솔리디티 개발 환경 구축 및 기본 문법, 메커니즘:

http://www.chaintalk.io/archive/lecture?sca=나도+dApp+개발

솔리디티 문법

https://m.blog.naver.com/lool2389/221291321984

스마트 컨트랙 간단 예제 코드

https://m.blog.naver.com/lool2389/221288477755

Dapp 개발 ~

https://m.blog.naver.com/lool2389/221298018082

솔리디티 **+ Oraclize** 듀토리얼:

https://medium.com/dnext-post/solidity-tutorial-1-252c9edf2f84

스마트 계약을 완성하는 마지막 퍼즐 한 조각, '오라클(Oracle)'

'블록체인 확장성 문제' 외에도 중요한 이슈가 하나 더 있다. 바로 '외부의 데이터를 블록체인 내부로 어떻게 들여와야 하는가?'에 대한 이슈다. 그럼 블록체인 상에 외부 데이터가 필요한 이유는 무엇일까? 바로 '스마트 컨트랙트(Smart Contract)'가 제대로 작동하기 위해서는 현실 세계에서 일어난 사건(외부 데이터)을 블록체인 네트워크가 인지해야 하기 때문이다.

참조로 스마트 컨트랙트(Smart Contract)란, 중개자 없이 거래 당사자끼리 디지털 명령어로 계약을 작성해서 특정 조건이 충족되면 계약 내용이 자동으로 이행되는 계약 자동화 시스템이다. 스마트 컨트랙트의 등장을 계기로, Dapp 들이 이더리움(Ethereum) 블록체인 플랫폼 위에 다양한 비즈니스 모델을 구축할 수 있게 되었다. 블록체인 2.0 의 시작이라고도 평가된다.

블록체인 기반이라 계약 내용이 모든 사람들에게 공개되기에 함부로 악의적인 행위를 하기 어렵고, 계약 조건만 충족되면 무조건 시행되기 때문에 계약 사항이 불이행 될 위험도 없다.

이처럼 중개인 수수료 없이, 개인과 개인이 수평적으로, 계약 불이행 위험 없이, 투명하고 강제적으로 이행되는 계약 시스템이라. 그러나 앞서 언급한 것처럼 스마트 컨트랙트가 제대로 작동하려면 외부 세계의 데이터가 꼭 필요하다. 기존의 중앙집중형 플랫폼의 경우에는 거래 당사자들의 거래 조건이 충족되었음을 보증해주는 '중개인'이 있었지만, 분산화된 네트워크인 블록체인은 중개인(Middle Man)에게 의존하지 않고 계약 이행 조건이 충족되었음을 스스로 인지해야 하기 때문이다.

'오늘 미세먼지가 많으니 마스크를 쓰라고 사용자에게 알림을 띄우는 것', '보험 가입자가 병에 걸렸으니 보험 약관에 의해 보험금을 지불하는 것', '기상악화로 비행기가 연착되어서 소비자에게 비행기표를 환불해주거나,

다른 비행기의 빈 좌석을 대신 제공하는 것' 등, 이 모든 것은 현실 세계에 대한 데이터 없이 진행할 수 없는 프로세스들이다. 외부 데이터를 가져오지 못하는 Dapp 은 극히 1 차원적인 솔루션만을 제공할 수 있을 뿐이다.

그렇다면 블록체인 네트워크 외부(Off-Chain)에서 데이터를 가져오는 게 왜 어려울까? 외부에서 데이터를 불러오는 경우 다음과 같은 이슈(Oracle problem)가 생기기 때문이다.

1) 어떤 채널에서, 어떤 주체로부터, 어떤 데이터를 수집할 것인가?

중앙집중형 플랫폼의 경우 신뢰할 수 있는 제 3 자가 존재한다. 즉 계약 당사자가 아니어도 계약 내용을 인지하고 있으며, 계약 당사자들이 계약 내용을 제대로 이행하는지를 감시하고 보증하는 중개인이 있다. 이들은 계약 이행에 필요한 정보를 제공해주며, 문제가 발생할 시에도 이를 중재한다.

그러나 스마트 계약은 '특정 조건이 충족될 경우 특정 행위가 자동으로 이행된다.'라는 깰 수 없는 약속일 뿐, 실제로 '특정 조건'이 충족되었는지를 확인하고 보증해주는 제 3 자가 없다. 무슨 채널을 통해 누구로부터 어떤 형식의 데이터를 수집할 것인지가 불명확한 것이다.

2) 데이터 전송 과정에서 위변조 가능성이 있는가?

블록체인 네트워크를 해킹하려면 어마어마한 채굴 자원이 필요하다.
사실상 해킹에 성공할만한 자원을 확보하는 것은 거의 불가능에 가깝고, 설령 확보하더라도 얻게 되는 이득에 비해 잃게 되는 시간/채굴 자원이 더 많을 것이다. 그렇기에 블록체인은 결과적으로 보안성이 우수하다는 특성을 지니고 있다.

그러나 외부 데이터를 블록체인에 전송하는 과정에서 해커가 개입하여 데이터를 위조한다면 어떨까? 전송된 데이터가 전송 도중 위변조된 데이터인지를 골라낼 수 있는 판단 근거가 없다면, 악의적인 행위에 의해 계약 조건이 충족되지 않았음에도 불구하고 거래가 진행될 수 있다.

3) 외부에서 가져온 데이터를 신뢰할 수 있는가? (데이터의 무결성)

애초에 외부 채널에서 가져온 데이터 자체에 오류가 있다면 어떨까? 다시 말해, 신뢰할 만한 채널을 통해 데이터를 가져오더라도 이조차도 사실과 무관한 데이터라면 어떨까? 단순한 기록 누락, 통계의 오류, 하드웨어 센서를 조작하여 데이터를 바꾸는 등, 여러 가지 요인에 의해 데이터 자체에 오류가 있을 가능성도 있다.

스마트 컨트랙트는 해당 데이터가 '사실'인지 아닌지를 자체적으로 판단하지 못한다. 그렇기에 잘못된 데이터가 입력된다 하더라도 계약은 그대로 이행되며, 블록체인의 비가역적인 특성(한 번 블록체인에 입력된 기록은 변경할 수 없다) 때문에 한 번 이행된 스마트 컨트랙트는 되돌릴 수 없다. 이와 같은 이유로 Dapp 이 제기능을 못하거나 큰 문제를 일으키면, 자연스레 Dapp 의 가치는 떨어질 것이다.

위에서 살펴본 것처럼 '외부 데이터를 블록체인 내부로 가져올 때' 발생하는 다양한 문제들을 '오라클 문제(Oracle problem)'라고 부른다.

---

3. 블록체인과 외부 세계를 연결하는 통로, 오라클(Oracle)의 유형

오라클은 외부 세계의 데이터를 블록체인 네트워크 내로 들여오는 기술이다. 즉, 현실 세계의 데이터를 찾아서 진위 여부를 확인하고 이

정보를 블록체인 네트워크에 제출하는 역할을 한다. 만약 스마트 컨트랙트으로 1 차원적인 솔루션을 구현하는 데에서 나아가, 더 다양한 분야에 활용하고자 한다면 오라클은 필수적이다.

오라클의 유형은 크게 하드웨어 오라클(Hardware Oracle)과 소프트웨어 오라클(Software Oracle)로 나뉜다.

1) 하드웨어 오라클

먼저 하드웨어 오라클이란 외부 변화를 감지해서 데이터를 송신하는 센서(sensor)를 의미한다. 센서는 온도, 음식의 부패 정도, 특정 물품의 파손율, 온실가스 배출량 등, 현실 세계의 다양한 정보들을 데이터화하여 블록체인 네트워크에 제공할 수 있다.

예를 들어, 회사 휴게실에 센서가 탑재된 커피 자판기가 있다고 가정해보자. 자판기 내부에 센서가 있으니 커피 원두가 얼마나 남았는지를 자동으로 인식할 수 있다. 이 경우에 자판기 관리자가 '자판기 내에 남은 원두의 양이 100g 이하가 되면 A 업체에서 자동으로 새로운 원두 5KG 를 주문한다'라는 스마트 컨트랙트를 셋팅할 수 있다. 즉, 센서가 현실 세계의 사건을 블록체인에 보고하게 함으로써 거래/구매 프로세스를 자동화할 수 있다.

2) 소프트웨어 오라클

두번째로 소프트웨어 오라클이란 온라인 상의 신뢰할 수 있는 채널(ex.기상청, 중앙은행, 항공사 홈페이지 등)로부터 데이터를 가져오는 것을 의미한다. 환율, 상품의 가격, 비행기 또는 기차의 연착 정보, 선거 결과 등, 웹 사이트에서 얻을 수 있는 데이터를 모아 블록체인 네트워크에 제공하는 것이다.

예를 들면 '특정 업체 주식의 주가가 0000 원에 도달하면 해당 주식을 10 주 구매한다.' 또는 '미국 달러와 한국 원화 사이의 외환율이 00% 이하가 되면 USD 달러를 00000 원 어치 자동으로 환전한다.'등의 스마트 컨트랙트를 셋팅할 수있다.

소프트웨어 오라클, 하드웨어 오라클 유형 외의 오라클도 존재한다. 그 중 하나가 블록체인 상에서 발생한 거래 데이터를 외부로 전송하는 '아웃바운드 오라클(Outbound Oracles)' 이다. 적용 예시로는 'A 가 자전거를 빌리는 대가로 B 에게 코인을 지불하면, 스마트 컨트랙트가 B 의 자전거 IOT 자물쇠 잠금장치를 일정시간 동안 해제'하는 역할을 할 수 있다.

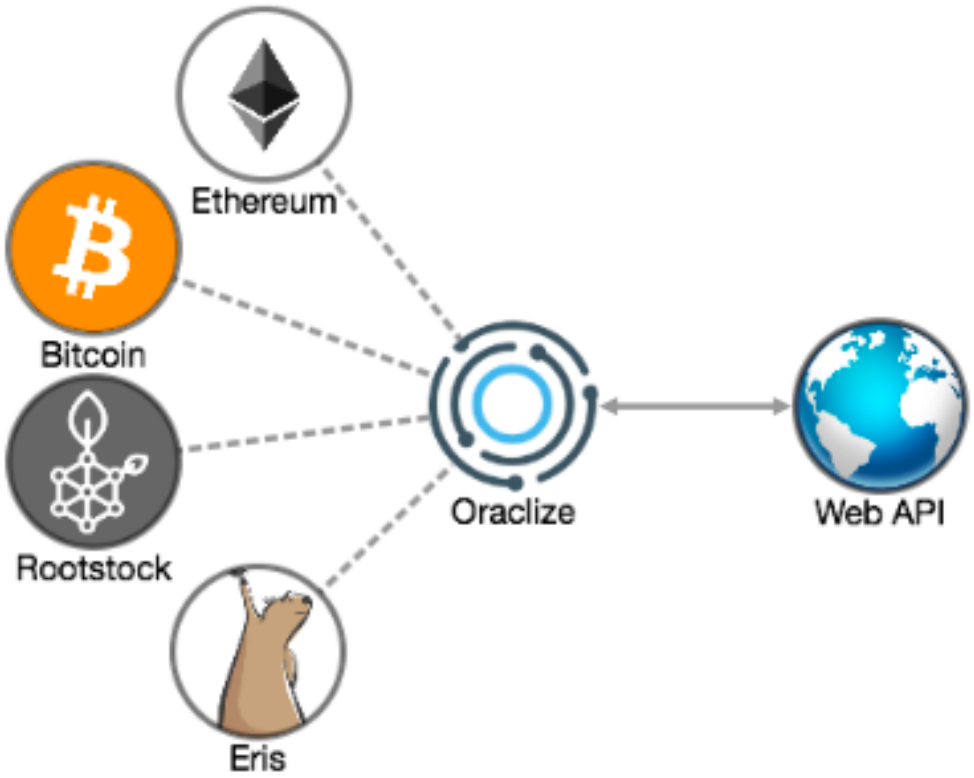
---

4. 오라클 문제(Oracle problem)는 어떻게 풀어야 할까?

현재까지 대부분의 Dapp 이 적절한 오라클 모델을 구현하는 데 많은 어려움을 겪고있는 것으로 보인다. 그도 그럴게 오라클은 복잡한 현실 세계의 데이터를 자동으로 수취하고, 이를 데이터화 해서, 스마트 컨트랙트까지 안전하게 전송하는 역할을 한다. 스마트 컨트랙트와 제대로 상호작용하는 모델을 만들기 위해서는 많은 시간을 들여 세밀하게 실험을 진행하고, 논리의 허점을 점차 보완해가야 할 것이다.

그렇다면 현재 단계에서 오라클 문제를 극복하고자 제시되고 있는 방법에는 어떤 것이 있을 까? 이번 지면에서는 '미들웨어(Middleware)'와 '지분 증명(PoS)'을 활용한 방식을 소개하고자 한다.

1) 미들웨어(Middleware)



이미지 출처: https://blog.oraclize.it/new-blockchain-integrations-and-beyond-e1a6d92bda85

먼저 오라클 전문 미들웨어란, Dapp 이 자체적으로 오라클 모델을 만들지 않아도 검증된 외부 데이터를 가져올 수 있게 타업체로부터 솔루션을 제공받는 것이다. 미들웨어는 스마트 컨트랙트가 작동하기 위해 필요로 하는 각종 측정 데이터를 API 형태로 제공한다.

그러나 미들웨어란 -기존 중앙집중형 플랫폼이 '중개인 수수료'를 부과해왔던 것처럼- 신뢰할 수 있는 제 3 자가 스마트 컨트랙트에 필요한 판단 근거(데이터)를 제공하는 솔루션이다. 따라서 미들웨어 사용이 보편화될 경우 또다른 '중개자'가 등장할 가능성을 배제할 수 없다. 또한 미들웨어가 제공하는 데이터 역시 무결성이 100% 보장되지 않는다는 점은 매한가지이다.

2) 지분 증명(PoS)

두 번째는 지분 증명(PoS)을 통해 오라클 모델을 구성하는 방법으로, Dapp 에 일정 지분을 보유한 노드를 통해 오라클 모델을 구축하는 것이다.

여기서 지분 증명(PoS)이란 합의 알고리즘으로, 특정 블록체인에 해당하는 지분(Stake)을 보유한 노드에 새로운 블록을 생성할 수 있는 권한을 부여한다. 지분이 많을수록 블록을 생성할 수 있는 기회가 더 높은 확률로 주어진다. 과반수(51%) 이상의 지분이 동의한 블록이 더 빠르고 길게 다음 블록을 형성한다.

[참조] 합의 알고리즘의 발전과 진화방향

만약 둘 이상의 (Dapp 의 일정 지분을 보유한) 노드들로 하여금 외부 데이터를 블록체인에 보고하게 하고, 이에 대한 보상으로 토큰을 제공하면 어떨까? 이때 보고된 데이터들 중에서 가장 많은 지분이 걸린 데이터를 '참'으로 인정하는 것이다. 나머지 노드가 보고한 데이터는 '거짓'으로 판정된다.그리고 블록체인 네트워크는 거짓 데이터를 보고한 노드에게서 토큰(지분)의 일부를 회수한다. 이렇게 걸러진 무결한 데이터는 스마트 컨트랙트를 이행하는데 필요한 추론의 근거로 사용된다. 즉, 보상과 패널티 구조를 세밀히 설계한다면 블록체인 네트워크만으로도 오라클 모델을 만들 수 있는 셈이다.

PoS 에 의한 오라클 모델을 조작하려면 공격자가 막대한 자금을 지불해서 Dapp 의 지분을 과반수 이상 사들여야 한다. 따라서 Dapp 을 공격하더라도 얻을 수 있는 기대 이익은 아주 적을 것이다. 만약 과반수 이상의 지분을 보유한 공격자가 데이터 조작에 성공하더라도, 스마트 컨트랙트가 제대로 작동하지 않는 한 Dapp 은 그 가치를 상실하게 된다. 따라서 Dapp 을 이탈하는 노드가 많아질 것이며 그럴수록 해당 Dapp 의 사용성은 근본적으로 무너질 것이다. 토큰의 가치가 하락하면 과반수 이상의 지분을 사들였던 공격자도 큰 금전적 손실을 입게 된다.

즉, 노드가 Dapp 의 가치를 훼손할 의향이 없고, 거짓 데이터 보고로 인한 패널티를 물지 않으려면, 사실에 근거한 데이터를 보고하고 리워드를 받는 편이 더 유리하다.

PoS 를 응용한 오라클 모델을 적용한 Dapp 의 사례로는 미래예측시장 플랫폼 'Augur(어거)'가 있다.

이미지 출처: Augur(어거) 백서

[참조] 집단지성으로 만드는 미래 예측 시장, Augur(어거)

Augur 의 유저들은 미래 사건의 결과에 대해 코인을 걸고 베팅한다. 그리고 미래에 이벤트의 결과가 확정되면 예측에 걸었던 베팅금을 잃거나(잘못된 예측), 리워드와 함께 걸었던 베팅금을 돌려 받는다(올바른 예측). 그럼 유저들의 베팅금을 배분하는 기준인 '실제 이벤트의 결과'를 보고하는 주체는 누구일까?

바로 'REP(평판토큰)'을 보유하고 있는 Reporter(보고자)들이다. 이들은 예측 시장에 생성된 베팅의 실제 결과를 입력할 수 있는 권한과 책임을 가지고 있다. 만약 Augur 플랫폼에 '00 지역의 0000 년도 00 월 강수량은 000mm 이상이다.'라는 예측시장이 형성되면 Reporter(보고자)가 일정 기간 내에 '00 지역의 0000 년도 00 월의 실제 강수량은 000mm 였다.'라는 데이터를 스마트 컨트랙트에 보고한다.

Augur 에 가짜 데이터를 보고한 Reporter 는 패널티로 토큰을 지불해야하며, 사실 데이터를 보고한 Reporter 는 REP 토큰과 예측시장에서 배팅되었던 금액의 일부를 수고료로 얻는다.

그러나 **PoS 에 의한 오라클 모델도 한계점**이 있다. 첫번째, 스마트 컨트랙트가 어떤 형식으로 데이터를 보고받을 것인지에 대한 기준을 사전에 지정해야 한다. 같은 데이터라도 그것을 보고할 수 있는 방식은 다양하기 때문이다. (ex. '1,2,3'과 같은 아라비아 숫자, '일, 이, 삼'과 같은 특정 언어로 표현한 숫자, '금주의 로또번호는 1,2,3 이다.'과 같은 문장형 답변, 이밖에도 O/X 또는 YES/NO 중 하나 선택 등)

두 번째, 무결한 데이터를 입력하기 위해 매번 다수로부터 충분한 양의 데이터를 보고받으려면 시간이 소요된다. 따라서 현실 세계에서 이미 스마트 컨트랙트 이행 조건에 준하는 결과가 발생했어도, 이를 블록체인 네트워크가 인지하기까지 시간이 오래 걸린다.

마지막으로, 노드가 올바른 데이터를 보고하도록 유도하기 위해선 매번 일정 수준의 보상(수수료)을 제공해야 한다. 만약 가짜 데이터를 보고함으로써 얻게되는 이득이 사실을 보고해서 얻게되는 보상보다 크면 노드가 제출한 데이터의 무결성을 담보할 수 없게 된다. 그러므로 이러한 수수료의 부담을 덜고자 부분적으로 PoS 에 의한 오라클 모델을 사용하자는 의견도 있다. 기본적으로는 하드웨어/소프트웨어 오라클을 사용하되, 오라클이 보고한 데이터에서 오류를 발견하고 이를 보고한 노드에게 보상(현상금)을 제공하는 방식인 것이다. 하지만 이 방법도 완전한 데이터 무결성을 보장하지는 못한다.

여기까지 살펴보았듯이, 오라클은 스마트 컨트랙트가 제대로 작동하기 위해서 꼭 필요한 시스템이다. 하지만 현재까지 논의된 방식과 이론들이 복잡한 현실세계에서 얼마나 효율적으로 적용될 것인지는 좀 더 지켜보아야 할 여지가 많다. 블록체인마다 서로 다른 합의 알고리즘을 사용하며, 전달하고자 하는 가치도 제각각이다. 더 효율적인 블록체인 네트워크를 구축하기 위해 다양한 시도가 이루어지고 있는 것처럼 오라클도 수많은 시행착오를 거치며 점차 보완되어갈 것이다.

오라클 문제가 무엇인가?

- 블록체인 안(온체인)에서 발생한 데이터가 아니라 블록체인 밖(오프체인)에서 발생한 데이터를 블록체인 안으로 가져올 때 신뢰성을 보장받을 수 없는데, 이러한 문제를 오라클 문제라고 합니다.

온체인, 오프체인 트랜잭션은 무엇인가?

- 온 체인 트랜잭션(On-chain Transaction)
  말 그대로 체인 위에 발생하는 트랜잭션입니다. 그런데 여기에서의 체인은 메인(단일) 블록체인 네트워크를 의미합니다.

예 : 비트코인, 이더리움 등의 자체 네트워크를 구성하고 있는 블록체인 내에서 발생하여 블록에 기록되는 트랜잭션들입니다.

- 오프 체인 트랜잭션(Off-chain Transaction)
  온 체인 트랜잭션의 정의를 이해하셨다면, 오프체인은 그냥 쉽게 이해하실 수 있습니다. 그냥 메인 블록체인이 아닌 곳에서 발생하는 트랜잭션인 것입니다.

예 : 이더리움 네트워크의 입장에서는 비트코인 네트워크에서 주고 받는 트랜잭션은 오프체인 트랜잭션인 것입니다. 반대로 비트코인 네트워크의 입장에서는 이더리움 네트워크에서의 트랜잭션이 오프체인 트랜잭션이 되겠죠.

- 온 체인(Onchain), 오프 체인(Offchain)이 무엇인가?
-

오라클라이즈는 무엇인가?

- 오라클 문제를 해결하기 위한 하나의 방안으로 오프체인—온체인 사이에 위치하는 미들웨어입니다. 신뢰할 수 있는 third party 역할을 하여 신뢰성을 높이려 하고 있습니다. 탈중앙화 된 방법은 아닙니다.
- Ethereum, Rootstock, R3 Corda, Hyperledger Fabric, EOS 에서 오라클라이즈를 이용할 수 있습니다.
-

오라클라이즈는 어떻게 사용하는가?

- 공식 문서
- 사용 예제
- 오라클라이즈 쿼리 테스트
-

데이터 소스 종류

- URL: 웹사이트나 API 를 호출한 값을 가져옵니다.
- WolframAlpha: 울프럼알파 서비스에 검색한 결과 값을 가져옵니다.
- IPFS: IPFS 에 저장 된 파일에 접근합니다.
- random: Ledger Nano S(하드웨어 지갑)에서 동작하는 안전한 어플리케이션으로부터 생성한 랜던한 바이트를 가져옵니다.
- computation: 연산의 결과를 가져옵니다.
- nested: 여러 종류의 데이터 소스 또는 같은 종류의 여러 데이터 소스에서 결과를 받아서 단일 결과를 가져옵니다.
- identity: query 자체를 반환합니다.
- decrypt: 오라클라이즈 개인키로 암호화 된 문자열을 복호화합니다.

파싱 가능한 데이터 형태

- JSON
- XML
- HTML(XPATH)
- Binary
-

가격

- 컨트랙트 당 첫번째 call 을 무료
- 두번째 call 부터는 value 에 ether 를 넣어서 호출해야 됨

		Proof type			
Datasource	**Base price**	None	TLSNotary	Android	Ledger
URL	**0.01$**	+0.0$	+0.04$	+0.04$	N/A
WolframAlpha	**0.03$**	+0.0$	N/A	N/A	N/A
IPFS	**0.01$**	+0.0$	N/A	N/A	N/A
random	**0.05$**	+0.0$	N/A	N/A	+0.0$
computation	**0.50$**	+0.0$	+0.04$	+0.04$	N/A