Közlekedési automatika

Biztonsági architektúrák

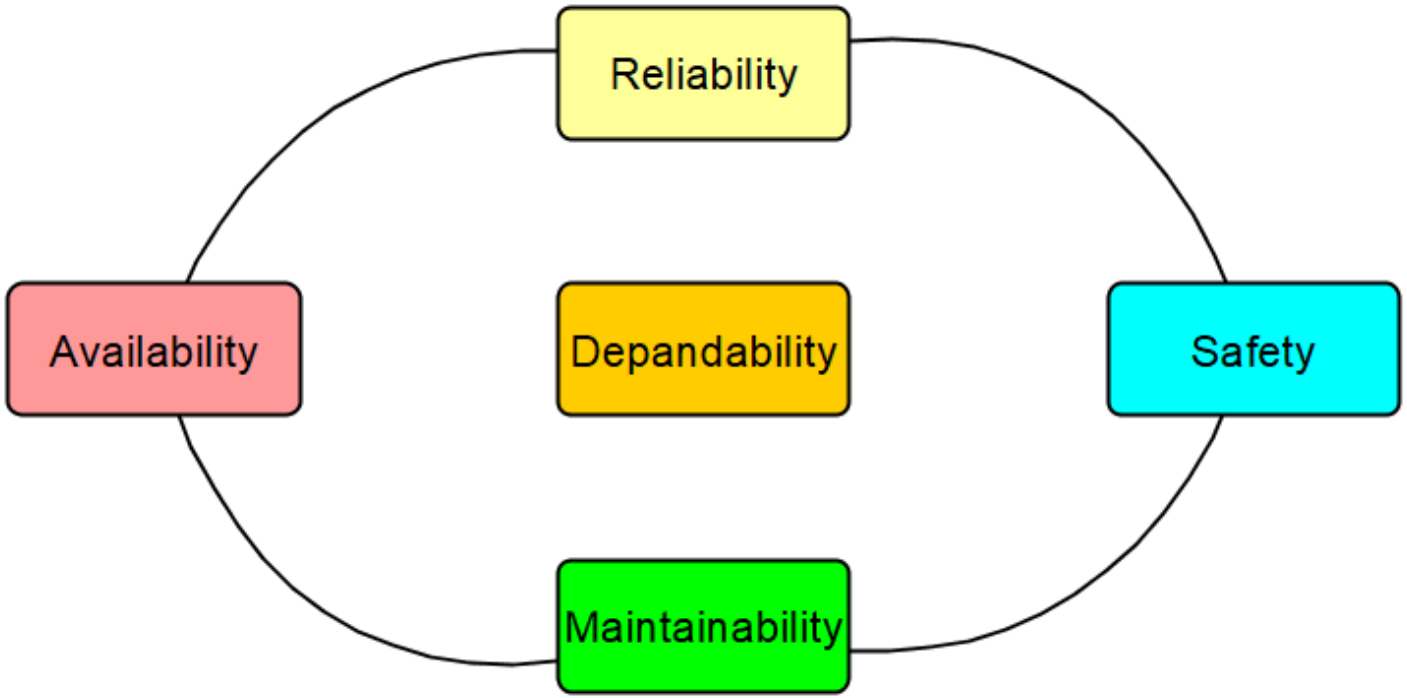
Dr. Sághi Balázs diasora alapján
összeállította, kiegészítette: Lövétei István Ferenc

BME Közlekedés- és Járműirányítási Tanszék
2019

Tartalomjegyzék

- Bevezetés – RAMS,
- Biztonsági architektúrák, alapelvek,
- Biztonsági architektúrák:
  - egy csatornás rendszerek,
  - több csatornás rendszerek,
- Komplex biztonsági architektúrák.

Bevezetés - Depandability



- Depandability:
  - a rendszer azon tulajdonsága, mely lehetővé teszi a szolgáltatása iránti bizalmat,
- Reliability – működőképesség:
  - R(t) – λ – T,
  - elemek, soros és párhuzamos rendszerek ,
  - védelem a véletlenszerű hibák ellen,
- Maintainability – karbantarthatóság:
  - M(t) – ha a rendszer karbantartható,
  - javítások, javítható rendszerek,
- Availability – rendelkezésre állás:
  - A(t), A_{ss} - ha a rendszer javítható,
  - modellezés pl. Markov folyamatokkal,
- Safety- biztonság:
  - biztonságintegritás, fejlesztési módszertanok,
  - életciklus modellek, védelem a szisztematikus hibák ellen,
  - biztonsági stratégiák a megbízhatóság és a rendelkezésre állás növelésére.

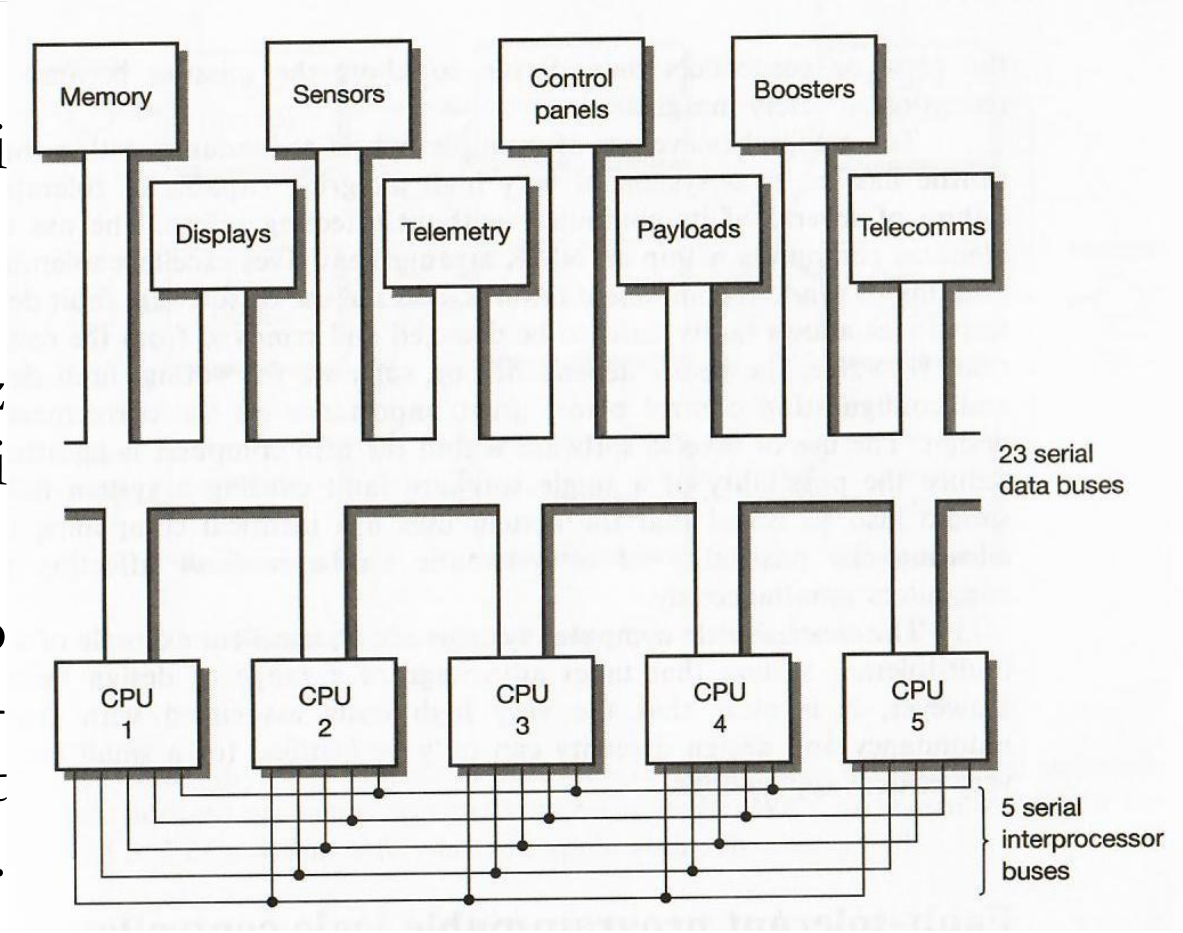
Bevezetés – Biztonsági architektúrák

- Cél:
  - megkívánt megbízhatóság és rendelkezésre állás elérése,
  - az adott biztonságintegritási szinten – THR (TFFR) és SIL (ASIL, DAL) követelmények alapján,
  - az elérhető leghatékonyabb, de szabványos fejlesztési módszerekkel, életciklus modellekkel,
  - a karantarthatóság és a költségek figyelembe vételével.

- Nincs: „Best Practice”:
  - egy alkalmazás (funkció) több, az alkalmazásnak megfelelő biztonsági architektúrával is elérhető,
  - a rendszerek belső felépítése egyszerre több biztonsági stratégiát is alkalmazhat.

Ez meg mi?

- 5 egyedi számítógép,
  - 4 többségi szavazóként konfigurálva,
  - ha a szavazásnál az egyik eltér, a többi „lekapcsolja”,
  - az ötödik számítógép alapesetben a nem-kritikus funkciókat valósítja meg (pl. kommunikáció).



az ábra forrása: Neil Storey, Safety-Critical Computer Systems, Addison-Wesley, 1996, England

Hát ez: ☺

- ha ezen múlik a nemzet büszkesége …. ☺ …. most az 1970-es évekről beszélünk …

az ábra forrása: link

Védelem a véletlenszerű meghibásodások ellen, példák

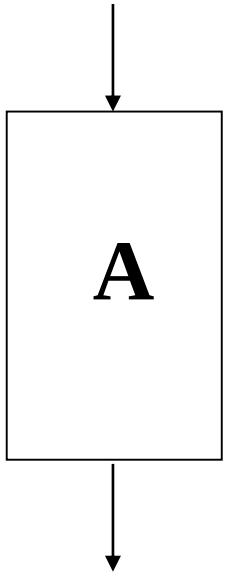
Technikák Intézkedések*	SIL 1	SIL 2	SIL 3	SIL 4
1. A biztonságorientált és nem biztonságorientált rendszerek szétválasztása	R	R	HR	HR
2. Egyszerű elektronikai felépítés ön-teszteléssel és ellenőrzéssel	R	R	-	-
3. Duál elektronikai felépítés	R	R	-	-
4. Összetett fail-safe jellegen alapuló duál elektronikai felépítés fail-safe összehasonlítással	R	R	HR	HR
5. Belső fail-safe jellegen alapuló egyszerű elektronikai felépítés	R	R	HR	HR
6. Reaktív fail-safe jellegen alapuló egyszerű elektronikai felépítés	R	R	HR	HR
7. Diverziter elektronikai struktúra fail-safe összehasonlítással	R	R	HR	HR
8. Az architektúra igazolása a hardver mennyiségi meg-bízhatósági elemzésével	HR	HR	HR	HR

MEGJEGYZÉS: Minden, szürke mezőben feltüntetett technika alternatívát jelent, azaz az „R” azt jelenti, hogy ezen technikák közül legalább egy választása ajánlott.

- Vasútbiztonsági szabványokban definiált biztonsági életciklus modell - MSZ EN 50129

Egy HW csatornás rendszerek, fail – safe működéssel

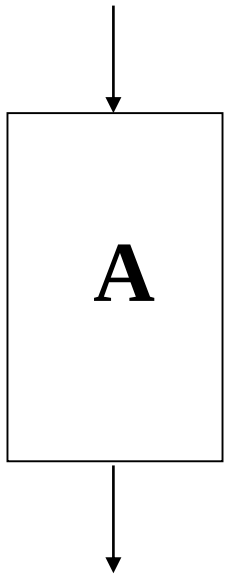
- klasszikus jelfogós berendezések. *(Nem életciklus modell szerint tervezettek.)*
- fail-safe: biztonsági állapotba való kényszerítés – energiamentes állapot,
- belső fail-safe jelleggel,
- önellenőrzések, öntesztek.

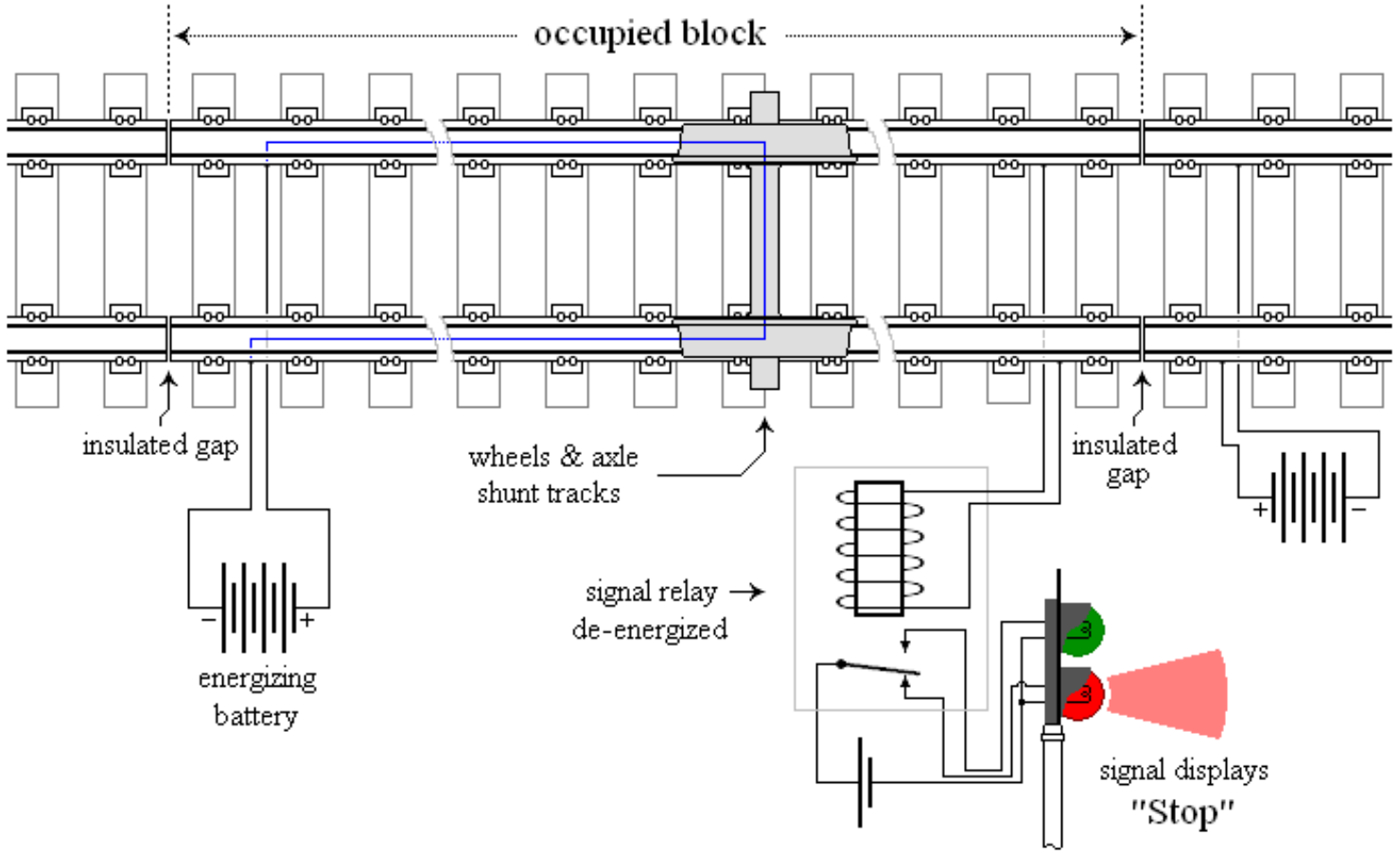


forrás: kjit.bme.hu

Belső fail-safe jellegen alapuló egyszerű elektronikai felépítés, inherens fail – safe működéssel

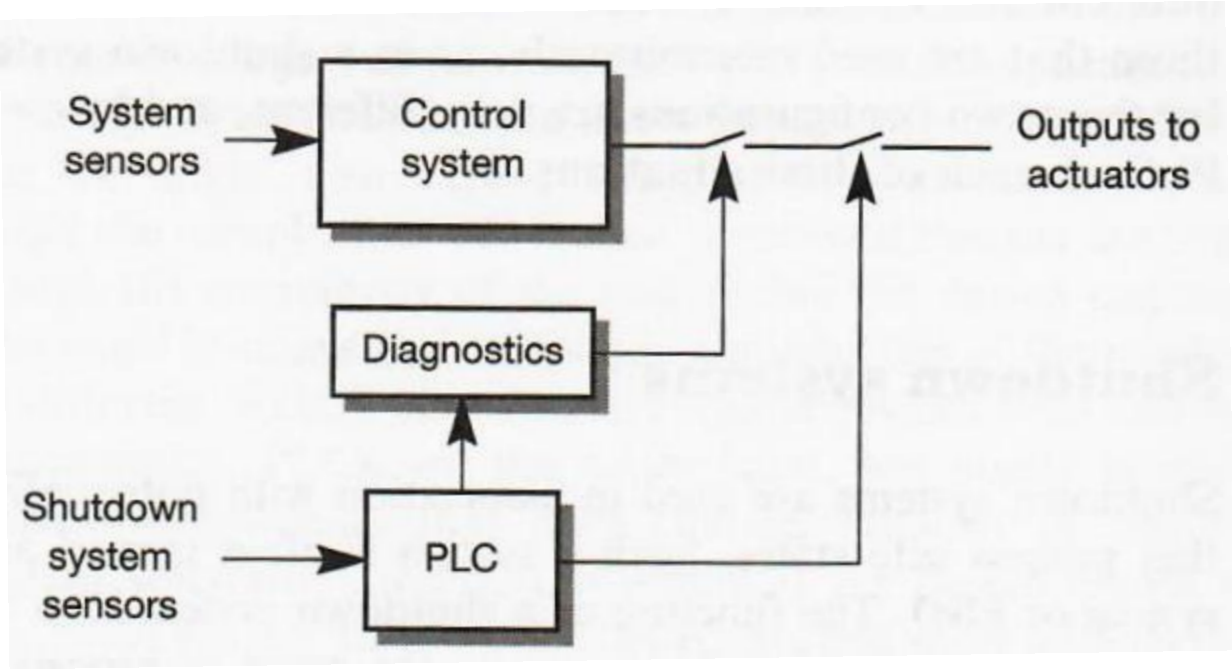
- egy hiba esetén a biztonság felé tér el.





1 HW, 1 SW

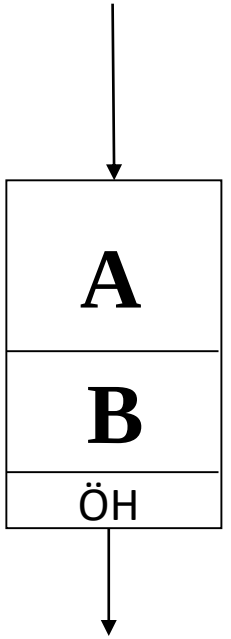
- lehet, hogy a szoftver jól van megírva,
- de a hardver véletlen hibái ellen semmi nem véd.
- a legkisebb hiba esetén szintén a biztonság felé tér el,
- pl. biztonsági lekapcsolás – diagnosztika szerepe nagy!



az ábra forrása: Neil Storey, Safety-Critical Computer Systems, Addison-Wesley, 1996, England

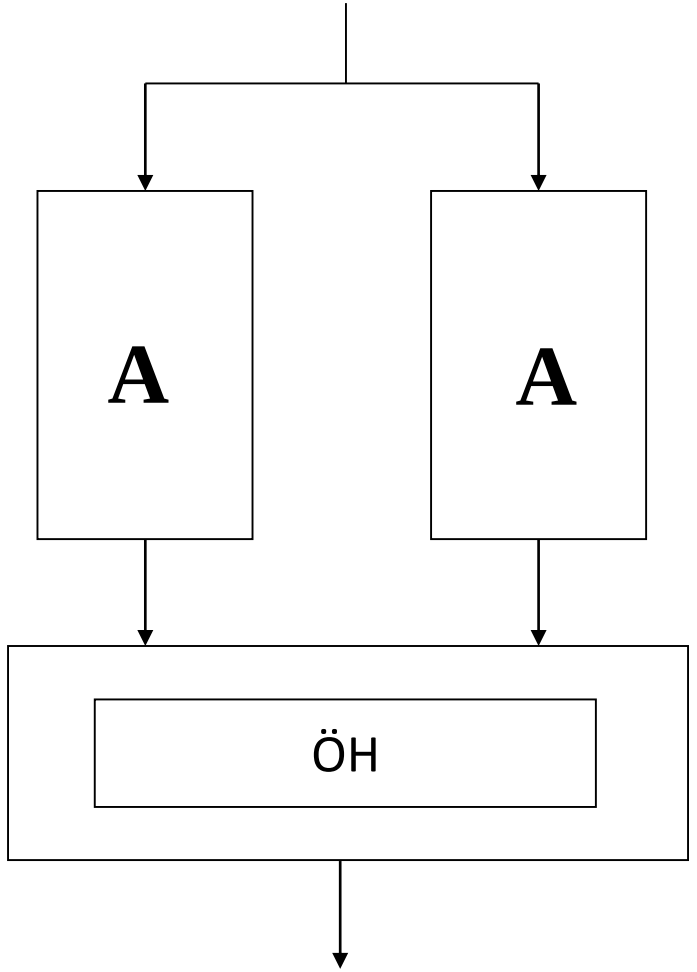
1 HW, 2 SW

- Két különböző (diverz) szoftver fut (A és B) ugyanazon a gépen.
- Két szoftver futhat párhuzamosan, vagy egymás után.
- Az összehasonlító felfedi, ha a két szoftver mást mond → felfedhetők a specifikációs és programozási hibák.
- Mivel a két program eltérő, ezért egy HW hiba nem egyformán hat a két szoftverre, így a véletlen HW hibák is felfedhetők.

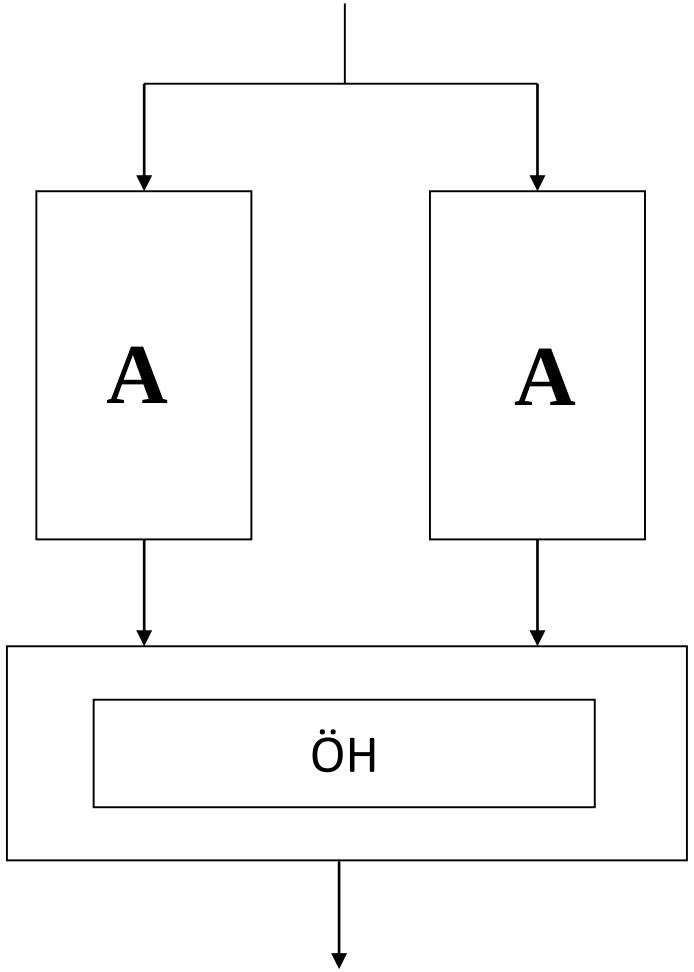


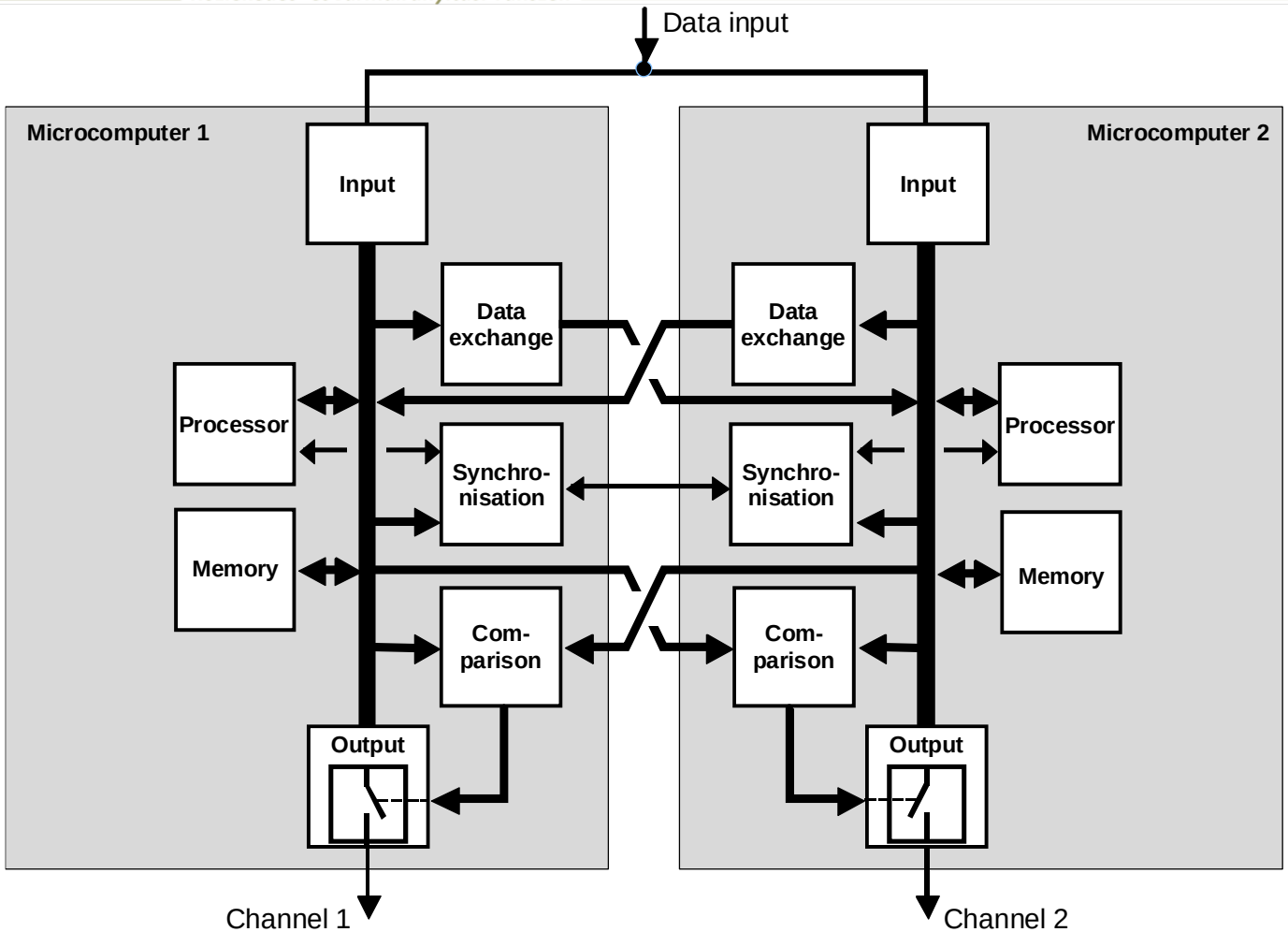
Kvázi fail-safe rendszer 1 HW, 1 SW

- 2 csatorna, csatornánként ua. HW és SW,
  - 2-ből 2 rendszer (2oo2).
  - Véd a hardver véletlen meghibásodásai ellen.
  - A szoftvert „eleve jóra” kell készíteni, mert az architektúra nem véd a specifikációs és programozási hibák ellen.

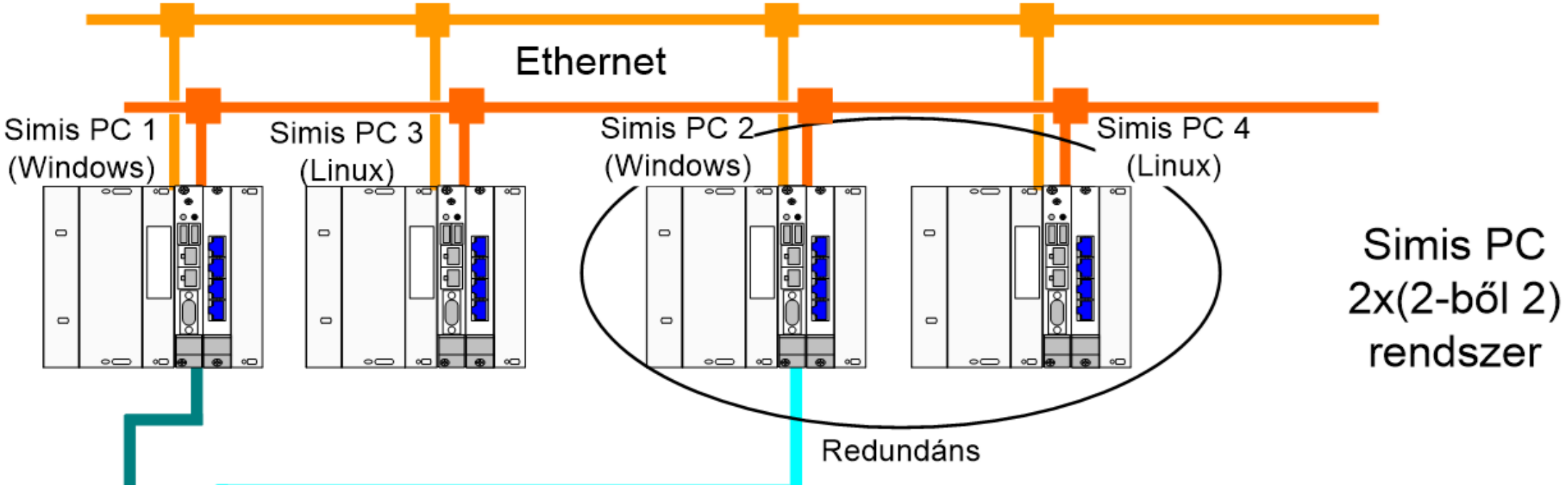


Összetett kompozit fail-safe rendszer, 2 HW, 1 SW





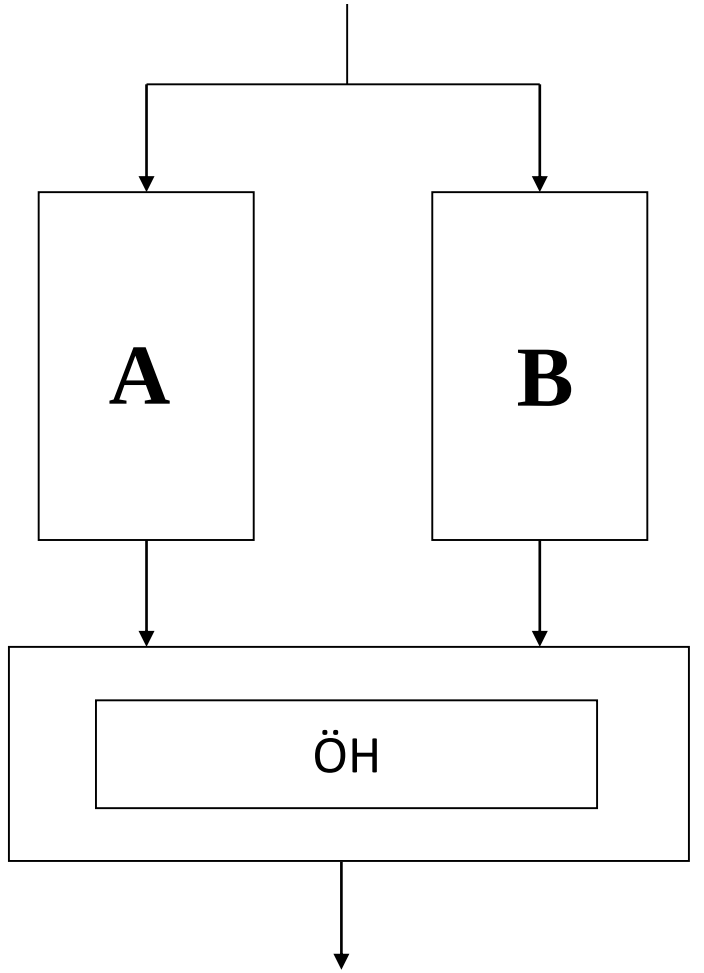
Összetett kompozit fail-safe rendszer, a rendelkezésre állás növelésével, példa



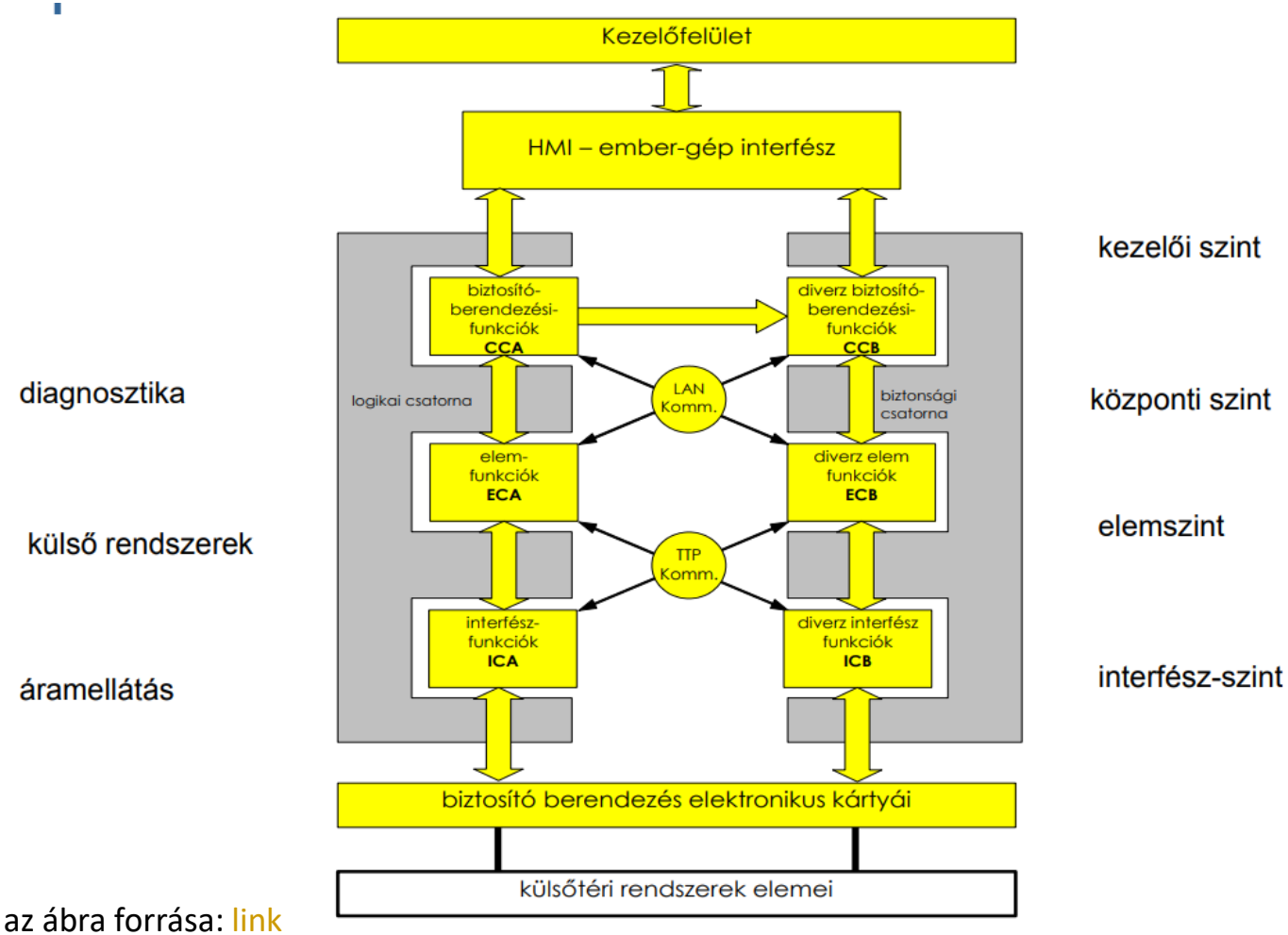
- megegyező HW és SW,
- az SW eltérő operációs rendszerre kerül lefordításra („némi” diverzitás), eltérő operációs rendszeren a fordítás után a SW szisztematikus hibái máshogy lépnek fel, a szavazás során felfedhető.

Reaktív fail-safe, 1 HW, 2 SW

- 1 hardver, 2 szoftver
  - Az architektúra véd a véletlen hardver hibák ellen és
  - a szoftver hibák ellen.
  - A két csatornában eltérő specifikációval, eltérő programnyelven kifejlesztett programok futnak
- Pl. Thales Elektra II.
  - rendelkezésre állás növelése: 2x(2oo2).



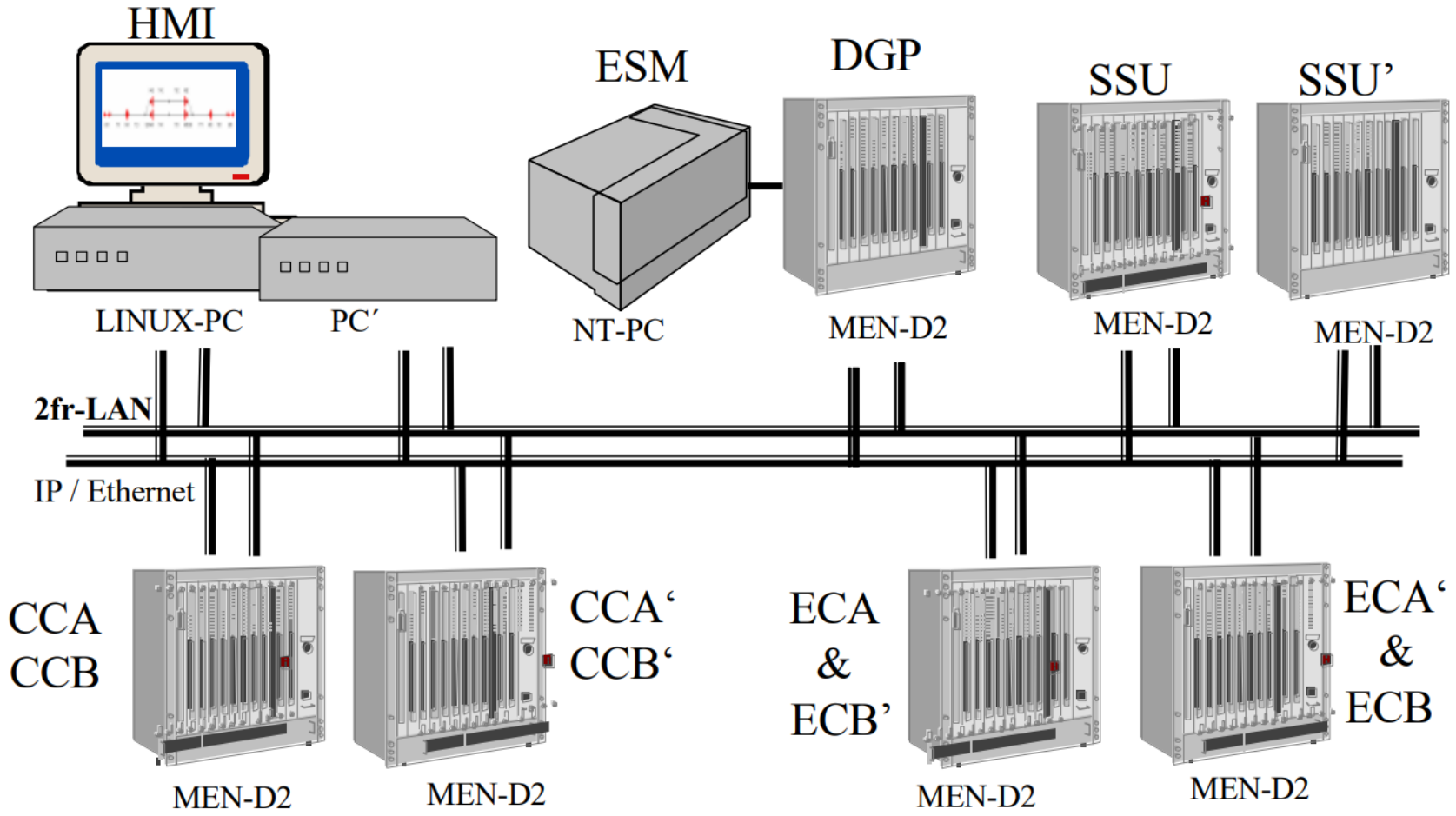
Reaktív fail-safe megoldás, példa



az ábra forrása: link

- logikai csatorna:
  - eljárás-orientált programnyelv,
- biztonsági csatorna:
  - szabály-orientált programnyelv,
  - az első csatorna működésnek ellenőrzése.

Reaktív fail-safe megoldás, példa

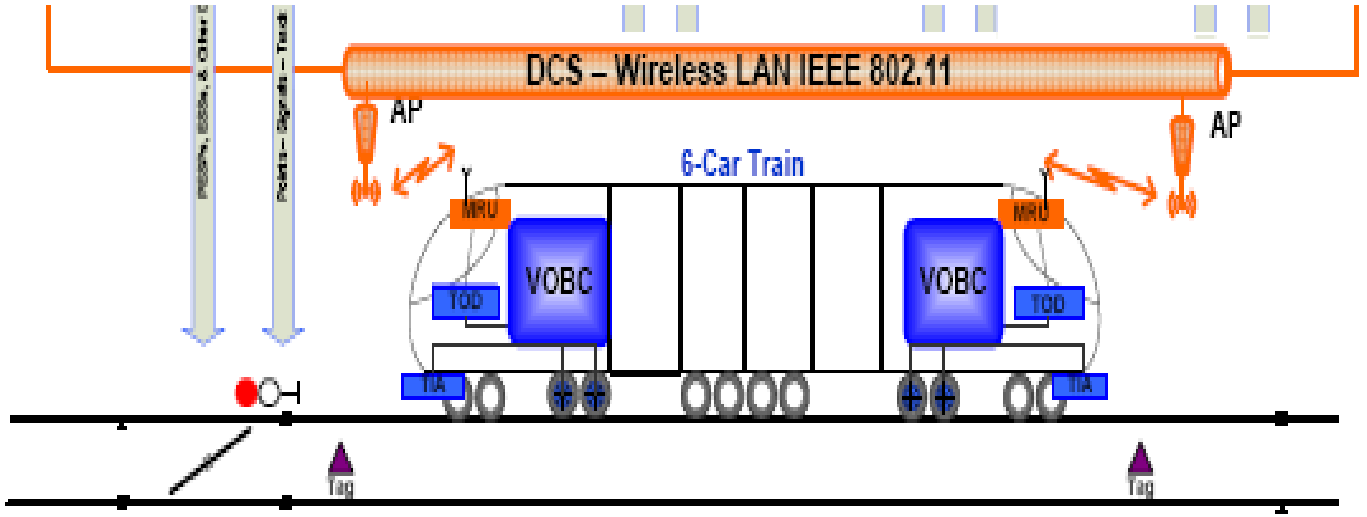


az ábra forrása: link

A rendelkezésre állás növelése

- A „k=n” architektúrák biztonságosak ugyan, de már egy hiba esetén is működésképtelenek.
- Módszerek a rendelkezésre állás növelésére → Tartalékolás
  - Egycsatornás rendszer: redundancia
  - 2v2 → 2×(2v2) (pl. SIMIS IS: SIMIS PC)
  - 2v2 → 2v3 (pl. SIMIS IS: ECC számítógépek)

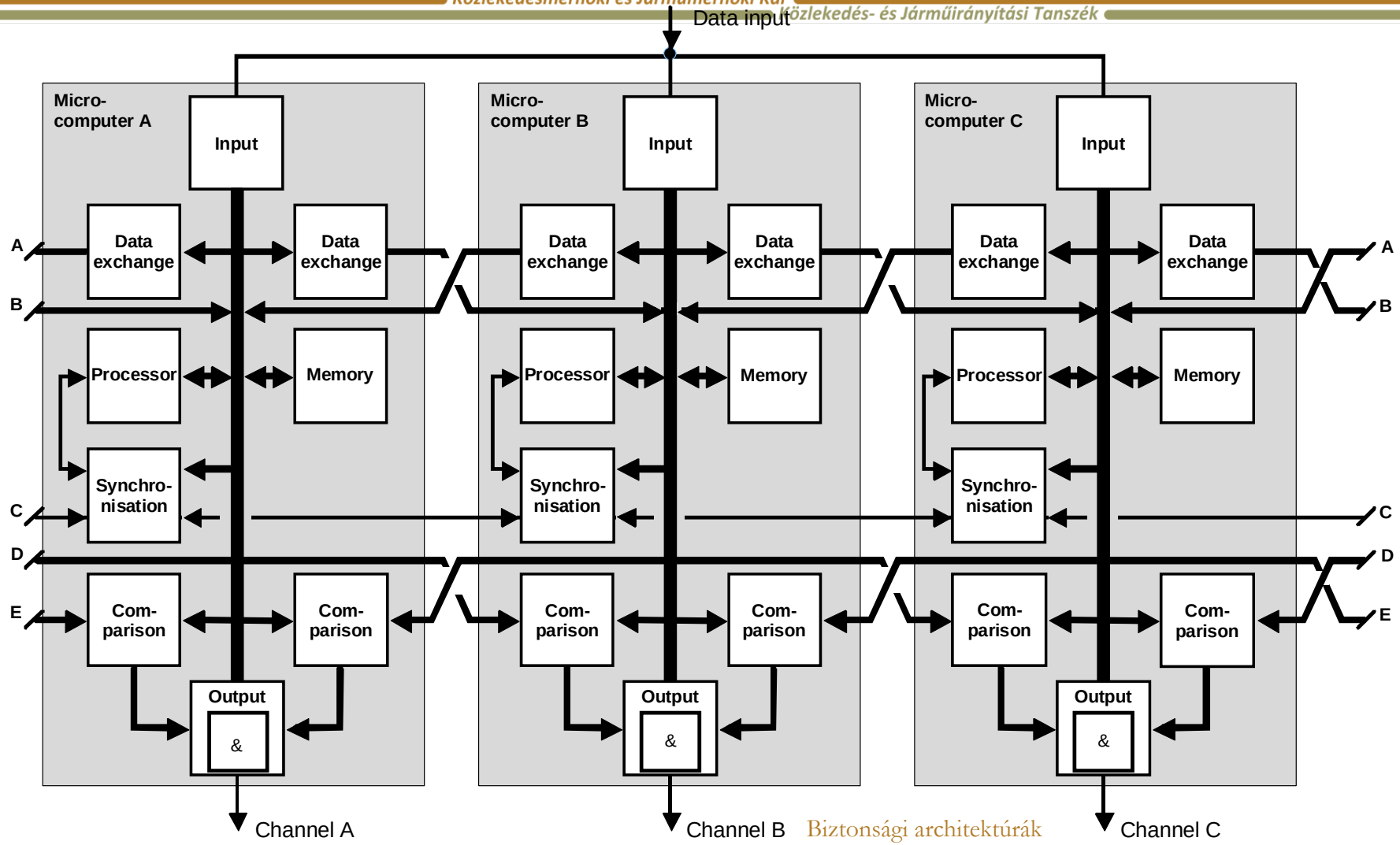
A rendelkezésre állás növelése: 2x(2oo2), példa



az ábra forrása: link

- VOBC: Vehicle On-Board Computer
  - vezető nélküli (driverless) közlekedtetés,
  - biztonságos vonatközlekedtetés,
  - automatikus fordulás,
  - automata ajtóvezérlés.

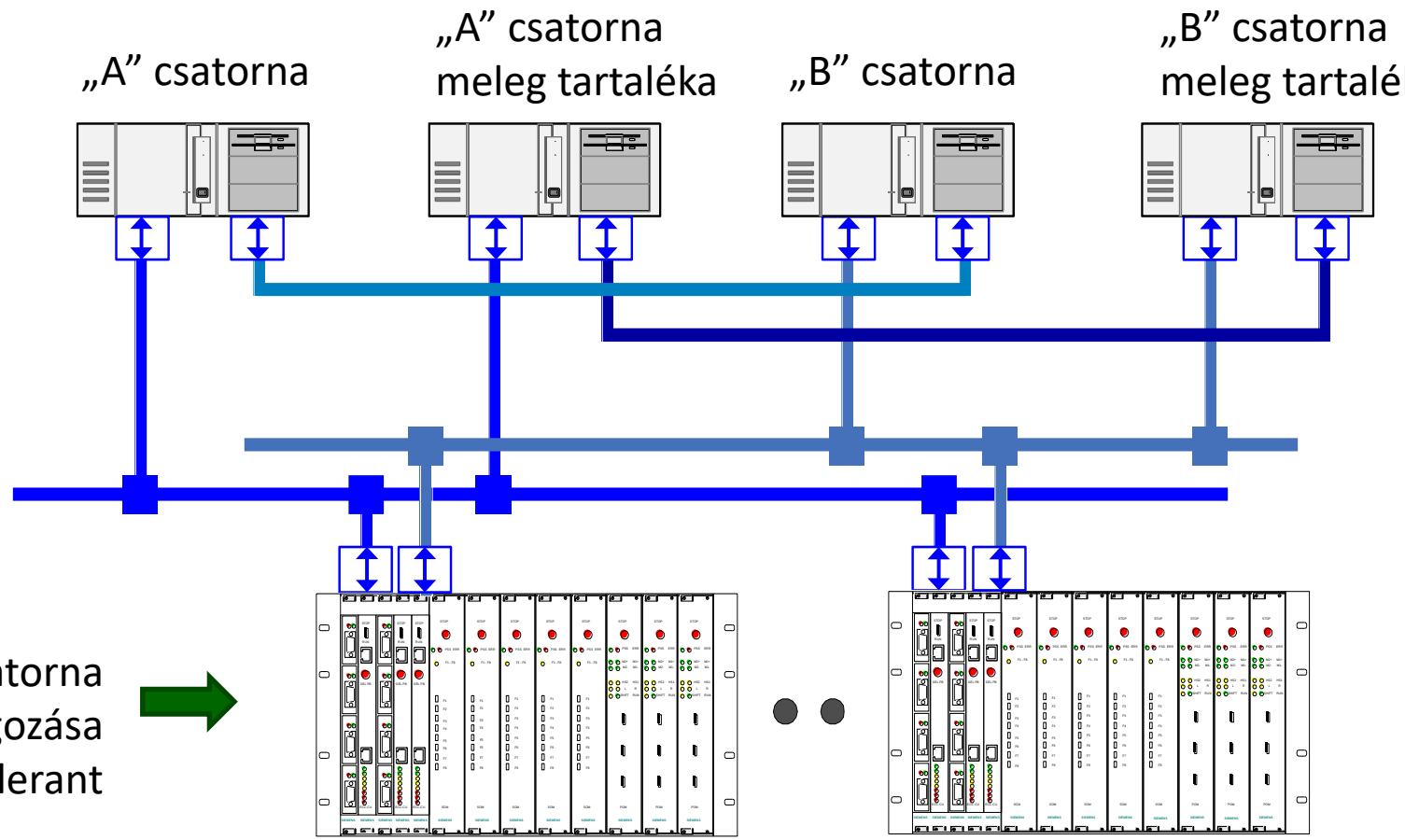
A rendelkezésre állás növelése: 2oo3



Összetett architektúra, példa

COTS

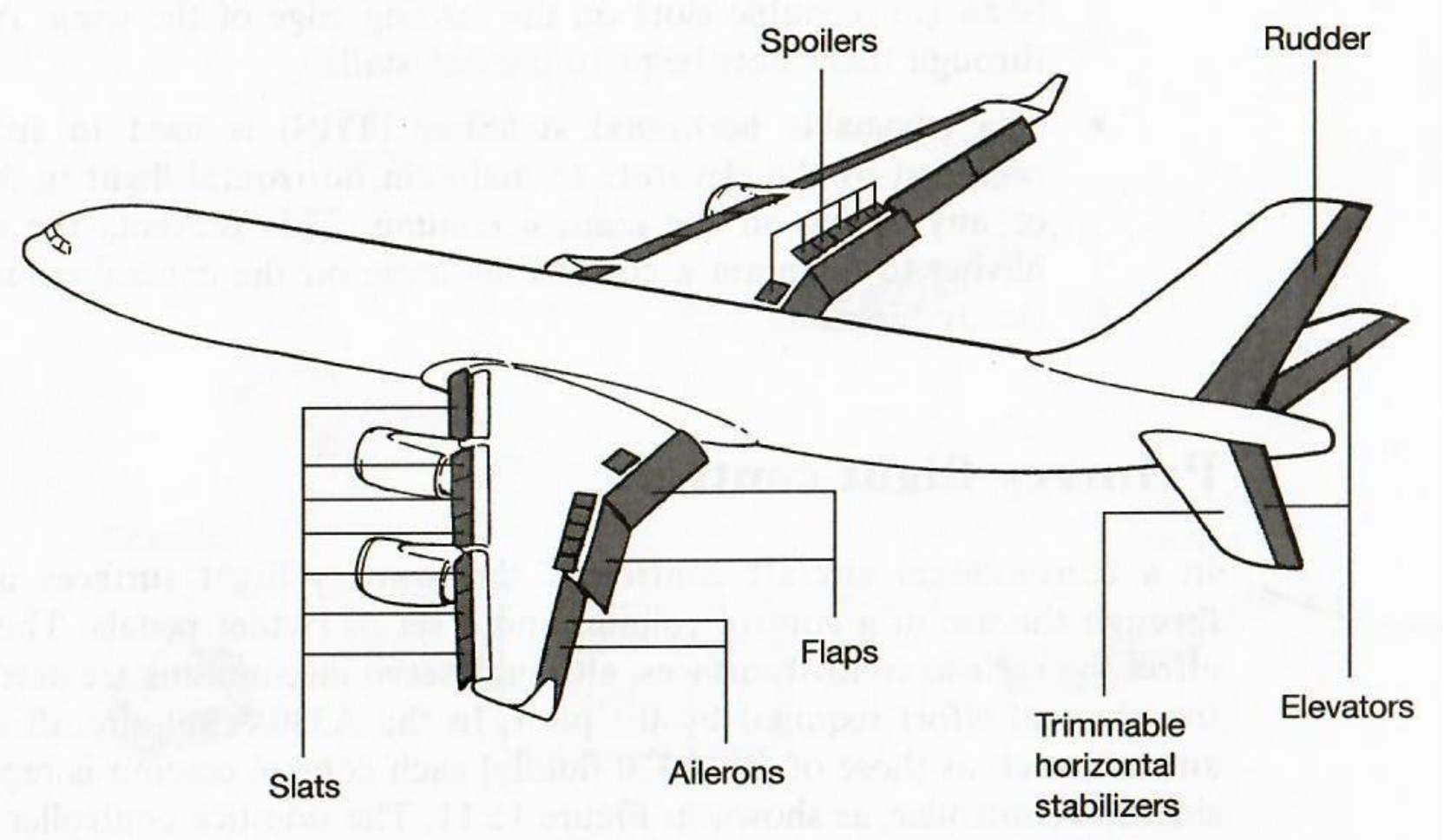
- PC HW (Intel)
- Operációs rendszer (Win2000/Linux)
- diverz fordítók
- diverz program kód



„A” és „B” csatorna összehasonlítása és feldolgozása kvázi fail-safe/fault tolerant rendszerben

kimenetek vezérlése

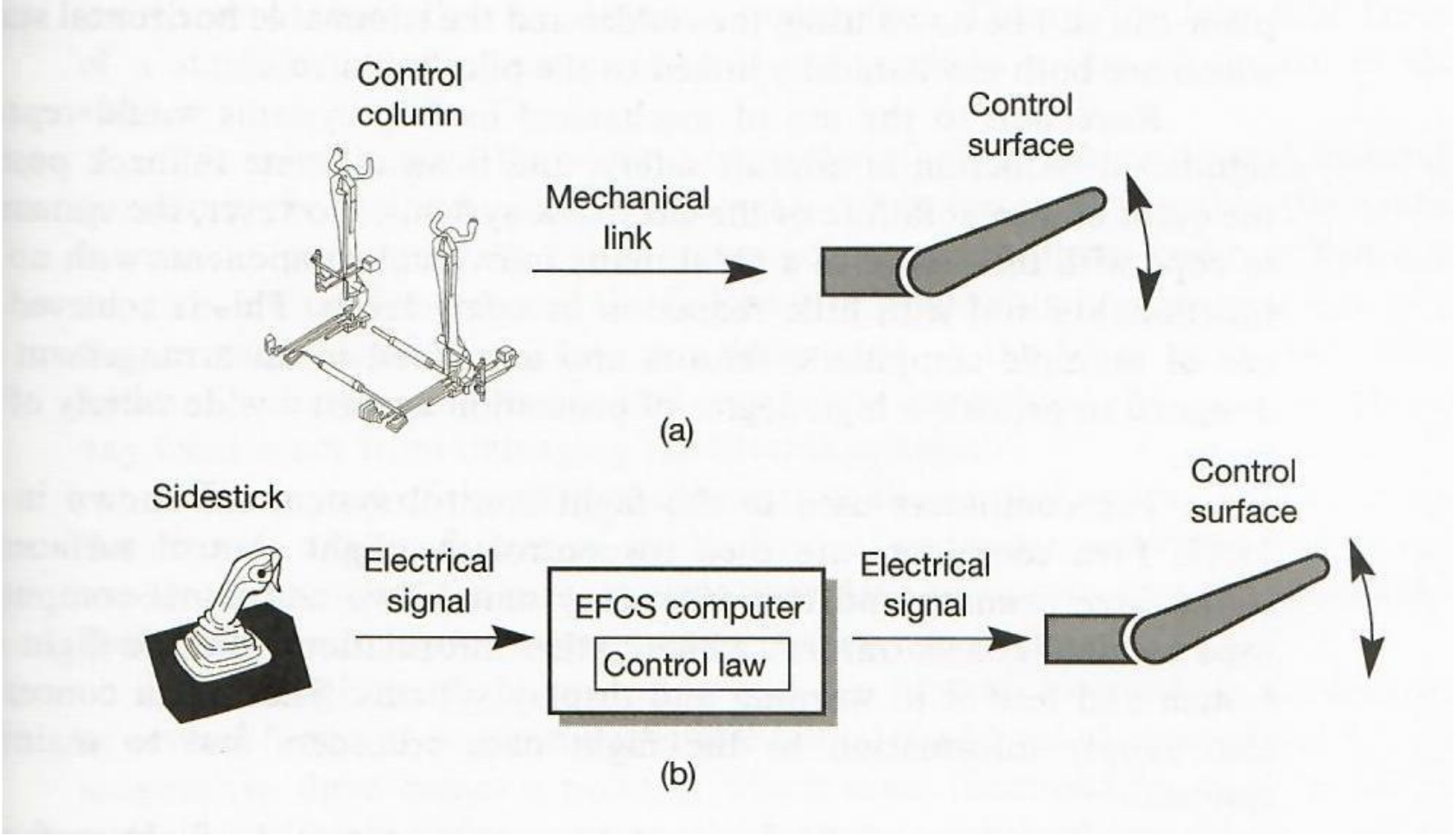
A330/A340 primary flight control system - példa



az ábra forrása: Neil Storey, Safety-Critical Computer Systems, Addison-Wesley, 1996, England

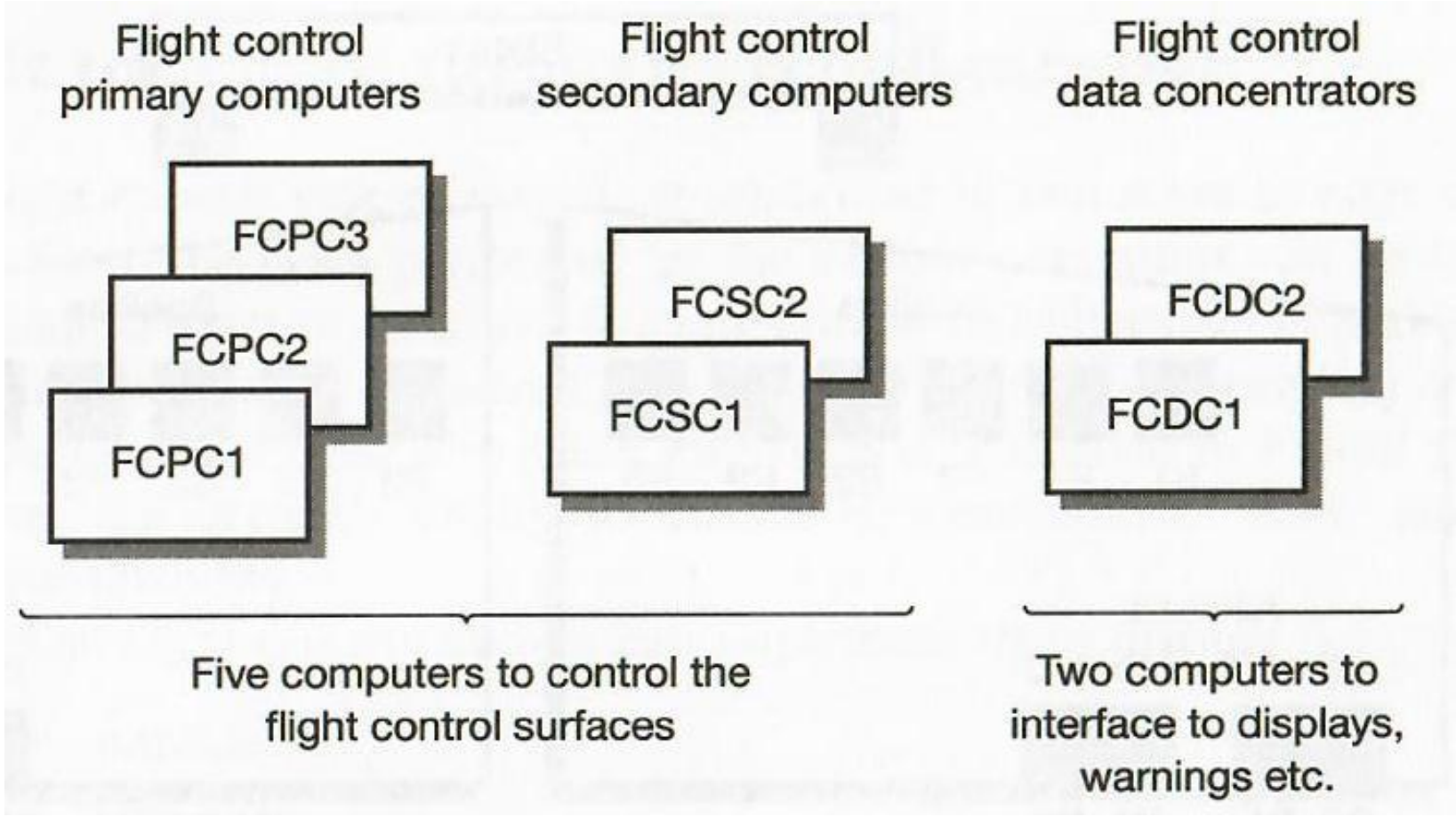
A330/A340 primary flight control system - példa

- hagyományos:
  - mechanikus kapcsolat,
- fly-by-wire:
  - EFCS: electronic flight control system, figyelembe véve a „repülésirányítási törvényeket”



az ábra forrása: Neil Storey, Safety-Critical Computer Systems, Addison-Wesley, 1996, England

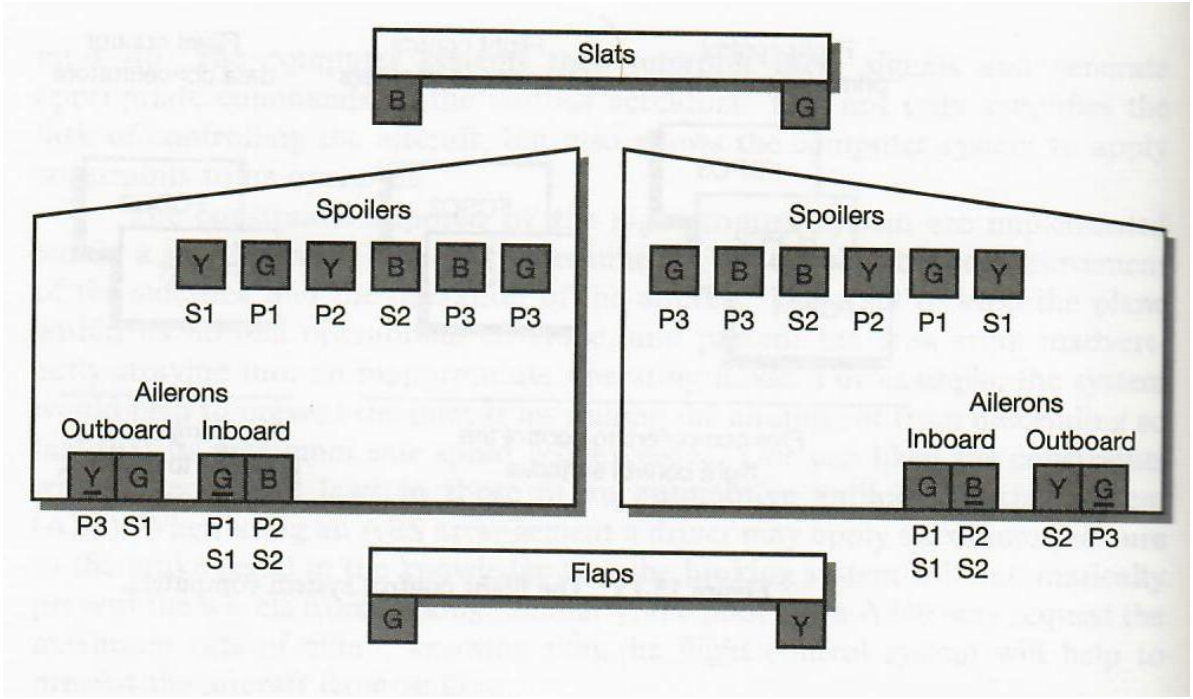
A330/A340 primary flight control system - példa

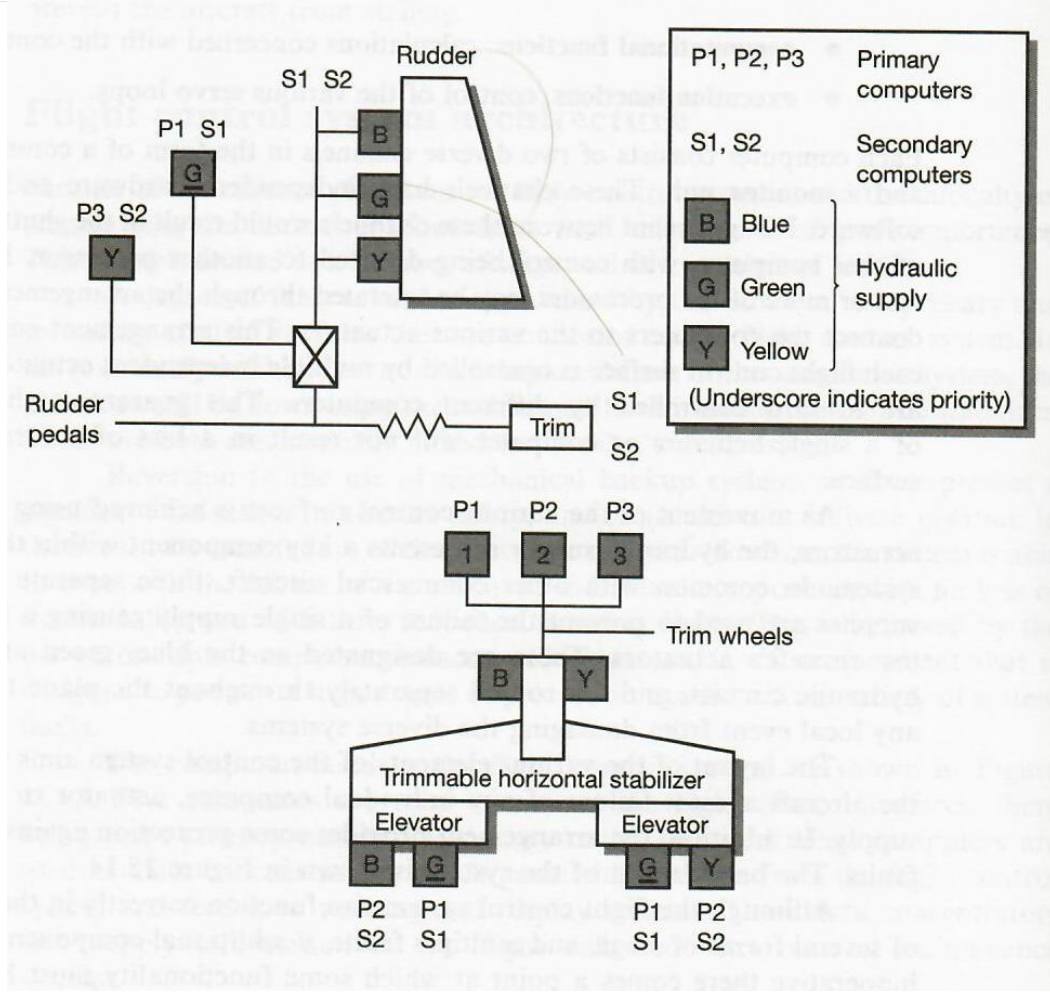


- minden számítógép tartalmaz két csatornát, a csatornák HW-e független, az SW diverz:
  - az egyik csatorna felel az utasításokért, a másik ellenőrzi az első eredményeit.

az ábra forrása: Neil Storey, Safety-Critical Computer Systems, Addison-Wesley, 1996, England

A330/A340 primary flight control system - példa

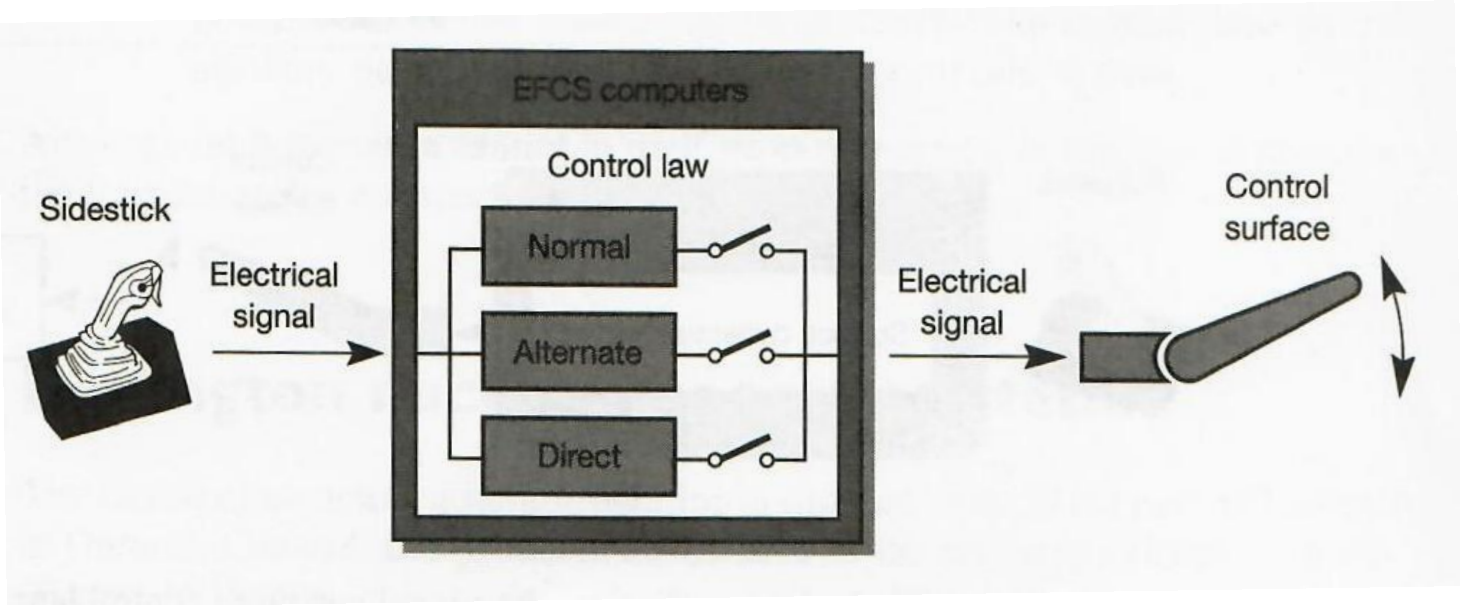




az ábrák forrása: Neil Storey, Safety-Critical Computer Systems, Addison-Wesley, 1996, England

A330/A340 primary flight control system - példa

- a rendszer képes alkalmazkodni, és váltani a repülési feltételek között:

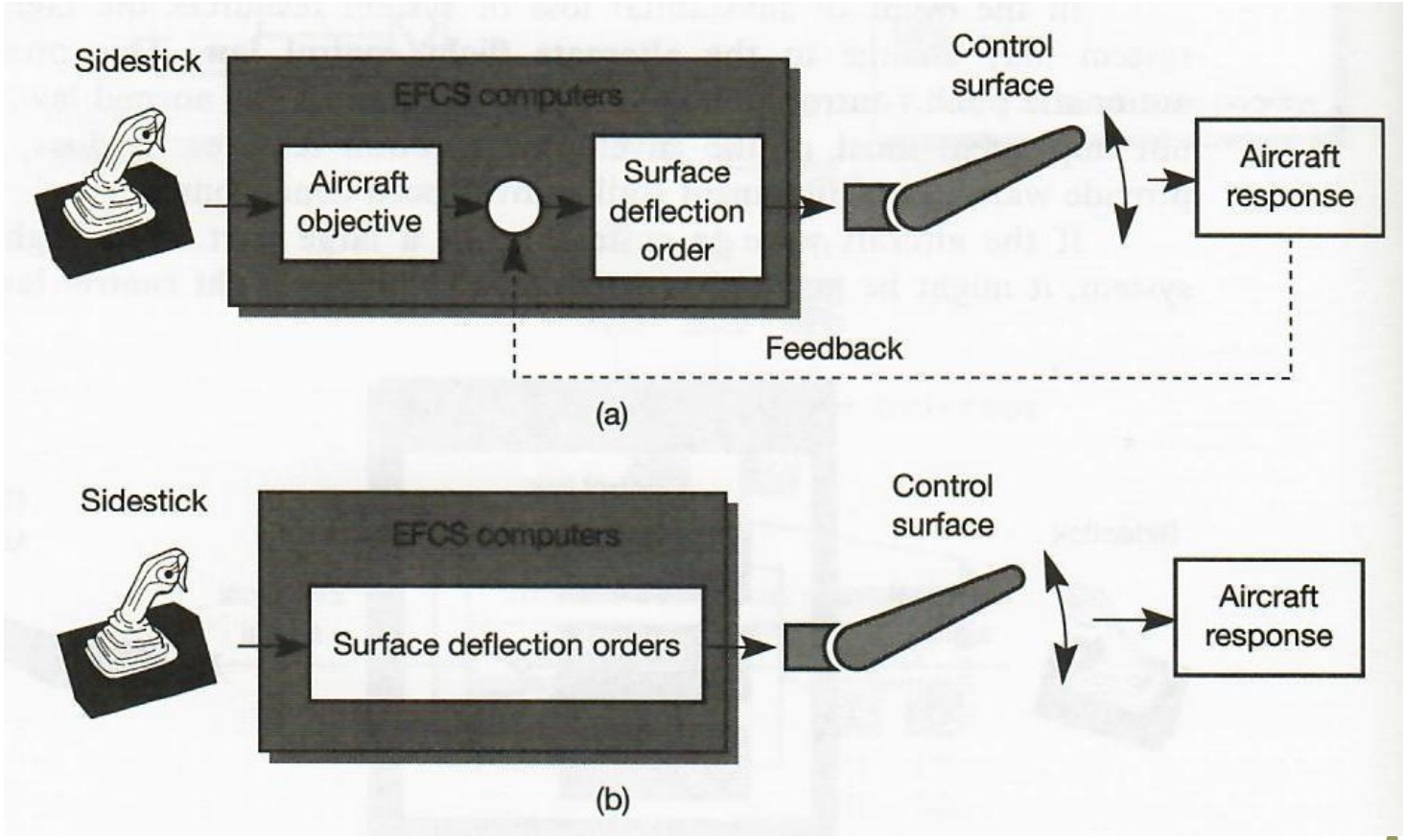


- alaphelyzetben: „normal” repülésirányítás:
  - sok automatizált funkció, jelentősen csökken a pilóták terhelése,
- jelentős veszteség esetén: „alternate” repülésirányítás:
  - csak stabilizálás, , és átesés közelében csak figyelmeztetés, a védelmi funkciók jelentős része inaktív,
- igen jelentős károk esetén „direct” repülésirányítás:
  - a sidestick közvetlenül vezérli a felületeket, minden védelmi funkció inaktív (csak a figyelmeztetések élnek).

az ábrák forrása: Neil Storey, Safety-Critical Computer Systems, Addison-Wesley, 1996, England

A330/A340 primary flight control system - példa

- különbség a „normal” és a „direct” repülésirányítás között:
  - zárt-hurkú (closed–loop) irányítás,
  - nyílt hurkú (open-loop) irányítás:
    - (a „direct” mód megfelel a hagyományos irányítási módnak).



az ábrák forrása: Neil Storey, Safety-Critical Computer Systems, Addison-Wesley, 1996, England

A330/A340 primary flight control system - példa

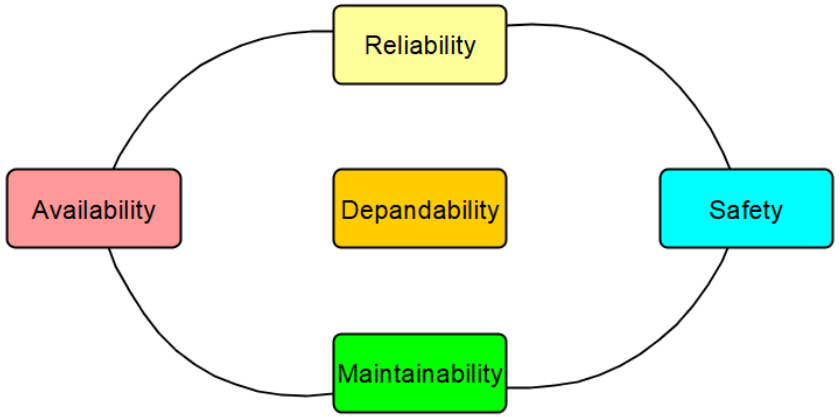
- pimary computers: Intel 80386:
  - 16 MHz órajel,
  - kb. 275 000 db tranzisztor (első verziók),
  - Assembler és PL/M programnyelvek,
  - program nagysága: kb. 800 kB,

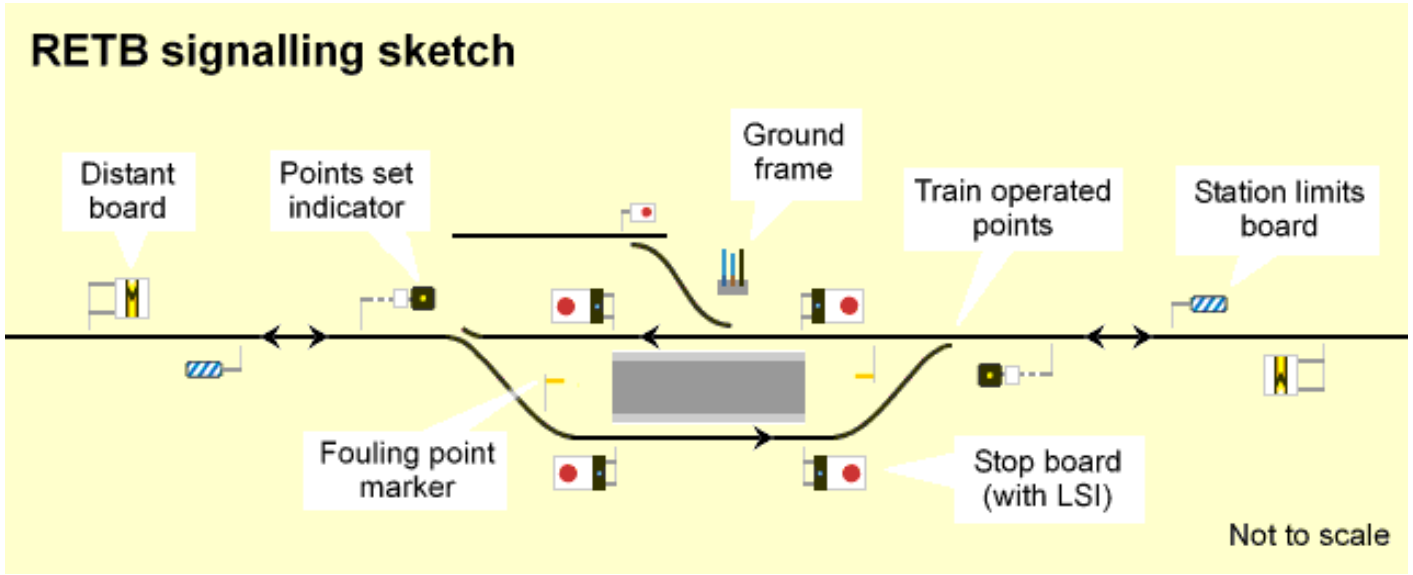
- secondary computers: Intel 80186:
  - 12 MHz órajel,
  - kb. 55 000 db tranzisztor,
  - Assembler és Pascal programnyelvek,
  - program nagysága: kb. 300 kB.

A330/A340 primary flight control system - példa

- fault-tolerant tulajdonság:
  - mechanikus: oldalkormány és trimmelhető vezérsík,
  - számítógépek: 5 db, ebből 2 típus, számítógépenként 2 független HW csatorna, diverz szoftverrel,
  - érzékelők: minden esetben min. 2,
  - beavatkozók: akár felületenként 3 db,
  - hidraulika: 3 független rendszer,
  - energiaellátás: 6 generátor, 2 akkumulátor, 5 busz rendszer.

Összefoglalás – Közlekedés automatika (BSc)





az ábra forrása: link

- Cél: a **megkívánt** működőképesség és rendelkezésre állás elérése,
  - az adott funkciókhoz **szükséges biztonságintegritási szinten, az adott specifikáció szerint,**
  - egy alkalmazás (funkció) több, az alkalmazásnak megfelelő biztonsági architektúrával is elérhető,
  - a rendszerek belső felépítése egyszerre több biztonsági stratégiát is alkalmazhat.

Köszönöm a figyelmet!

Biztonsági architektúrák

Lövétei István Ferenc
(lovetei.istvan@mail.bme.hu)
Dr. Sághi Balázs