



CHAIN
CARE

PROBLEMLER



01

DEPOLAMA SORUNU

Sağlık kuruluşlarında hastaların verileri karışık ya da eksik ve bulması zor bir şekilde depolanıyor.



03

ZAMAN

Doktor uzaktaysa hastaların ufak bir işlemi bile gerçekleştirilemez.

02

SİBER GÜVENLİK

Hastanelere yapılan siber saldırılar kişisel verilen korunmasına karşı büyük bir risk teşkil ediyor.



DATA
PRIVACY

04

MAALİYET-ÇEVRE KİRLİLİĞİ

Verilerin kağıtlara karışık bir şekilde depolanması gereksiz kağıt harcamaya sebep olduğu için doğayı kirletiyor.



ÖRNEK PROBLEM

Beklenmedik bir durumdan dolayı hastanede bulunamayan bir doktor olan Amanda Hanım'dan randevusu olan Henry Bey'in rutin kontrol ve ilaç yenilemesi için hastaneye gittiğini varsayıyalım.

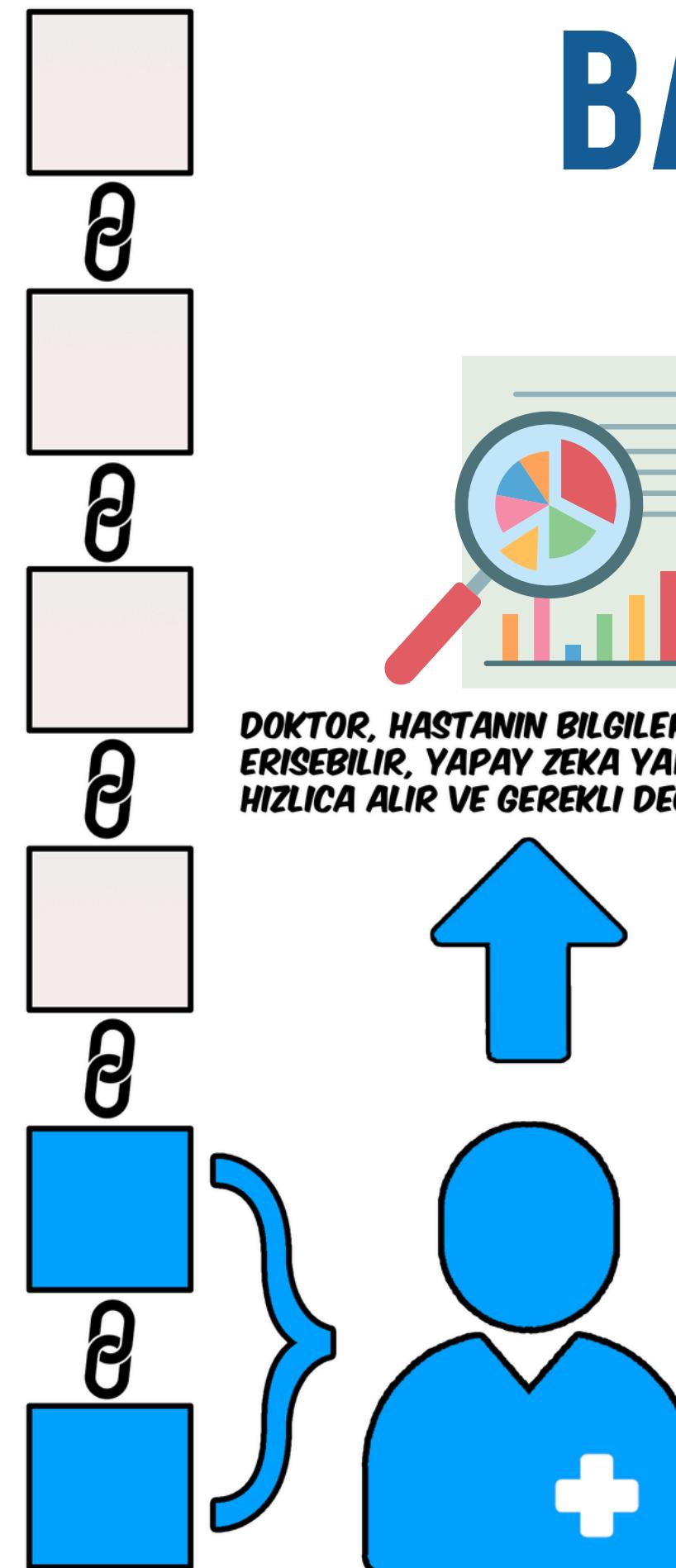
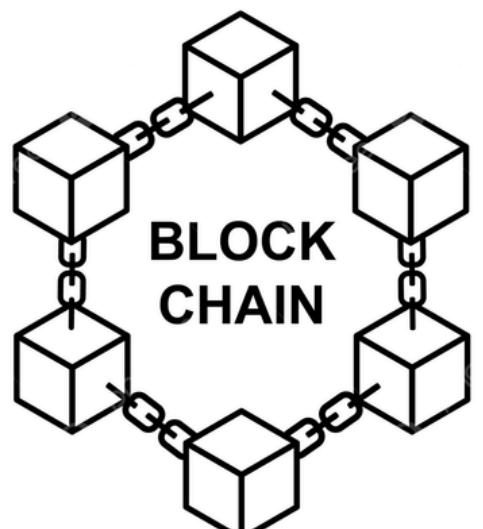
ChainCare sistemi sayesinde Dr. Amanda, ofisinde bulunmasa bile hastane sistemine güvenli bir bağlantı ile uzaktan dilediği cihaz ile erişir. Yapay zeka destekli sistem, Henry Bey'in son test sonuçlarını ve sağlık durumunu bir araya getirir ve olgu şeklinde Dr. Amanda'ya sunar.

Dr. Amanda, bu değerler üzerinden kendi yorumunu yapar ve Hasan Bey'in ilaçlarının yenilenmesi gereğine karar verir. Sistemin sunduğu uzaktan hasta kimlik doğrulama özelliği* ile, Henry Bey'in kimliğini doğrulayan Dr. Amanda, yeni reçeteyi dijital olarak yazar ve sisteme kaydeder. Henry Bey, herhangi bir eczanededen bu reçeteyi kolayca temin edebilir.

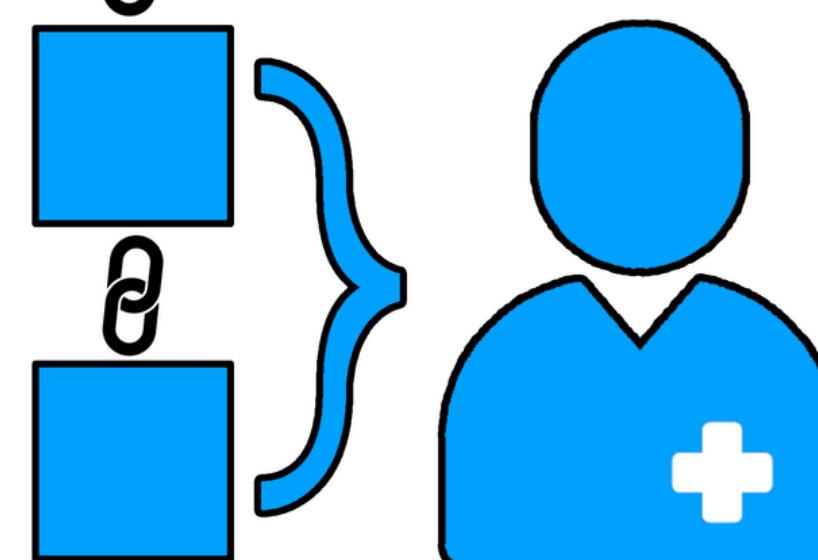
Bu süreçte, hemşire ve doktor eksikliği nedeniyle olası bir tedavi gecikmesi önlenmiş, insan hatası riski en aza indirilmiş ve hastanın sağlığı güvende tutulmuştur.



*: Blockchain ile yalnızca seçili kişilerin erişebileceğii ve NFT ile giriş yapabileceğii bir veri sistemi

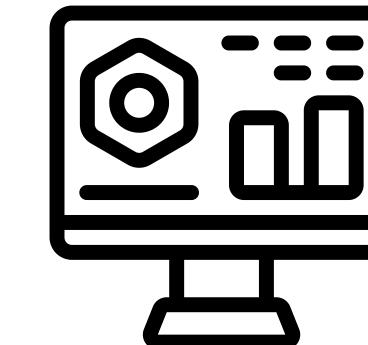


DOKTOR, HASTANIN BILGILERINE UZAKTAN GÜVENLE ERİSEBİLİR, YAPAY ZEKA YARDIMIYLA DURUMUN ÖZETİNİ HIZLICA ALIR VE GEREKLİ DEĞİŞİKLİKLERİ YAPABİLİR



DOKTORDA YALNIZCA KENDİ HASTALARININ BLOKLARINA DÜZENLEME ERİSMİ BULUNUR

BASIT ALGORİTMA



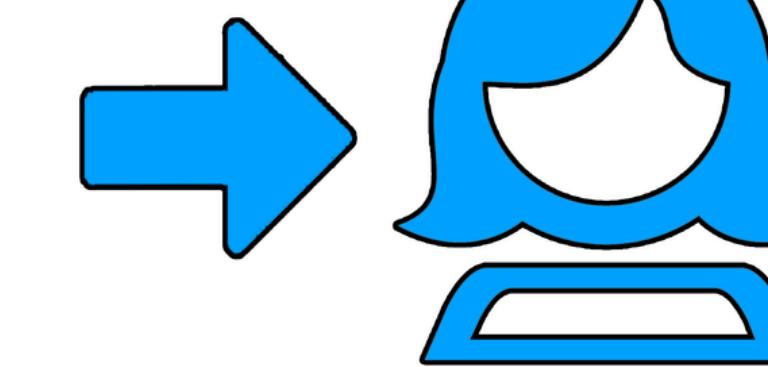
DEĞİŞİM BLOCKCHAIN'E UPDATE OLARAK YANSITILIR VE YENİ BİLGİ KAYDEDİLİR



HEMSİRE, HASTAYA GEREKLİ DOZU VERDİKTEN SONRA İŞLEM ONAYI KOLAYCA VERİLİR VE TABLOYA YANSITILIR



HEMSİREDE, YALNIZCA KENDİ HASTALARININ YER, SAAT VE DOZ VERİSİ BULUNUR. TABLOYA YANSITILIR VE ZAMAN GELİNCE BİLDİRİM GÖNDERİLİR



CHAINCARE KODU

Hasta bilgileri ekleme/düzenleme

```
contract PatientData {
    struct Patient {
        string name;
        string lastName;
        string doctor;
        string nurse;
        string medicine;
        uint dose;
        string time;
        string day;
    }
}

mapping(address => Patient) private patients;

function addPatient(address _address, string memory _name, string memory _lastName, string memory _doctor, string memory _nurse, string memory _medicine, uint _dose, string memory _time, string memory _day) public {
    patients[_address] = Patient(_name, _lastName, _doctor, _nurse, _medicine, _dose, _time, _day);
}

function getPatient(address _address) public view returns (string memory, string memory, string memory, string memory, string memory, uint memory, string memory, string memory) {
    Patient memory p = patients[_address];
    return (p.name, p.lastName, p.doctor, p.nurse, p.medicine, p.dose, p.time, p.day);
}
```



CHAINCARE KODU

Görüntüleme ve editleme yetkileri

```
function viewData(address _patient) public view hasAccess(_patient, AccessLevel.View) returns (
    string memory name,
    string memory lastName,
    string memory doctor,
    string memory nurse,
    string memory medicine,
    string memory dose,
    string memory time,
    string memory day
) {
    Patient storage patient = patients[_patient];
    return (
        patient.name,
        patient.lastName,
        patient.doctor,
        patient.nurse,
        patient.medicine,
        patient.dose,
        patient.time,
        patient.day
    );
}
```



CHAINCARE KODU

Hesap bilgileri ve yetkilendirme

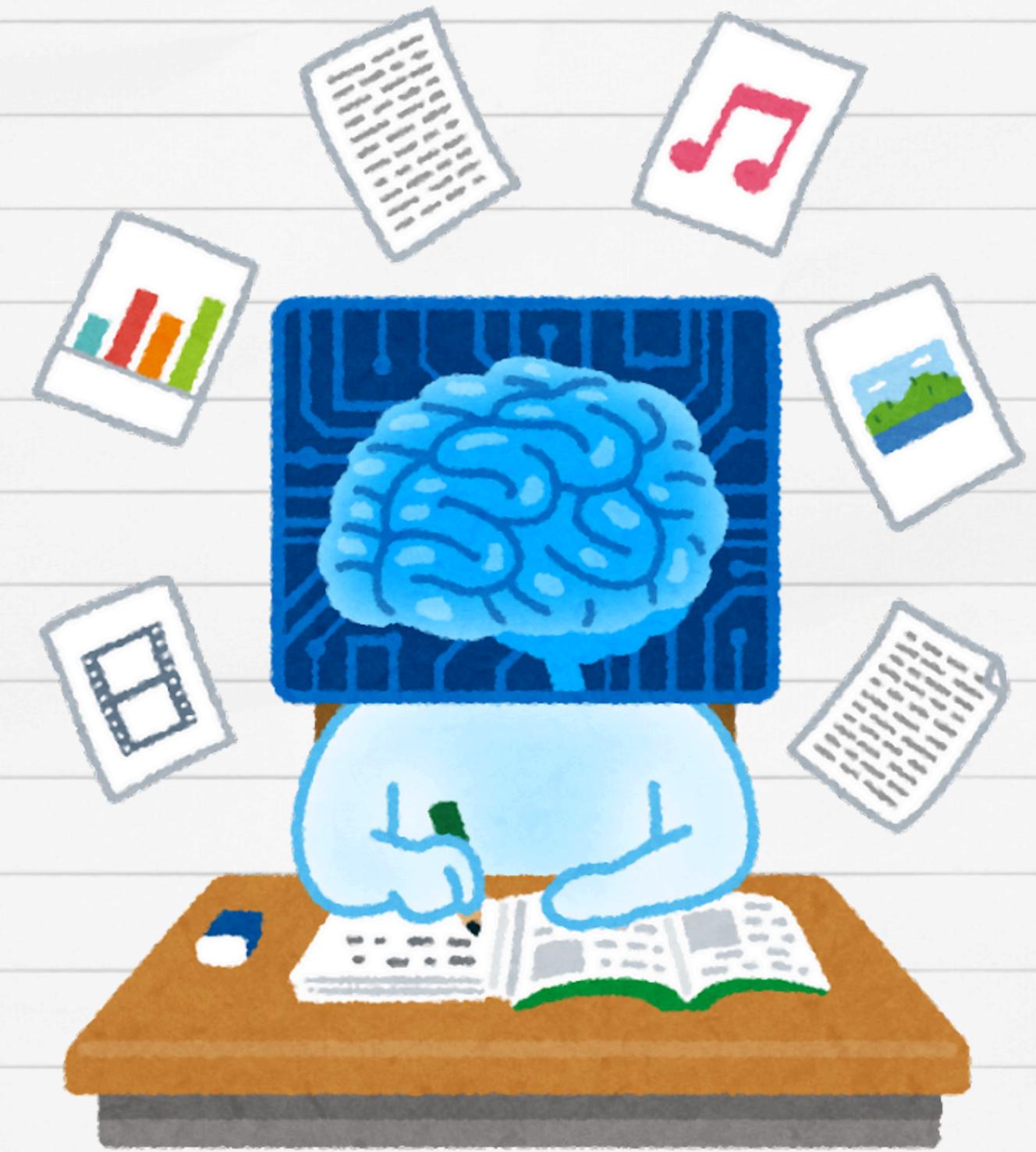
```
enum AccessLevel { NoView, View, Edit }

mapping(address => Patient) private patients;

modifier onlyOwner(address _patient) {
    require(msg.sender == patients[_patient].owner, "Not authorized");
}

modifier hasAccess(address _patient, AccessLevel _required) {
    require(
        msg.sender == patients[_patient].owner ||
        patients[_patient].authorizations[msg.sender] >= _required,
        "Access denied"
}
```





CHAINCARE VE AI

Yapay zeka, okunan değer verilerini derleyip olgu halinde doktora sunar, ayrıca veriyi tek bir yerde bulundurarak hemşirelerin gözünden kaçmasıyla oluşacak olası bir insan kaynaklı hatanın önüne geçebilir.



NEDEN CHAINCARE?



CHAINCARE

Düşük veri boyutu, sanal depolaması ve olası insan hatalarını önlemeyle ortalama bir hastanenin giderlerini kısaltarak aylık yaklaşık

\$100K

zarar engeller.

KAĞIT MASRAFLARI

Ortalama bir hastanenin tüm kağıt ve türevleri için yaptığı harcamayı kısaltarak datanın tamamen sanal ortamda blockchain ile depolayanmasını sağlayabiliriz.

ZAMAN KAZANCI

Veri depolama sürecinde oluşabilecek insan hatalarını engelleyecek yapay zeka sistemi ve otomatik veri depolama ile normalde harcanan zamanı azaltabiliriz.

GÜVENLİK

BlockChain güvenliği ile hastanın kişisel verilerin korunmasını ve olası siber saldırırlara karşı güvende olmasını sağlayabiliriz.

DEPO ALANI KAZANCI

BlockChain tarafından sağlanan düşük veri boyutu ve küçük depo ihtiyacı sayesinde database masrafları düşürülür.

HUKUKA UYGUNLUK

ChainCare, yapay zeka desteği ile olası insan hatalarının önüne geçerek ihmal davaları gibi hukuksal olarak hastaneleri tehdit edebilecek unsurların olasılığını azaltabilir.

GÜVENCE

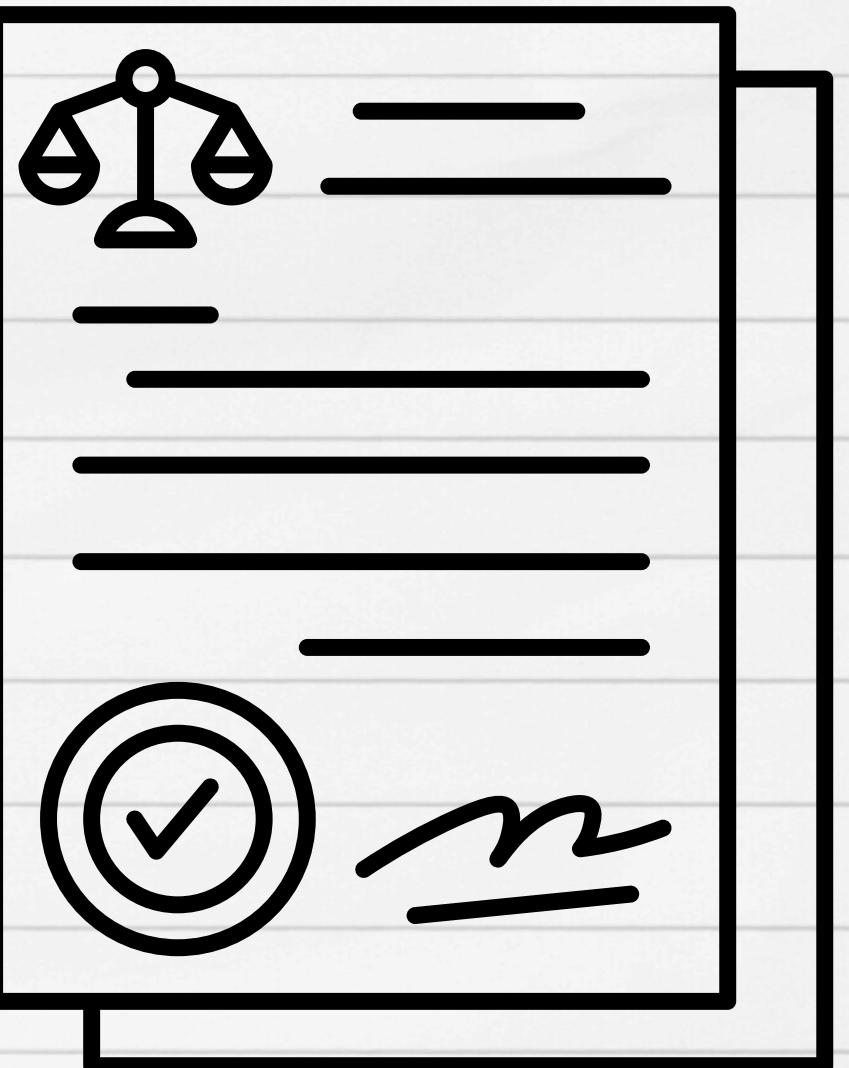
BlockChain sistemi sayesinde hasta, verilerinin hangi hesaplar tarafından işlenebileceğini seçebilecek ve hastanın veri güvenliği güvence altında olacak.



Hukuka Uygunluk

- **KVKK 4. Maddesi:** Dürüstlük, meşru amaç, veri minimizasyonu, güncellik, saklama süresi, güvenlik.
- **GDPR (General Data Protection Regulation) 7. maddesi:** Meşru amaç, dürüstlük, güncellik.

Blockchain teknolojisi bu iki madde uyarınca verilerin amacına uygun bir şekilde kaydedilmesini ve takip edilmesini sağlar.



Hukuka Uygunluk

- KVKK'nın 7. maddesi: Verilerin silinmesi, yok edilmesi anonimleştirilmesi.
- GDPR'nin 17. maddesi: 'Right To Erasure' veya "unutulma hakkı"

'Zero Knowledge' ile kullanıcı verilerini "silmek" istediğiinde, şifreleme anahtarını imha edebilir. Bu işlem, veriyi pratik olarak okunamaz hale getirir, çünkü anahtar olmadan veriye erişim imkansız olur. Böylece verilerin yok edilmesine ve unutulma hakkına uymuş olur.





**CHAIN
CARE**

TEŞEKKÜRLER