라이브러리

구분	단위	이름
블록체인	플랫폼/인프라	Ethereum
Web UI	Front-end	react.js
API Server	Back-end	node.js
스마트 컨트랙트	Back-end	Solidity
인터넷 프로토콜	Back-end	IPFS

B. SW

이름	내용
docker	블록체인 가상 환경
docker-compose	블록체인 가상 환경
geth	블록체인 네트워크 구성
tuffle.js	블록체인 네트워크 구성
Solidity	스마트 컨트랙트
react.js	IPFS
node.js / npm	서버
vscode	개발 IDE
rest	스마트 컨트랙트 개발 IDE
git	개발 협력

3. 세부 사항

3.1 블록체인 네트워크

A. Public 블록체인

누구든지 참여 가능한 개방형 블록체인 네트워크다.

모든 데이터와 트랜잭션, 권한을 공유하기에 탈중앙화적인 특성이 있다.

그러나 많은 노드가 참여할수록 속도가 저하되고 네트워크 확장이 어렵다.

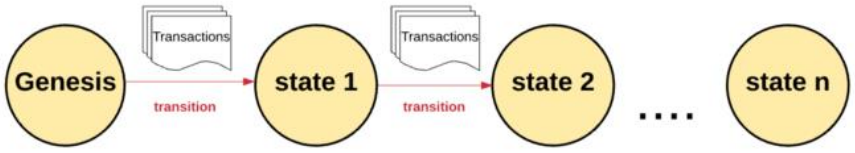
B. Private 블록체인

허가된 조직이나 개인들만 참여할 수 있는 폐쇄형 블록체인 네트워크다.

주로 기업에서 활용되며 중앙 기관이 대다수의 권한을 가지고 있다.

그만큼 확장이 간편하며 거래속도가 빠르고 인증을 거친 증명자만 참가하기에 합의 알고리즘 또한 간단하다.

C. 이더리움



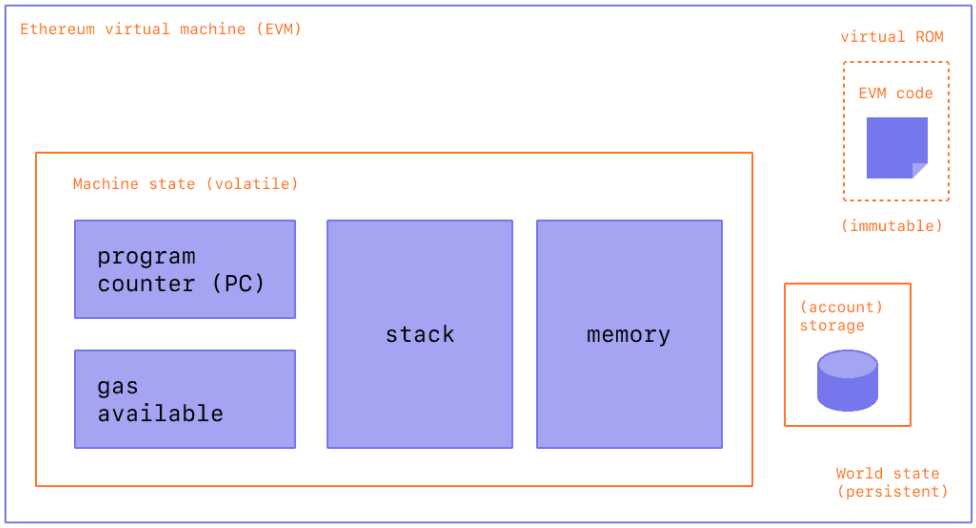
[그림 9] 이더리움 상태 머신 구조

이더리움은 public 블록체인의 일종으로 트랜잭션에 기반한 상태 머신이다.

이더리움 상태 머신은 아무런 트랜잭션이 기록되지 않은 제네시스 상태에서 출발한다.

이더리움의 상태 전이는 스택 기반 가상머신 EVM에 의해 처리되며 하이레벨 프로그래밍 언어로 작성된 스마트 컨트랙트를 컴파일한 바이트코드를 실행한다.

3.2 EVM



[그림 10] EVM 구조

EVM은 이더리움 블록체인 네트워크의 노드들이 공유하는 가상머신이다.

EVM은 모든 이더리움 노드에서 로컬 인스턴스로 실행되지만 EVM의 모든 인스턴스는 동일한 초기 상태에서 작동해 동일한 최종 상태를 생성하기에 시스템 전체가 단일 컴퓨터로 작동한다.

모든 형태의 알고리즘을 처리할 수 있는 튜링 기계로서 먼저 들어온 데이터(Byte code)를 우선적으로 처리하는 스택 구조를 가진다.

저장공간은 휘발성 저장공간인 Memory와 비휘발성 저장공간인 ROM과 Storage로 구성되어 있으며 Memory는 함수 호출 및 로컬변수를 저장하고 ROM은 영구히 저장되는 바이트 코드(스마트 컨트랙트), Storage는 스마트 컨트랙트의 상태변수를 저장한다.

본 과제에서 자체 개발한 스마트 컨트랙트 내부 method를 거쳐 상태변수를 수정하는 방식으로 Storage에 접근해 용도별 데이터를 기록 및 조회할 예정이다.

3.3 스마트 컨트랙트

```solidity
pragma solidity ^0.8.13;

// 1. contract 선언
contract HelloWorld{

    // 2. 상태 변수 선언
    string public str;

    //3. 생성자 선언
    constructor (string memory _str) {
        str = _str;
    }

    // 4-1. method 선언(set)
    function setStr(string memory _str) public {
        str = _str;
    }

    // 4-2. method 선언(get)
    function say() public view returns (string memory){
        return str;
    }
}
```

[그림 11] 스마트 컨트랙트 코드 예시

스마트 컨트랙트는 바이트 코드로 컴파일 되어 EVM을 통해 모든 노드에서 실행되는 불변적인 프로그램이다.

일반적으로 솔리디티(Solidity), 바이퍼(Viper)와 같은 고급언어로 작성된다.

바이트코드로 컴파일된 후 컨트랙트 생성 트랜잭션을 사용하여 이더리움 블록체인 네트워크에 배포되며 이후 변경이 불가능하다.

수정을 위해선 기존 인스턴스를 삭제 후 수정한 인스턴스를 재배포해야 한다.

배포된 스마트 컨트랙트는 EOA(External Owned Account)의 트랜잭션으로만 호출된 경우에만 실행된다.

Contract를 기본 객체로 삼는 구조이며 내부 method로만 상태변수에 접근하도록 하며 휘발성 데이터는 EVM의 메모리, 비휘발성 데이터는 EVM의 Storage에 저장한다.

본 과제에서 구현할 스마트 컨트랙트는 아래와 같다.

- Auction 컨트랙트

  거래별로 경매 내역을 상태 변수 배열에 저장한다.

  (거래->내역 mapping 방식)

  DApp을 통해 생성된 내역 조회 트랜잭션을 수신하면 사용자가 지정한 내역 정보를 제공하는 트랜잭션(view)을 생성한다.

- Item 컨트랙트

  상품별로 상품 정보를 상태 변수 배열에 저장한다.

  (상품 -> 정보 mapping 방식)

  DApp을 통해 생성된 상품 정보 조회 트랜잭션을 수신하면 사용자가 지정한 내역 정보를 제공하는 트랜잭션을 생성한다.

- Supply 컨트랙트

  거래별로 유통 경로 정보를 상태 변수 배열에 저장한다.

  (거래 -> 유통 경로 mapping 방식)

  DApp을 통해 생성된 유통 경로 조회 트랜잭션을 수신하면 사용자가 지정한 유통 정보를 제공하는 트랜잭션을 생성한다.

3.4 IPFS

IPFS란 분산형 파일 시스템에 데이터를 저장하고 인터넷으로 공유하기 위한 프로토콜이다. P2P 방식으로 대용량 파일과 데이터를 공유하며 해시 테이블 형식으로 데이터에 접근한다.

중앙화된 서버에서 데이터를 관리하는 HTTP와 달리 탈중앙화된 네트워크이다.

본 과제에선 이미지와 같이 블록에 기록하기엔 용량이 큰 파일을 IPFS 업로드해 해싱시키는 용도로 활용한다.

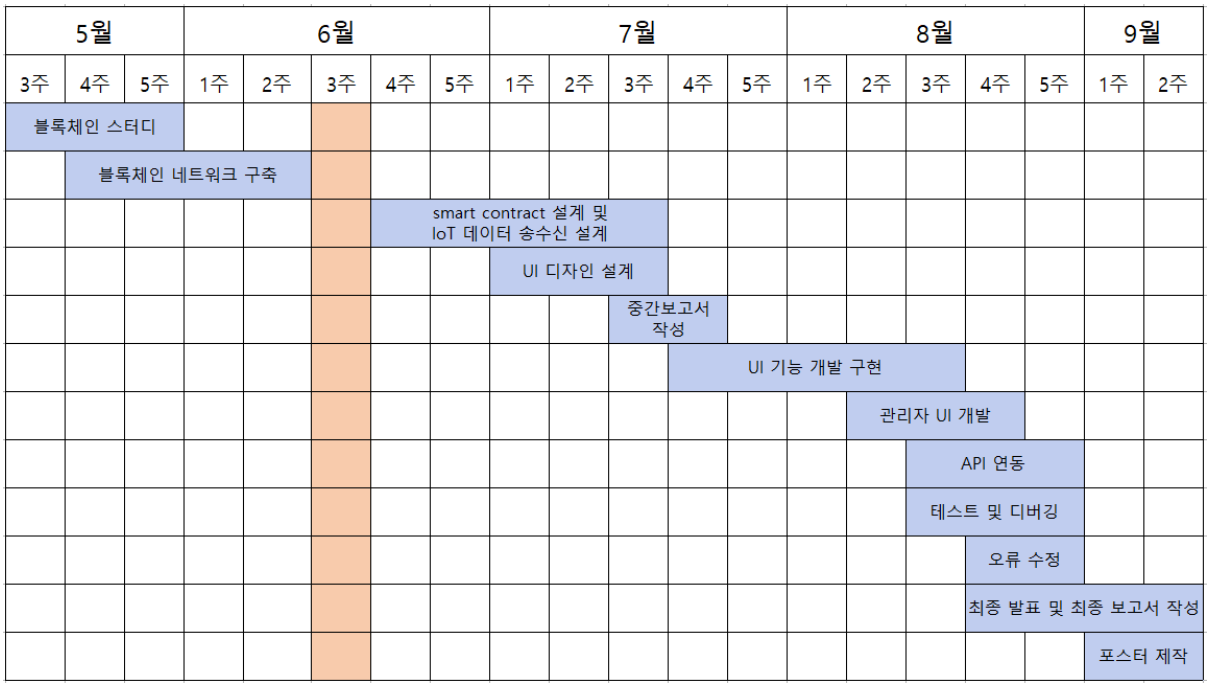
3.5 현실적 제약사항

실제 메인넷에서의 이더리움 가격은 비싸고 변동성이 크기 때문에 화폐의 역할을 하기 힘들고 블록을 생성할 때 경제적, 시간적 비용 또한 많이 소요된다.

따라서 로컬 네트워크를 구현해 경제적, 시간적 제약을 없애고 실제 가치가 존재하지 않는 이더리움을 사용한다.

4 개발 일정 및 역할분담

4.1 개발일정



4.2 역할분담

이름	역할 분담
류지환	- 블록체인 네트워크 구축 - 스마트 컨트랙트 개발
엄혜림	- UI 디자인 설계 및 개발 - 관리자 UI 설계
박서현	- 스마트 컨트랙트 개발 - 관리자 UI 개발
공동	- 테스트 및 디버깅 - 오류 수정 - 시스템 테스트 - 보고서, 문서 작성 - 발표 심사 및 시연 준비