



2020-1 블록체인인의 실무 응용1

# 블록 체인을 이용한 중고거래 내 보상형 리뷰 시스템

김수환 김종웅 윤성원 윤여환 장원철 정한나



1

**중고시장의 성장과 부작용**

2

**신뢰도에 대한 문제의식**

3

**보상형 리뷰 시스템의 제안**

“

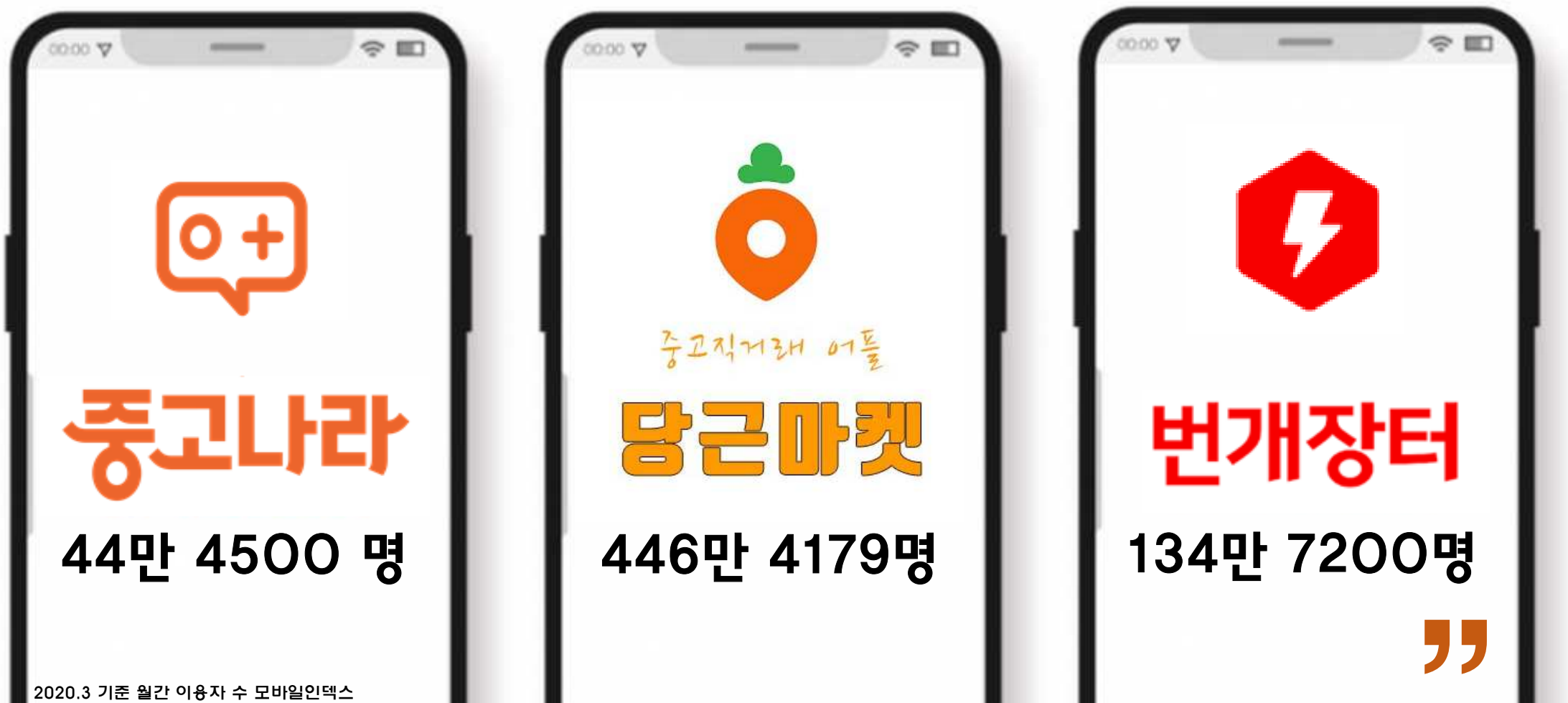
## 중고거래란?

중고거래는 **중고품을 사고 파는 행위**를 전반적으로 지칭하며 개인간의 거래, 단체간의 거래 등 다양한 형태로 존재

보통 감가상각을 깊게 고려하지 않는 **개인대 개인의 거래**를 지칭한다

”

# “ 중고거래 시장

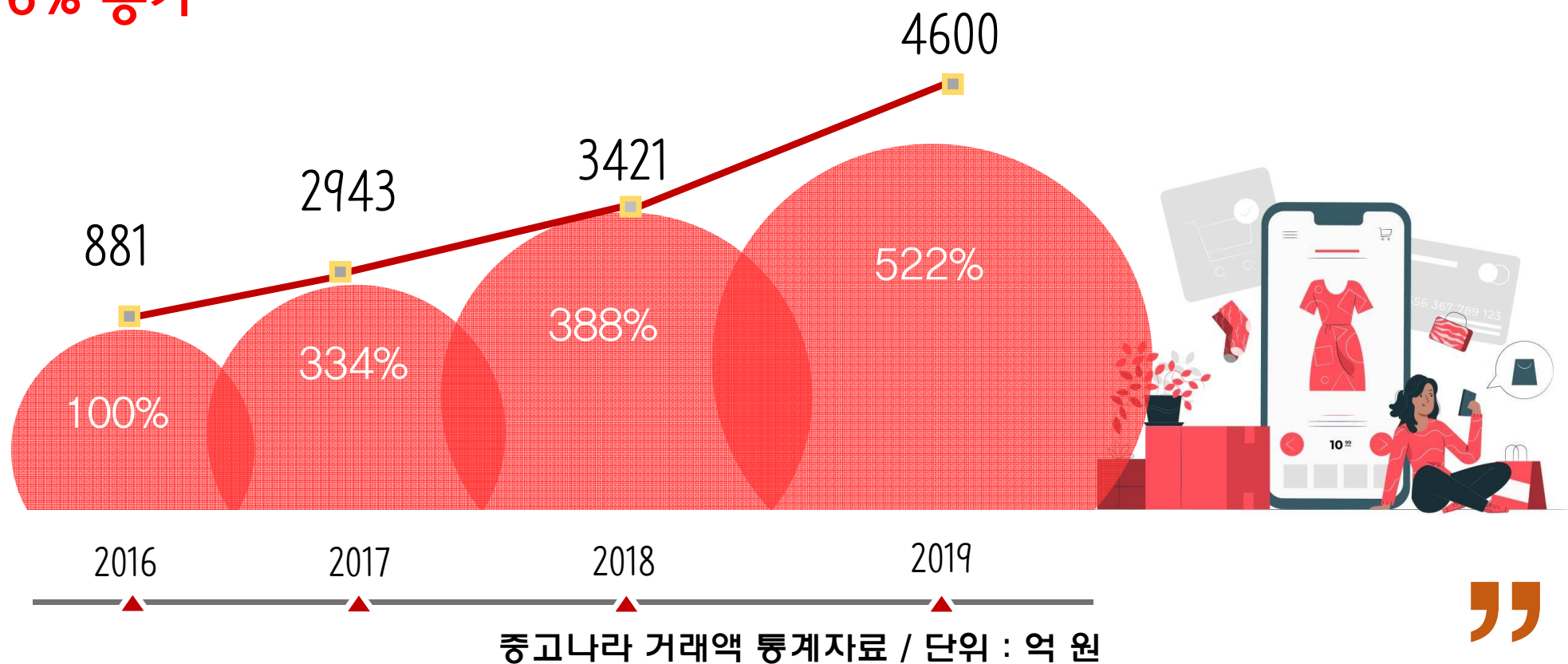


2020.3 기준 월간 이용자 수 모바일인덱스

“

# 중고시장의 성장

2020년 올해 3월 전체 중고거래 앱 사용자는 492만 명, **지난해 1월 대비 76% 증가**

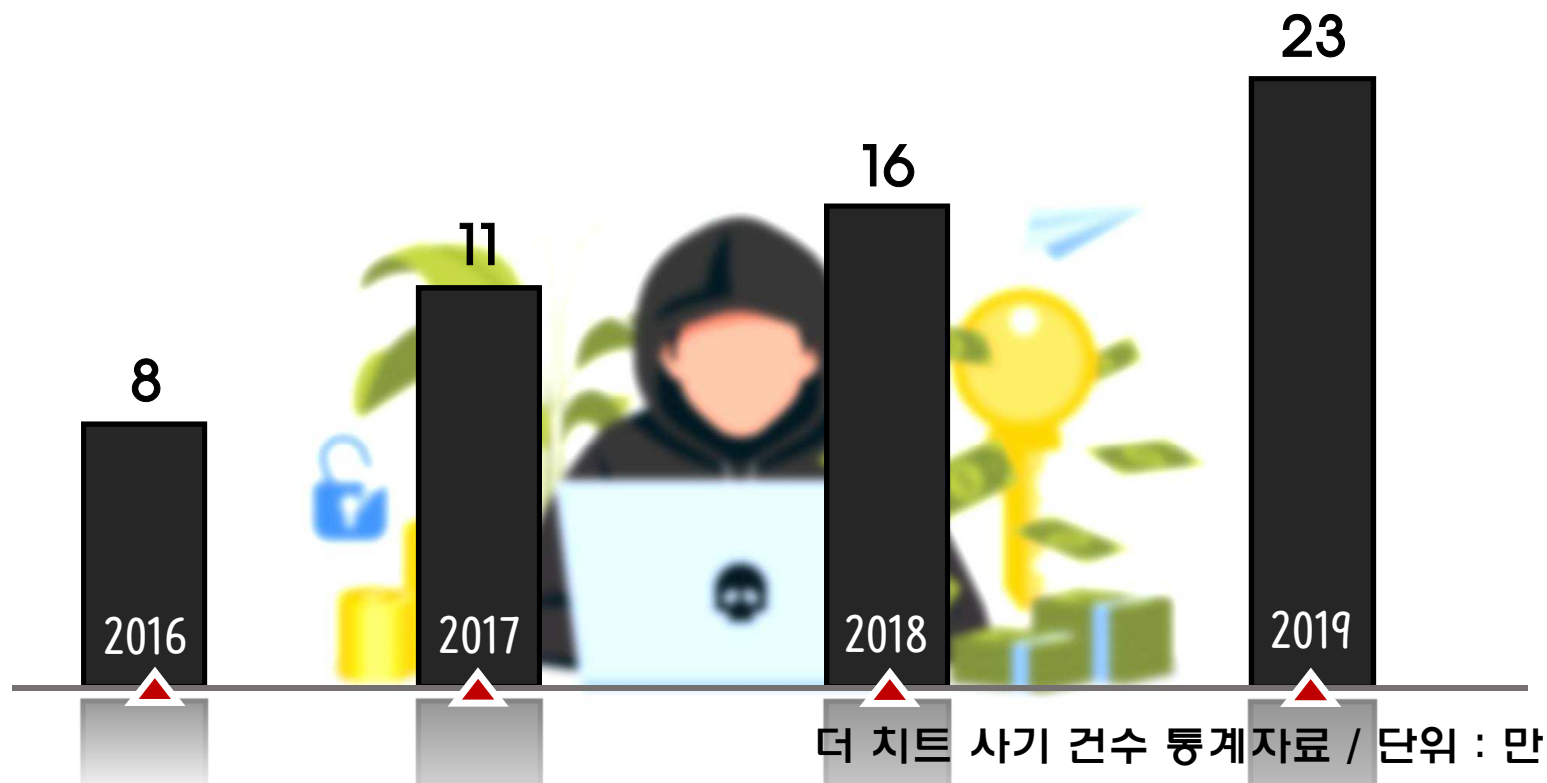


”

“

# 사기피해의 증가

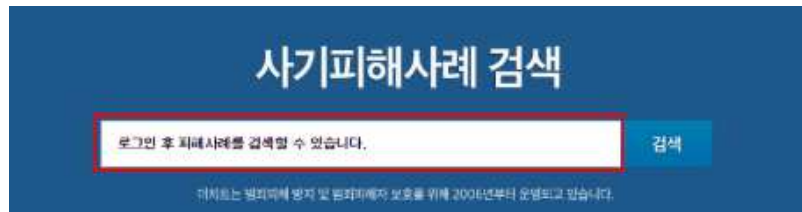
2006년 부터 2015년까지의 누적 사기보다 최근 **3년간의 건수가 2배** 이상 정도로 급속도로 증가



”

“

# 사기 방지?



## 더 치트 사기 조회

더 치트 조회를 통해 판매자의 전화  
번호, 통장의 사기 이력 조회

NAVER

안전쇼핑



## 안전 거래 시스템

안전 거래 시스템을 이용 계좌 사  
기 방지 및 원활한 환불 보장

”

“

# 사기 방지?



**THECHEAT** 

**NAVER**  
안전쇼핑 

사기피해사례 검색

로그인 후 피해사례를 검색할 수 있습니다. **검색**

이치트는 범죄피해 방지 및 범죄피해자 보호를 위해 2008년부터 운영되고 있습니다.

**더 치트 사기 조** **거래 시스템**

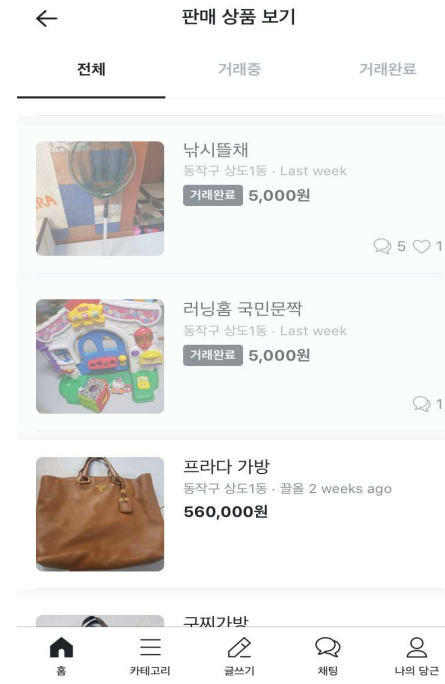
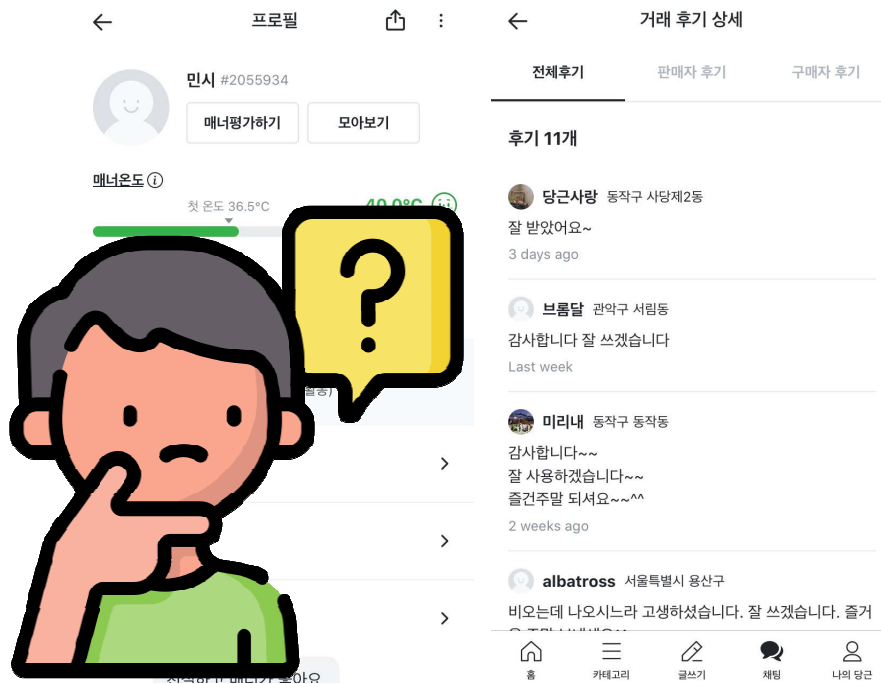
더 치트 조회를 통해 편 **시스템을 이용 계좌 사**  
번호, 통장의 사기 **원활한 환불 보장**

”

“

# 신뢰의 문제

## 이 판매자를 과연 신뢰할 수 있는가?



”

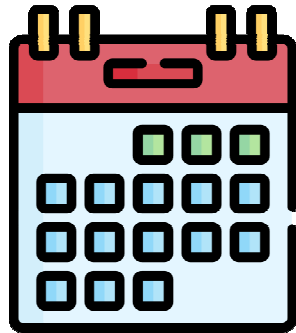
“

# 신뢰의 문제

이 판매자를 과연 신뢰할 수 있는가?



활동 지역



가입 일자



판매 기록

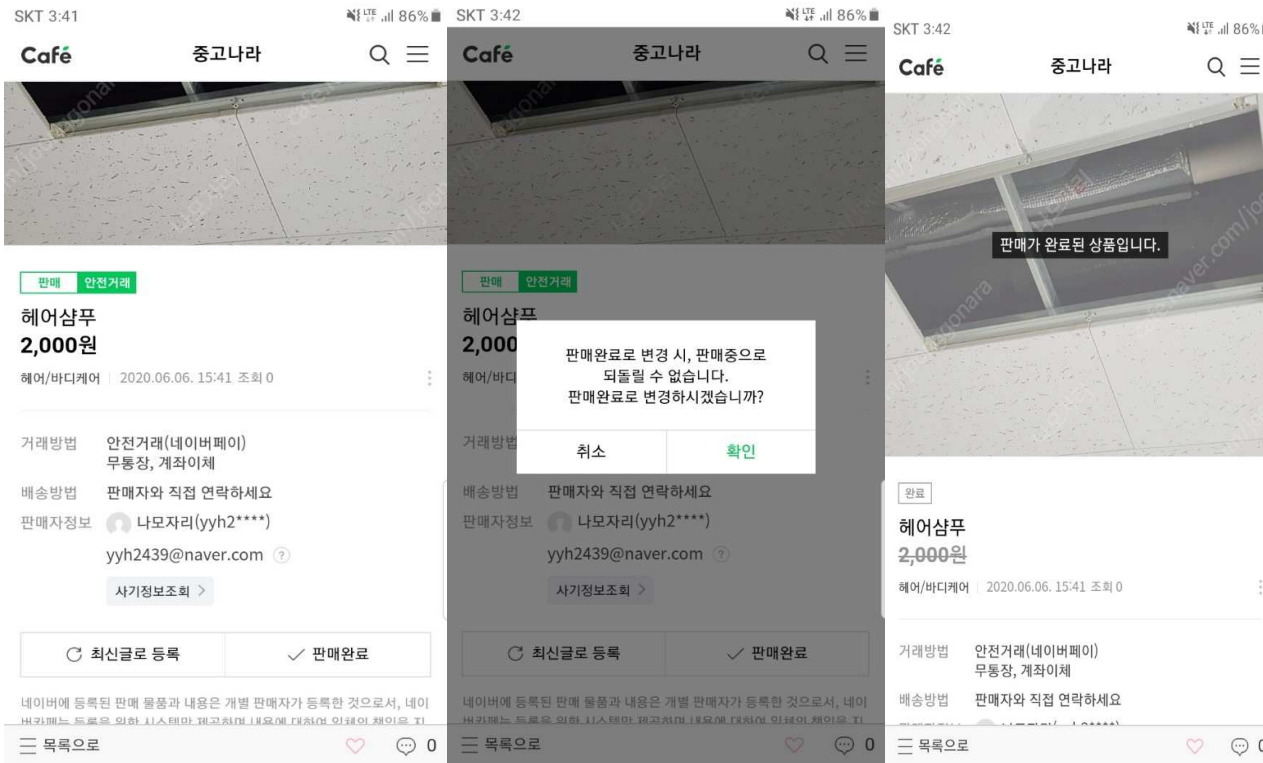


거래 후기

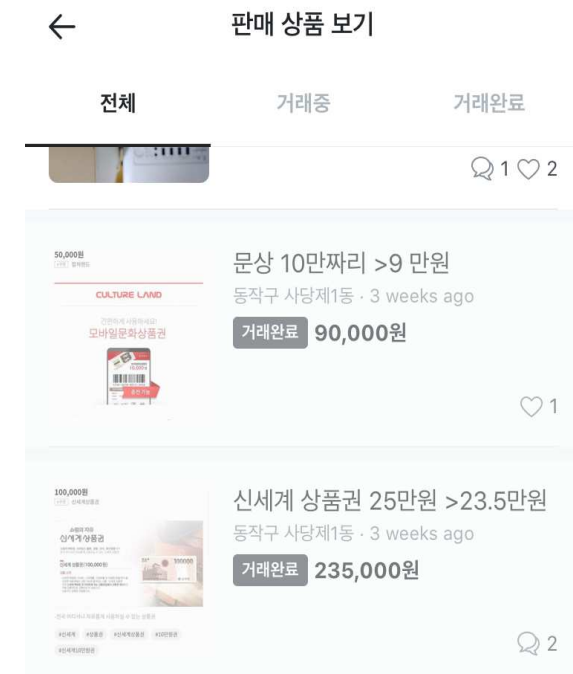
”

“

# 문제의식-판매 기록



\* 중고나라 - 판매 게시판에 가격과 제목명, 무관한 사진만을 올려도 자체 판매가 되는 과정



구매자 없이 자체 판매 완료 가능 문제

정상 판매자 행세

“

# 문제의식-거래 후기



<
거래 후기 상세

전체 후기
판매자 후기
구매자 후기

후기 2개

codi
서울특별시 송파구

멀리서 오셨는데 고생 하셨습니다.  
잘쓰세요~~

3 days ago

이니마니또
사당동

테스트 후 잘 샀습니다

3 weeks ago

후기 작성률 약 17%

거래에 비해 매우 적은 거래 후기

“

# 아이디어



”

“

## 워크 플로우

---

거래발생



상호평가 및  
스마트 컨트랙트



판매자 및  
구매자의 신뢰도  
지표 반영

”

# “ 워크 플로우

거래발생

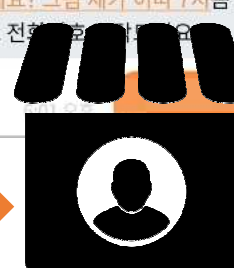


상호평가 및  
스마트 컨트랙트

Buyer



Seller



거래 발생



”

# “ 워크 플로우



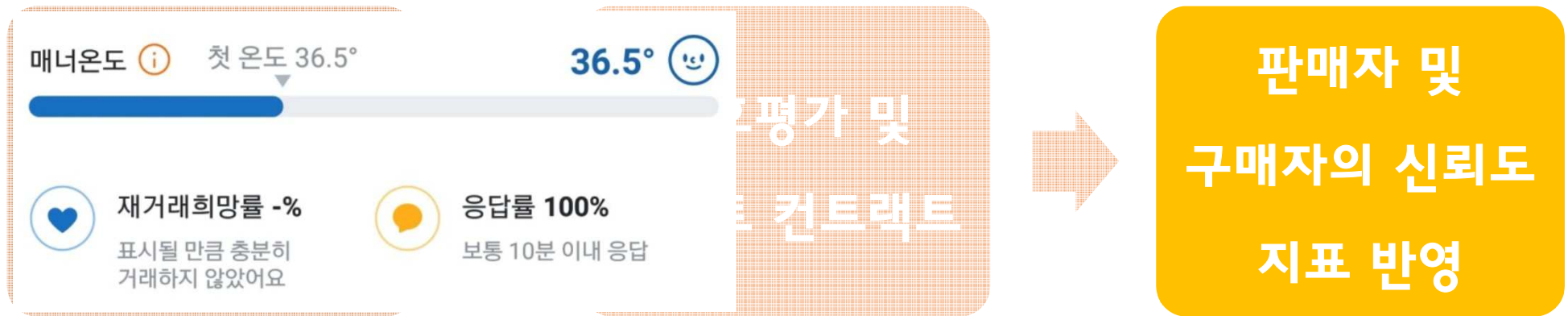
## Smart contract

- 평점 점수가 smart contract 이용해 전달
- Contract 가 일어남에 따라 거래 횟수 또한 증가



“

# 워크 플로우



—장기휴면계정 선별 및 다중 계정간 거래를 통해 신뢰도 지표 반영

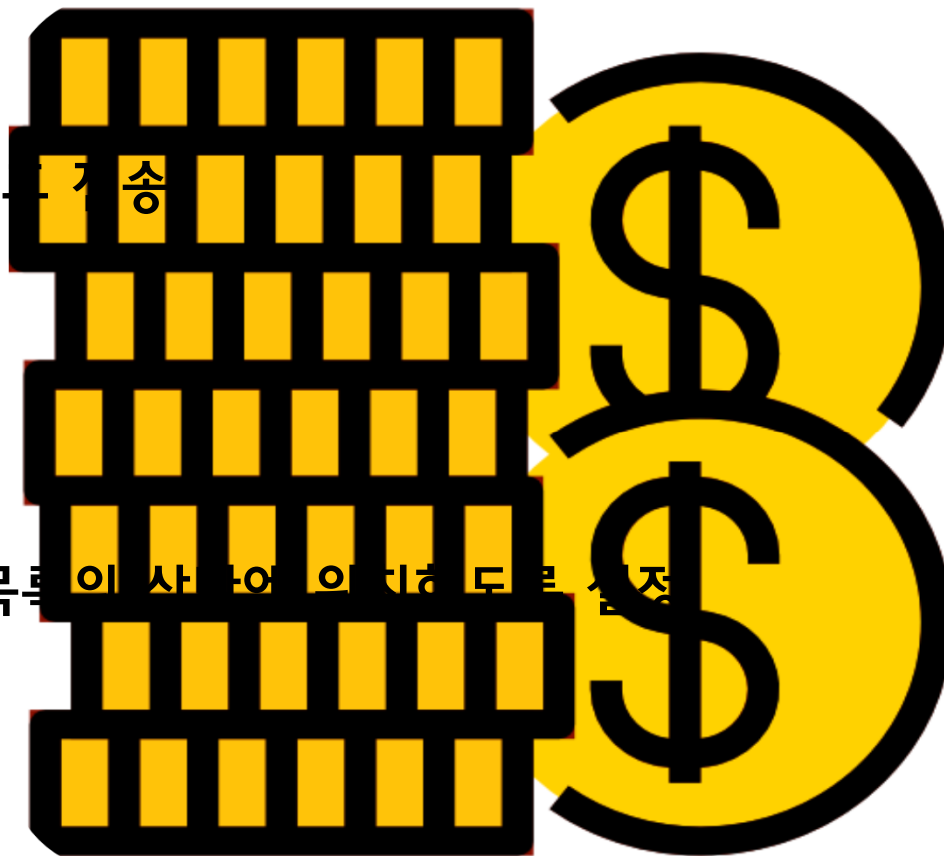
”

“

# 리뷰의 보상

## SCORE

- 1-10 사이의 점수가 평가의 보상으로 상품 제공
- 받은 점수의 양은 신뢰도의 지표의 역할
- 블록마켓에서의 보상
  - 신뢰도가 높은 자의 상품이 상품목록에 상위에 위치하도록 설정



”

“

# 스마트 컨트랙트

```
1 pragma solidity ^0.4.22;
2
3 contract Second_hand {
4     address public minter;
5
6     mapping (address => uint256) public scoreForUser;
7     mapping (address => uint256) public NumOfFraud;
8
9     event Evaluate(address indexed _from, address indexed _to, uint amount);
10
11     constructor() public {
12         minter = msg.sender;
13     }
14
15     // Scores of fraud users would be reset to 0
16     // Can be only called by the contract creator
17     function reset_scores(address fraud) public{
18         require(msg.sender == minter);
19         scoreForUser[fraud] = 0;
20         NumOfFraud[fraud] ++;
21     }
22
23     // The scoring function is only accessible from other scorers
24     // The scoring range is from 0 to 10
25     function scoring(address _from, address _to, uint score) public {
26         require(_from==msg.sender);
27         require(0 < score && score < 10);
28         scoreForUser[_to] += score;
29         emit Evaluate(_from, _to, score);
30     }
31 }
32
33 }
```

유저들의 점수 및 거래 회수를 전역 변수로 관리 :

점수 : **scoreForUser**

거래 횟수 : **NumOfTrade**

**scoring** 함수 : 중고거래 후

- 평가 시 별점을 주는 사람에 대한 신원 확인을 통해 자신이 아닌 남이 평가할 수 있도록 require문 이용
- 0~10 사이의 점수를 부여하며
- 거래 횟수 1회 증가
- Evaluate 이벤트를 emit 하여 log 관리

”

“

# 스마트 컨트랙트

```
1 pragma solidity ^0.4.22;
2
3 contract Second_hand {
4     address public minter;
5
6     mapping (address => uint256) public scoreForUser;
7     mapping (address => uint256) public NumOffFraud;
8
9     event Evaluate(address indexed _from, address indexed _to, uint amount);
10
11     constructor() public {
12         minter = msg.sender;
13     }
14
15     // Scores of fraud users would be reset to 0
16     // Can be only called by the contract creator
17     function reset_scores(address fraud) public{
18         require(msg.sender == minter);
19         scoreForUser[fraud] = 0;
20         NumOffFraud[fraud] ++;
21     }
22
23     // The scoring function is only accessible from other scorers
24     // The scoring range is from 0 to 10
25     function scoring(address _from, address _to, uint score) public {
26         require(_from==msg.sender);
27         require(0 < score && score < 10);
28         scoreForUser[_to] += score;
29         emit Evaluate(_from, _to, score);
30     }
31 }
32
33 }
```

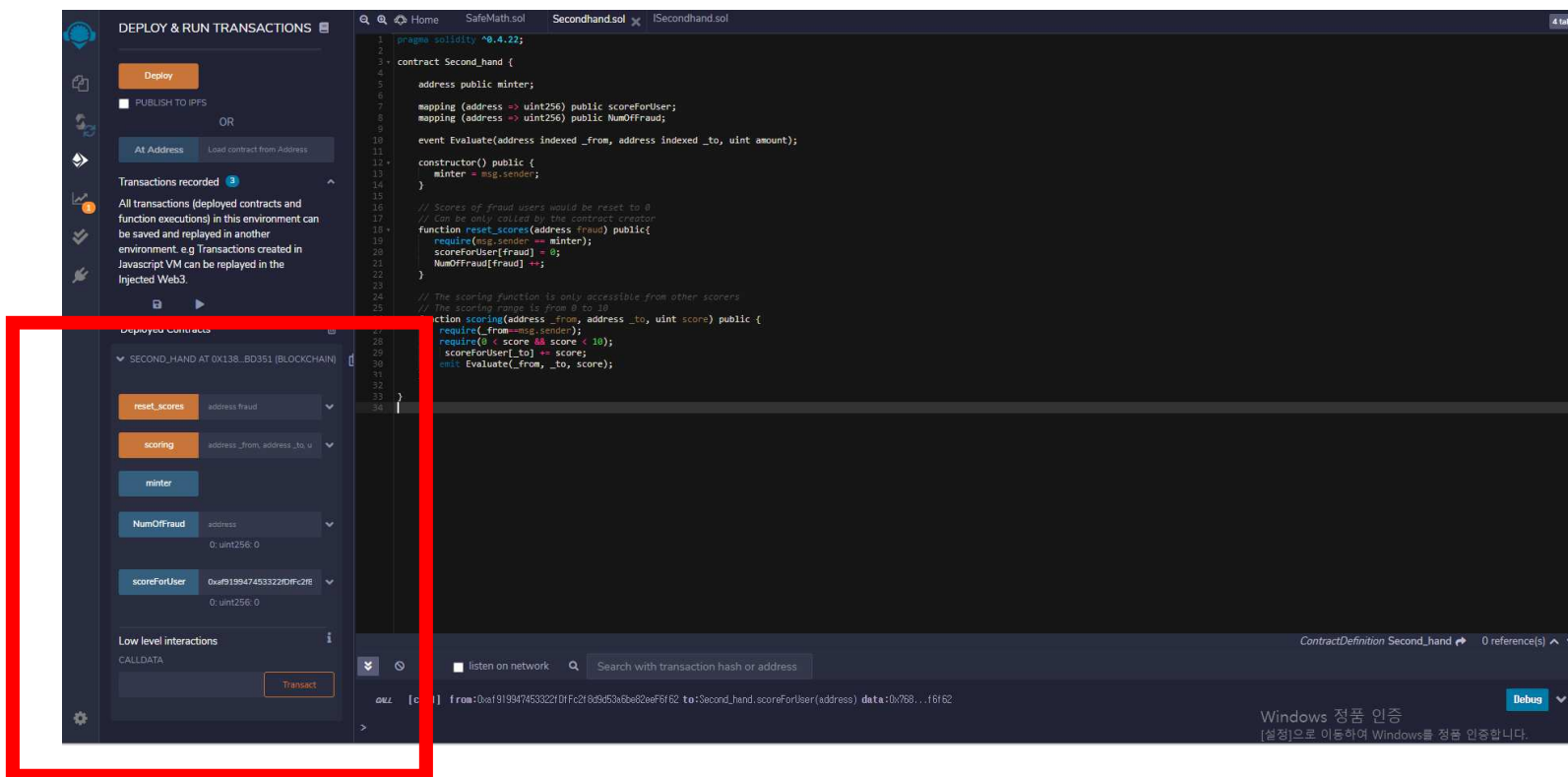
Secondhand 에 대한  
ERC20 interface의 code

```
1 pragma solidity ^0.4.24;
2
3
4 /**
5  * @title ERC20 interface
6  * @dev see https://github.com/ethereum/EIPs/issues/20
7  */
8 interface ISecondhand {
9
10     function scoring(address _from, address _to, uint score)
11         external returns (bool);
12
13     function reset_scores(address fraud)
14         external returns (bool);
15
16
17     event Evaluate(
18         address indexed _from,
19         address indexed _to,
20         uint amount
21     );
22 }
```

”

“

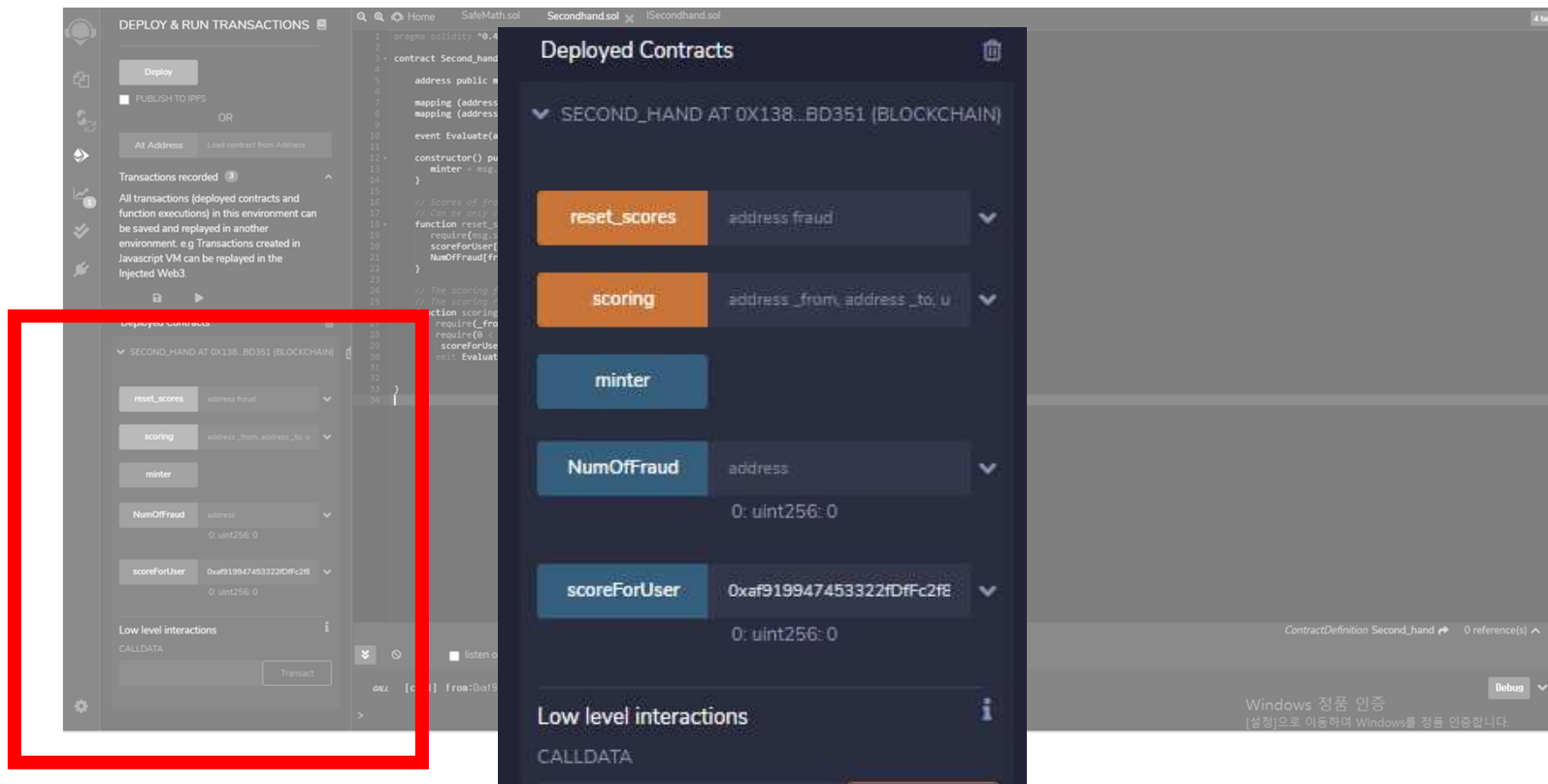
# 스마트 컨트랙트



”

“

# 스마트 컨트랙트



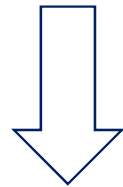
”

“

# 스마트 컨트랙트

The screenshot shows a web interface for a smart contract. At the top, there's a dropdown menu set to 'SECOND\_HAND AT 0X138...BD351 (BLOCKCHAIN)'. Below it, there are two buttons: 'reset\_scores' and 'address fraud'. A red box highlights the 'scoring' section, which contains a form with three fields: '\_from:' with the value '0xaf919947453322fDfFc2f8d9d53a6be82eeF6f62', '\_to:' with '0x88bcD55E9DB4d4b2449eE', and 'score:' with '7'. There is a 'transact' button next to the 'score' field. Below the 'scoring' section, there's a 'minter' button. Then, there's a 'NumOffFraud' section with an 'address' dropdown and a value '0: uint256: 0'. Below that, there's a 'scoreForUser' section with an 'address' dropdown and a value '0x88bcD55E9DB4d4b2449eE', and a 'call' button. A red box highlights the bottom of the interface, showing the value '0: uint256: 7'.

지갑 1 : 0xaf919947453322fDfFc2f8d9d53a6be82eeF6f62



scoring ( 지갑 1, 지갑 2, 7 )  
7점 이 실제로 부여되는지 확인

지갑 2 : 0x88bcD55E9DB4d4b2449eEE4404cf7813c519eb89

지갑 2 의 scoreForUser의 값이 7 만큼 증가한 것을 확인

”

“

# 스마트 컨트랙트

SECOND\_HAND AT 0x138...BD351 (BLOCKCHAIN)

reset\_scores address fraud

**scoring**

\_from: "0xaf919947453322fdffc2f8d"

\_to: "0x88bcD55E9DB4d4b2449eE"

score: "11"

transact

scoring ( 지갑 1, 지갑 2, 11 )

0~10점 범위를 벗어나는 점수의 경우 transaction fail

Transaction Details

PASSIVE INCOME FROM YOUR CRYPTO  
WITH UNIQUE ALGORITHM RJV<sub>12</sub>

Overview State Changes

[ This is a Ropsten Testnet transaction only ]

Transaction Hash: 0x1b4ca32bdb3b5a881d379b6cb78ec3fff3c52372bf6840352c7e1f067016f49f

Status: Fail

Block: 8097055 1 Block Confirmation

Timestamp: 49 secs ago (Jun-15-2020 11:10:03 AM +UTC)

From: 0xaf919947453322fdffc2f8d9d53a6be82eef6f62

To: Contract 0x1385a720a61490b7fa41f47c9d0c0cf5efbbd351 Warning! Error encountered during contract execution (Reverted)

Value: 0 Ether (\$0.00)

Transaction Fee: 0.000044626 Ether (\$0.000000)

Click to see More

”

“

---

Q & A

”