SAFEPAPA

White Paper

June 29 2021

SAFEPAPA 메인넷의 정식 명칭은 파파라치(PAPARAZZI)로 범법자들을 쫓아다니며 사진을 찍는다는 의미를 담고 있습니다. 현재는 행정안전부 안전신문고 제보를 통해서 많은 사회의 변화를 꾀하고 있지만 어렵고 복잡한 과정으로 인하여 국민들의 외면을 받고 있는것이 현실입니다. 남녀노소를 막론하고 저희 SAFEPAPA를 이용하여 누구나 안전에 신속하게 제보함으로 그에 따른 보상과 사회질서의 견인차 역할을 톡톡히 할 수 있을 것으로 제안을 하게 되었습니다. PAPARAZZI는 SAFEPAPA의 사명을 실현시킬 핵심기술로, 범법행위에 대한 데이터 수집, 관리 및 활용이 이루어지는 생태계를 만들게 됩니다. 그 동안 시민들이 겪었던 문제점은 PAPARAZZI를 통해 개선 및 해결되고, 우리 사회는 완벽하게 재창조 될 것입니다. PAPARAZZI 블록체인 상에서는 노드들의 합의를 바탕으로 내용을 기록, 검증하기 때문에 한 번 성공적으로 기록된 내용은 위조 및 변조가 불가능합니다. 사진정보의 해시값을 PAPARAZZI에 기록하고 이를 통해 범법행위에 대한 무결성 및 소유권 검증을 할 수 있는 것이 PAPARAZZI의 핵심 기능입니다. 이는 PAPARAZZI 생태계의 경제적 바탕이 되는 독자적인 암호화폐(PAZ코인)를 기반으로 원활하게 활용될 수 있습니다.

BACKGROUND

현 실태 및 문제점

행정안전부의 안전신문고는 국민들에게 신고만 종용 할 뿐 적극적으로 참여하기 어렵게 만들었다. 신고시:1점 신고이후 만족도 등록 평가시:1점으로 정말이지 어렵게 마일리지 적립 포인트가 쌓이고 적립되고 있다. 그러나 아직까지 마일리지 적립자 활동우수자들에 대한 어떠한 포상이나 상장등 인센티브는 없고 전무한 실정으로 문제점이 대두가 되고 있는 실정이다.

보상체계(INCENTIVES)

사용자의 투표를 많이 받은 검증자는 블록 생성 순서가 더 자주 돌아오고, 득표를 많이 할 수록 가중치가 늘어나 검증자로 선정될 확률이 높아집니다. 검증자는 SAFEPAPA를 통해 정부에서 제공하는 지역화폐및 상품권을 받을 수 있으며 사용자들간 교환을 통해 이익을 얻을 수 있습니다. 검증자 본인은 스스로 자신의 수수료율을 설정할 수 있으며 해당 검증자에게 투표한 사용자(위임자)에게는 이렇게 책정한 수수료를 제외한 후 투표 비율대로 보상을 배분합니다. 투표한 사용자(위임자)는 검증자의 정보와 블록생성 수수료를 토큰스왑 이후 검증자 선출시 PAPARAZZI 익스플로러에서

확인할 수 있습니다. PAPARAZZI 상에 스테이킹(Staking)된 비율에 따라 인플레이션 비율은 조정됩니다. 스테이킹한 코인의 비율이 상대적으로 낮아지면 인플레이션 비율이 높아지고, 반대로 스테이킹한 코인 비율이 상대적으로 높아지면 인플레이션 비율은 낮아집니다. 따라서 PAPARAZZI의 검증자들이 스테이킹을 많이 할수록 시중에 유통되는 SAFAPAPA코인의 수량이 줄어들고, 가치를 유지하게 됩니다.

기대효과 :

SAFEPAPA를 통해서 우리의 사회는 안전에 관련하여 신속하게 제보하고 신속하게 관련기관에서 해결하여서 삶의 기폭제가 되어야 한다. 파파라치제도 활성화 차원에서도 매우 더더욱 기여하여 국민들의 삶의 질이 업그레이드 될 것 이며 될 수 있도록 지속적인 방법들이 꾸준하게 진행이 되어야 합니다. 국민들이 제보를 열심히 하고 있는데 관련기관에서는 보여 주시기식 일회성으로 비춰지고 지나가면 정말로 안된다는 것입니다. 우리의 일상생활 안전에 국민모두가 관심을 갖도록 해서 안전 불감증에서 벗어나는 계기가 될 수 있어서 기대효과는 매우 좋을 것입니다.

Technology

공익 목적의 제보나 신고를 독려하는 취지로 만들어진 플랫폼이니만큼 제보자에 대한 익명성 보장이 매우 중요하다. 퍼블릭 블록체인의 경우 보상 지급이나 거래시에 계좌나 개인정보가 노출될 위험성이 있으므로 프라이빗 블록체인의 일종인 하이퍼레져 패브릭(hyper ledger fabric)을 기반으로 하여 시스템을 만든다.

하이퍼 레져는 많은 장점을 가지고 있다.

1. falut tolerance, 공인인증, 데이터베이스 기능에 대해 블록체인에 다 구현이 되는데 이 과정에서 업무 프로세스가 늘어나는 것이다. 기존 중앙 시스템은 따로따로 구현이 되었지만, 블록체인은 일괄적으로 업무 프로세스가 되어 있다.

2. 이더리움 같은 경우 가스비, 하이퍼레져같은 경우는 공유 원장을 저장할 하드가 필요하다. 하이퍼레져 같은 경우는 가스비가 안들게 구축을 할 수 있지만, 초기 만들 때 비용이 들어간다. 리눅스를 설치하고 하이퍼레져를 설치하는 것은 무료이다.

3. 하이퍼레져는 즉 허가받은 사람 만이 접근을 할 수 있다. 하이퍼레져의 장점은 접근에 대한 제어를 할 수 있다는 것이다. 예를 들어서 투표 시스템 같은 경우 쓰기는 가능하지만 읽기가 가능하지 않다. (자기는 투표를 할 수 있지만, 다른 사람의 투표를 못본다는 것) 즉 Permissionless 라는 것이다.

4. 암호화 기능

유저가 트랜잭션을 요청을 할 때 인풋 데이터에 대해 암호화를 시켜 체인코드로 전송을 한다. 이 때 누군가가 끼어들 수 없게 하는 것이고, 체인코드는 암호화를 시켜서 수행을 시키고 최종적으로 데이터들 또한 암호화가 이루어진다. 즉 이중 삼중으로 암호화를 한다는 것이다. 이더리움 같은 경우 계속해서 암호화 알고리즘을 추가를 한다는 것이 가스 차원에서 부담 스럽다. 비트코인은 코드상으로 이러한 알고리즘을 적용하기 부담 스럽기 때문에 하이퍼레져가 아무래도 괜찮다

이러한 장점들은 우리가 구상한 시스템을 만드는데 아주 적합하다.

채널(Channel) :

1. 블록체인 네트워크에 아무나 못들어옴

2. 채널에 아무나 못들어옴

3. 채널 내에서 private을 설정하여 접근을 권한을 할 수 있다. (공산품이 농산품을 못들어오기에)

4. 같은 채널 내에서 private설정, 누군가가 데이터를 전송하는 부분에 있어서 이 데이터를 접근할 수 있는 범위를 설정을 해줄 수 있음. (access control이 더 정교하게 이루어진다.)

프라이빗 데이터는 API가 다르다. 항상 접근 메소드 이름에 프라이빗이 붙는다.

채널에 접속을 하기 위해서는 일단, 피어가 누구인지 알아야한다. (역할 부터 등등)SDK가 게이트웨이 역할을 한다. 게이트웨이를 통해서 채널과 연결을 한다. 그 채널을 통해서 체인코드와 연결이 된다. 그 다음에 스마트컨트랙을 연결하여 리턴을 받고 해당하는 스마트컨트랙트의 트랜잭션을 불러서 아까와 같은 과정을 수행하고 결과를 리턴을 받는다.

합의 알고리즘

하이퍼레져의 종류 중 FABRIC을 사용을 할 것이고, 합의 알고리즘은 RAFT를 사용하는 것인데, 이것은 리더가 하자는 식으로 진행이 되는 합의 알고리즘이다. 작년 연말까지는 long term support라는 것이 있는데, 이는 이 버젼을 오래쓰게 될 것이라는 것이데, 결국은 2.0 버젼을 사용을 할 것이다.

AP

---+---

smart Contract

---+---

응용 프로그램

---+---

API , SDK

---+---

SystemChain code (데이터 접근, 합의 p2p, 맴버십 서비스)

Middle ware (linux - docker)

---+---

DB(Level DB,

---+---

하이퍼레져 구조

DB :

기존의 블록체인과 달리, 하이퍼레져의 블록체인 안에는 데이터베이스가 존재한다. couchDB가 그것이다. 이더리움에서는 데이터의 역할을 하는 것이 있어도 데이터베이스가 존재하지 않다. 여기 counchDB는 블록체인 안에 들어있는 데이터베이스이다. 블록체인에 들어와 있는 것을 수정 삭제가 안되는데, 이것이 장점이었는데 (위변조 불가능) 이것이 단점도 존재한다. 즉 정보보호에서 잊혀질 권리에 대해 부합하는 것이다. 즉 틀린 정보에 대해서도 수정이 불가능 했던 것이다. 하지만 이 하이퍼레져는 데이터베이스가 있기에 수정, 삭제가 가능해진 것이다. 이는 블록체인 안에 존재하지만,

기존의 데이터베이스와 다르다는 것이다.

스마트 컨트랙트 :

하이퍼레져는 스마트 컨트랙트와 데이터와 네트워크를 따로 나누어 놓은 것이다. (이더리움 같은 경우 전부 합쳐져서 블록에 담겼다면)

Fabric Ledger 를 보면 world state + Block Chain 를 합쳐서 Ledger라고 부른다.

여기서 트랜잭션을 read / write 라고 한다. 즉 어떤 액션이 일어난 히스토리라고 보면된다. 어떤 데이터를 읽고 나서 쓴다고 한다는 과정인 것이다. Chanel configurations 블록은 처음에 블록체인을 초기화 할 때 channel configuration에 대한 내용이 담겨 있다. (제네시스 블록이라고 생각하면됨) 그 뒤로 일반적인 read/write에 대한 정보가 블록에 담기게 된다. DB의 종류는 LevelDB/ CouchDB가 있는데, 전자는 심플한 쿼리는 것과 후자는 복잡한 쿼리문이 가능하다. 즉 데이터베이스도 원하는 부분 정도로 설정이 가능하다.

하이퍼레져 구성요소

: Peer는 committing 트랜잭션을 담당한다. Endorse가 트랜잭션을 수행하고 order가 블록을 생성하고 validate가 이 앞에 두 사람의 역할을 검증을 한다. Endorse는 트랜잭션을 수행하고 거래당사자가 그곳에 사인을 하는 것이다.

트랜잭션 매커니즘

1. 클라이언트가 브록체인 네트워크에 접속을 하여 트랜재션을 일으킨다.

2. endorse로 이동하여 사인에 대한 표시가 기제된다. (트랜잭션이 하나가 처리가 됨)

3. 처리된 트랜잭션이 order로 이동을 한다. (트랜잭션 충돌이 발생하면 트랜잭션 취소를 한다)

4. order는 이러한 처리된 트랜잭션들을 모아서 블록을 생성을 한다. (간접생성)

5. 이렇게 전체적인 트랜잭션 과정 중 충돌에 대한 부분을 확인을 하는 작업을 해시 값을 통해서 피어들이 관리 감독을 한다.

6. 최종 승인이 되면 블록이 생성이 된다. (진짜 생성)

Endorse의 디테일 :

클라이언트가 트랜잭션을 요청을 하면, Endorsement policy에 따라 트랜잭션을 일으키려면 누군가의 서명이 필요하다는 것의 역할을 하는 수행이 생긴다. (서명 거래) 트랜잭션 요청을 받은 이는 실제로 트랜잭션을 수행을 하고 (아직 ledger에 업뎃은 안됨) RW set으로 응답을 한다. 이러한 트랜잭션을 ordering에게 보내면 모아서 블록을 생성한다. 여기서 합의 하는 알고리즘은 SOLO(single node, development), Kafka(crash falut tolerance) 를 사용을 한다.(2.0버젼에서는 RAFT) 여기서 실패한 것에 대해서는 블록체인에 기제가 되지만 데이터베이스에는 들어가지 않는다. (실패한 것도 조차 이력이다) 데이터베이스는 실제로 예약한 부분이라 이중으로 예약을 할 수는 없기에 블록체인에만 기록을 하는 것이다. ECCC가 트랜잭션을 수행하고 order가 블록을 만들고 VSCC commtting peer가 검증을 한다.

Order의 디테일 :

트랜잭션이 몰려오면 블록을 하나 만드는 것이다. 블록을 보통 홀수로 만드는 것이 좋다. 왜냐하면 기본 원칙이 투표방식이라 1:1 비율로 갈리면 답이 안나올 수 있다. 채널이 분리되면 다른 채널에 들어갈 수가 없다. (다른 차원) 그렇기에 채널이 찢어지면 ledger 또한 다른 것을 이용하게 되는 것이다

하이퍼레져의 서비스 ?

1. 신원확인 :

하이퍼레져 블록체인에 참여하는 user들의 신원을 확인하는 과정으로 지원을 하는 것이 MSP, 등록이 있다.

Client 는 전자지갑에 접근 카드가 있어서, 클라이언트가 네트워크에 접속을 하기위해 MSP가 제공한 Identifi가 전자지갑에 존재해야 접속을 할 수 있다. 하이퍼레져 네트워크에 참여하는 당사자들은 모두 전자지갑을 가지고 있다. (PKI) 이것이 서명될 때도 사용을 하니깐, 통신라인에서 데이터를 주고받을 때 암호화를 해서 데이터를 주고 받는 기능도 있다. (채널, 유저, 운영자, peer, order, channel)

2. 원장, 거래 (ledger, transaction) :

분산원장, 네트워크 프로토콜, endorsement검증, 합의 서비스 (RAFT), 보안 및 암호화 서비스

3. 스마트 컨트랙트

: 비즈니스 로직, 보안 레지스트

네트워크 구축

: 처음에는 Order를 생성을 한다. Configuration, 해당 명령어가 존재를 하는데 (docker-compose[-f orderer.yml] 이라고 한다. 이 configuration파일대로 구성을 해달라는 것이다. 이 명령어를 통해서 Order를 생성을 하는 것이다. 그 다음에는 command로 peer node start를 검색을 하면 peerNode들이 생성이 된다. 그 이후 chainCode들이 install 이 되고 나면, channel을 생성을 해야한다 (peer channel create) 그 다음. 이 채널과 연결을 해야한다. (peer channel join) 아까는 Install 을 하였다면, 프로그램을 설치를 했지 수행을 한 것이 아니니, peer cahincode instatiate -P policy 를 통해서 프로그램을 실행하고 서명을 누가 할지 지정을 하는 것이다.

체인코드 진행 과정 :

체인코드가 진행하는 과정은, 블록체인 관리하는 자가 deploy를 하고, (프로그램을 수행 시켰다면) 체인코드를 호출을 하고 읽고 쓰고, 저장을 하고 원장에 기록되면 체인코드에 데이터를 조회를 할 수

있다. 읽는 것은 디비에서 읽어오지만, 쓰는 과정은 피어들의 합의 과정이 포함된 블록체인의 매커니즘을 바탕으로 쓰게 되었다.