

스마트컨트랙트를 활용한 투명하고 신뢰받는 주유소 만들기

(II - B 분야)

이현우, 임지훈, 최승주

목차

1. 현존하는 주유소의 문제점
2. 시스템 제안
3. 작동 매커니즘
4. 프로토타입

현존하는 주유소의 문제점

가짜 석유 판매:

서울시 구로구의 D주유소는 올해 2월 서울시 관악구 서울대학교 정문 앞 GS건설공사 현장에서 자동차용 경유에 등유 등을 30% 섞은 가짜 석유 제품을 판매한 행위로 6개월간의 사업 정지 행정 처분을 받았다.

주유시 정량을 속이는 문제:

경기도 고양시의 D주유소는 휘발유와 경유를 각각 리터당 250ml, 230ml씩 적게 판매해 2개월 간의 사업 정지 처분을 받았다.

16년 기준, 가짜 석유를 판매하거나 정량을 속여 석유를 판매해 적발된 주유소가 191곳인 것으로 조사됐다. 15년도 기준(146곳 적발) 보다 30% 더 많은 규모다.

정유사의 품질 관리 프로세스



품질 서비스 센터

석유제품 기술지원 및 고객고충처리 등 석유시장 품질관리 서비스를 현장에서 신속히 제공할 수 있도록 전국 6개권역에서 각종 지원 업무를 수행중, 주요업무는 품질고충을 예방하기 위한 사전 품질점검 활동과 발생된 품질고충을 신속히 처리하기 위한 활동

품질 고충 예방활동

- * 주유소 품질 Monitoring 활동 (유사석유제품 유통 방지, 혼유 예방 및 수분오염 방지 등)
- * 송유관 수하 제품을 포함하여 물류센터 입고 및 출하 제품 검사
- * 수요처별 사전 기술지원 및 기술지원 결과 D/B구축을 통한 철저한 사후 관리로 품질관련 사고 재발방지 유도
- * 판매원 제품지식 교육

품질 고충 처리활동

- * 당사 제품에 대한 품질고충 발생 시 품질

석유제품 품질 규격

법규	주무 기관	내용	점검 기관	관리
석유 및 석유 대체연료 사업법	산업통상자원부	석유제품 (휘발유, 등/경유,B-C,아스팔트) 액화석유가스(LPG)	한국석유관리원	정유사 월 1회 점검 주유소 수시점검
대기환경보전법	환경부	석유대체연료(바이오디젤)에 대한 법적 품질관리 기준 및 관리규정, 처벌 기준 등 명시	각 지방 환경유역청	각 권역별 수시 점검

석유제품 품질관리 과정

원유 구매단계, 제품 생산단계, 제품 유통단계에서 모두 진행된다. 구매단계에서는 원유마다 다른 특성을 분석하여 석유제품 가이드라인을 만들고, 생산단계에서는 지속적인 모니터링과 품질관리 항목을 점검하여 법적 규격을 충족시킨다. 최종적으로 규격을 통과한 제품만이 공장에서 전국 물류센터로 이동하여 최종 소비자로 보내진다. 제품 수송 시 발생할 수 있는 품질저하 및 석유 혼합 사고를 막기 위해 유통단계에서도 철저한 품질관리가 이루어진다.

주유소 품질관리

- * 주유소 수시 점검
- * 식별제를 첨가해 불법석유 유통 방지
- * 휘발유 증기압, 경유 유동점 등 계절에 따라 규격이 변동될 때 법규 위반여부 확인
- * 품질 고충 발생시 주유소와 차량연료 분석을 통해 정확한 원인을 파악후 신속 대응

환경품질등급제

주유소에 기름이 공급되기까지의 검증 프로세스는 충분
주유소에 기름이 **공급되고 난 후의 검증 프로세스는 미흡**
또한 품질고충이 일어났을 때는 이미 피해를 본 이후임.

WHY BLOCKCHAIN?

기름의 유통 과정 및 유통 후 유량의 변동 과정을 기록

보통 주유소에서 불법행위를 저지를 때는 주유기를 조작하여 정량을 속이거나 기름 탱크에 불순물을 섞거나 가열해서 부피를 늘리는 등의 수법을 사용하고 있다.

위변조를 하기 어렵고 제한적

일반적인 DB와는 다르게 블록체인은 생성되었던 데이터를 삭제하거나 바꾸는 행위는 매우 어렵다. 데이터를 관리할 수 있는 권한이 일반 DB보다는 훨씬 제한적이다.

정유사에 대한 신뢰부족

식별이 어려운 가짜 경유가 대규모로 만들어져 공급이 가능했던 것은 가짜경유 원료를 제조 및 유통시킨 석유정제 회사와 이윤을 추구한 채 원재료를 공급한 국내 대형 정유회사의 부도덕함이 한 몫을 했기 때문으로 검찰 수사 결과 드러났다.



시스템 제안

시스템 목적

- * 주유소 내에서 기름의 성분 변조가 불가능한 시스템
- * 주유소 내에서 주유량의 변조가 불가능한 시스템

스마트컨트랙트에 반영되는 데이터

기름 탱크의 성분 데이터

기름에 불순물이 섞여 있는지를 판별해주는 용도

기름 탱크의 기름 잔량 데이터

주유기에서 빠져나간 기름양과 탱크 내 남은 잔량을 비교해주는 용도

주유기에서 빠져나간 주유량 데이터

주유기에서 빠져나간 기름양과 탱크 내 남은 잔량을 비교해주는 용도

시스템 구성

블록체인 형태	프라이빗 블록체인 (Private Blockchain)
합의 알고리즘	권한 증명 (Proof of Authority)
블록체인 코어	이더리움 (go-ethereum)
제공하는 서비스	블록 탐색기



검증자 노드

트랜잭션의 검증 및 블록 생성



주유소 노드

기름탱크의 성분 및 잔량 전송



Rpi3:

이더리움 네트워크에 트랜잭션만 발생시킬 수 있는 스펙을 가진 노드

Tank:

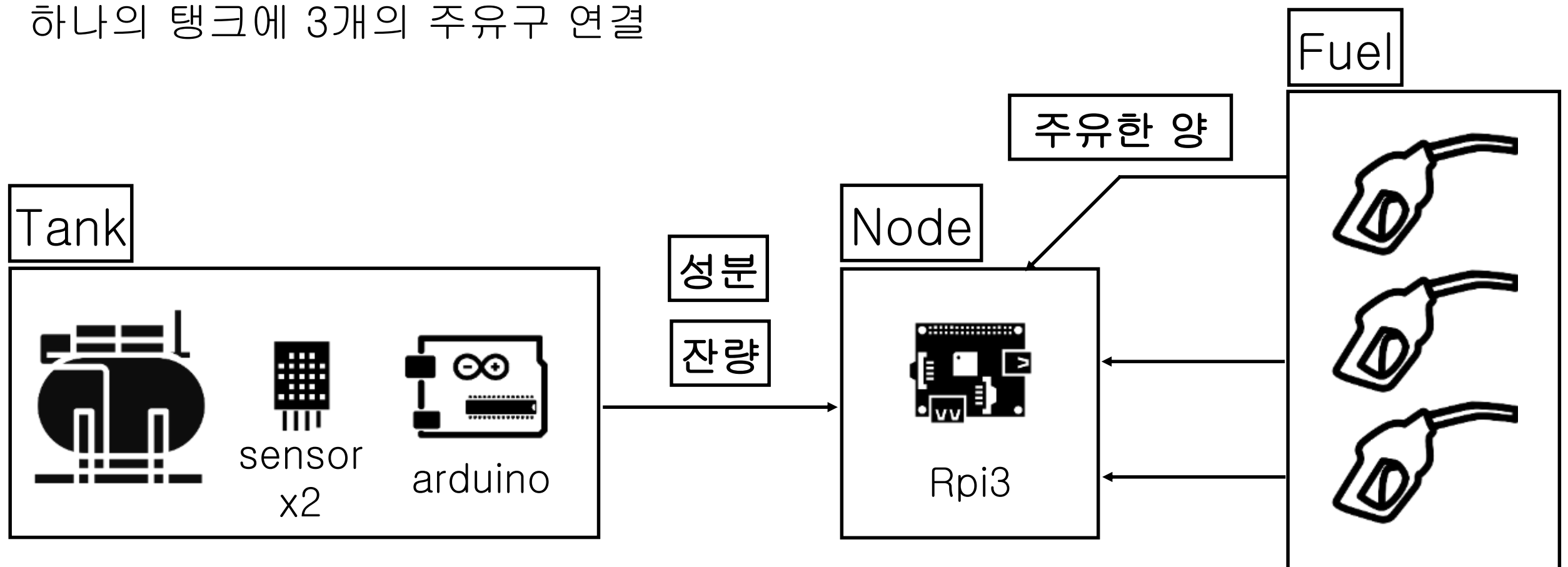
유류를 담는 탱크는 수위센서, 식별 센서, 측정 정보를 Rpi3로 전송하는 아두이노가 부착

Fuel:

하나의 탱크에 3개의 주유구 연결



주유소 노드





검증자 노드

Availability Zone A

부트노드 A

검증자 V1

검증자 V2

Availability Zone B

부트노드 B

검증자 V1

검증자 V2

네트워크 유지를 위해 필요한 최소 검증 노드 수: $\text{int}(N / 2) + 1$

N: 현재 작동중인 검증 노드 수, 현재 구성에서 $N = 2$

최소 검증 노드 수: $\text{int}(2 / 2) + 1 = 2$

하나의 Zone이 다운되었을 때 존속가능 여부: **가능**



검증자 노드

Availability Zone A

부트노드 A

검증자 V1

검증자 V2

Availability Zone B

부트노드 B

검증자 V3

검증자 V4

네트워크 유지를 위해 필요한 최소 검증 노드 수: $\text{int}(N / 2) + 1$

N: 현재 작동중인 검증 노드 수, 현재 구성에서 $N = 4$

최소 검증 노드 수: $\text{int}(2 / 2) + 1 = 3$

하나의 Zone이 다운되었을 때 존속가능 여부: **불가능**

작동 매커니즘

정상적인 루틴

1. 빈 탱크에 정유사로부터 경유를 공급받아 탱크를 채운다.



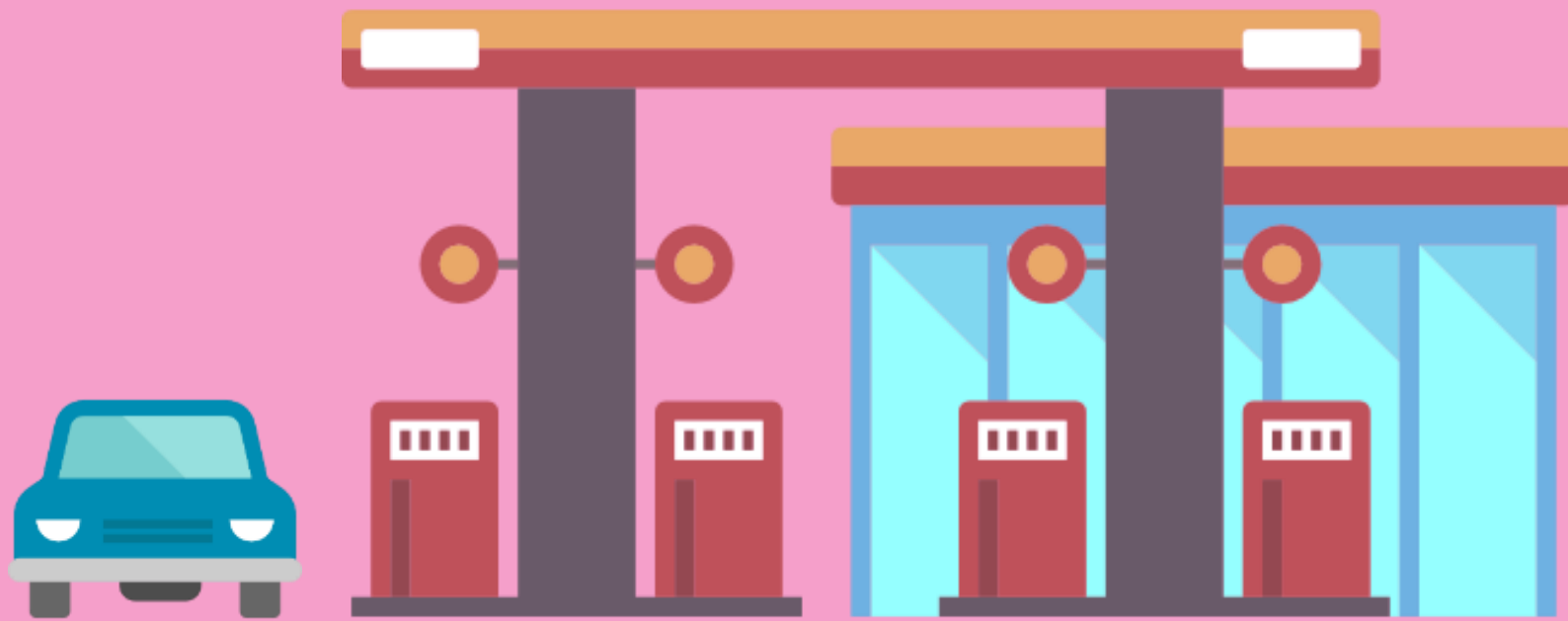
정상적인 루틴

2. 정상 영업을 시작한다.



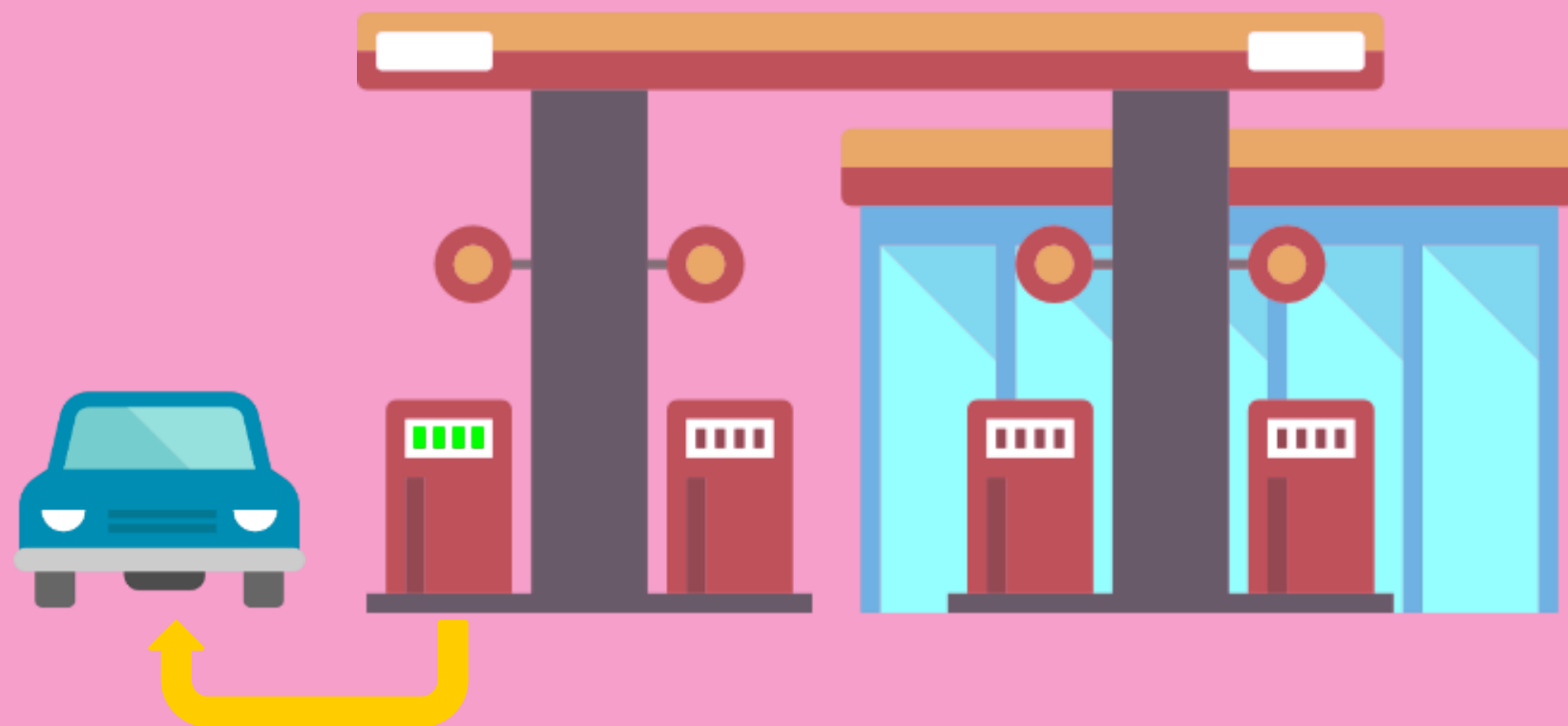
정상적인 루틴

3. 주유가 필요한 차량이 주유소에 들어온다.



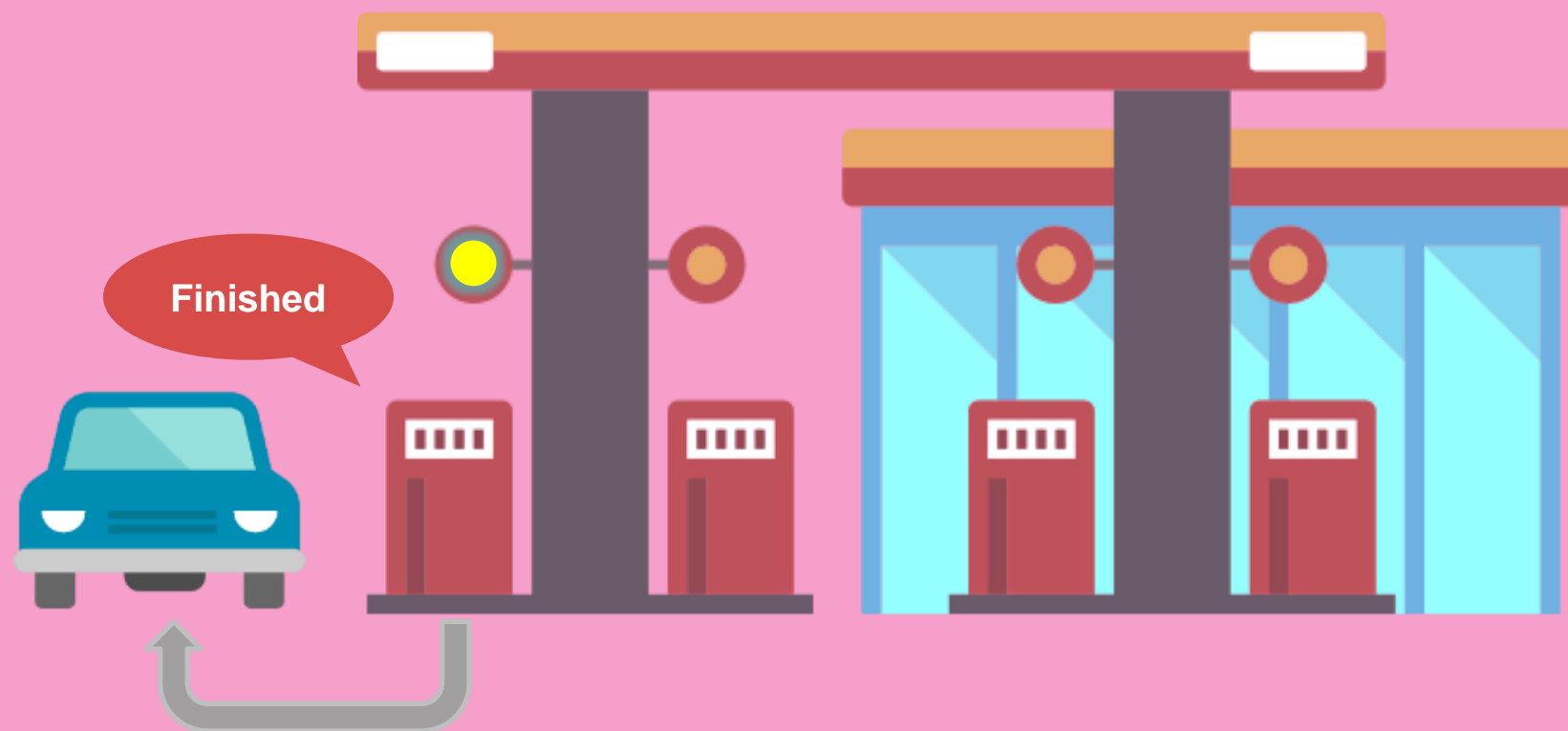
정상적인 루틴

4. 주유기에서 주유가 시작된다.



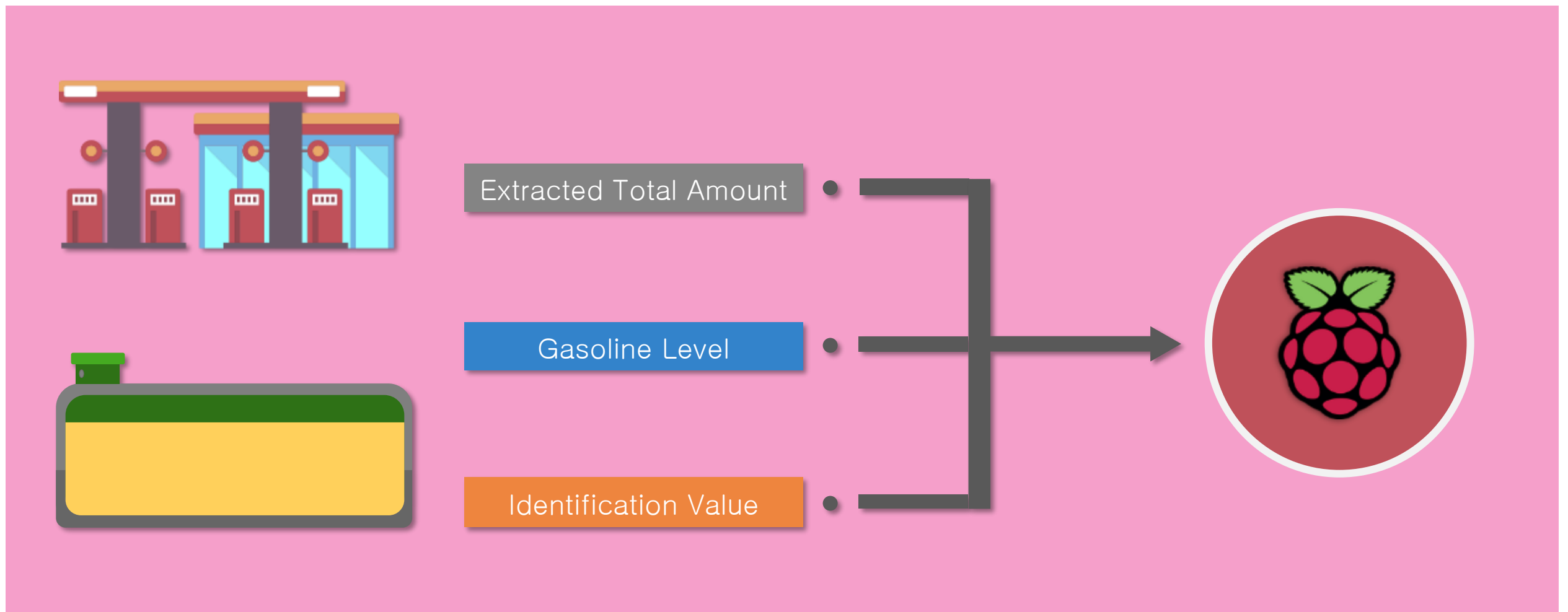
정상적인 루틴

5. 모든 주유기에서 주유가 끝날때 까지 기다린다.



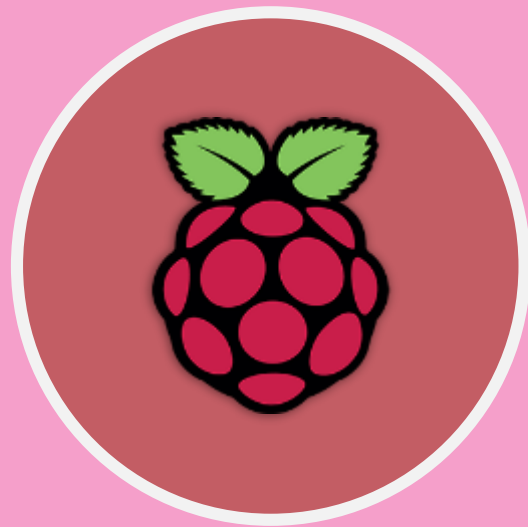
정상적인 루틴

6. 주유가 종료되면 데이터가 아두이노에서 라즈베리파이로 전송된다.



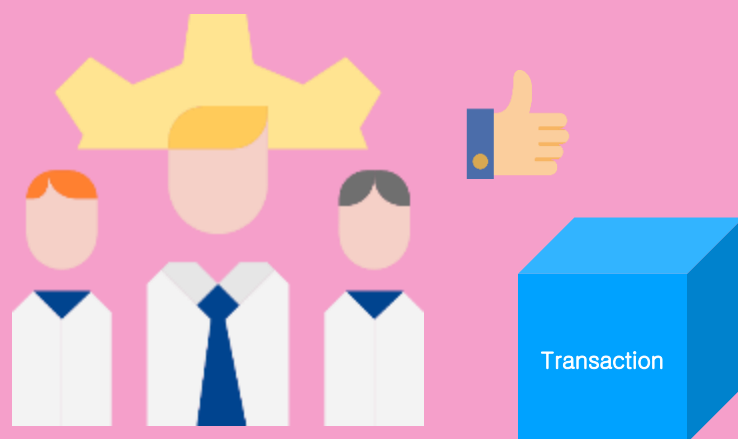
정상적인 루틴

7. 라즈베리파이는 전송받은 데이터를 스마트 컨트랙트에 반영하기 위해 트랜잭션 발생

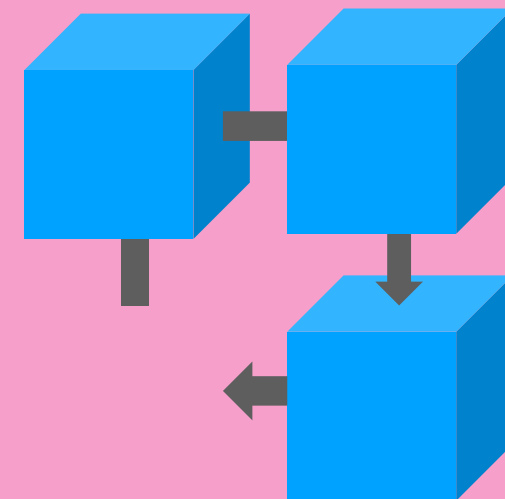


정상적인 루틴

8. 검증자 노드는 발생한 트랜잭션을 검증 후 블록에 담아 블록체인에 기록



Supervisor
Node



Blockchain

정상적인 루틴 요약

하나의 기름 탱크에 연결되어 있는 모든 주유기에서 주유가 종료될 때까지 대기한다.

주유가 종료되면 다음과 같은 3개의 데이터가 아두이노에서 라즈베리파이로 전송된다.

- * 탱크의 경유의 잔여량(Gasoline Level)과 탱크 내 경유의 식별 값(Identification Value)
- * 탱크와 연결된 모든 주유기에서 빠져나간 총 주유량 (Extracted Total Amount)

라즈베리파이는 전송 받은 **경유 잔여량, 경유 식별 값, 총 주유량**을 스마트컨트랙트에 반영하기 위해 트랜잭션을 발생시킨다.

주유기가 동작하지 않을 때는 10분에 한 번 데이터가 기록 및 검증된다.

예외 상황 처리

1. 주유 전 측정한 탱크 내 경유 잔여량 > 주유량 + 주유 후 탱크 내 경유 잔여량 (오차 범위 적용)

외부에서 비정상적인 방법으로 경유가 유입되었다는 것을 판단할 수 있다. 정유사로부터 기름을 공급받는 상황 이외에 기름이 추가된 것이므로 불순물이 섞였을 가능성이 존재한다.

2. 주유 전 측정한 탱크 내 경유 잔여량 < 주유량 + 주유 후 탱크 내 경유 잔여량 (오차 범위 적용)

비정상적인 방법으로 경유가 유출되었다는 것을 판단할 수 있다. 주유기에서 빠져나간 것이 아니라 인위적으로 빠져나간 것이므로 비정상적인 유출이라 할 수 있다.

3. 10분 전 측정한 경유 잔여량 > 현재 경유 잔여량

10분 전 측정한 경유 잔여량이 10분 뒤 측정된 경유 잔여량보다 많다면, 비정상적인 방법으로 외부에서 기름이 유입되었다는 것을 확인할 수 있다.

4. 10분 전 측정한 경유 잔여량 < 현재 경유 잔여량

10분 전 측정한 경유 잔여량이 10분 뒤 측정된 경유 잔여량보다 적다면, 비정상적인 유출이 있었다는 것을 확인할 수 있다.

예외 상황 처리(1) - 유사 석유 제조

1. 주유 전 측정한 탱크내 경유 잔여량 > 주유량 + 주유 후 탱크 내 경유 잔여량 (오차 범위 적용)

주유 전 탱크내 경유 잔여량이 100L, 주유량이 20L인 상태에서 주유 후 탱크 내 경유 잔여량이 90L인 상황을 생각해보자. 원래 주유 후에 탱크 내 경유 잔여량은 80L여야 하지만 90L로 10L가 초과된 상황이다.

외부에서 비정상적인 방법으로 경유가 유입되었다는 것을 판단할 수 있다. 정유사로부터 기름을 공급받는 상황 이외에 기름이 추가된 것이므로 불순물이 섞였을 가능성이 존재한다.

$$\begin{array}{ccccccc} & & & & & 10 \text{ L} & \text{초과 이상 발생} \\ 100 \text{ L} & - & 20 \text{ L} & = & 90 \text{ L} \\ \text{주유 전 탱크내 경유 잔여량} & & \text{주유한 양} & & \text{주유 후 탱크 내 경유 잔여량} \end{array}$$

예외 상황 처리(2) - 불법적인 기름 추출

2. 주유 전 측정한 탱크내 경유 잔여량 < 주유량 + 주유 후 탱크 내 경유 잔여량 (오차 범위 적용)

주유 전 탱크내 경유 잔여량이 100L, 주유량이 20L인 상태에서 주유 후 탱크 내 경유 잔여량이 70L인 상황을 생각해보자. 원래 주유 후에 탱크 내 경유 잔여량은 80L여야 하지만 70L로 10L가 부족한 상황이다. 주유기 이외의 방법으로 기름이 빠져나갔으므로 비정상적인 기름 유출이 일어났다는 것을 알 수 있다.

$$\begin{array}{ccccccc} & & & & & 10 \text{ L} & \text{미만 이상 발생} \\ 100 \text{ L} & - & 20 \text{ L} & = & \cancel{80} 70 \text{ L} \\ \text{주유 전 탱크내 경유 잔여량} & & \text{주유한 양} & & \text{주유 후 탱크 내 경유 잔여량} \end{array}$$

예외 상황 처리(3) - 유사 석유 제조

3. 10분 전 측정한 경유 잔여량 > 현재 측정된 경유 잔여량

10분 전 측정한 경유 잔여량이 10분 뒤 측정된 경유 잔여량보다 많다면, 비정상적인 방법으로 외부에서 기름이 유입되었다는 것을 확인할 수 있다.

```
function refueling(m){
  _timerCount[m.machine] = 0;
  _timer[m.machine] = setInterval(function(){

    let thisMachine = _gasMachine["machine" + m.machine];
    _timerCount[m.machine]++;
    thisMachine.useLiter = thisMachine.useLiter + 1;

    let percent = _timerCount[m.machine]/thisMachine.nowRefuelingAmout;
    ...

    if(_timerCount[m.machine]==thisMachine.nowRefuelingAmout){
      clearInterval(_timer[m.machine]);
    }
    ...
    //node send
    const url = "http://127.0.0.1:3000/usedgasamount/"+sum;

    fetch(url, {
      mode: 'no-cors',
      method: 'get'
    })
    .then(function(response) {
      response.text().then(function(text) {
        return 0;
      });
    })
    .catch(function(error) {
      console.log(error);
    });
  }
}, 600000);
```

예외 상황 처리(4) - 불법적인 기름 추출

4. 10분 전 측정한 경유 잔여량 < 현재 측정된 경유 잔여량

10분 전 측정한 경유 잔여량이 10분 뒤 측정된 경유 잔여량보다 적다면, 비정상적인 유출이 있었다는 것을 확인할 수 있다.

```
function refueling(m){
  _timerCount[m.machine] = 0;
  _timer[m.machine] = setInterval(function(){

    let thisMachine = _gasMachine["machine" + m.machine];
    _timerCount[m.machine]++;
    thisMachine.useLiter = thisMachine.useLiter + 1;

    let percent = _timerCount[m.machine]/thisMachine.nowRefuelingAmout;
    ...

    if(_timerCount[m.machine]===thisMachine.nowRefuelingAmout){
      clearInterval(_timer[m.machine]);
    }
    ...
    //node send
    const url = "http://127.0.0.1:3000/usedgasamount/"+sum;

    fetch(url, {
      mode: 'no-cors',
      method: 'get'
    })
    .then(function(response) {
      response.text().then(function(text) {
        return 0;
      });
    })
    .catch(function(error) {
      console.log(error);
    });
  }
}, 600000);
```

예외상황에 대한 이벤트 로그

불법적인 기름 추출

- 주유 전 측정한 탱크 내 경유 잔여량 < 주유량 + 주유 후 탱크 내 경유 잔여량 (오차 범위 적용)
- 10분 전 측정한 탱크 내 경유 잔여량 < 현재 경유 잔여량

주유소 이름: SK_ENERGY

기름 종류: DIESEL

이상 양: 1990

이벤트 상태: Tank has lower amount of gas. Possible abnormal gas extraction occurred

유사 석유 제조

- 주유 전 측정한 탱크내 경유 잔여량 > 주유량 + 주유 후 탱크 내 경유 잔여량 (오차 범위 적용)
- 10분 전 측정한 탱크 내 경유 잔여량 < 현재 경유 잔여량

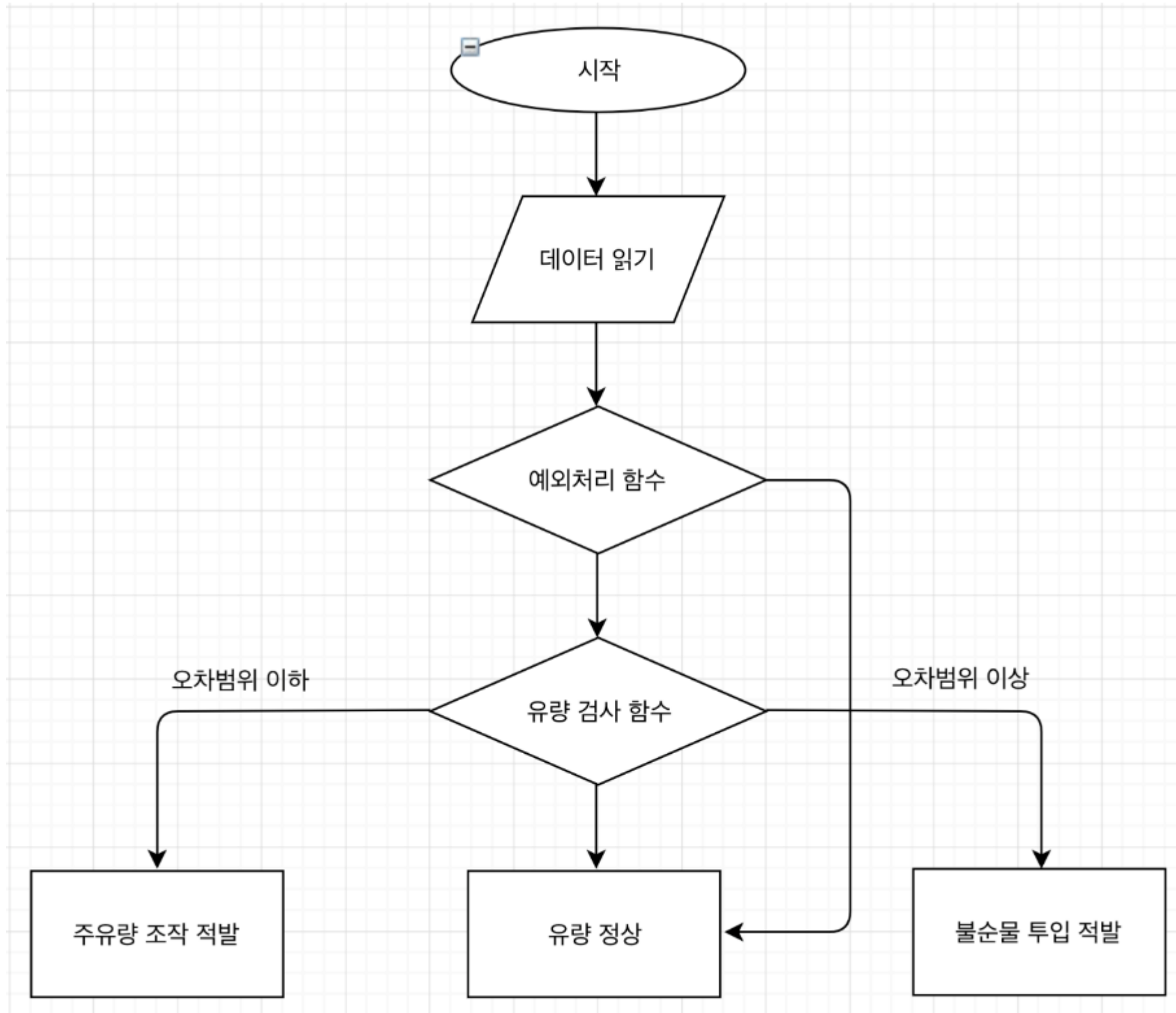
주유소 이름: SK_ENERGY

기름 종류: DIESEL

이상 양: 880

이벤트 상태: Tank has higher amount of gas. Possible abnormal liquid mixing occurred

스마트컨트랙트 순서도



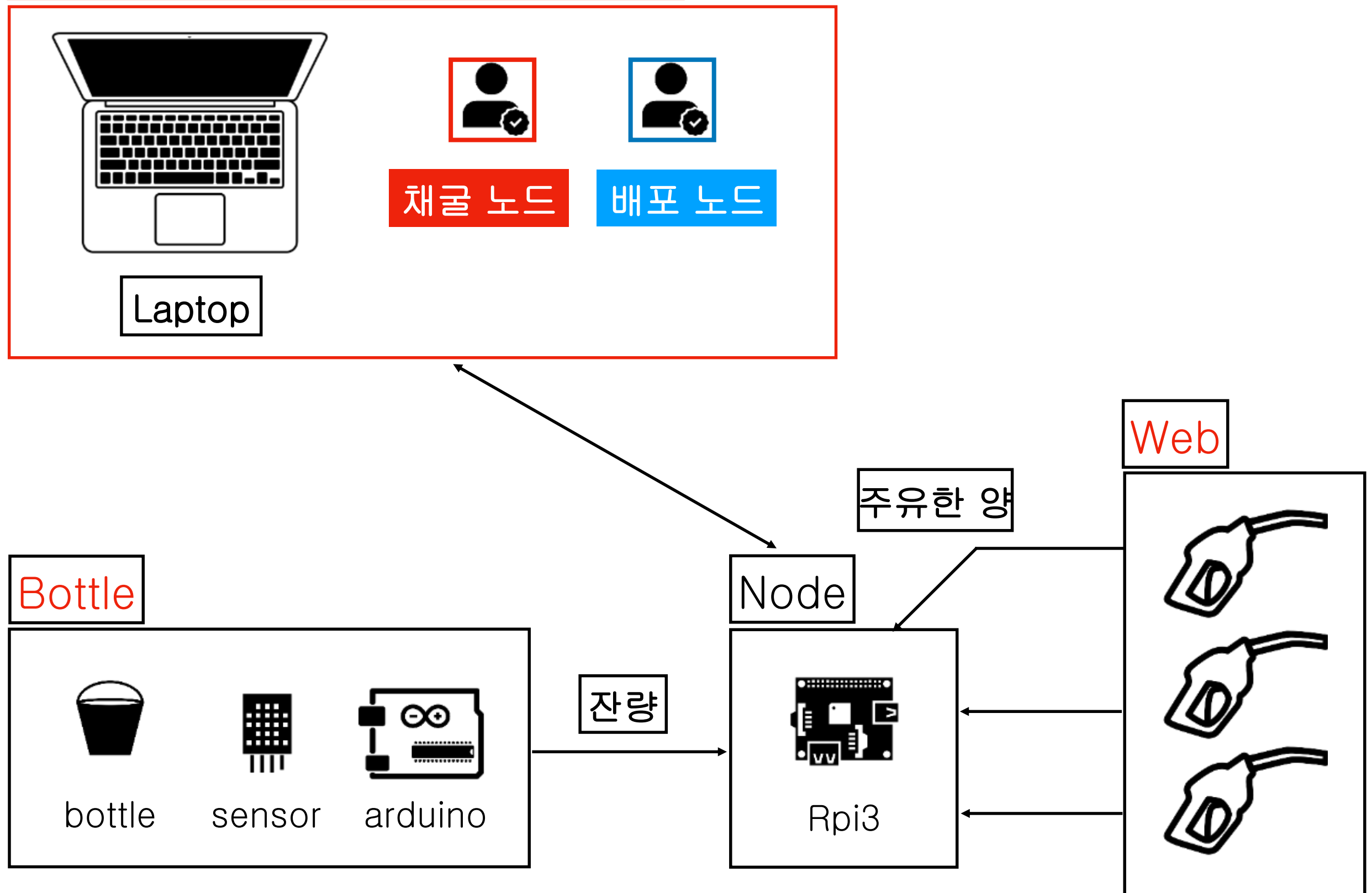
프로토 타입

프로토타입 구성

- * ~~가름의 성분 변조~~가 불가능한 시스템
- * ~~주유량의 변조~~가 불가능한 시스템

블록체인 형태	프라이빗 블록체인 (Private Blockchain)
합의 알고리즘	작업 증명 (Proof of Work)
블록체인 코어	이더리움 (go-ethereum)
제공하는 서비스	블록 탐색기

Private Proof Of Work Blockchain



Private Proof Of Work Blockchain



채굴 노드:

주유소 노드에서 발생한 트랜잭션을 블록체인 내에 기록

배포 노드:

주유소의 데이터가 기록되고 갱신될 스마트 컨트랙트를 배포

Rpi3:

이더리움 네트워크에 트랜잭션만 발생시킬 수 있는 스펙을 가진 노드

Bottle:

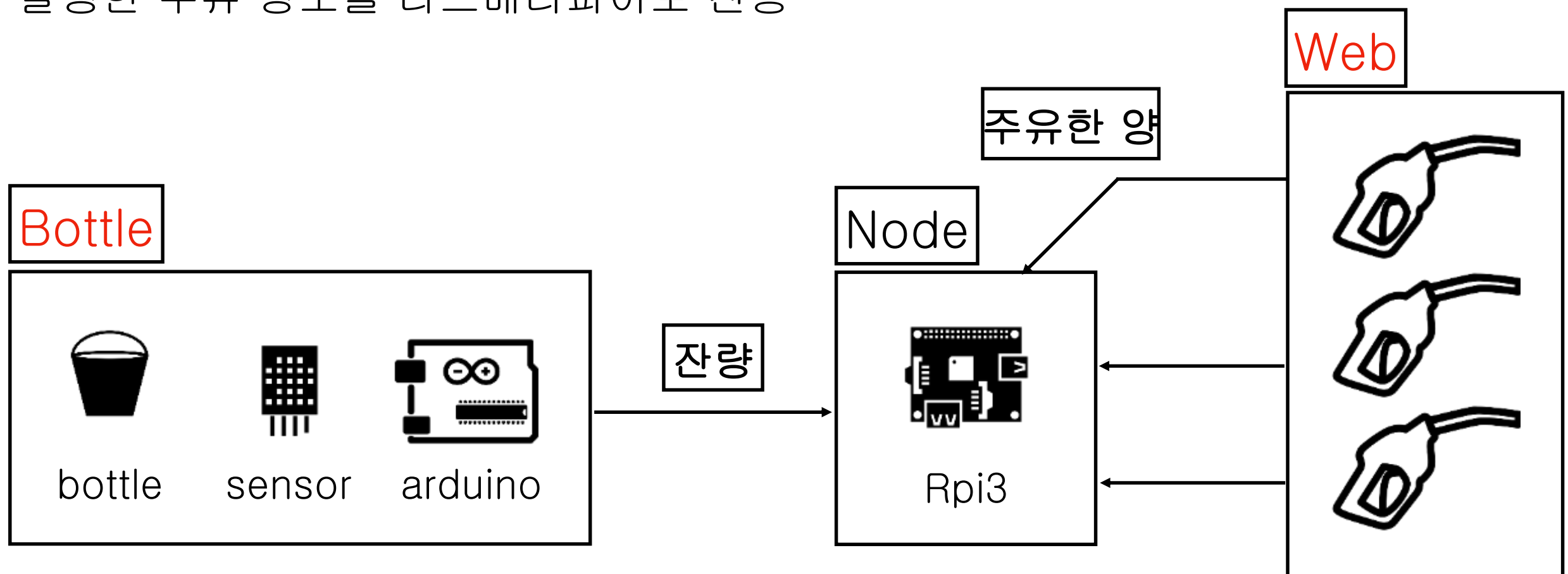
물을 담고 있는 물통

Web:

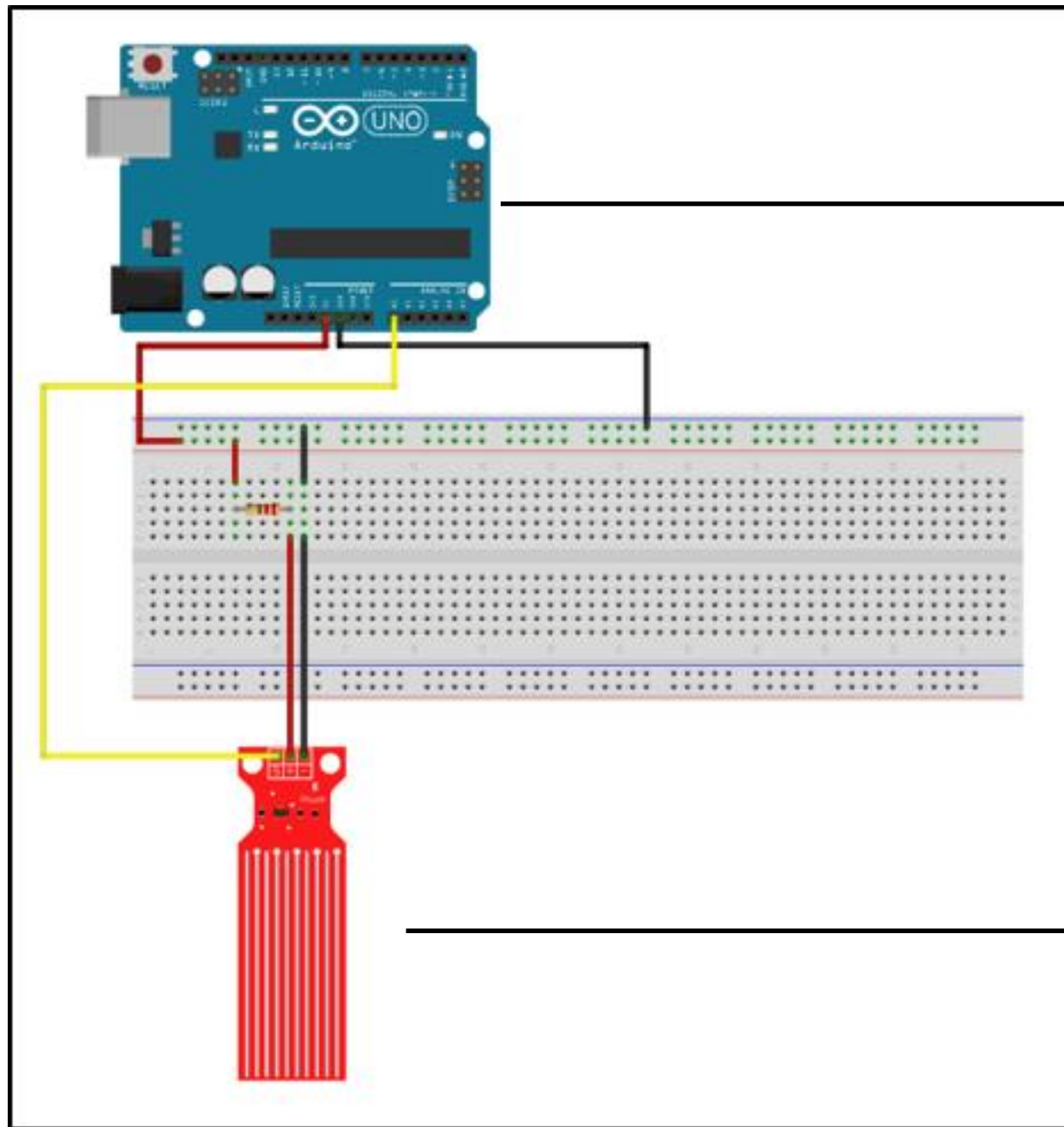
웹으로 구현한 가상의 주유구 3기
발생한 주유 정보를 라즈베리파이로 전송



주유소 노드



아두이노 회로 구성



아두이노:
센서값을 라즈베리 파이로
시리얼 통신을 통해 전송

수위 측정 센서:
Bottle에 있는 물의 높이 측정

웹으로 구현된 3대의 주유기

Gas station

Oil


1L 1,000WON

Diesel

1L 800WON

1st Machine

Gas




50000

Won ▾

END

2st Machine

Gas



300


Litter ▾

START

END

3st Machine

Gas



20000

Won ▾

START

END

40

프로토 타입 시연

주유소 기름 공급 - 1

기름을 공급하는 스마트 컨트랙트 함수 호출



YouTube Premium

검색



```
k  
1 > ct.stopCleaning.sendTransaction(330, {from:eth.accounts[0],gas:450000});  
1h "0x33b5d0db17957845b18b23bba4c8fb1854f10c5019eaf0550ce15b46aeb46723"  
f7 > ct.beforeFuelAmount()  
> ct.devEmptyFuel.sendTransaction(330, {from:eth.accounts[0], gas:450000});  
ock "0x8e8e413830221ff4cf5e77a9927bc20075870f19b32e3db7bfb99ac04a1d988f"  
h=0 > ct.beforeFuelAmount()  
lock 0  
sh=c > ct.startFueling(217, {from:eth.accounts[0], gas:450000});  
"0xb4a58ec48aec83b74f580c79494eaa31c0910f1c847f2df255693128a753f777"  
> []
```

MacBook



0:00 / 2:17



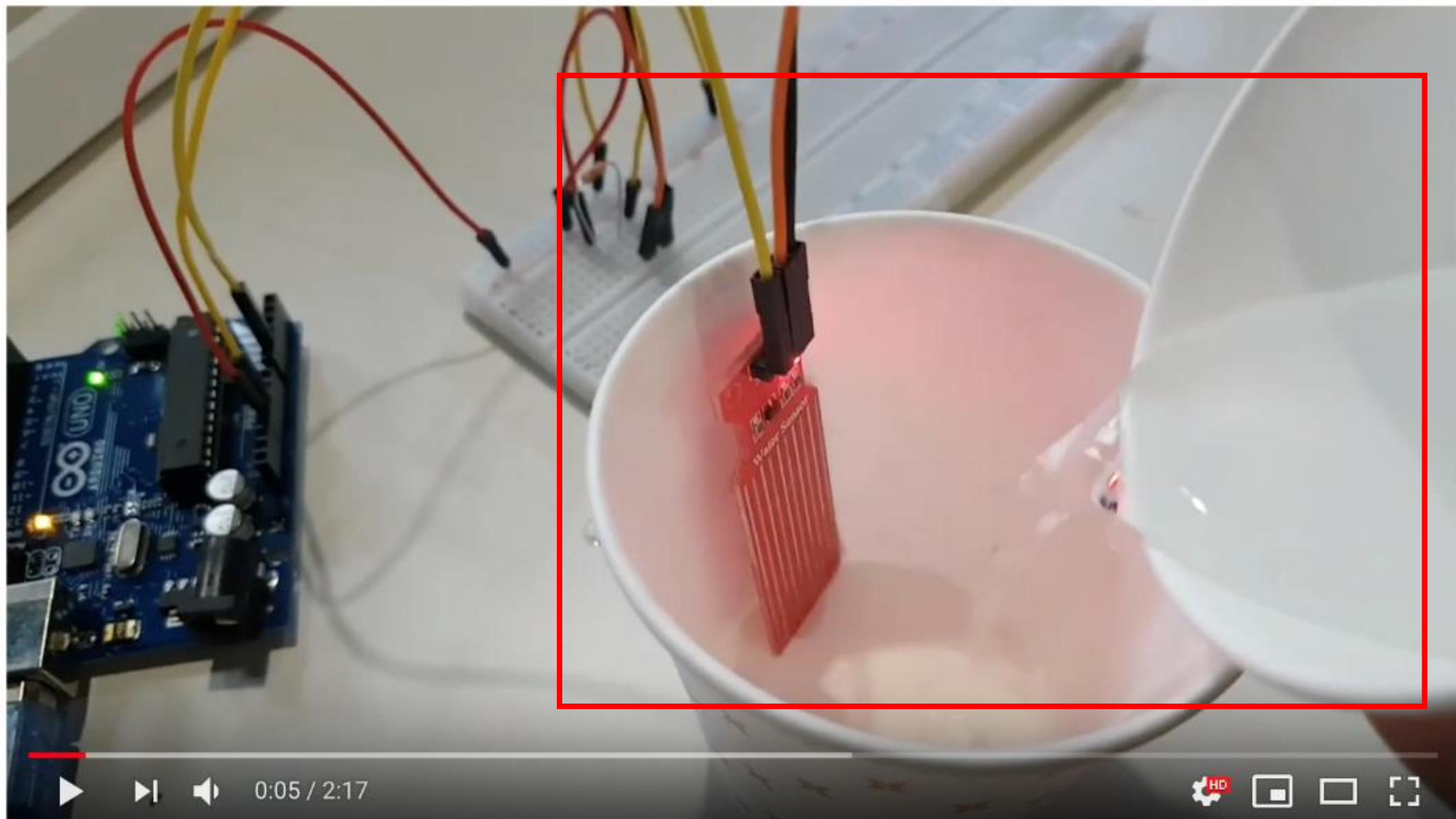
주유소 기름 공급 - 2

기름 탱크에 기름을 주입



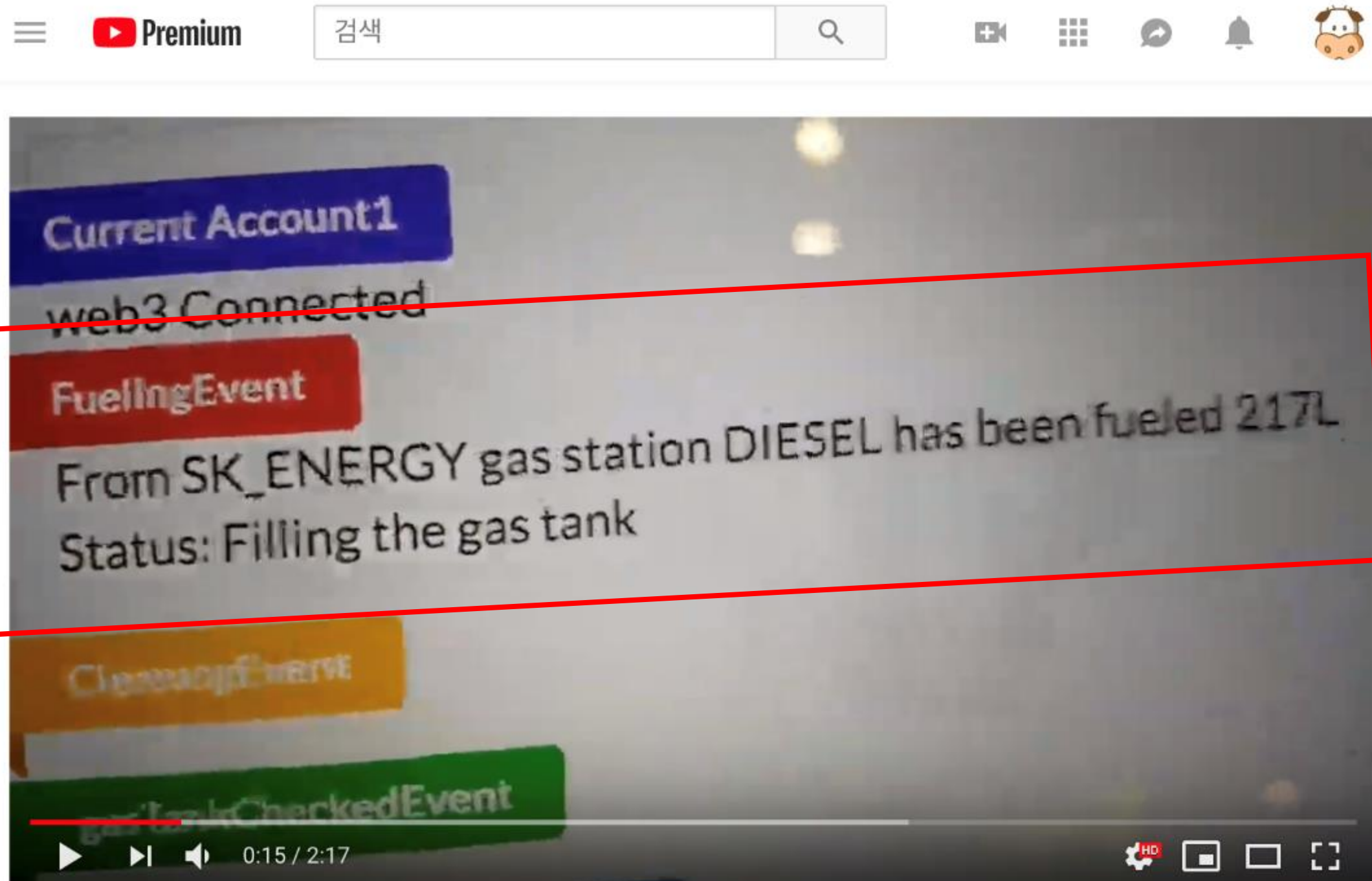
Premium

검색



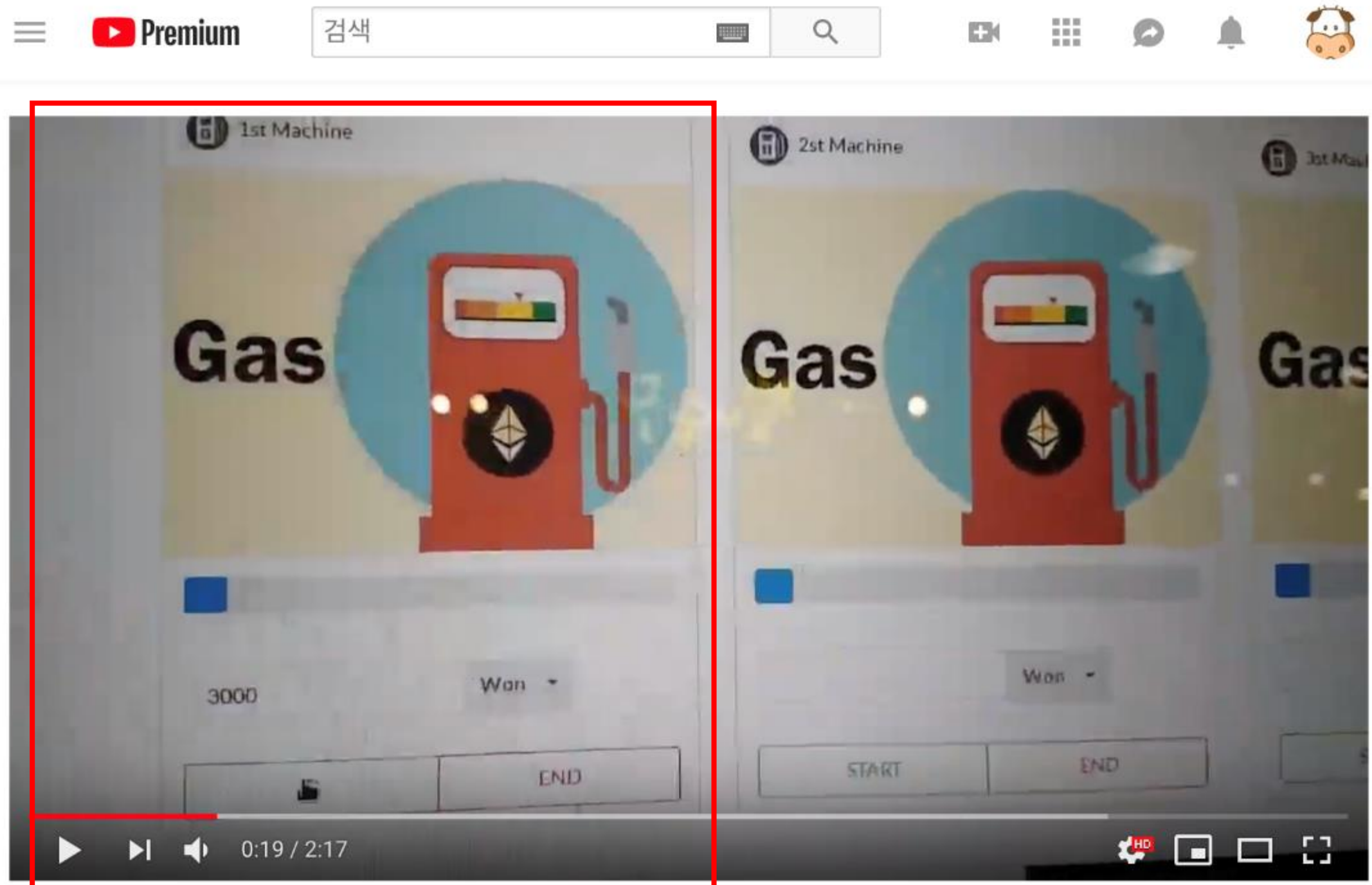
주유소 기름 공급 - 3

정상적으로 공급이 되었음을 알려주는 이벤트 로그



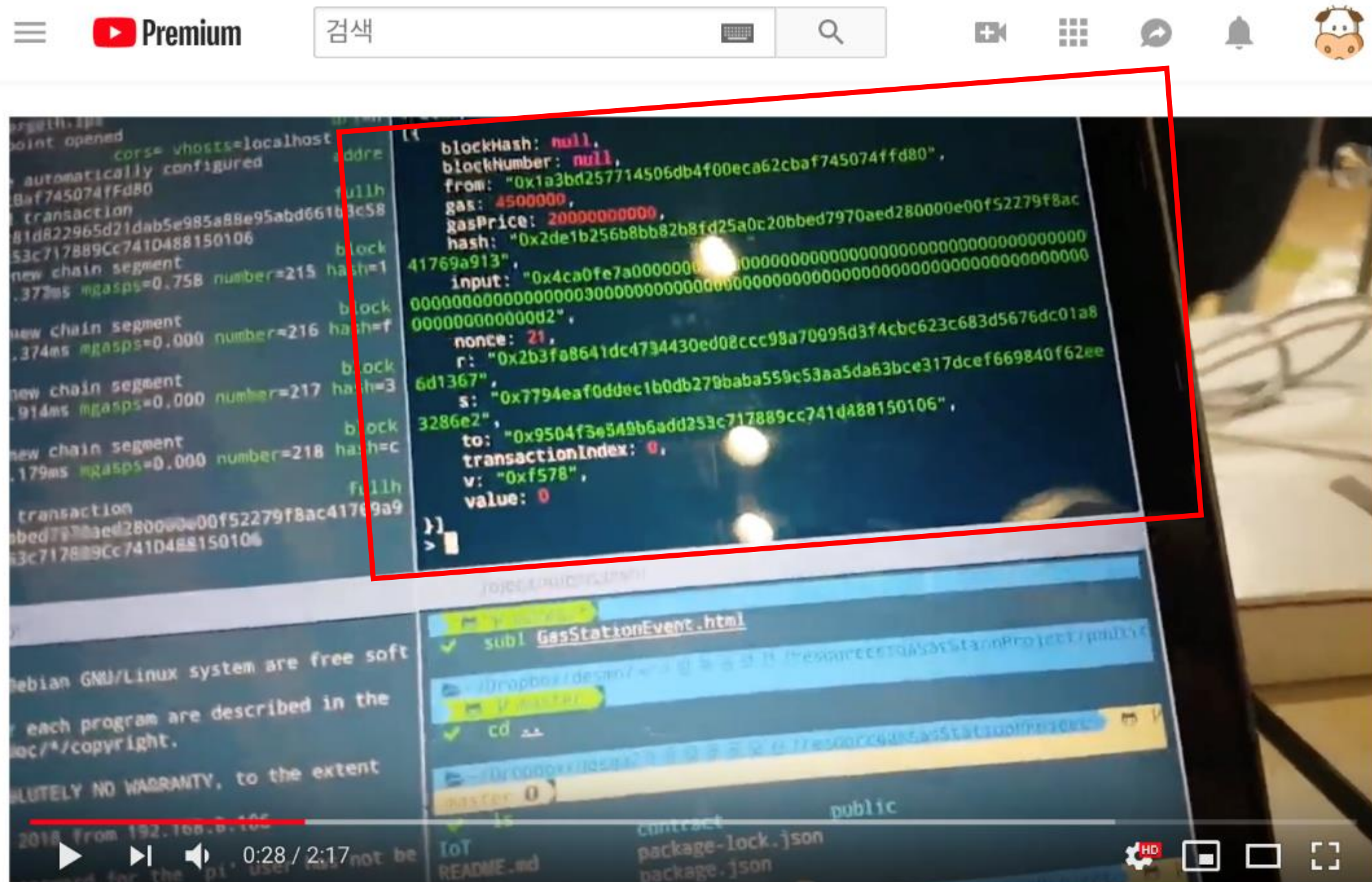
주유 과정 - 1

주유기에서 주유가 시작된다.



주유 과정 - 2

주유가 종료되면
총 주유량, 탱크 내 경유 잔여량 데이터를 블록체인에 기록



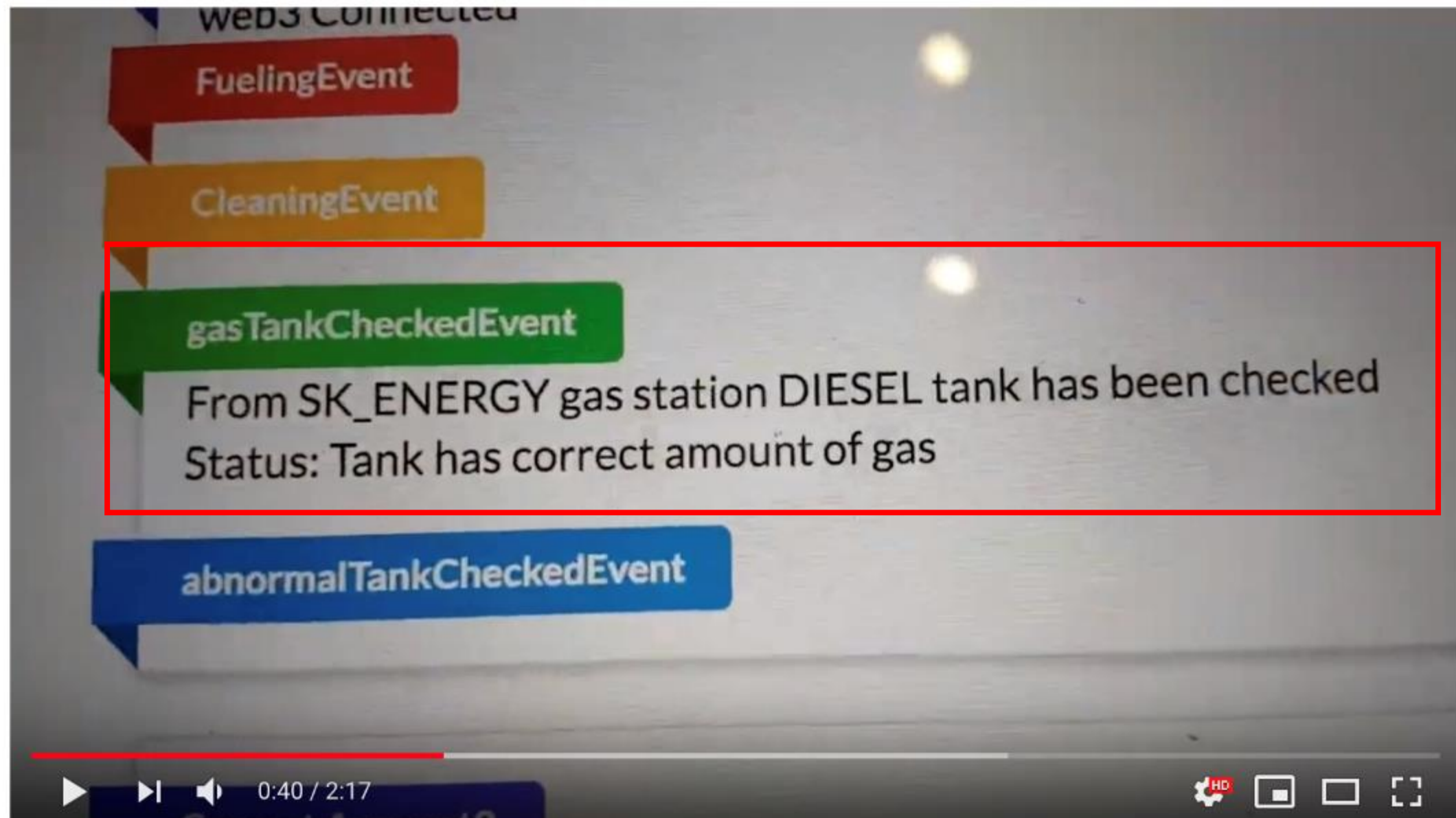
정상적인 상황

주유 전 탱크 내 경유 잔여량 = 주유량 + 주유 후 탱크 내 경유 잔여량



Premium

검색



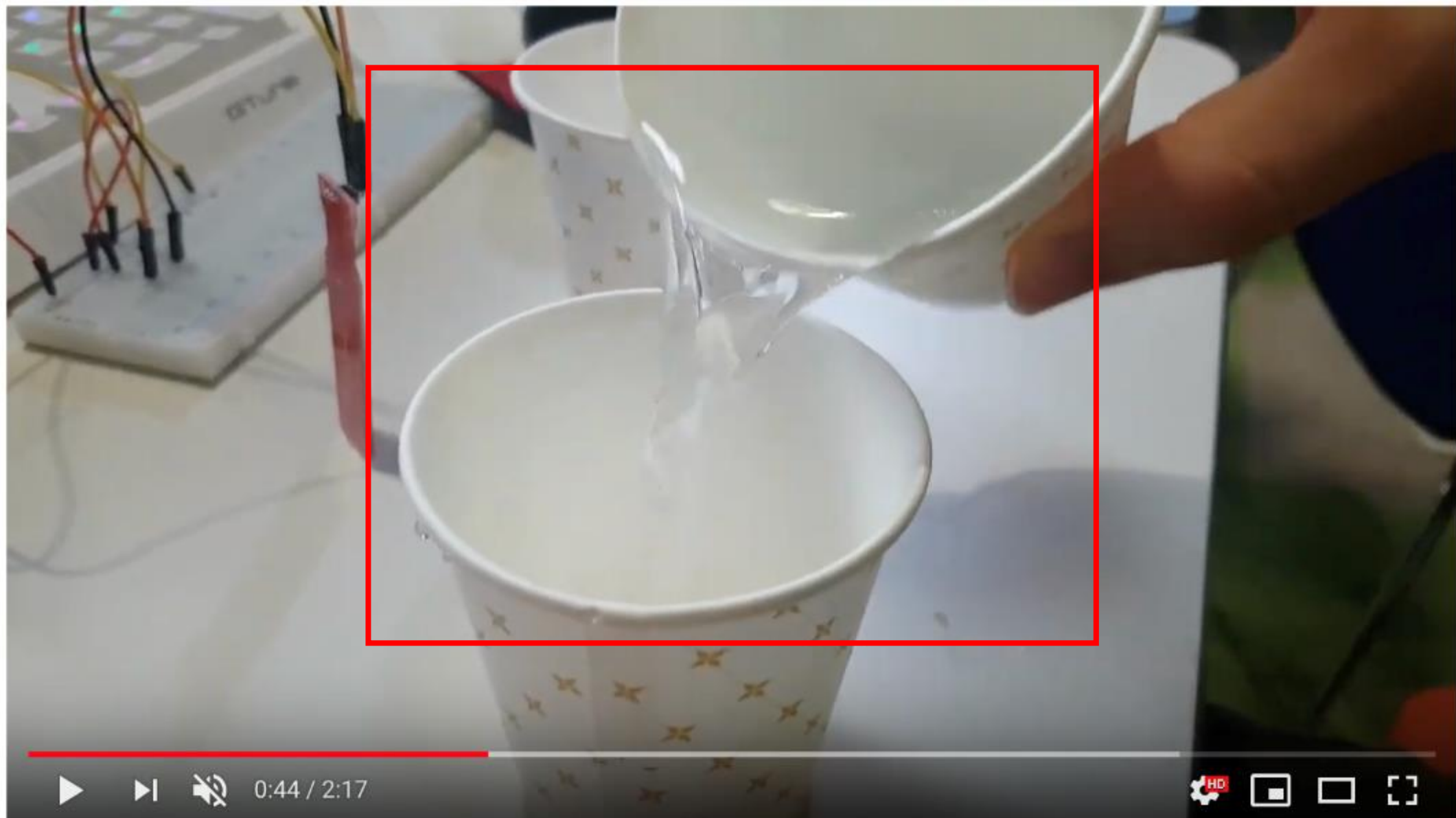
예외적인 상황(1) - 1

주유 전 탱크 내 경유 잔여량 > 주유량 + 주유 후 탱크 내 경유 잔여량
불법적인 기름 추출이 발생



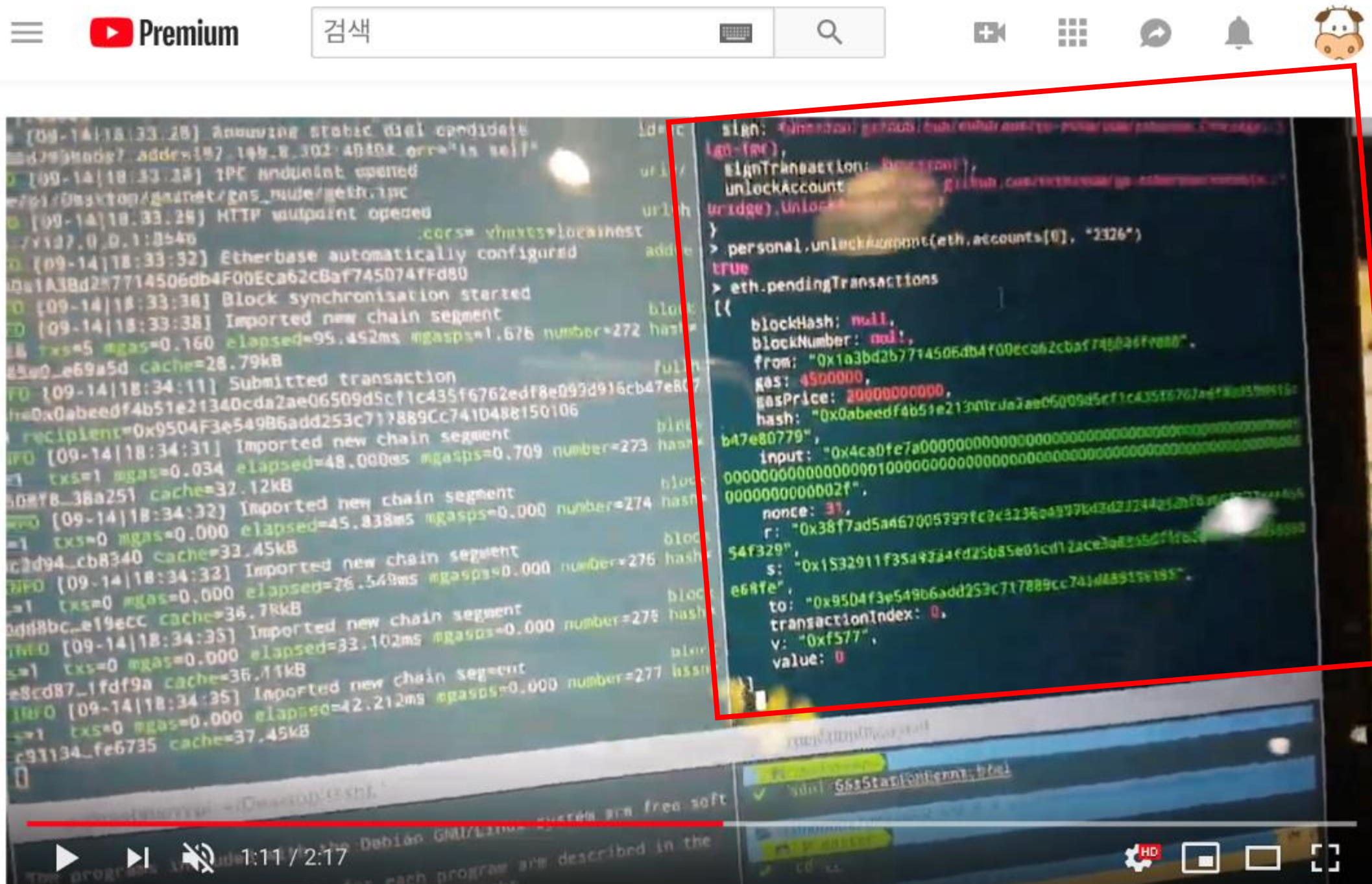
Premium

검색



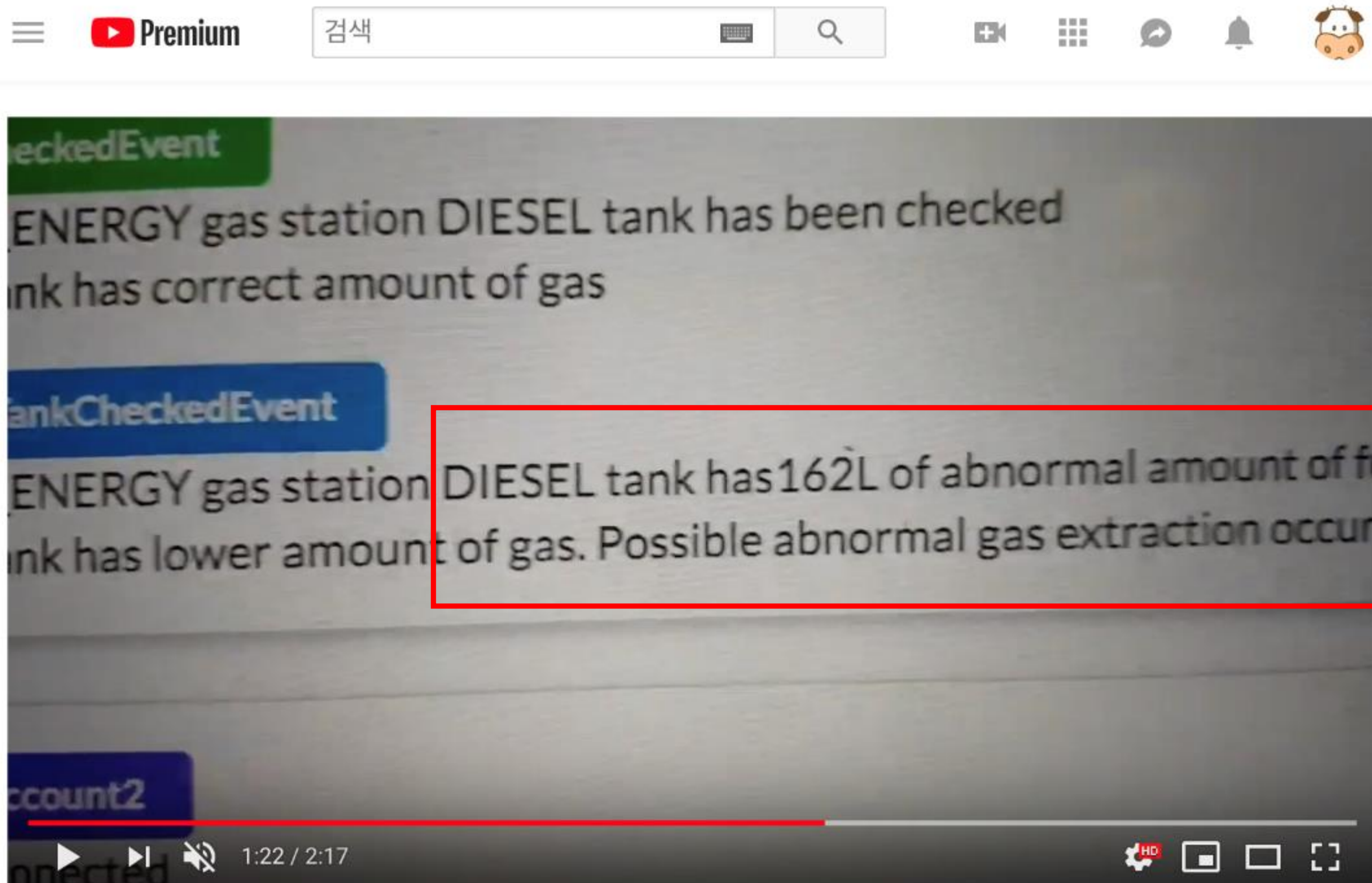
예외적인 상황(1) - 2

총 주유량, 탱크 내 경유 잔여량 데이터를 블록체인에 기록



예외적인 상황(1) - 3

162L 만큼의 불법적인 유출이 있었다는 것을 블록체인에 기록
(불법적인 기름 추출)



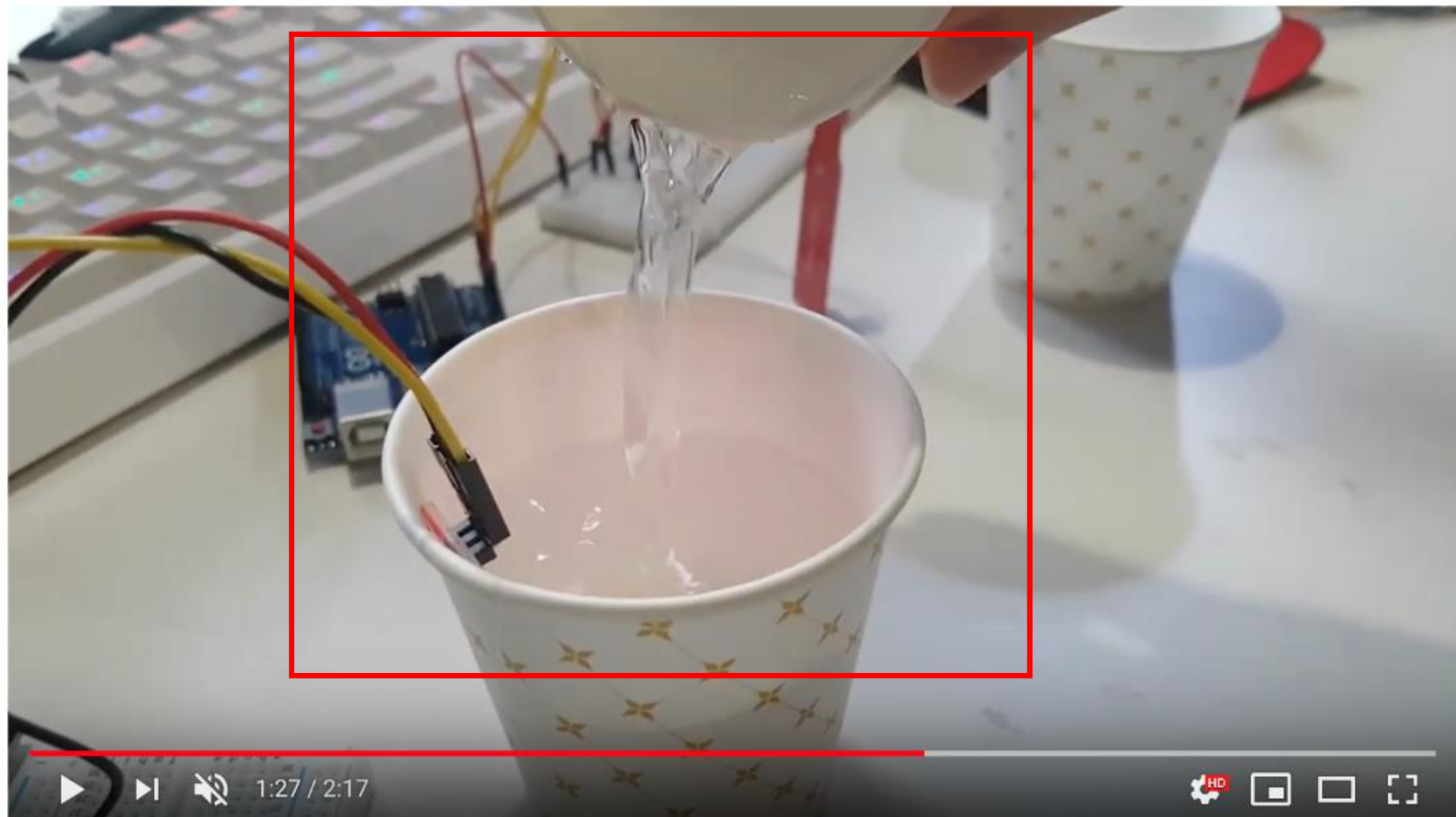
예외적인 상황(2) - 1

주유 전 탱크 내 경유 잔여량 < 주유량 + 주유 후 탱크 내 경유 잔여량
불법적인 외부 유입 발생 (유사 석유 제조)



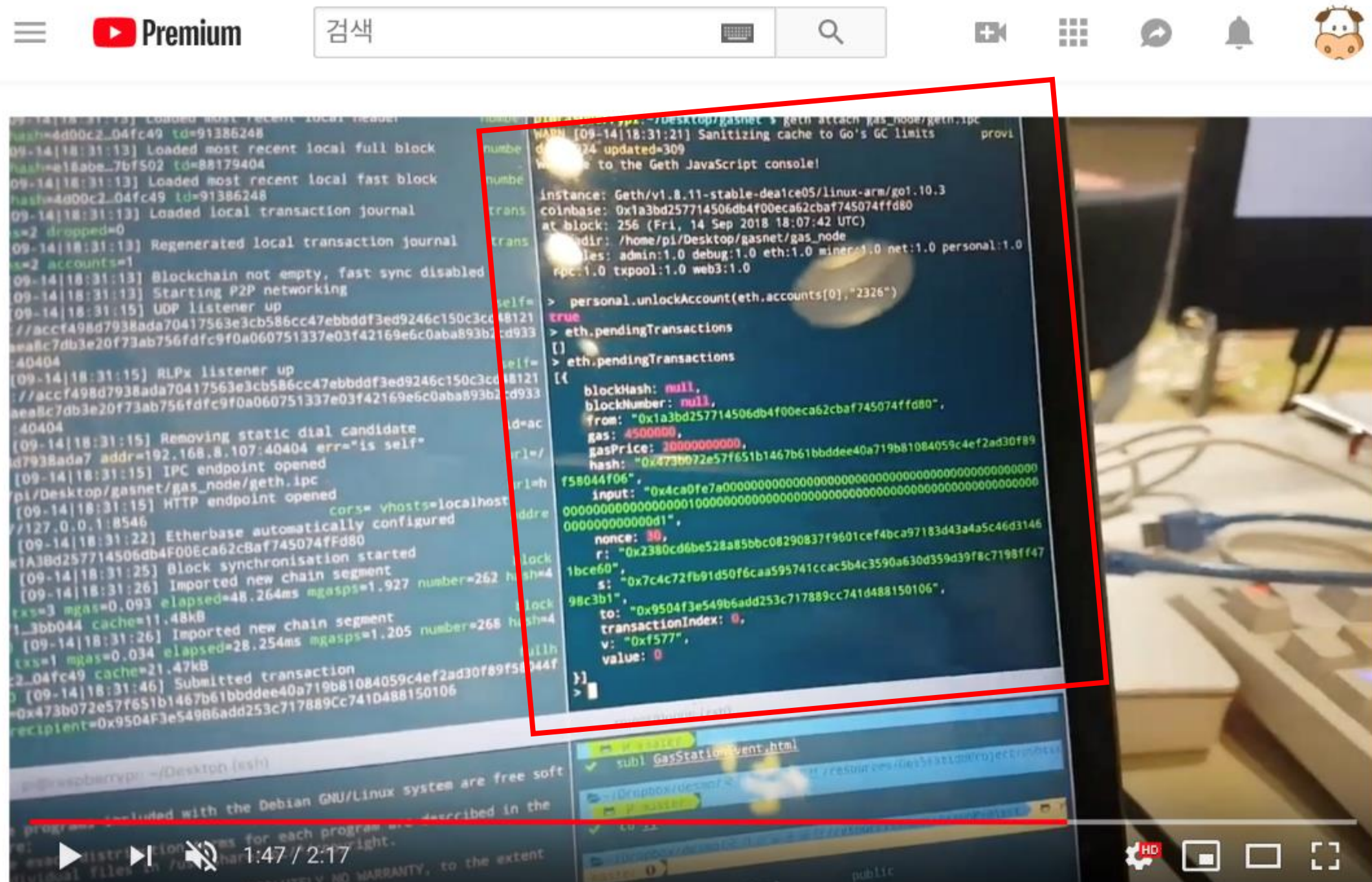
Premium

검색



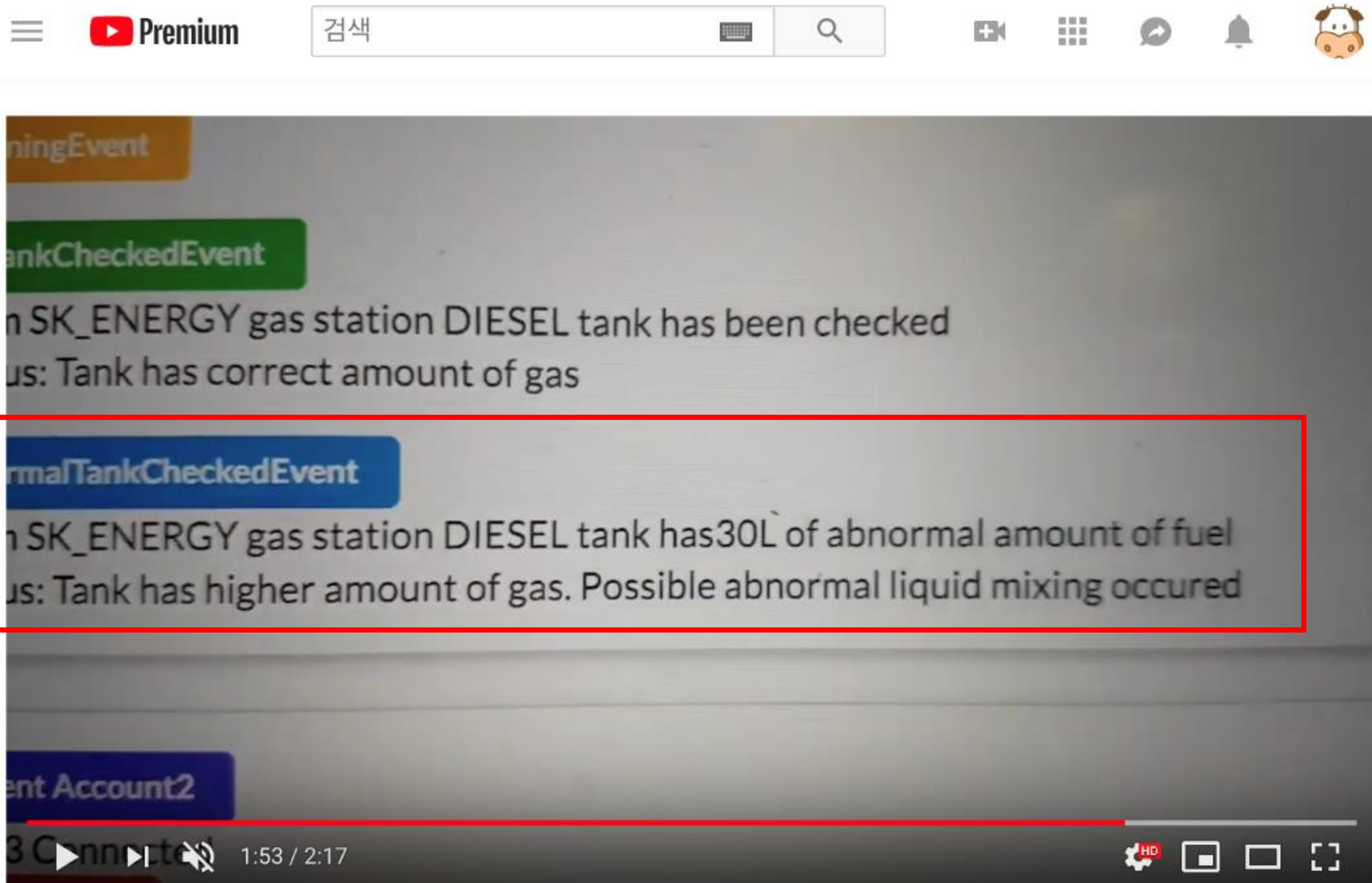
예외적인 상황(2) - 2

총 주유량, 탱크 내 경유 잔여량 데이터를 블록체인에 기록



예외적인 상황(2) - 3

30L 만큼의 불법적인 외부 유입이 있었다는 것을 블록체인에 기록
(유사 석유 제조)



Current Account1

web3 Connected

FuelingEvent

From SK_ENERGY gas station DIESEL has been fueled 244L
Status: Finished filling the gas tank

CleaningEvent

From SK_ENERGY gas station DIESEL tank is being cleaned and had 330L of gas.
Status: Finished cleaning the tank

gasTankCheckedEvent

From SK_ENERGY gas station DIESEL tank has been checked
Status: Tank has correct amount of gas

abnormalTankCheckedEvent

From SK_ENERGY gas station DIESEL tank has 162L of abnormal amount of fuel
Status: Tank has lower amount of gas. Possible abnormal gas extraction occurred

시스템의 취약점과 솔루션

1. 센서의 고장 및 조작

센서가 고장나거나 조작되면 비정상적인 데이터가 발생할 것이고 이 또한 블록체인에 기록이 되면서 예외상황을 발생시킨다. 따라서 검증자 노드가 이를 파악하고 적절한 조치를 취할 수 있다. 사전 방지를 완벽하게 할 수는 없지만 **사후 이상을 감지하고 대처할 수는 있다.**

2. 센서 데이터의 신뢰성

무게 센서 및 초음파 센서 등을 종합적으로 활용하여 값을 산출하여 신뢰성을 높일 수 있다.

3. 노드의 고장

노드가 고장나면 블록체인에 트랜잭션이 발생하지 않게 되고 이는 검증자 노드에서 파악이 가능하다. 사전 방지를 완벽하게 할 수는 없지만 **사후 이상을 감지하고 대처할 수는 있다.**

4. 이상이 생긴 체인 데이터에 관한 수정 권한

경쟁 정유사 그리고 한국 석유관리원으로 검증 노드를 구성하여 데이터에 관한 권리를 분산시켜 신뢰를 높인다. 특정 주체가 프라이빗 블록체인에 가지는 권한을 분산하여 관리한다.

감사합니다.