

# **Bütünlük Politikaları**

Yrd. Doç. Dr. Özgü Can

# Bütünlük Politikası

- Bütünlük politikaları, **bütünlüğün sağlanmasına** odaklanmaktadır.
- Bir çok ticari ve endüstriyel firma verinin ortaya çıkmasından çok **verinin doğruluğu** ile ilgilenmektedir.

# Amaç

Lipner\* tarafından tanımlanan gereksinimler:

1. Kullanıcılar kendi programlarına yazamazlar, ancak mevcut üretim programlarını ve veritabanlarını kullanmalıdır.
2. Programcılar, programlarının geliştirimini ve testini üretimin olmadığı bir sistemde gerçekleştirmelidir.
  - Veriye erişmek istiyorlarsa, veri programcılara özel süreçler yardımı ile verilmelidir. Ancak, programcılar bu veriyi kendi geliştirme sistemlerinde kullanmalıdır.

\* *Non-Discretionary Controls for Commercial Applications, 1982*

# Amaç

3. Programlar, geliştirme ortamından üretim sistemine yüklenirken özel süreçler takip edilmelidir.
4. 3. adımdaki özel süreç kontrol edilmeli ve denetlenmelidir (auditing).
5. Yöneticiler ve denetleyiciler (auditor), oluşturulan sistem durumuna (state) ve günlüklerine (logs) erişim yetkilerine sahip olmalıdır.

# Amaç

- Bu gereksinimler çeşitli işlem prensiplerini ortaya çıkarmaktadır:
  - Görev Ayrılığı (Separation of Duty)
  - İşlev Ayrılığı (Separation of Function)
  - Denetleme (Auditing)

# Görev Ayrılığı (Separation of Duty)

Bir işlemi gerçekleştirmek için  
iki ya da daha fazla adıma ihtiyaç varsa,  
**en az iki farklı kişi** adımları gerçekleştirmelidir.

# Görev Ayrılığı (Separation of Duty)

- Geliştirilen programın, geliştirme ortamından üretim sistemine taşınması kritik bir işlemdir.
- Program geliştirilirken;
  - Uygulama programcısı geçersiz bir varsayım (assumption) yaptığında



Programın yüklenmesi farklı bir kişi tarafından yapıldığında, **hatanın yakalanması ihtimali** daha fazladır.

# İşlev Ayrılığı (Separation of Function)

- Uygulama geliştiriciler;
  - Üretim verisini bozmamak için, yeni programları üretim sistemlerinde geliştirmemektedir.
  - Üretim verisi, geliştirme sisteminde de kullanılmamaktadır.
- Verinin duyarlılığına bağlı olarak, uygulama geliştiriciler ve test işlemlerini gerçekleştirenler “**sanitized**” üretim verisi kullanmaktadır.

**NOT:** Geliştirme ortamı, mevcut üretim ortamı ile mümkün olduğunca benzer olmalıdır.



# Denetleme (Auditing)

- Denetleme, **hangi** **eylemlerin** gerçekleştirildiğini ve bu eylemlerin **kimler tarafından** gerçekleştirildiğini belirlemek için *sistemin analiz edilmesi sürecidir*.
- Denetlemenin temeli günlüklerdir (logs).
- Programlar, geliştirme sisteminden üretim sistemine taşınırken, günlükler ve denetleme önem kazanmaktadır.

# Bütünlük Politikası

- Ticari ortam askeri ortamdan farklıdır.
- Askeri ortam;
  - Erişim yetkileri, kategoriler ve güvenlik seviyelerine göre belirlenmektedir.
- Ticari ortam;
  - Kişinin belirli bir bilgiye ihtiyacı varsa, bu bilgi ona verilir.

# Bütünlük Politikası

- Ticari ortam;
  - **Bell-LaPadula** kullanılabilir. Ancak;
    - Çok fazla sayıda **kategoriye ve güvenlik seviyesine** ihtiyaç duyulacaktır.



**Kategori ve güvenlik seviyesi yönetimi zorlaşır.**


**Modelin karmaşıklığını artacaktır.**

# Bütünlük Politikası

- Güvenlik seviyeleri ve kategoriler;
  - Askeri ortamda → Merkezi (Centralized)
  - Ticari ortamda → Dağıtılmış (Decentralized)

# Bütünlük Politikası

- Ticari ortamda;
  - Duyarlı bilgi gizli tutulmaktadır.
  - Sınırlı miktarda bilgi kamuya açılmaktadır.



Sınırlı miktardaki veriden duyarlı bilgiye ulaşılabilir.

Uygulanacak model tarafından engellemelidir.

# Biba Modeli

- 1977 yılında, Kenneth J. Biba tarafından geliştirilmiştir.
- Formel bir durum geçiş sistemidir.
- Erişim denetim kuralları, veri bütünlüğünü garantiler.

# Biba Modeli

- Özneler ve veri, bütünlük seviyelerine göre gruplanır.
- **Yüksek bütünlük seviyesindeki** veri, **alt bütünlük seviyesinde** ki veriden daha **güvenilirdir** (*trustworthy*).

# Biba Modeli

- **S**: Özneler kümesi
- **O**: Nesneler kümesi
- **I**: Bütünlük seviyeleri

$i(s) = i(o)$  ise, okuma  
ve yazma işlemlerine  
izin verilmektedir.

1. Eğer  $i(s) \leq i(o)$  ise,  $s \in S$  öznesi  $o \in O$  nesnesi okuyabilir. [**No Read Down**]
2. Eğer  $i(o) \leq i(s)$  ise,  $s \in S$  öznesi  $o \in O$  nesnesine yazabilir. [**No Write Up**]
3. Eğer  $i(s_2) \leq i(s_1)$  ise,  $s_1 \in S$  öznesi  $s_2 \in S$  yi yürütebilir (execute).



# Biba Modeli

## Simple Integrity Axiom

- [**No Read Down**] Özne, kendi bütünlük seviyesinden daha alt bir bütünlük seviyesine sahip olan bir nesneyi okuyamaz.

## \* (Star) Integrity Axiom

- [**No Write Up**] Özne, kendi bütünlük seviyesinden daha yukarıda bir bütünlük seviyesine sahip olan nesneye yazamaz.

# Bell LaPadula & Biba

## BLP

- Gizliliği korumaktadır.
- Okuma (Reads) ile ilgilenir.
- **No Read Up, No Write Down**
- **High Water Mark**

Farklı güvenlik seviyelerindeki  
iki nesne birleştirildiğinde



Birleştirilmiş nesne **en yüksek**  
güvenlik seviyesindeki nesnenin  
güvenlik sınıflandırılmasına  
sahip olur.

## Biba

- Bütünlüğü korumaktadır.
- Yazma (Writes) ile ilgilenir.
- **No Read Down, No Write Up**
- **Low Water Mark**

Farklı güvenlik seviyelerindeki  
iki nesne birleştirildiğinde



Birleştirilmiş nesne **en düşük**  
güvenlik seviyesindeki nesnenin  
güvenlik sınıflandırılmasına  
sahip olur.

# Clark-Wilson Modeli

- 1987 yılında, David Clark ve David Wilson tarafından geliştirilmiştir.
- Ticari ortamda ki temel endişe;
  - Verinin bütünlüğüve
  - Veri üzerinde gerçekleştirilen işlemlerin bütünlüğü

# Clark-Wilson Modeli

- İşlemler sonucunda **tutarlılık** (*consistency*) koşulları sağlanmalıdır.
- İyi biçimlendirilmiş işlemlerde, sistem tutarlı bir durumdan başka bir tutarlı duruma geçecektir.
  - ÖR: Bir hesaptan diğer bir hesaba para aktarımı
    - Her bir işlemten sonra, hesapların tutarlı bir durumda olması gerekir.

# Clark-Wilson Modeli

- ÖR: Kuruma gelen bir faturanın ödenmesi
  1. Ödeme işlemini talep edilmesi ve ödemenin hangi hesaptan yapılacağının belirlenmesi
  2. Faturanın geçerliliğinin onaylanması
  3. Ödeme emrinin verilmesi
- Sahte fatura düzenlenmesini önlemek için
  - Bu işlem birden fazla kişi tarafından yapılmalıdır



**Görev Ayrılığı (Separation of Duty)**

# Clark-Wilson Modeli

- İşlem hareketlerinin (transaction) doğru bir şekilde gerçekleştirildiği onaylanmalıdır (certify).
- Görev ayrılığında prensip:
  - Onayı verenin ve işlemi gerçekleştirenin **farklı kişiler** olmasıdır.

# Clark-Wilson Modeli

- Bir işlem hareketinin veriyi bozması için:

- iki farklı kişinin de aynı hatayı yapması

ya da

- kişilerin iyi biçimlendirilmiş işlem hareketine onay vermek için anlaşmış

olması gerekir.

# Clark-Wilson Modeli

- Clark-Wilson modelinde;

Sistemdeki veriler:

- **Kısıtlandırılmış Veri Öğeleri**

*Constrained Data Items (CDI)*

- **Kısıtlandırılmamış Veri Öğeleri**

*Unconstrained Data Items (UDI)*

olarak kümelenir.



# Clark-Wilson Modeli

- Kısıtlandırılmış veri öğeleri (CDI):
  - Bütünlük kontrollerine maruz kalan veriler
- Kısıtlandırılmamış veri öğeleri (UDI):
  - Bütünlük kontrollerine maruz kalmayan veriler
- Bütünlük kısıtları → CDI'ların değerlerini kısıtlar.

# Clark-Wilson Modeli

- ÖR: Banka hesapları
  - Banka hesaplarının bakiyeleri → CDI
    - Bütünlüğü bankacılık işlemleri için önemlidir.
  - Banka müşterisine verilen hediyeler → UDI
    - Bütünlüğü bankacılık işlemleri için önemli değildir.

# Clark-Wilson Modeli

Clark-Wilson modeli iki yordam kümesi tanımlar:

## 1. **Bütünlük Doğrulama Yordamı**

*Integrity Verification Procedure – IVP*

## 2. **Değişim Yordamı**

*Transformation Procedure - TP*

# Clark-Wilson Modeli

- Bütünlük doğrulama yordamı (IVP):
  - IVP çalıştığı zaman, CDI'ların bütünlük kısıtlarına uyduğunu test eder.
  - Bu durumda sistem geçerli (valid) durumdadır.

# Clark-Wilson Modeli

- Değişim yordamı (TP):
  - Sistemdeki verinin geçerli durumunu başka bir geçerli duruma değiştirir.
  - İyi biçimlendirilmiş işlem hareketlerini gerçekleştirir.

# Clark-Wilson Modeli

- Model, [9 adet] kural kümelerinden oluşur:
  - Onaylama Kuralları (Certification Rules, **CR**) [5 adet]
  - Uygulama Kuralları (Enforcement Rules, **ER**) [4 adet]

# Clark-Wilson Modeli

ÖR: Banka hesabı

- Hesapların bakiyeleri → CDI
- Hesap bakiyelerinin dengeli olduğunun kontrol edilmesi → IVP
- Para çekme, para yatırma ve para transferi işlemleri → TP
- Hesapların doğru bir şekilde yönetildiğinin garantilenmesi gerekmektedir.

# Clark-Wilson Modeli

Clark-Wilson modeli, bu gereksinimleri iki **onaylama kuralı (certification rule, CR)** ile sağlamaktadır:

- **CR1:** Herhangi bir IVP çalıştığında, bütün CDI'ların geçerli bir durumda olduğunu garantilemelidir.
- **CR2:** TP, birbiri ile ilgili bazı CDI kümeleri için, bu CDI'ları bir geçerli durumdan başka bir geçerli duruma dönüştürmelidir.



# Clark-Wilson Modeli

- CR2, CDI'lar kümesini belirli bir TP ile ilişkilendiren bir onaylanmış ilişki tanımlar.

**C:** Onaylanmış ilişki (certified relation)

Banka örneği için;

$(\text{bakiye}, \text{hesap}_1), (\text{bakiye}, \text{hesap}_2) \dots (\text{bakiye}, \text{hesap}_n) \in C$

# Clark-Wilson Modeli

- TP'nin CDI üzerinde çalışma onayı yoksa, CDI'yı bozabilir.
- Bu nedenle;
  - Sistem, TP'nin CDI üzerinde çalışma onayı yoksa, CDI üzerinde çalışmasını önlemelidir.



**Uygulama kuralı (enforcement rule - ER)** tanımlanır.

# Clark-Wilson Modeli

- **ER1:** Sistem onaylanmış ilişkilerin sürekliliğini sağlamalı ve sadece onaylanmış TP'lerin CDI üzerinde değişiklik yapabileceğini garantilemelidir.



Eğer bir TP işlemi olan  $f$ , CDI üzerinde işlem gerçekleştirirse  $\rightarrow (f, o) \in C$  'dir.

# Clark-Wilson Modeli

- Odacının banka müşterilerinin hesapları üzerinde işlem yapmasına izin verilmemektedir.
- Bu nedenle;

Model, TP'yi gerçekleştirecek kullanıcıyı belirtmelidir.



**ER2**

# Clark-Wilson Modeli

- **ER2:** Sistem, her bir TP ve CDI kümesi ile ilgili bir kullanıcıyı ilişkilendirmelidir. TP, ilişkilendirilmiş kullanıcının yerine, CDI'lara erişebilir.

Eğer kullanıcı belirli bir TP ve CDI ile ilişkilendirilmemiş ise, TP kullanıcının yerine, CDI'lara erişemez.



Üçlü (Triple) tanımı → **(kullanıcı, TP, {CDI kümesi})**

# Clark-Wilson Modeli

- (kullanıcı, TP, {CDI kümesi}) üçlüsü;
  - Kullanıcı, TP ve CDI arasında ki ilişkileri belirler.
  - Bu ilişkiye izin verilen dersek: A (allowed)
    - A izin verilen ilişkileri belirtmektedir.

Bu ilişkilerin de onaylanması gerekmektedir.



**CR3**

# Clark-Wilson Modeli

- **CR3:** İzin verilen ilişkiler, görev ayrılığı prensibi gereksinimlerini karşılamalıdır.
- Sistem, kullanıcı kimliğinin doğruluğunu garantilemelidir.



**ER3**

# Clark-Wilson Modeli

- **ER3:** Sistem, TP'ye erişmek isteyen her kullanıcının kimliğini doğrulamalıdır (authenticate).



# Clark-Wilson Modeli

- Model, kullanıcı sisteme bağlandığında **kimlik doğrulama** gerektirmez.
  - Çünkü, kullanıcı UDI'larla ilgili değişiklik yapabilir.
- Kullanıcı, CDI üzerinde değişiklik yapacaksa, bunu TP aracılığı ile gerçekleştirir. Bunun için;
  - Kullanıcı izin verilen (allowed) olarak onaylanmalıdır. → **ER2**
    - Bu işlem için, kullanıcının kimlik doğrulanmasına gereksinim duyulmaktadır. → **ER3**

# Clark-Wilson Modeli

- İşlem hareketi temelli sistemlerde;
  - Her bir işlem günlüklerde (log) tutulur.
    - Böylelikle, denetleyici işlem hareketlerini izleyebilir.
- Clark-Wilson modelinde;
  - Log → CDI olarak değerlendirilir.
  - Her bir TP log'a ekleme yapabilir, ancak üzerine yazamaz.



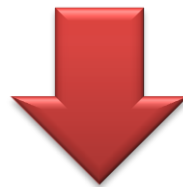
**CR4**

# Clark-Wilson Modeli

- **CR4:** Bütün TP'ler, işlemle ilgili yeterli bilgiyi günlüğe (log) eklemelidir.

# Clark-Wilson Modeli

- Sisteme bir bilgi girildiğinde, bu bilgi güvenilir olmayabilir. → UDI
  - ÖR: Kullanıcının yatırdığı para ile yatırdığı miktar için girdiği tutar birbirine eşit olmayabilir. (*Çelişki*)
- Bu bilginin (UDI), CDI'ya dönüştürülmesi gerekir.



**CR5**

# Clark-Wilson Modeli

- **CR5:** Herhangi bir TP, bir UDI'yi girdi olarak alıp, UDI'nın bütün geçerli değerleri için, sadece **geçerli dönüşümleri/değişimleri gerçekleştirir** ya da **hiçbir dönüşüm/değişim gerçekleştirmez**.

Değişim ya UDI'yi reddeder ya da onu CDI'ya dönüştürür.

# Clark-Wilson Modeli

- **ER4** → ER2 ve ER3'deki ilişkilerin **bütünlüğünün sağlanması** için *görev ayrımını (separation of duty)* uygular.
- Eğer bir kullanıcı TP yaratabiliyor ve bazı varlıklar ile kendisini bu TP ile ilişkilendirebiliyorsa (**ER3**), bu TP'nin bütünlük kısıtlarını ihlal edecek yetkilendirilmemiş işlemler gerçekleştirmesini sağlayabilir.

**ER4**



# Clark-Wilson Modeli

- **ER4:** Sadece TP'nin onaylayıcısı (certifier) bu TP ile ilişkilendirilmiş varlıklar listesini değiştirebilir.

Bir TP'nin hiçbir onaylayıcısı ya da bu TP ile ilişkilendirilmiş hiçbir bir varlık, bu varlık ile ilgili izinleri yürütemez.

# Clark-Wilson Modelinin Katkıları

1. Kurumlar, veriyi **sınıflandırmak** zorunda değildir.
  - Görev ayrılığı yaklaşımını uygular.
2. Onaylama (certification) kavramı uygulama (enforcement) kavramından farklıdır.
  - Her birinin kendi kural seti vardır.



# Clark-Wilson Modeli

- Clark-Wilson modeli, uygulama kurallarına uyulacağını garantiler.
- Onaylama kuralları;
  - Dış müdahaleye gereksinim duyar.
  - Karmaşıktır.
  - Hataya ya da eksikliklere yatkındır.
    - Çünkü, onaylayıcılar neye güvenileceği ile ilgili varsayımlarda bulunurlar. ← **Modelin zayıflığı**

# Gereksinimler & Clark-Wilson Modeli

**G1** *Kullanıcılar kendi programlarına yazamazlar, ancak mevcut üretim programlarını ve veritabanlarını kullanmalıdır.*

- Kullanıcıların TP'lerin onaylamasını gerçekleştirmeye izni yoksa ve sadece güvenilir personel bunu gerçekleştirebiliyorsa, o zaman **CR5** ve **ER4** bu gereksinimi uygular.

# Gereksinimler & Clark-Wilson Modeli

**G1** *Kullanıcılar kendi programlarına yazamazlar, ancak mevcut üretim programlarını ve veritabanlarını kullanmalıdır.*

- Normal kullanıcılar;
  - Onaylanmış TP'leri yaratamazlar
  - Üretim veritabanına erişmesi için programlara yazamazlar



Mevcut TP'leri ve CDI'ları kullanmalıdırlar.

Üretim  
programları      Üretim  
veritabanları

# Gereksinimler & Clark-Wilson Modeli

**G2** *Programcılar, programlarının geliştirimini ve testini üretimin olmadığı bir sistemde gerçekleştirmelidir.*

- Bu gereksinim yordamsaldır.
  - Teknik kontroller, bunu sağlayamaz.
- Ancak, özel süreç aracılığı ile üretim verisi sağlanması “**sanitize**” için TP kullanımına karşılık gelmektedir.

# Gereksinimler & Clark-Wilson Modeli

**G3** *Programlar, geliştirme ortamından üretim sistemine yüklenirken özel süreçler takip edilmelidir.*

- Programların üretim sistemine kurulması, kurulum için bir TP kullanımını ve onaylama için güvenilir personel gerektirmektedir.

# Gereksinimler & Clark-Wilson Modeli

**G4** *3. adımdaki özel süreç kontrol edilmeli ve denetlenmelidir (auditing).*

- **CR4** → Programın kurulmasındaki denetlemeyi sağlar.
- **ER3** → Kurulumu gerçekleştirecek güvenilir personelin kimlik doğrulmasını gerçekleştirir.
- **CR 5 ve ER4** → Kurulum yordamını denetler.

# Gereksinimler & Clark-Wilson Modeli

**G4** *Yöneticiler ve denetleyiciler (auditor) ,*

*oluşturulan sistem durumuna (state) ve günlüklerine(logs)*

*erişim yetkilerine*

*sahip olmalıdır.*

- Günlük (log), CDI olduğundan, yöneticiler ve denetleyiciler ilgili TP aracılığı ile sistem günlüklerine erişebilirler.
  - Benzer şekilde sistem durumuna da erişebilmektedirler.

# Sonuç

Clark-Wilson modeli,  
Lipner'in tanımladığı  
**gereksinimleri** karşılamaktadır.



# Biba & Clark-Wilson Modeli

## Biba

- Bütünlük seviyelerini nesnelere ve öznelerle bağlamaktadır.

## Clark-Wilson

- Bütünlük seviyelerini nesnelere ve öznelerle bağlamaktadır. Ancak,
- Nesnelerin iki seviyesi vardır:
  - Kısıtlanmış ya da yüksek (CDI)
  - Kısıtlanmamış ya da alt (UDI)
- Öznelerin iki seviyesi:
  - Onaylanmış (TP)
  - Onaylanmamış (Geriye kalan bütün yordamlar)

# Gereksinimler & Clark-Wilson Modeli

İki model arasındaki temel farklılık  
**onaylama kurallarından** kaynaklanmaktadır.

## Biba

- Onaylama kuralı yoktur.
- Güvenilir özneler, sistemin eylemlerinin modelin kurallarına uyduğunu garantiler.
- Güvenilir varlıklar ya da onların eylemlerini onaylayan bir düzen ya da yordam bulunmamaktadır.

## Clark-Wilson

- Varlıkların ve eylemlerin uyması gereken gereksinimler vardır.

# Gereksinimler & Clark-Wilson Modeli

Bütünlük seviyelerinde meydana gelen değişikliklerin idaresi kritiktir.

## Biba

- Güvenilir varlık, bir süreçte iletilen her bir girdiyi bütünlük seviyesi girdiden **daha yüksek** olan bir süreçte iletir.
- Pratik değildir.

## Clark-Wilson

- Güvenilir varlık, verinin **daha yüksek** bir seviyeye iletilmesini onaylamalıdır.
- Güvenilir varlık her veriyi onaylamaz.
  - Verinin daha yüksek seviyeye iletimi ile ilgili olan metodu onaylaması yeterlidir.
  - Böylece, veri iletilebilecektir.
- Daha pratiktir.