



유니블럭조

이다훈 김재원 김혜진 손민성 엄찬진 허종문





मनुष्य कुछ अपना नहीं कर सकता,।  
인간은 **무언가를** **소유할 수 없으며**,  
सभी स्वामित्व ईश्वर के हैं।  
모든 소유권은 **신에** 귀속되어 있다.

현대 사회에서의 지적 재산권

중세 봉건 사회에서의 분할 소유권

고대 로마 토지 소유권

모든 소유권은 신에게 귀속

사회 구성원의 인식변화와 사회 고도화를 통해 합의를 통해  
소유권 범위 변화가능성이 有

## 소유물(所有物) ?

민법 제 98조

---

(물건의 정의)

본법에서 물건이라 함은 유체물 및 전기 기타 관리할 수 있는 자연력을 말한다.





만약 행성이 소유물이 된다면,  
누가 이 소유권을 공증하여야 할까?



## 행성은 공동 유산이다

달과 기타 천체를 포함한 외기권의 탐색과 이용은 그들의 경제적 또는 과학적 발달의 정도에  
관계없이  
모든 국가의 이익을 위하여 수행되어야 하며 모든 인류의 활동 범위이어야 한다.  
- UN 우주 조약 제 1조 -



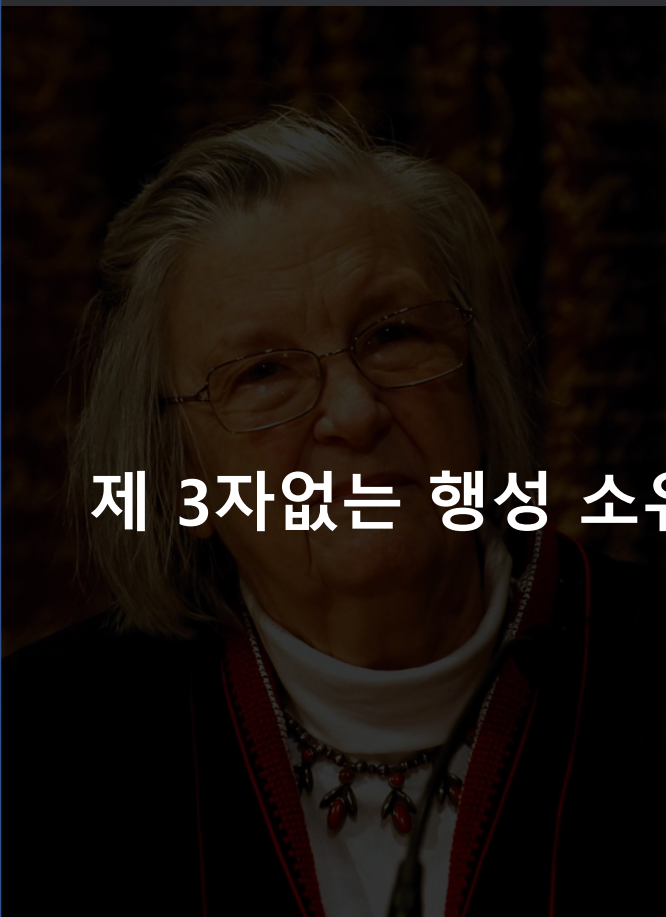


“공유 자원의 문제는 국가나 시장이 해결하는 것이 당연한 것이 아니라,  
제3의 방안으로 자체적으로  
해결할 수 있다”

-노벨 경제학 수상자 엘리너 오스트롬 [공유의 비극을 넘어] 중-

제 3의 방안?

공증을 해주는 주체가 없이  
지구인 전체 네트워크가 공증을  
한다면?



제 3자없는 행성 소유권 공증을 **블록체인의 탈중앙을 통해** 구현해보고자 했습니다.  
“공유 자원의 문제는 국가나 시장이 해결하는 것이 당연한 것이 아니라,  
해결할 수 있다”

-노벨 경제학 수상자 엘리너 오스트롬 [공유의 비극을 넘어] 중-

공동 유산인 만큼,  
지구인 참여 네트워크가 공증해줄 수 있다면?



행성 자원 소유 기대권 거래 플  
랫폼



---

**SPACE**

---

즐거, My Galaxy

---



Uni-Block

# 행성 자원 소유 기대권 ?

## 행성 자원 소유 기대권

~~아직 행성 자원과 관련하여 감독 기관이 없어서 행성의 소유권을 주장할 수 있는 권리~~

BUT



멀지 않은 미래에는 충분히 행성 자원 소유 기대권 거래 플랫폼



# 행성 자원 소유 기대권 거래 플랫폼으로서 과제

## ① 행성 소유 기대권의 투명성 및 영원한 입증



영원히 보존되는 기록되는  
영속성



위/변조의 불가능

## ② 행성의 소유물화 가속

- ✓ 개발 성공시 환급되는 이더리움의 동결 해제에 대한 갈망(보증금 개념)
- ✓ 토큰을 사게 되면 클라우드 펀딩처럼 우주개발업체의 개발자금으로 들어간다.

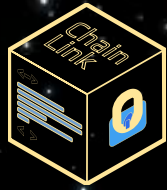
# 행성 소유 기대권 거래 플랫폼 거래 흐름도



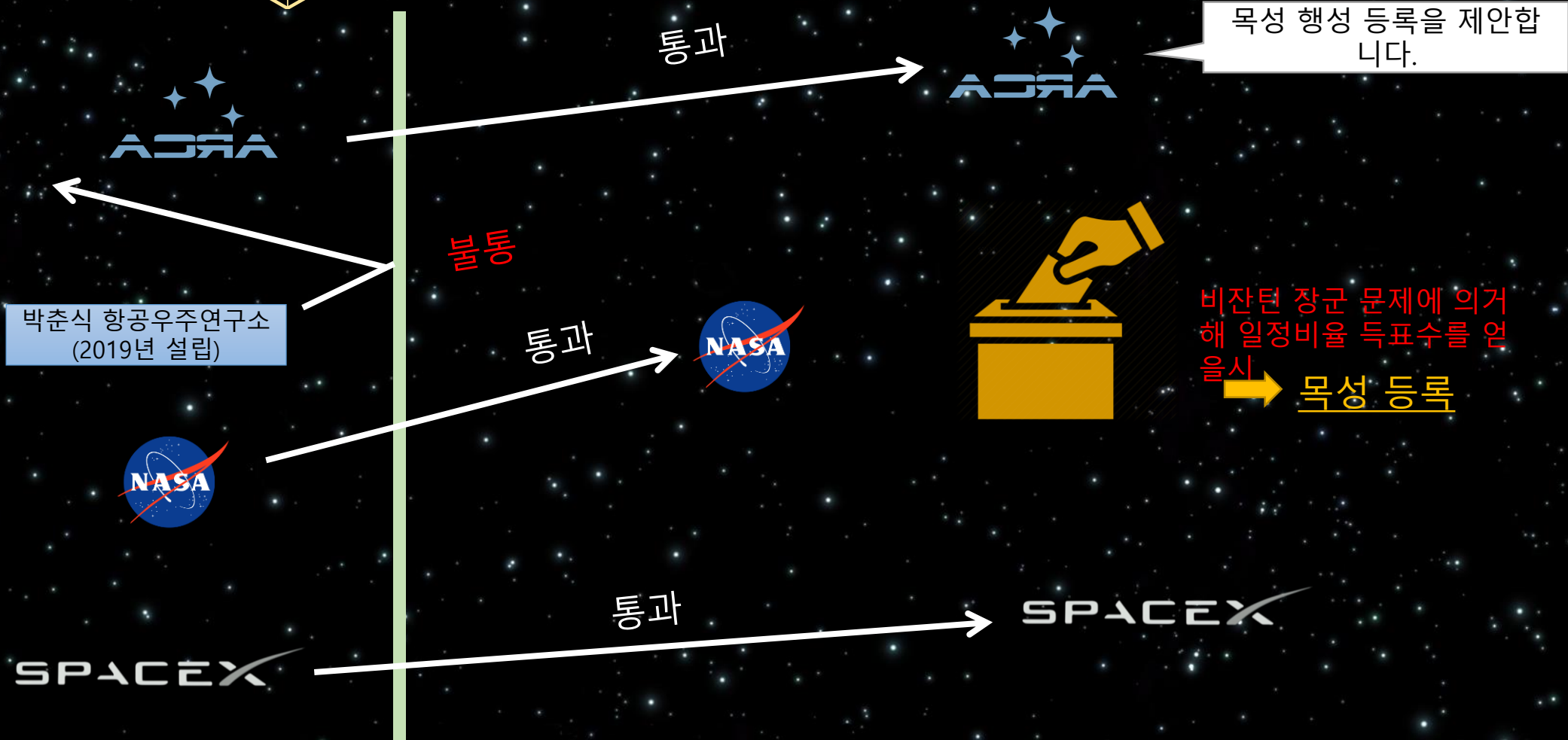


# 거버넌스 등록

우주 분야에서 전문성을 지닌 업체들을 통해 무분별한 행성 등록을 피하기 위함



거버넌스 등록:  
5년 이상 존속 여부



# 행성 개발권 경매

## 비공개 경매로 진행

거버넌스들이 이더리움으로 입찰표를 제시,  
최고입찰가를 제시한 거버넌스가 제시한 이더리움에 대한 개발권을 낙찰받음

## 투명한 경매 과정

모든 경매 과정은 블록체인에 기록  
분산시스템으로 네트워크상의 모두에게 공증

## 낙찰 기업은 크라우드 펀딩 기회 획득

낙찰된 기업은 민간 투자자들의 크라우드 펀딩을 독자적으로 받을 수 있는 권한을 얻습니다. -> 기업의 참여 및 개발 가속화 유인

입찰표

희망매수금:  
80 ETH



입찰표

희망매수금:  
90 ETH



입찰표

희망매수금:  
100 ETH

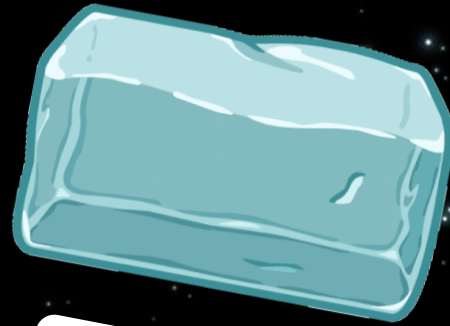


낙찰



# 행성 개발권 경매

동결된 이더리움은  
개발 성공시 동결이 해제됨

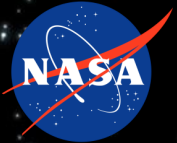


개발 완료에 대한 경제적인 유인  
제공  
행성 자원 개발 가속화



# 행성 토큰 클라우드 펀딩 절차

개발권을 낙찰받은 기관에 투자할 용의가 있는 개인 투자자들은 이더리움을 투자, 해당 행성 토큰을 배당금으로 취득



목성개발권을 낙찰받았습니다. 목성 자원에 투자하실 분들 계신가요?

스마트 컨트랙트

!@#%&\*(^&^  
%\$#@#%&^&^  
%\$#@#%&^&^  
%\$%&^&^&^&^  
\$#

오 괜찮은데?



나도



그럼 나도



개발 성공시 해당기관은 동결된 이더리움을 돌려받으며,  
개발 성공여부는 투자자들의 투표를 통해 결정되며, 투표는 보유한 토큰으로 진행됨



# 개발범위

---

- 투표 시스템
- 경매 시스템
- ERC20 토큰 생성 (galaxy 생성)
- 사용자 안드로이드 프론트
- 삼성 플랫폼 SDK
- Solidity

# 거버넌스 등록

---

```
{  
  "co" : "1 zo",  
  "coInfo" : "handsome",  
}
```



1 zo

```
function registration(string memory coName, string memory coInfo) public {  
  Models.Company memory company = Models.Company(coName, coInfo);  
  _candidates[msg.sender] = CandidateInfo(false,  
    true,  
    coName,  
    coInfo,  
    new CandidateVote(company, address(this)));  
}
```

```
{  
  "co" : "uniblock",  
  "coInfo" : "awful",  
}
```



uniblock





```
if(bullet){  
    if(_candidates[newCo].candidateVote.agree(_companys.sizeOf())){  
        toBeGovernance(newCo);  
        emit voteToCompanyCall(voteResult.SUCCESS);  
    }  
}  
else{  
    if(_candidates[newCo].candidateVote.disagree(_companys.sizeOf())){  
        emit voteToCompanyCall(voteResult.SUCCESS);  
    }  
}
```



uniblock

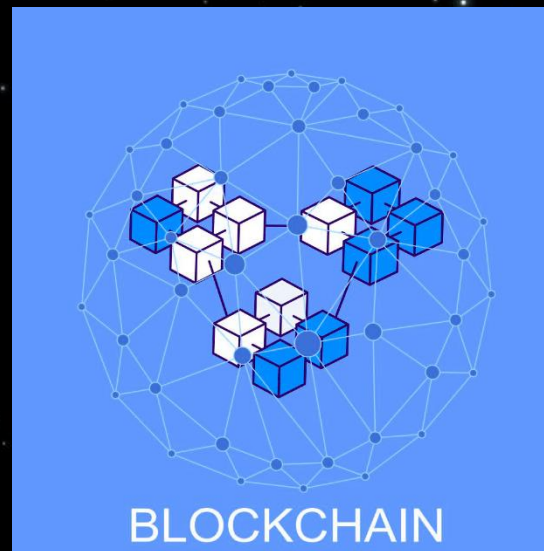
```
function toBeGovernance(address newCo) internal{  
    _companys.push(_coKey, true);  
    _coAddr[_coKey] = newCo;  
    _coKeyMap[newCo] = _coKey;  
    _coKey++;  
}
```

# 행성 투표

---

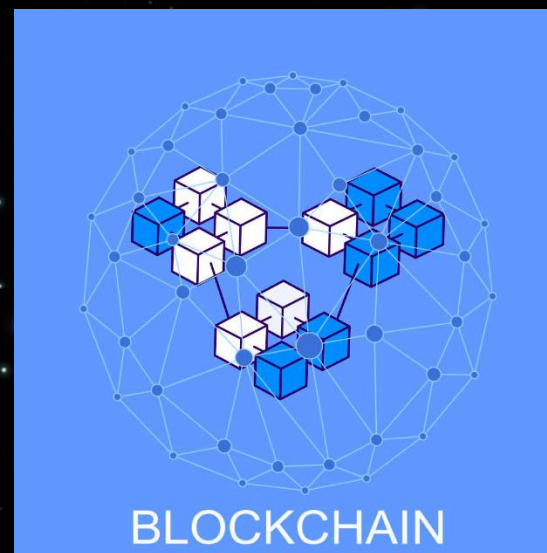
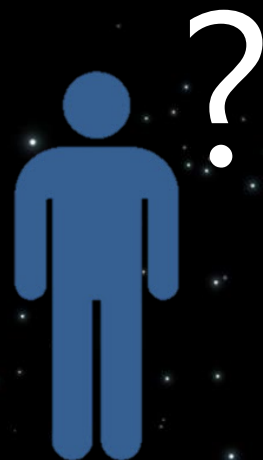






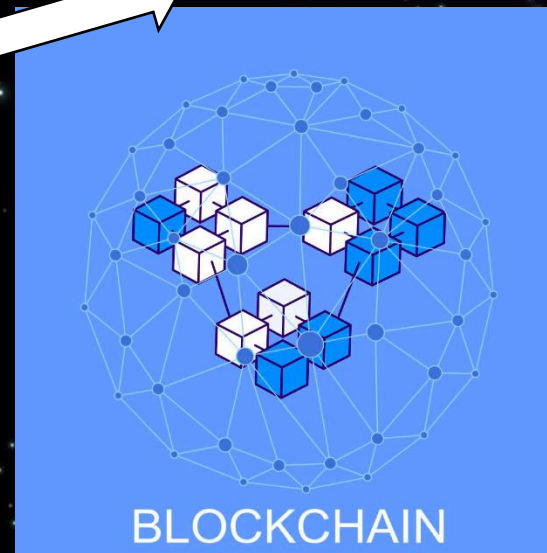


$H(k + \text{diamond})$





$$H(k + \text{diamond})$$







# 클라우드 펀딩

---



행성의 소유 기대권 == ERC20 컨트랙트





## Galaxy 컨트랙트에 이더를 동결시켜 놓는다

```
function genGalaxy(string memory name, string memory symbol, uint256 value) payable public {
    require(
        msg.sender == _highestBidder &&
        address(_galaxy) == address(0) &&
        msg.value == _highestAmount &&
        _proofBiddingDeadline > now,
        "not permissioned"
    );
    Galaxy g = new Galaxy(name, symbol, msg.sender, address(this));
    g.setDeposit.value(_highestAmount)();
    _galaxy = g;
}
```

Galaxy 컨트랙트는 ERC20 표준을 준수하는 컨트랙트

```
function buyToken() payable public returns(bool){  
    _owner.transfer(msg.value);  
    uint256 ETH = (10 ** 18);  
    uint256 zeroDotOne = ETH / 10;  
    _mint(msg.sender, msg.value/zeroDotOne);  
}
```

테스트용 코드.

이더에 따라 일정 비율로 토큰을 생성해 준다.



# 시연 영상

---

# 수익 모델

모든 거래 발생 시

수수료를 통한 이익 창출