

1

2

FARKLARI

1-Aslında en temel fark birisinin daha basit birisinin daha komplike olmasından kaynaklanıyor ve birinde 4 func çağırırken birinde 16.Ama temel işlevlerini baz aldığımızda mantık olarak aynı işleri yapmaktadır.

2- Kontratları sadece diyagram olarak değilde pisacode olarak incelediğim zaman aslında birinin markette çıkmaya hazır olmaya yakın bir dizi olarak gördüm.

3-Fonksiyonlarda kullandığımız parametreleri daha fazla ve daha karışık

4-Örnek olarak setAdmin fonksiyonlarını karşılaştıracak olursak

   function setAdmin(address newAdmin) external {

        require(msg.sender == admin); //escrow App2

        require(newAdmin != address(0));

        admin = newAdmin;

        user\_balances[newAdmin] = user\_balances[admin];

        user\_balances[admin] = 0;

    }

 function setAdmin(address \_newAdmin) external onlyAdmin {

        require(\_newAdmin != address(0)); //1

        admin = \_newAdmin;

    }

Zorlandığım yerler ve çözümlerim

1-

    function addEtherToTransaction(uint \_transaction\_id) external payable onlyTransactionOwner(\_transaction\_id) CheckIfTransactionNotZero(\_transaction\_id) CheckIfTransactionActive(\_transaction\_id) { //araştır CheckIfTransactionNotZero ve CheckIfTransactionActive

Kısaca eth işlemlerini takip eder. Burada fonksiyonun yanına eklediğimiz ekstra eklentiler daha önemlidir.

Kullanıcıların takibini kolaylaştırır.

CheckIfTransactionNotZero = Kodda kullanılan [Checked aritmetiği](https://docs-soliditylang-org.translate.goog/en/v0.8.17/control-structures.html?_x_tr_sl=en&_x_tr_tl=tr&_x_tr_hl=tr&_x_tr_pto=sc" \l "unchecked) nedeniyle 0 miktarlı işlem oluşturulmamasını sağlıyor

**Ek bilgi**; Solidity 0.8.0'dan önce, aritmetik işlemler, ek kontroller getiren kitaplıkların yaygın kullanımına yol açan eksik veya taşma durumunda her zaman ekstra işlemler yapılırmış.

CheckIfTransactionActive = İşlemin aktif olup olmadığını kontrol edici.

2-

function confirmTransaction(uint \_transaction\_id) external onlyTransactionOwner(\_transaction\_id) CheckIfTransactionNotZero(\_transaction\_id) CheckIfTransactionActive(\_transaction\_id) { //checl fonk araştır

Transfer onaylama

3-

    function TransferPayment(address \_receiver, uint \_amount) internal { //dön bak

Transfer ödeme fonksiyounu

4-function forceCancelTransaction(uint \_transaction\_id) external payable onlyAdmin {

Adminin zorla bir işlemi onayladığı bir fonksiyon. Kullanıcı işlem oluşturup onaylamazsa yada cancel etmezse admin işlemi force cancel edebiliyor.