

# A szolgáltatásbiztonság alapfogalmai

Majzik István  
majzik@mit.bme.hu

<http://www.mit.bme.hu/oktatas/targyak/vimim146/>

# Tartalomjegyzék

- A szolgáltatásbiztonság fogalma
- A szolgáltatásbiztonságot befolyásoló tényezők
- A szolgáltatásbiztonság eszközei

# Motiváció: Hibamentes működés

- Szolgáltatási szint szerződések (SLA):
  - Ügyfél által elvárt jellemzők (pl. rendelkezésre állás)
  - Telekom szolgáltatások szerver rendszerei („carrier grade”):  
„Öt kilences”: 99,999% (5 perc/év kiesés)
- Biztonságkritikus rendszerek:
  - Szabvány előírások a hibák gyakoriságára
  - Biztonságintegritási szintek (Safety Integrity Level)

SIL	Biztonságkritikus funkció hibája / óra
1	$10^{-6} \leq \text{THR} < 10^{-5}$
2	$10^{-7} \leq \text{THR} < 10^{-6}$
3	$10^{-8} \leq \text{THR} < 10^{-7}$
4	$10^{-9} \leq \text{THR} < 10^{-8}$

Ha 15 év az élettartam, akkor ez alatt kb. 750 berendezésből 1-ben lesz hiba

Hiba nélküli működés ~ 11.000 év??

# Elkerülhetetlen: Hibahatások

**Fejlesztési folyamat**



**Működő termék**



- Tervezési hibák
- Implementációs hibák



- Hardver hibák
- Konfigurációs hibák
- Kezelői hibák

# Elkerülhetetlen: Hibahatások

**Fejlesztési folyamat**



**Működő termék**



- Tervezési hibák
- Implementációs hibák



- Hardver hibák
- Konfigurációs hibák
- Kezelői hibák

**Fejlesztési folyamat jellemzői:**

- Jobb minőségbiztosítás, jobb módszertanok
- De növekvő bonyolultság, nehezebb ellenőrzés

**Szokásos becsült értékek 1000 kódsorra:**

- Jó kézi fejlesztés és tesztelés: <10 hiba marad
- Automatizált fejlesztés: ~1-2 hiba marad
- Formális módszerek használata: <1 hiba marad

# Elkerülhetetlen: Hibahatások

**Fejlesztési folyamat**



**Működő termék**



- Tervezési hibák
- Implementációs hibák



- Hardver hibák
- Konfigurációs hibák
- Kezelői hibák

## Technológia korlátai:

- Jobb paraméterek, jobb anyagok
- De növekvő bonyolultság (érzékenység)

## Szokásos becsült értékek:

- CPU:  $10^{-5}$ ... $10^{-6}$  hiba/óra
- RAM:  $10^{-4}$ ... $10^{-5}$  hiba/óra
- LCD: ~ 2...3 év élettartam

# Elkerülhetetlen: Hibahatások

**Fejlesztési folyamat**



**Működő termék**



- Tervezési hibák
- Implementációs hibák

- Hardver hibák
- Konfigurációs hibák
- Kezelői hibák



**Verifikáció és  
validáció a  
tervezés során**

**Hibatűrés  
működés közben**

# A hibamentesség jellemzése

- Felhasználó: **Szolgáltatás** jellemzői érdeklik
  - Szolgáltatásminőség:
    - Használhatóság, rendelkezésre állás, javíthatóság,...
  - Termékminőség: ezt nyújtja a gyártó
    - Előállítási folyamat (ISO 9000)
- **Szolgáltatásbiztonság** (dependability):
  - Milyen biztonsággal képes ellátni feladatait a rendszer?
  - **Képesség: igazoltan bízni lehet a szolgáltatásban**
    - Igazoltan: elemzésen, méréseken alapul
    - Bizalom: szolgáltatás az igényeket kielégíti
  - Összetett fogalom

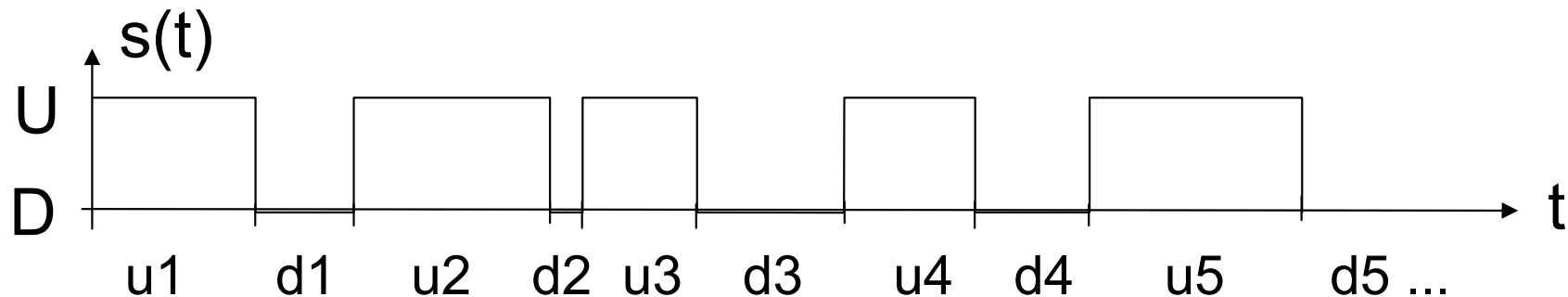


# Szolgáltatásbiztonság jellemzői

- Alapjellemzők:
  - **Rendelkezésre állás**: (javítva) használatra kész szolgáltatás
  - **Megbízhatóság**: folyamatos hibamentes szolgáltatás
  - **Biztonságosság**: katasztrofális következmények (baleset, káreset) nélküli szolgáltatás
  - **Integritás**: hibás változ(tat)ás elkerülése
  - **Karbantarthatóság**: javítás és fejlesztés lehetősége
- (Adat)biztonsághoz:
  - **Bizalmasság**: nincs jogosulatlan információközlés
- Terjedő fogalom: **Resilience**
  - Szolgáltatásbiztonság + adatbiztonság + dinamikus (adaptív, mobil, ad-hoc) működés

# Megbízhatósági mértékek: Várható értékek

- **Állapot particionálás:**  $s(t)$  rendszerállapot
  - Hibás (**D**) - Hibamentes (**U**) állapotpartíció



- **Várható értékek:**
  - **Első hiba bekövetkezése:**  $MTFF = E\{u_1\}$   
(mean time to first failure)
  - **Hibamentes működési idő:**  $MUT = E\{u_i\}$   
(mean up time)
  - **Hibás állapot ideje:**  $MDT = E\{d_i\}$   
(mean down time)
  - **Hibák közötti idő:**  $MTBF = MUT + MDT$   
(mean time between failures)

# Megbízhatósági mértékek: Időfüggvények

- Valószínűség időfüggvények

- Rendelkezésre állás:

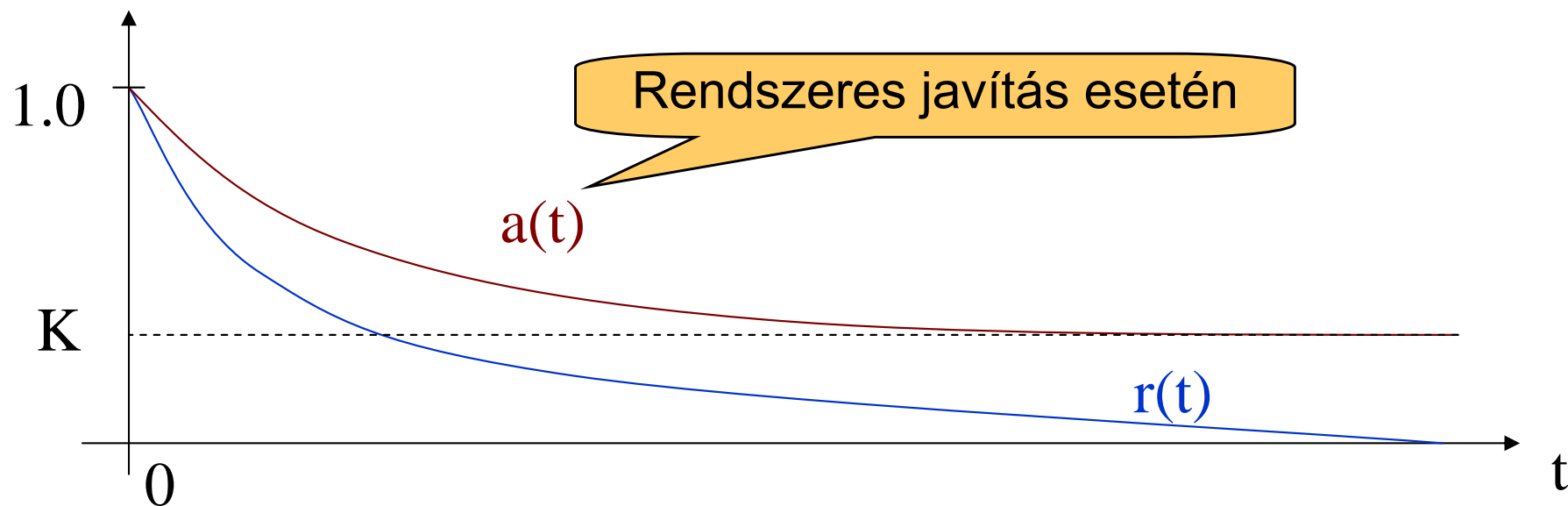
$$a(t) = P\{ s(t) \in U \} \quad (\text{közben meghibásodhat})$$

- Készenlét:

$$K = \lim_{t \rightarrow \infty} a(t) \quad (\text{aszimptotikus})$$

- Megbízhatóság:

$$r(t) = P\{ s(t') \in U, \forall t' < t \} \quad (\text{nem hibásodhat meg})$$



# Megbízhatósági mértékek: Alkatrész szinten

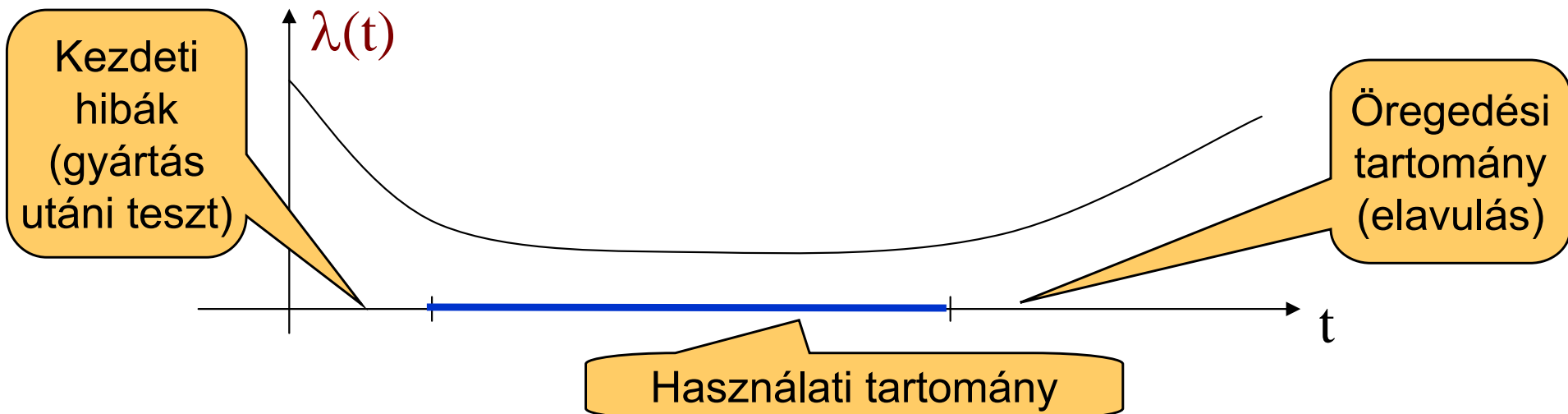
- **Meghibásodási tényező (gyakoriság):**
  - A rendszer mekkora valószínűséggel fog éppen  $t$ -ben elromlani, feltéve, hogy  $t$ -ig jól működött:

$$\lambda(t) = \frac{P\{s(t + \Delta t) \in D \mid s(t) \in U\}}{\Delta t}, \quad \text{miközben } \Delta t \rightarrow 0$$

másként felírva:

$$\lambda(t) = -\frac{1}{r(t)} \frac{dr(t)}{dt}, \quad \text{így } r(t) = e^{-\int_0^t \lambda(t) dt}$$

- Elektronikai alkatrészekre:



# Alkatrészek használati tartománya

## Elektronikai alkatrészek használati tartományában:

- Konstans a meghibásodási tényező:

$$\lambda(t) = \lambda$$

- Megbízhatóság:

$$r(t) = e^{-\lambda t}$$

- Első hiba bekövetkezése:

$$\text{MTFF} = \frac{1}{\lambda}$$

# Rendelkezésre állás követelményei

Rendelkezésre állás	Max. kiesés egy év alatt
99%	~ 3,5 nap
99,9%	~ 9 óra
99,99% („4 kilences”)	~ 1 óra
99,999% („5 kilences”)	~ 5 perc
99,9999% („6 kilences”)	~ 32 másodperc
99,99999%	~ 3 másodperc

**Munkaállomásokból álló elosztott rendszer,  
minden munkaállomás 95%-os rendelkezésre állású:**

- 1 szgép: 95% a rendelkezésre állás
- 2 szgép: 90%
- 5 szgép: 77%
- 10 szgép: 60%

# Tartalomjegyzék

- A szolgáltatásbiztonság fogalma
- A szolgáltatásbiztonságot befolyásoló tényezők
- A szolgáltatásbiztonság eszközei

# Befolyásoló tényezők

- **Hibajelenség** (failure):  
A specifikációnak nem megfelelő **szolgáltatás**
  - Értékbeli / időzítésbeli, katasztrofális / „jóindulatú”
- **Hiba** (error):  
Hibajelenséghez vezető **rendszerállapot**
  - Lappangó → detektált
- **Meghibásodás** (fault):  
A hiba feltételezett **oka**
  - **Hatás**: alvó → aktív
  - **Fajta**: véletlen vagy szándékos, időleges vagy állandósult
  - **Eredet**: fizikai/emberi, belső/külső, tervezési/működési



# Tipikus meghibásodások

Fajta	Eredet	Hely	Fázis	Idő	Példa
Véletlen	Fizikai	Belső	Működés	Állandó- sult	Komponens hiba
Véletlen	Fizikai	Külső	Működés	Időleges	Tranziens hiba
Véletlen	Emberi	Belső	Tervezés	Állandó- sult	Tervezési hiba
Véletlen	Emberi	Külső	Működés	Időleges	Kezelési hiba
Szándé- kos	Emberi	Külső	Működés	Állandó- sult	Rongálás
Szándé- kos	Emberi	Külső	Működés	Időleges	Behatolás

# Szoftver hibák

- Szoftver hiba: **Állandósult, tervezési** hiba
- **Aktiválás** a működési profil függvénye
  - Adott bemeneti tartomány (program útvonal) aktivál
- **Becslési módszerek:**
  - Megbízhatóság arányos:  
**Tesztelés után bennmaradó hibák számával**
  - Bennmaradó hibák száma arányos:  
**Időegység alatt detektált hibák számával** a tesztelés végén
  - Statisztikai módszerekkel becsülhető,  
meddig kell a tesztelést folytatni  
adott megbízhatóság eléréséhez

# Hatáslánc

- **Meghibásodás → Hiba → Hibajelenség**
  - Pl. szoftver esetén:
    - Meghibásodás: progr. hiba: csökkentés helyett növel
    - Hiba: vezérlés ráfut, változó értéke hibás lesz
    - Hibajelenség: számítás végeredménye rossz
  - Pl. hardver esetén:
    - Meghibásodás: kozmikus sugárzás egy bitet átbillent
    - Hiba: hibás memóriacella olvasása
    - Hibajelenség: robotkar a falnak ütközik
- **Rendszer hierarchiaszint szerepe**
  - Alsó szintű **hibajelenség** felsőbb szinten **meghibásodás**
    - Kimenet beragadása egy chip szintjén hibajelenség
    - Rendszer szintjén meghibásodás (chip mint komponens)

# A hatáslánc befolyásolása

- **Meghibásodási tényező csökkentése**
  - Jobb minőségű komponensek
  - Szigorúbb fejlesztési folyamat (ellenőrzés, tesztelés)
  - A meghibásodás-mentesség nem garantálható (csökkenő chipméretek, bonyolultabb programok)
- **Hibajelenség kialakulásának megakadályozása**
  - Rendszerstruktúra kialakítása: redundancia
- **Hibatípusok:**
  - Előre figyelembe vehető hibák:  
optimális **kezelés a tervezési folyamat során**
  - Előre figyelembe nem vehető hibák:  
megfelelő **rendszerstruktúra** kialakítása szükséges

# Tartalomjegyzék

- A szolgáltatásbiztonság fogalma
- A szolgáltatásbiztonságot befolyásoló tényezők
- A szolgáltatásbiztonság eszközei

# Eszközök

- **Hiba megelőzés:** Meghibásodás megakadályozása
  - Fizikai hibák: jó minőségű alkatrészek, árnyékolás,...
  - Tervezési hibák: **verifikáció, jól meghatározott folyamatok**
- **Hiba megszüntetés:**
  - Prototípus fázis: **tesztelés**, diagnosztika, javítás
  - Működés közben: **monitorozás**, javítás
- **Hibatűrés:** Szolgáltatást nyújtani hiba esetén is
  - Működés közben: **hibakezelés, redundancia**
- **Hiba előrejelzés:** Hibák és hatásuk becslése
  - Mérés és „jóslás”, megelőző karbantartás

# Hiba megelőzés és megszüntetés

## A tervezés során végzett ellenőrzések

- **Verifikáció (igazolás):** „Jól tervezem-e a rendszert?”  
A rendszer(modell) megfelel-e a specifikációnak
  - **Formális verifikáció:** a rendszermodell és a követelmények matematikai objektumok
  - **Tesztelés:** végrehajtható modell vagy kód
  - **Szimuláció:** rendszer és környezet modellje alapján
- **Validáció (érvényesítés):** „Jó rendszert terveztem-e?”  
A rendszer megfelel-e az elvárásoknak
  - **Validációs tesztelés:** prototípus, termék alapján
  - Szimuláció **valós környezetben**
  - **Mérések**

# Hibatűrés

- Akármilyen jó is az ellenőrzés a tervezés során, a szolgáltatásbiztonság nem garantálható
- Fel kell készülni a **működés közbeni** hibákra!
  - **Időleges hardver** hibák (ld. zavarérzékenység)
  - **Fel nem derített (teszteletlen) szoftver** hibák
  - Figyelembe nem vett **komplex interakciók**
- **Hibatűrés**: Szolgáltatást nyújtani hiba esetén is
  - Működés közbeni autonóm hibakezelés
  - Beavatkozás a **meghibásodás** → **hibajelenség** láncba
  - Rendszertechnikai megoldások (+ megbízható alkatrész)
- **Alapfeltétel: Redundancia** (tartalékolás)
  - Többszörös erőforrások a hibás komponensek kiváltására



# A redundancia megjelenése

## 1. Hardver redundancia

- Többlet hardver erőforrások

## 2. Szoftver redundancia

- Többlet szoftver modulok

## 3. Információ redundancia

- Többlet információ a hibajavítás érdekében

## 4. Idő redundancia

- Ismételt végrehajtás, hibakezelés többlet ideje

Együttes megjelenés!

# A redundancia típusainak összehasonlítása

<b>Redundancia / tulajdonság</b>	<b>Hideg tartalék (passzív redundancia)</b>	<b>Langyos tartalék (másodlagos funkciók)</b>	<b>Meleg tartalék (aktív redundancia)</b>
<b>Alapelv</b>	Csak hiba esetén aktiválva	Csökkentett terheléssel működik	Ugyanúgy működik, mint az elsődleges
<b>Előnye</b>	Nem hibásodik meg a passzív komponens	Kisebb meghibásodási tényező	Gyorsan átveheti az elsődleges helyét
<b>Hátránya</b>	Lassan veszi át az elsődleges helyét	Közepes sebességű feladat átvétel	Azonos meghibásodási tényező
<b>Példa</b>	Kikapcsolt tartalék számítógép	Naplózó számítógép belép elsődlegesként	Árnyék számítógép

# 1. Hardver redundancia

- Hardver állandósult hibák esetén
  - Cél az egyszeres hibapont (single point of failure, SPOF) elkerülése
- Megjelenése:
  - **Eleve a rendszerben lévő** redundáns komponensek
    - Elosztott rendszer, adaptív átkonfigurálás
  - Hibatűréshez **betervezett** redundancia (tartalékolás)
    - Kettőzés
    - TMR: Triple-modular redundancy
    - NMR: N-modular redundancy

## 2. Szoftver redundancia

Használat:

### 1. Szoftver tervezési hibák esetén:

- Ismételt végrehajtás nem segít...
- Redundáns modulok: **eltérő tervezés** szükséges **variánsok**: azonos specifikáció, de
  - eltérő algoritmus, adatstruktúrák
  - más fejlesztési környezet, programnyelv
  - elszigetelt fejlesztés

### 2. Időleges (hardver) hibák esetén:

- Ismételt végrehajtás esetén a hiba nem jelentkezik
- Hibahatások kiküszöbölése a fontos

# 3. Információ redundancia

- **Hibajavító kódolás**

- Memóriák, háttértárak, adatátvitel esetén
- Pl. Hamming-kód, Reed-Solomon kódok
- Korlátozott hibajavító képesség
  - Hosszú idejű adatstabilitás rossz lehet (“felgyűlnek” a hibák)
  - Háttértárak: “memory scrubbing”  
folyamatos **olvasás és javítva visszaírás**

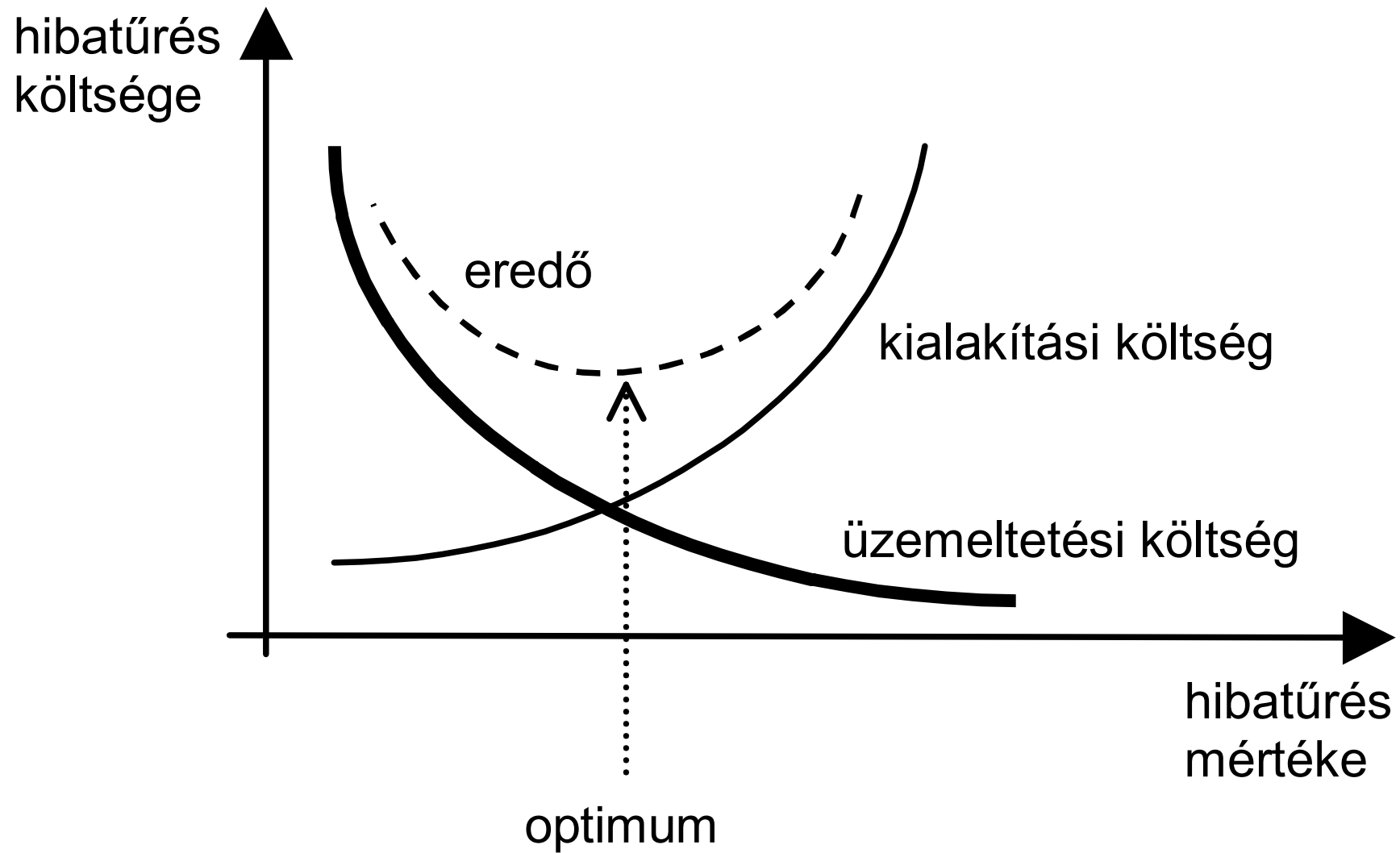
- **Többpéldányos (elosztott) adattárolás**

- Probléma: Hozzáférések konzisztenciájának biztosítása
  - Egypéldányos sorosíthatóság

## 4. Idő redundancia

- Tiszta eset: **utasítás újrapróbálás** (retry)
  - Alacsony hardver szinten: processzor utasítás
  - Időleges hibák esetén hatásos
- Idő redundancia “velejárója” a többi típusnak
  - **Valós idejű rendszerek**: tervezési szempont, hogy mennyire garantálható a hibakezelés ideje
    - Állandósult hardver hibák: maszkolás, meleg tartalék
    - Időleges hardver hibák: előrelépő helyreállítás
    - Szoftver tervezési hibák: N-verziós programozás

# Költségoptimalizálás



# Összefoglalás

- Szolgáltatásbiztonság

- Jellemzők: Megbízhatóság, rendelkezésre állás, biztonság, integritás, karbantarthatóság
- Hatáslánc: Meghibásodás → hiba → hibajelenség
- Eszközök: Hiba megelőzés, hiba megszüntetés, hibatűrés, hiba előrejelzés

- Hibatűrés

- Redundancia megjelenése: Szoftver, hardver, idő, információ redundancia
- Redundancia típusa: Meleg, langyos, hideg tartalék