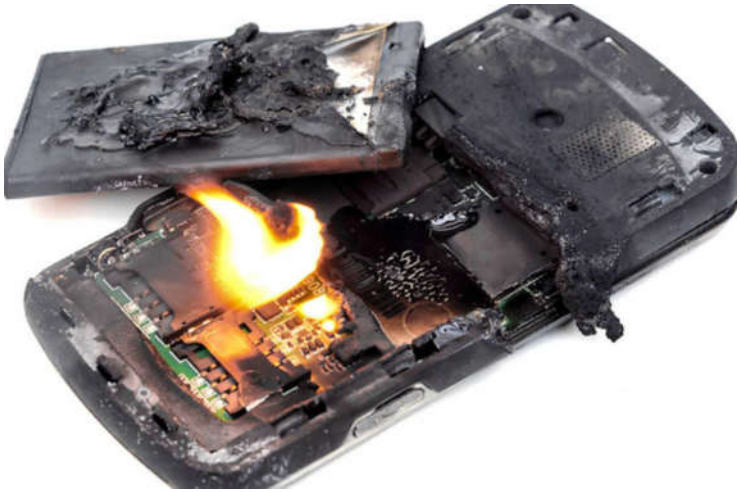
Safety architectures

Guillaume VIBERT

CPE – part 1/2

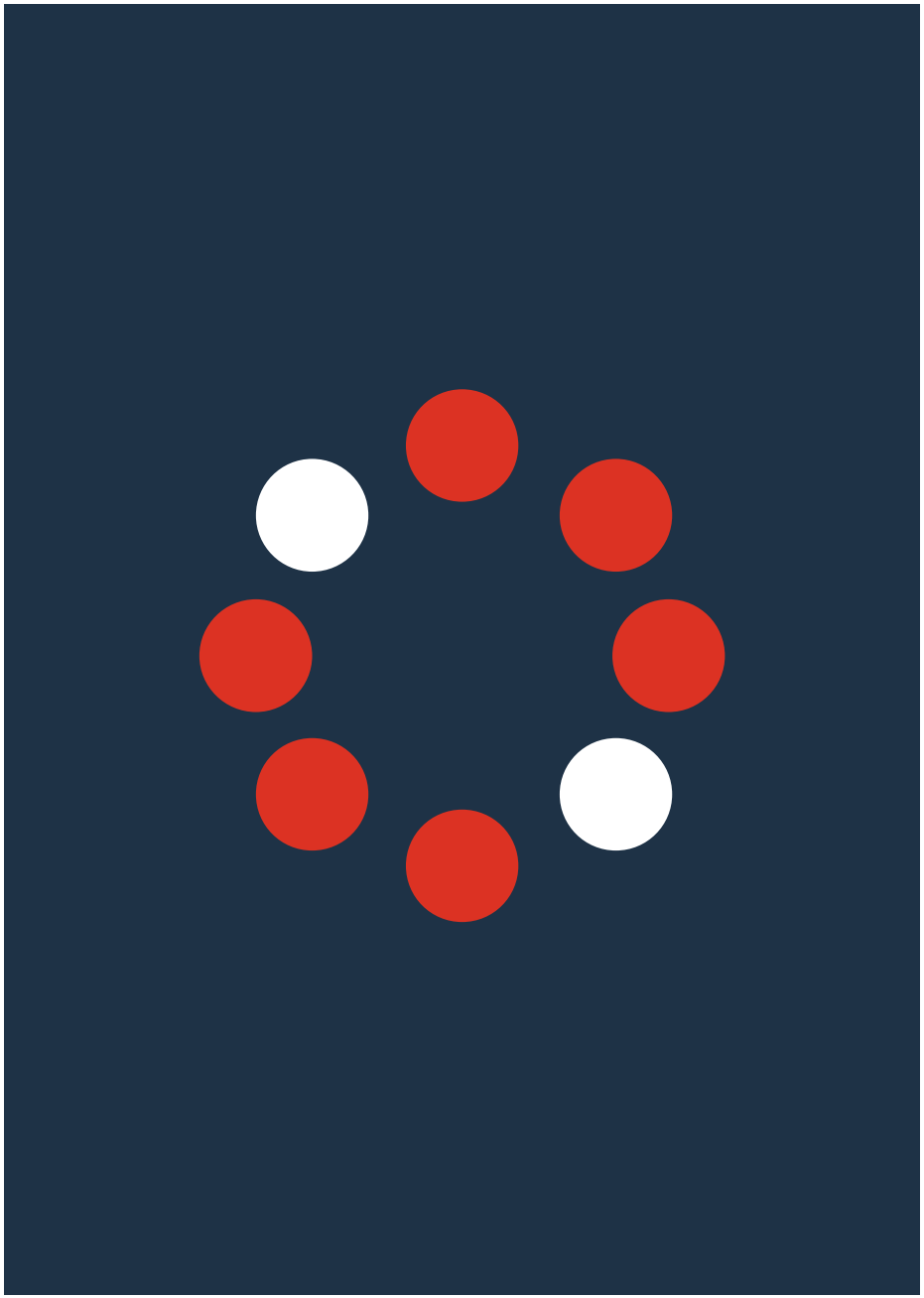
ALSTOM
• mobility by nature •

Welcome



Gare Montparnasse

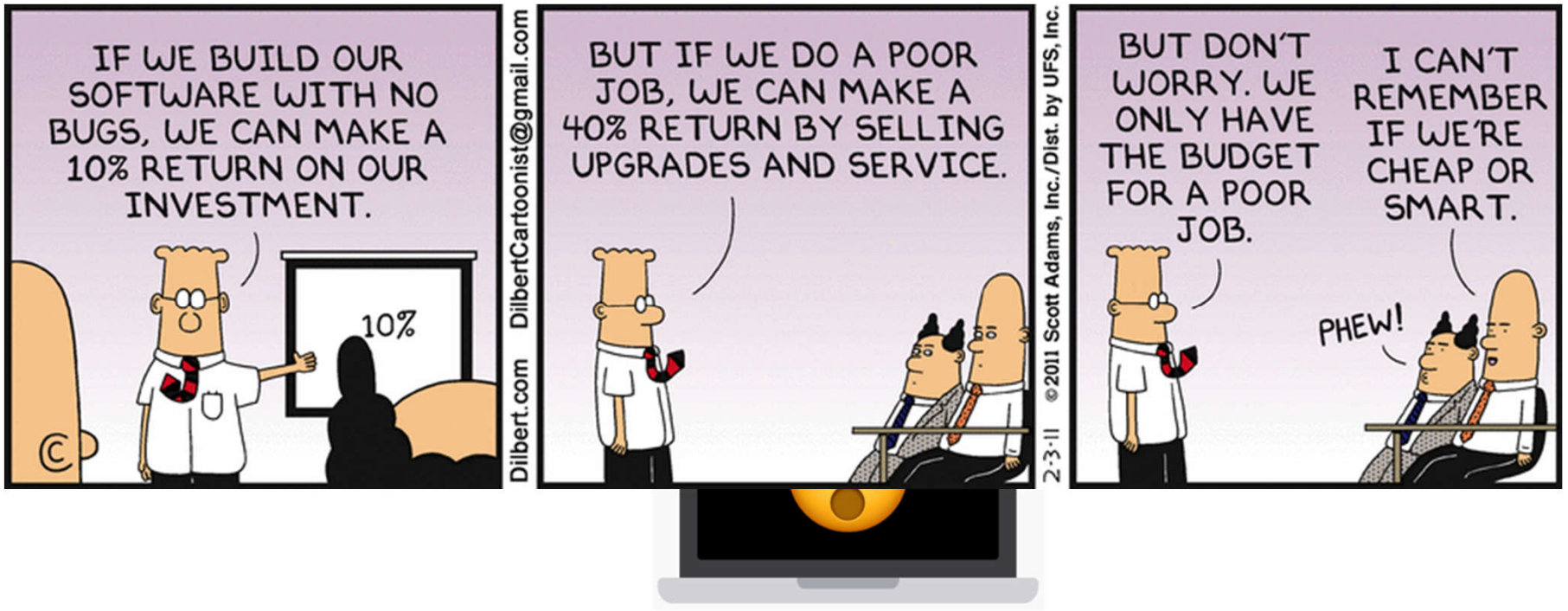
Paris 1895



Agenda

1. RAMS
2. Functional safety
3. Safety architectures

Dysfunctional approach – why?



Dysfunctional approach – why?

01 RAMS

Definitions

RAMS

- Reliability
- Availability
- Maintainability

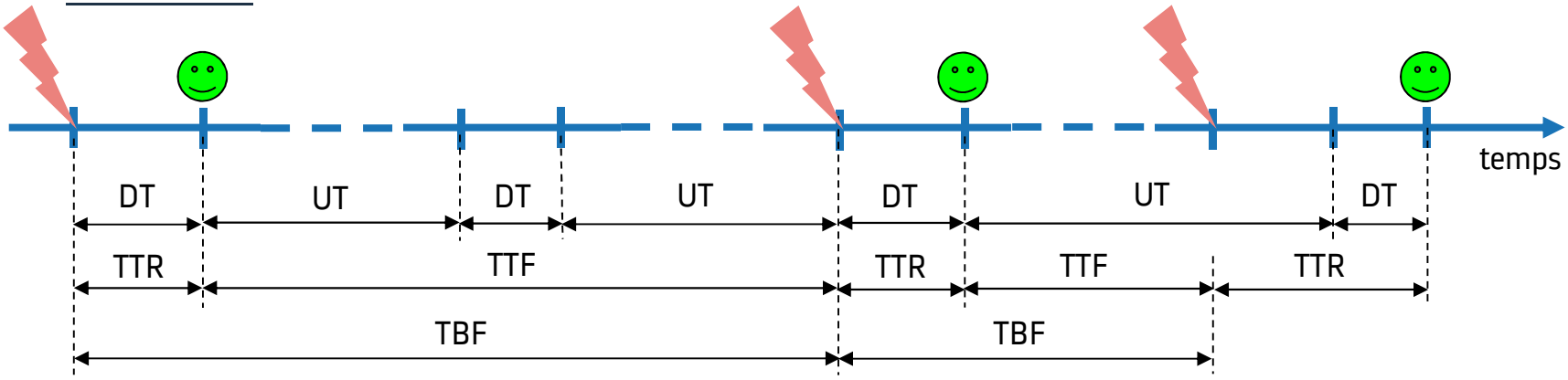
- Safety

FMDS

- Fiabilité
- Maintenabilité
- Disponibilité

- Sûreté de fonctionnement

Metrics



- **MDT** = Mean Down Time (moyenne des DT)
- **MUT** = Mean Up Time (moyenne des UT)
- **MTTF** = Mean Time To Failures (moyenne des TTF)
- **MTBF** = Mean Time Between Failures (moyenne des TBF)
- **MTTR** = Mean Time To Restore (moyenne des TTR)
- **A** = Availability
  = MUT / (MUT + MDT)

RAM: examples of requirements

Reliability

- Bogie 300.000 km
- Train door 100.000 cycles
- Embedded computer 25.000h – 200.000h

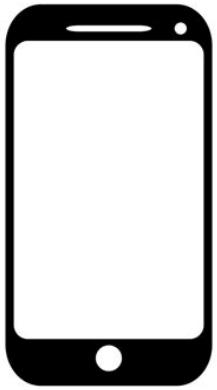
Availability

- 99.9999%

Maintainability

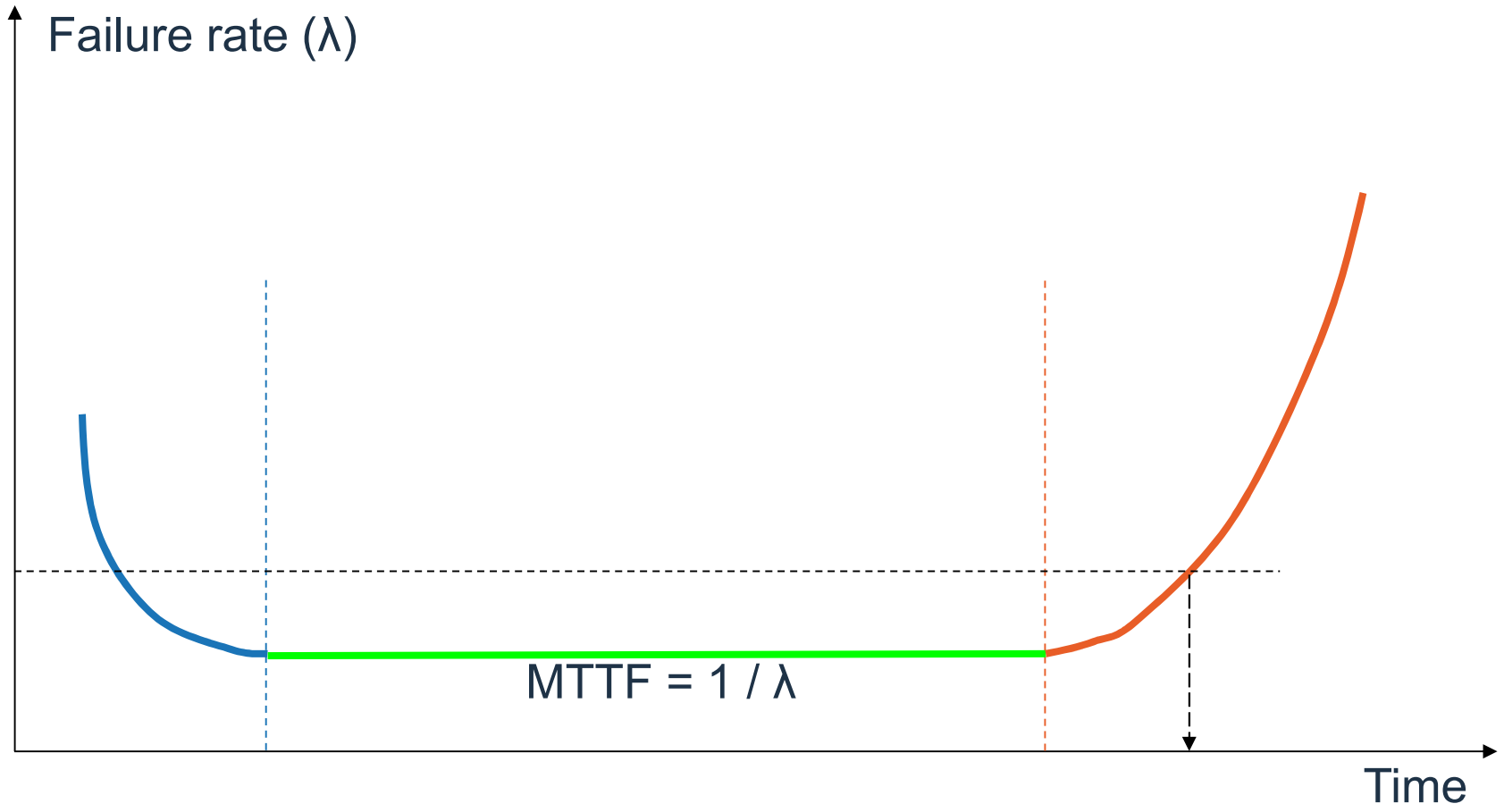
- MTTR 15 minutes – 2 tools maximum – no IT skills

RAM: how to increase performances

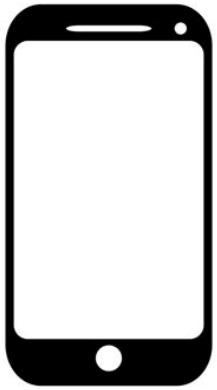


Please connect!

Bathtub curve



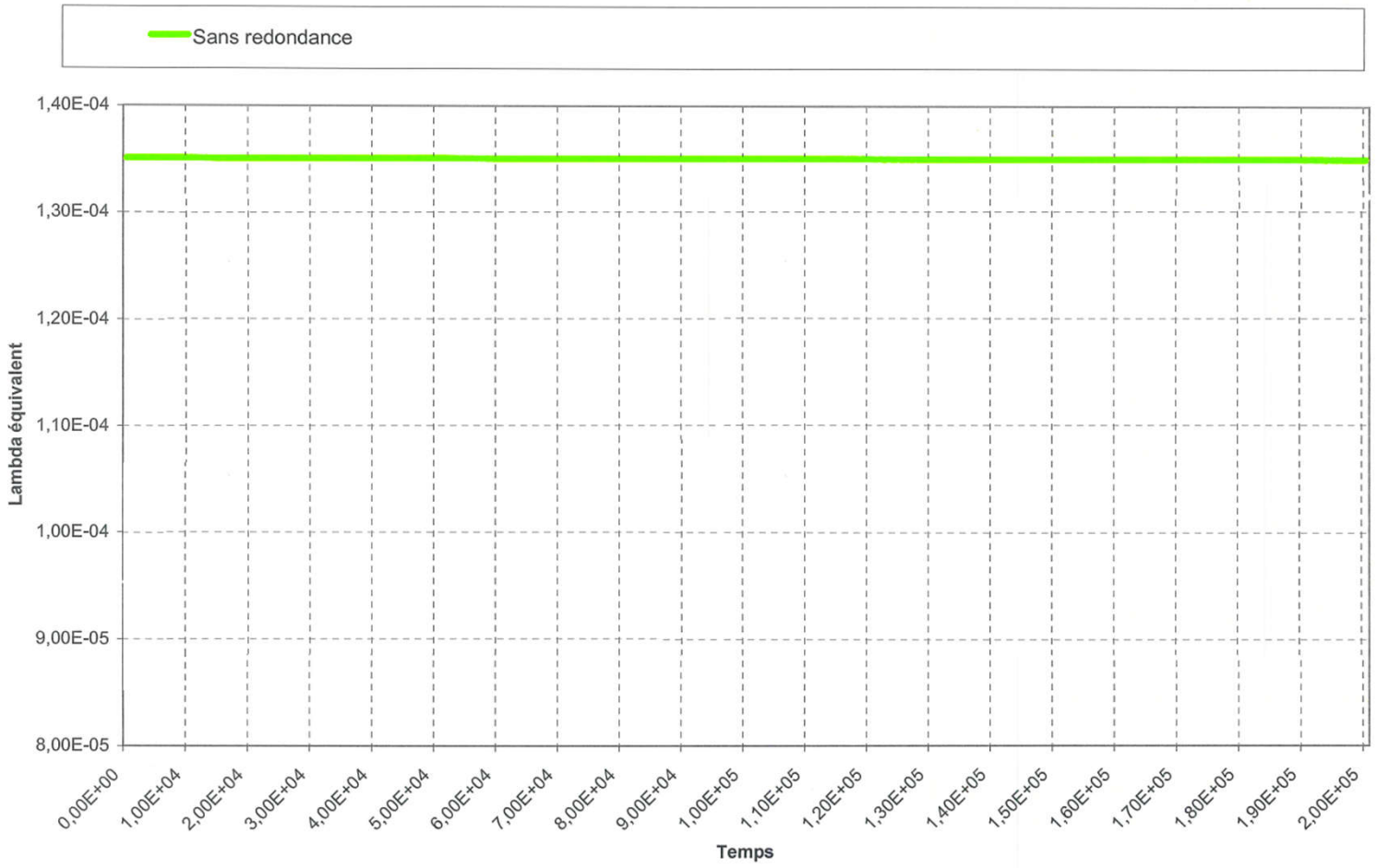
The bathtub curve



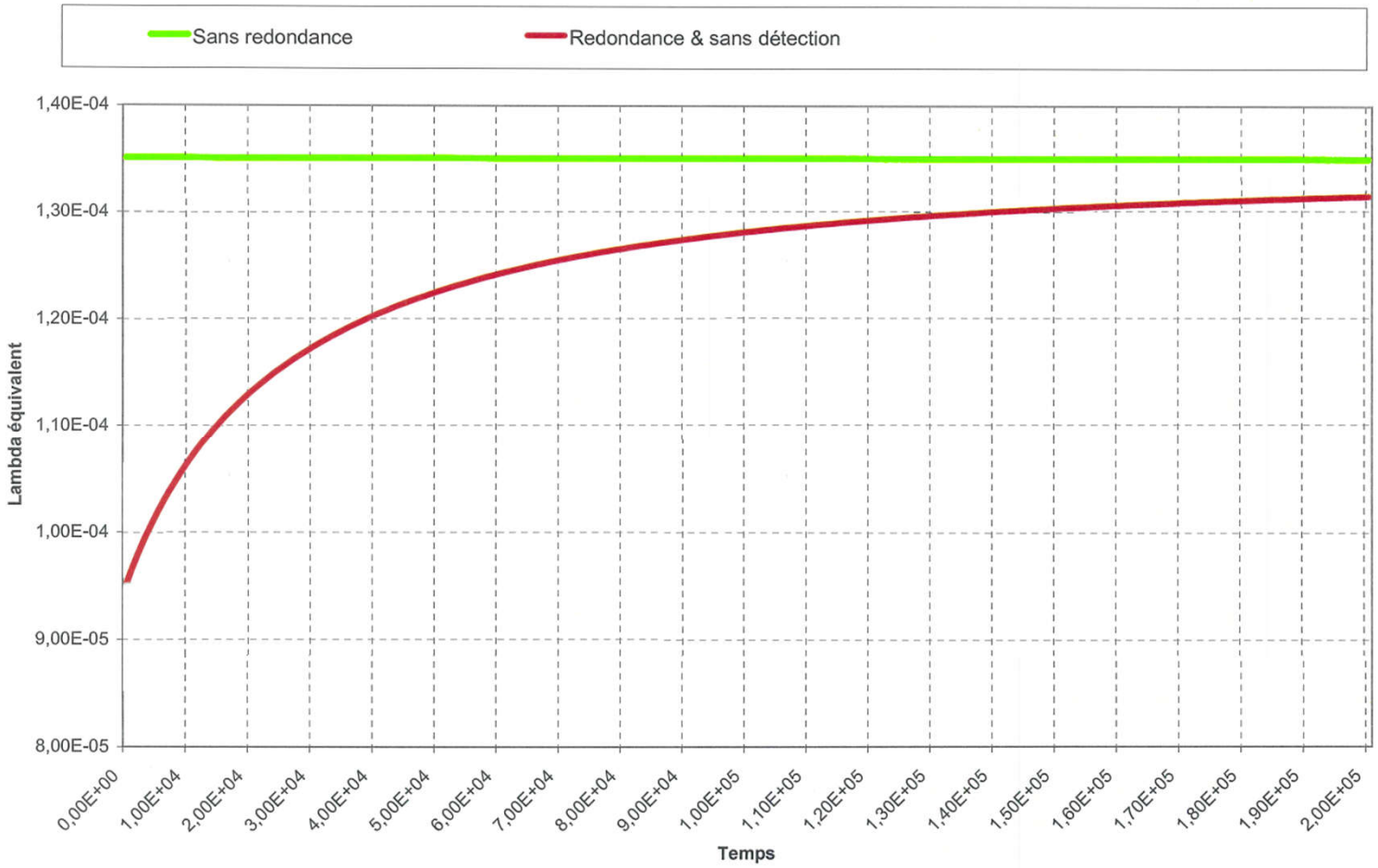
Please connect!

Redundancy

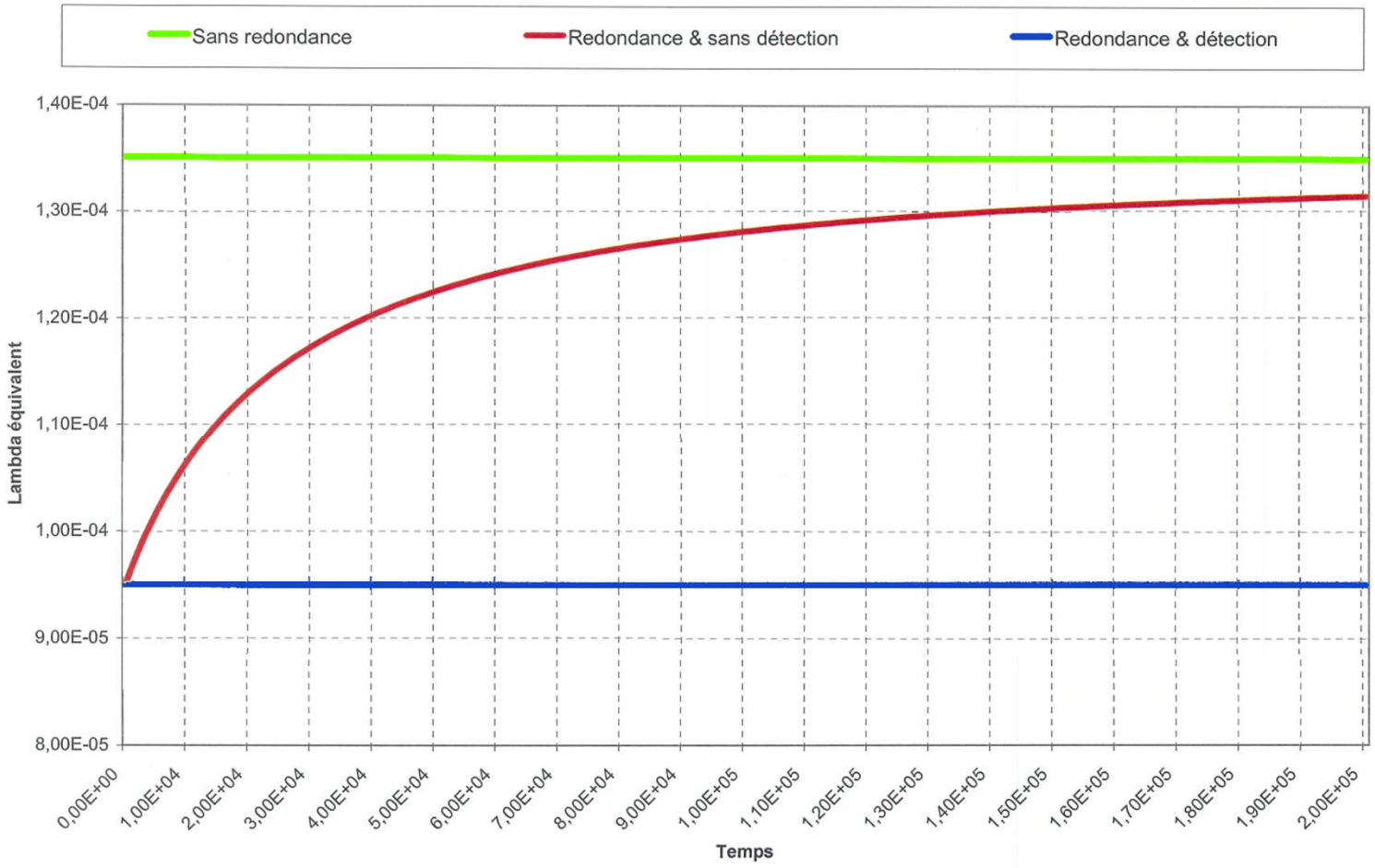
Redundancy



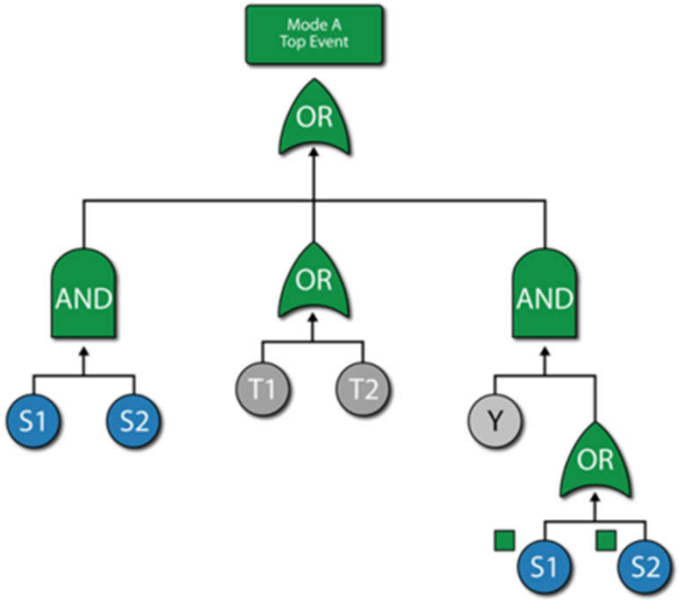
Redundancy

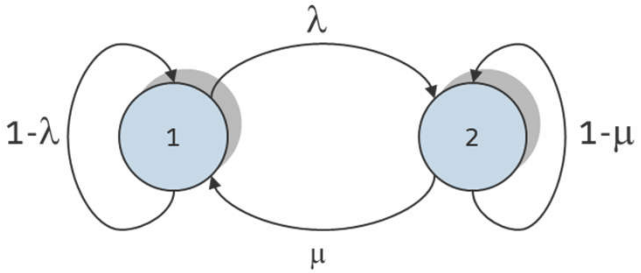


Redundancy



RAM: methods





$$\lambda = \underbrace{\left\{ \left\{ \pi_U \times \lambda_0 \right\} \times \left\{ \frac{\sum_{i=1}^{y} (\pi_t)_i \times \tau_i}{\tau_{on} + \tau_{off}} \right\} \right\}}_{\lambda_{die}} + \underbrace{\left\{ \left(2.75 \times 10^{-3} \times \sum_{i=1}^{z} (\pi_n)_i \times (\Delta T_i)^{0.68} \right) \times \lambda_B \right\}}_{\lambda_{package}} + \left\{ \underbrace{\pi_I \times \lambda_{EOS}}_{\lambda_{overstress}} \right\} \times 10^{-9} / h$$

Functional safety - Definition

IEC61508

- **Freedom from unacceptable risk**

Risk is a combination of

- the probability of occurrence of harm
- the severity of that harm

Zero risk doesn’t exist.

Risk

Risk acceptance (or tolerability):

- to reduce severity or/and occurrence of harm within a given environment

How ?

- Severity: protection (e.g. Airbag)
- Occurrence: preventive measures (e.g road safety, signalling)

Risk_Acceptance.asf

Safety requirement

- Boundary Hazard

- Tolerable hazard rate
  - Probability, defined in h^{-1}

Break

ALSTOM

02

Functional Safety

Faults

Random fault

- quantifiable
- unpredictable
- hardware fault

Systematic fault

- not quantifiable
- predictable in a given context
- inherent to a system
- consequence of a human error

Faults: random or systematic?

Please connect!

Faults (1)

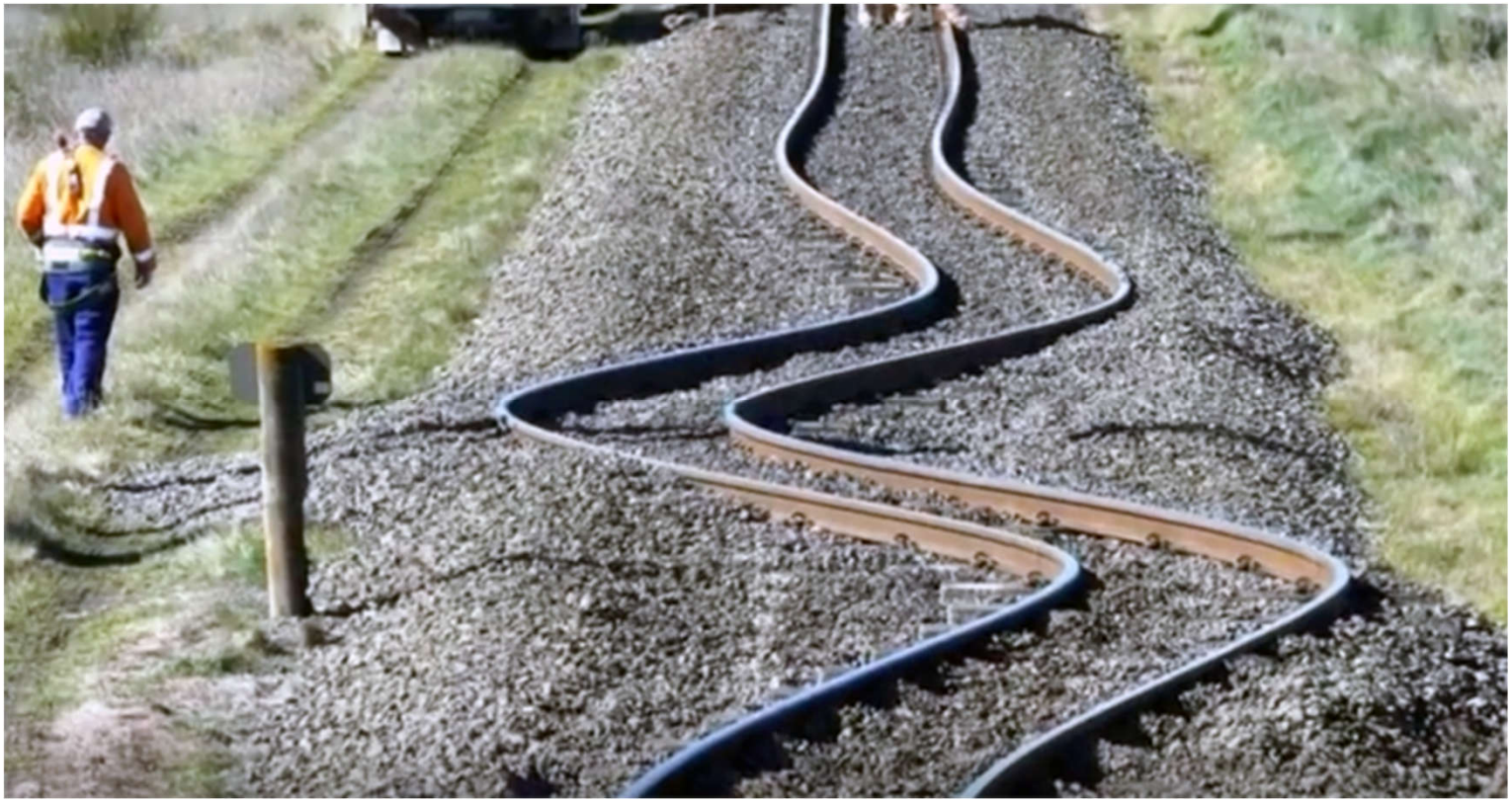
Random / systematic fault?

Faults (2)

Random / systematic fault?

Faults (3)

Random / systematic fault?

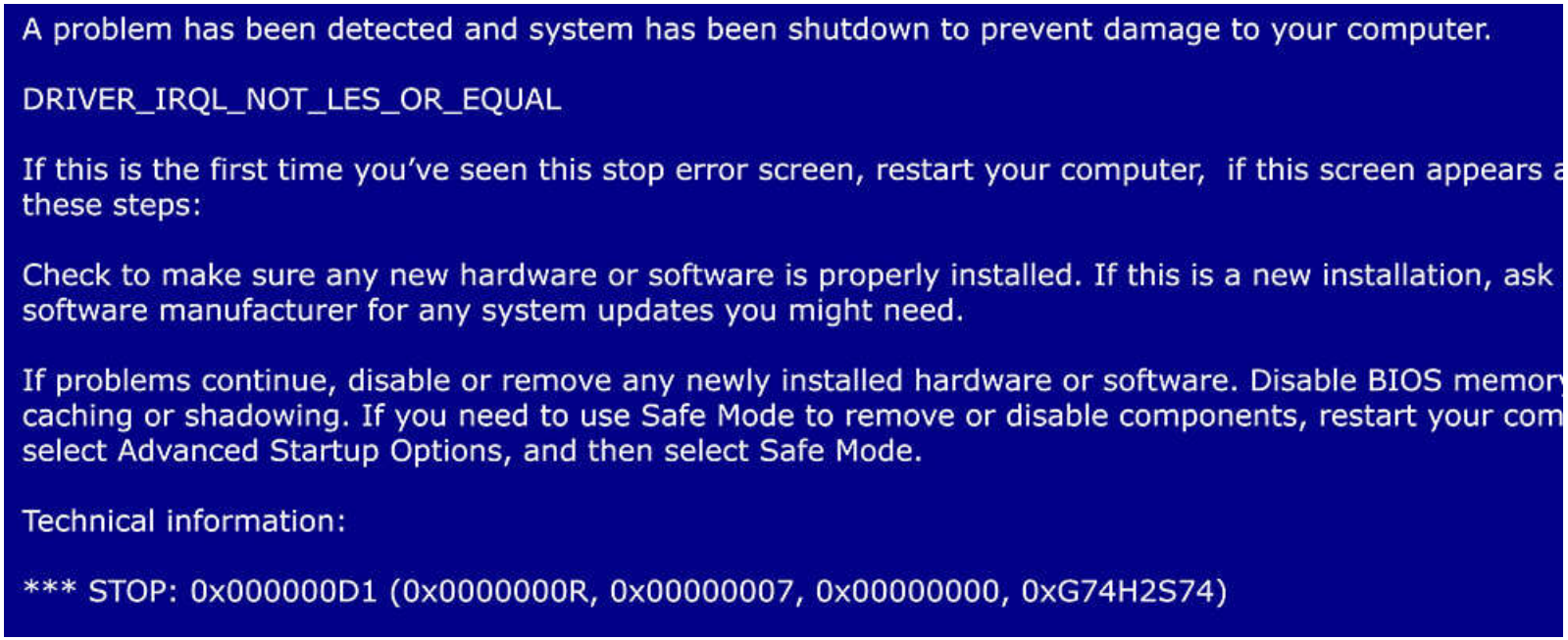


Faults (4)

Random / systematic fault?

Faults (5)

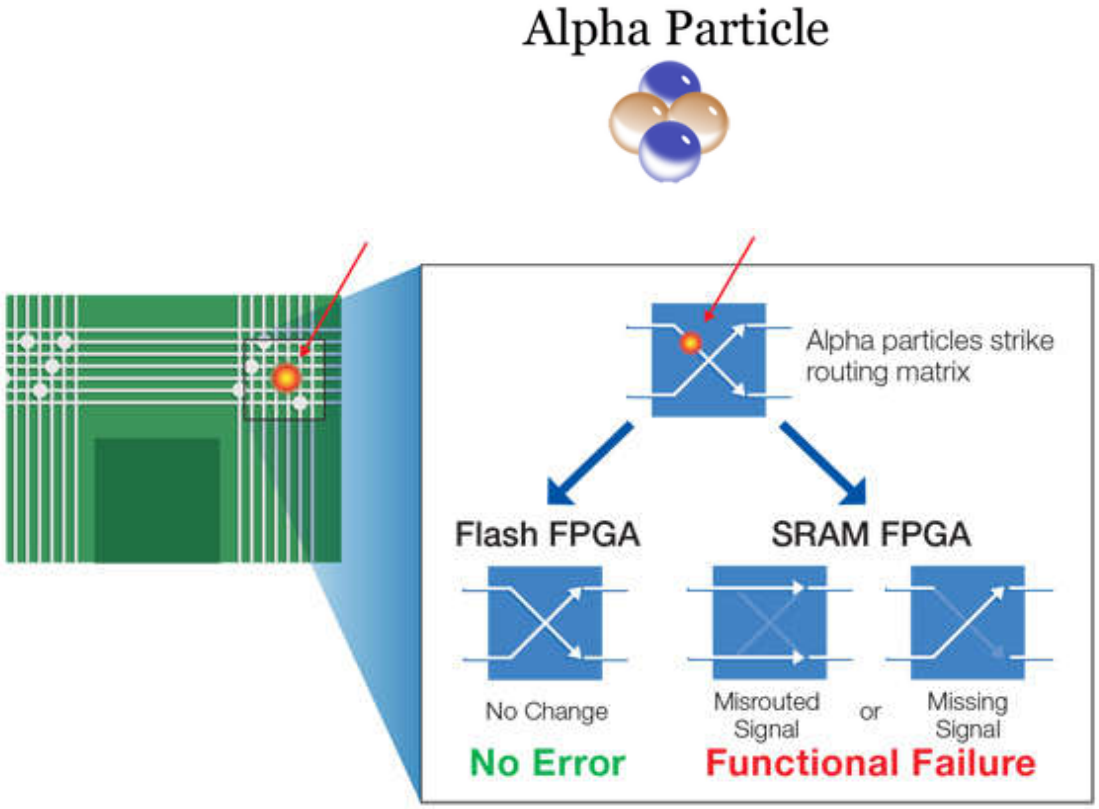
Random / systematic fault?



ALSTOM

Faults (6)

Random / systematic fault?

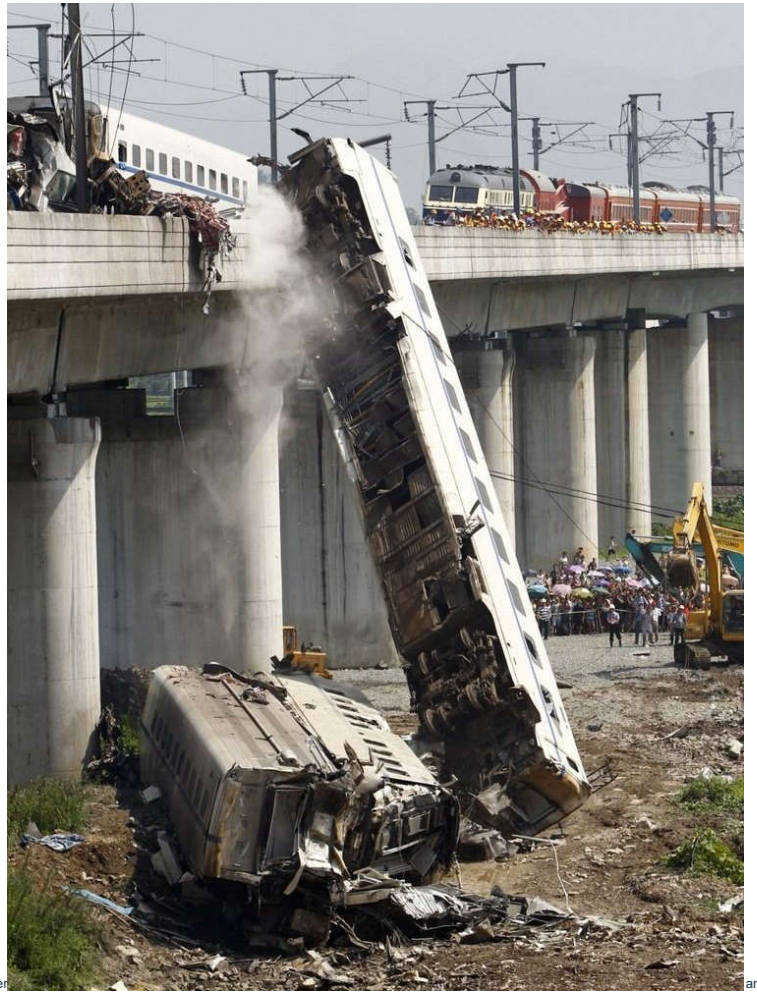


Faults (7)

Random / systematic fault?

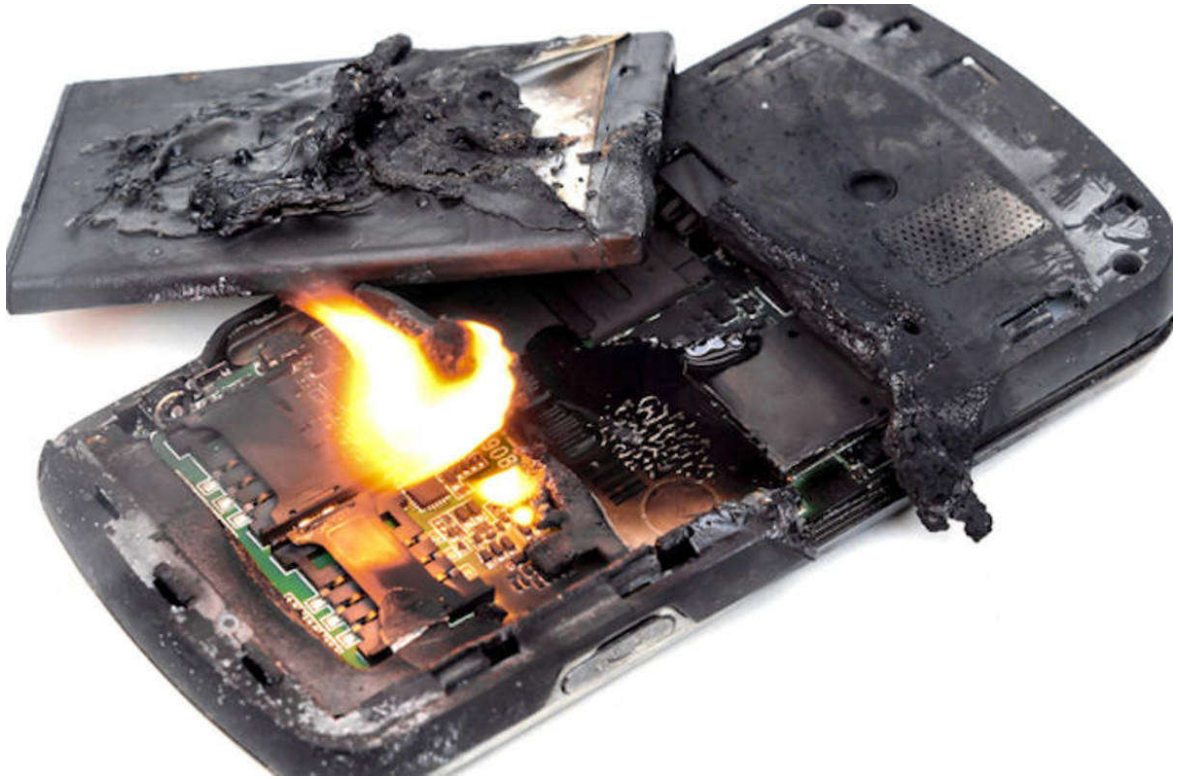
Faults (8)

Random / systematic fault?



Faults (9)

Random / systematic fault?



Countermeasures

Random fault

- ?

Systematic fault

- ?

Faults: how to mitigate?

Please connect!

Countermeasures

Random fault

- Architecture

Systematic fault

- Process

Tolerable Hazard Rate h^{-1}	Safety integrity level
$10^{-9} \leq THR < 10^{-8}$	4
$10^{-8} \leq THR < 10^{-7}$	3
$10^{-7} \leq THR < 10^{-6}$	2
$10^{-6} \leq THR < 10^{-5}$	1

Inherent fail safety

- A safety-related function can be performed by a single item, provided all the credible failure modes of the item are non-hazardous*.

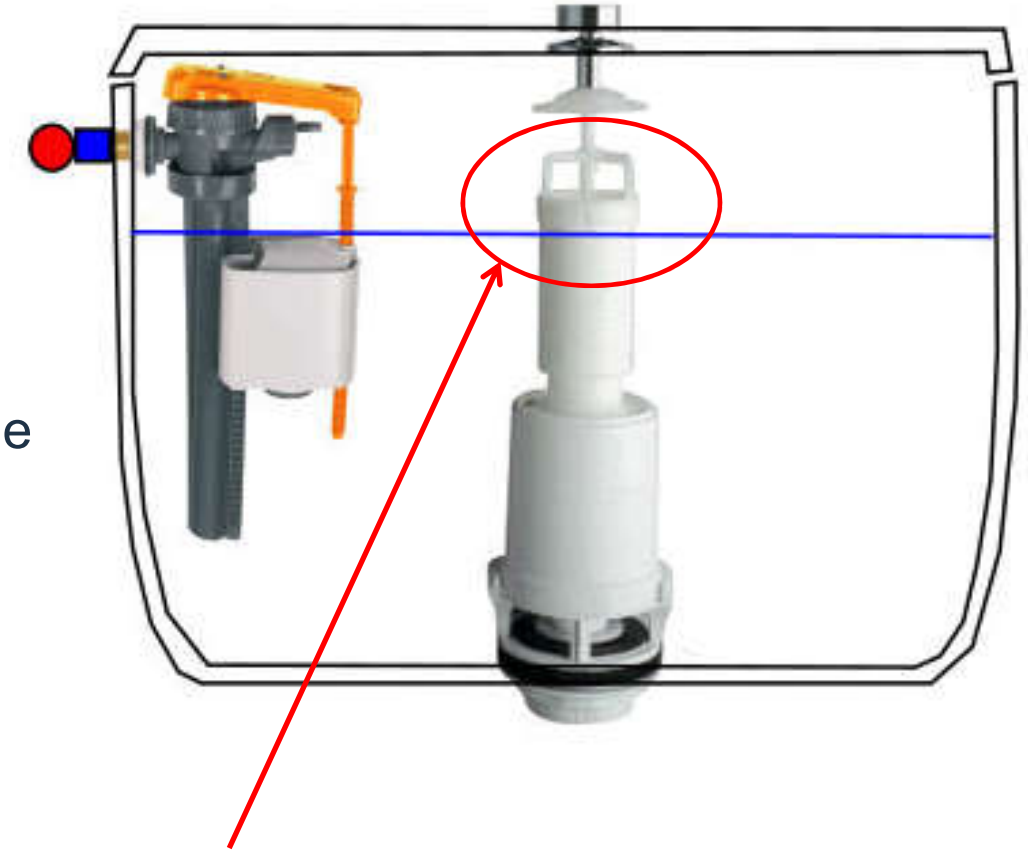
- **Principle:**
  - By construction, the hazardous fault is **physically incredible**.
  - Relies on **a physical characteristic** (absence of energy, thermal characteristics, gravity, buoyancy …)

*EN50129

Inherent fail safety: example

Toilet flush

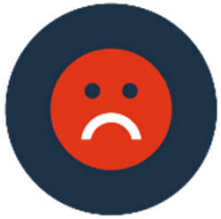
- Hazard: water overflow.
- Principle: thanks to hole, the water is evacuated and will not overflow outside.



Inherent fail safety

- Enforces design simplicity!
  - Cheap
  - Reliable

- Limited to **simple functions**



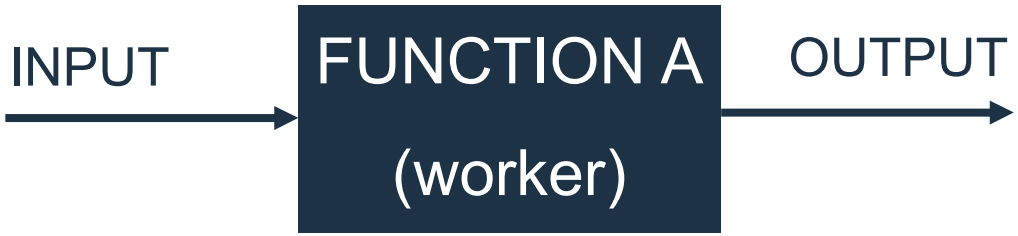
Reactive fail safety

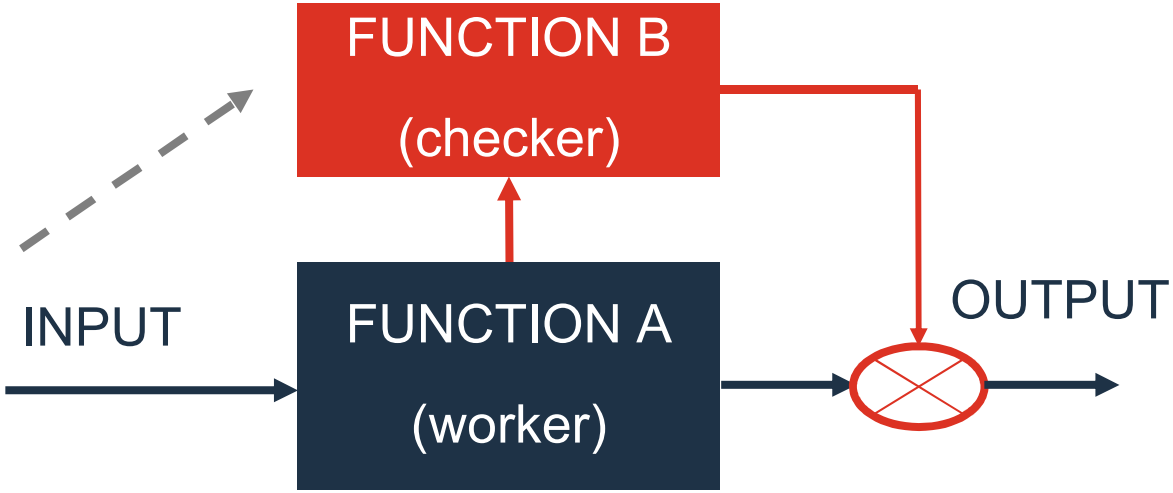
- A safety-related function can be performed by **a single item, provided its safe operation is assured by rapid detection and negation** of any hazardous fault*.

- **Principle:**
  - The function is realised by one item (worker) and then controlled and negated by a second one (checker).

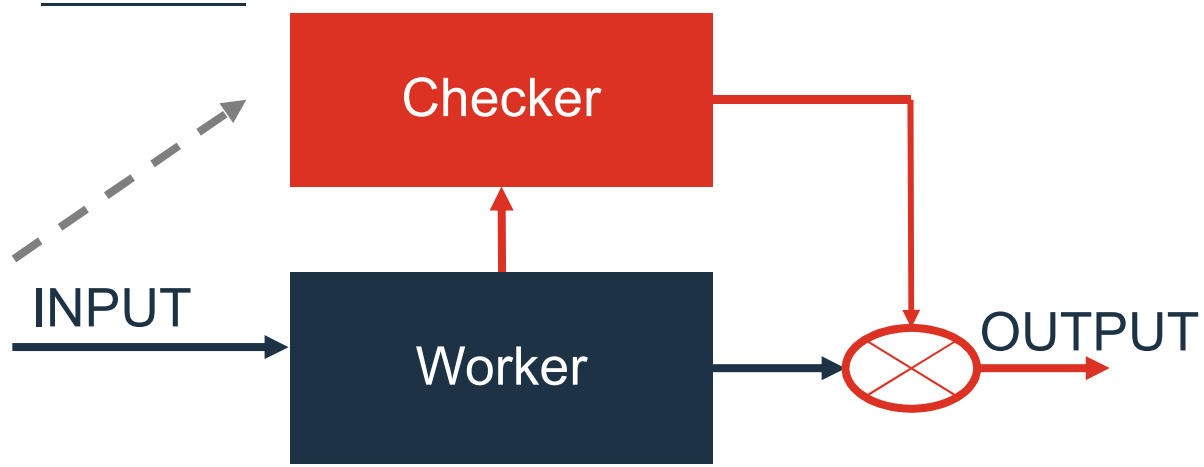
*EN50129

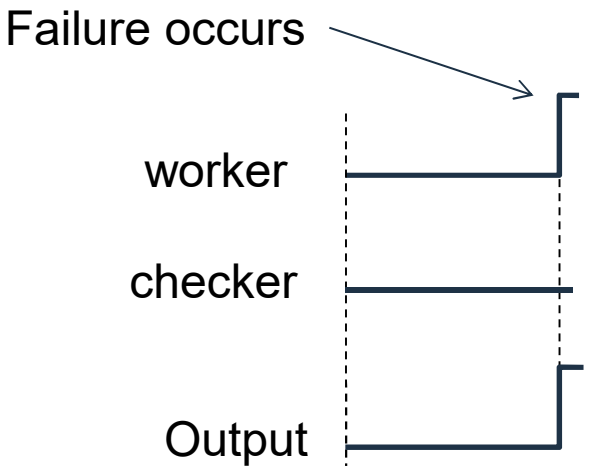
Reactive fail safety: example



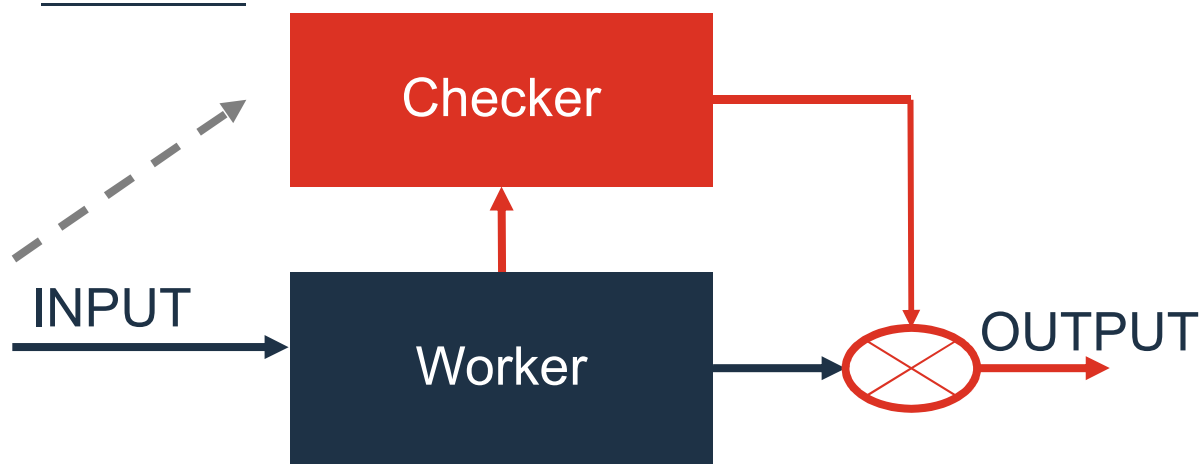


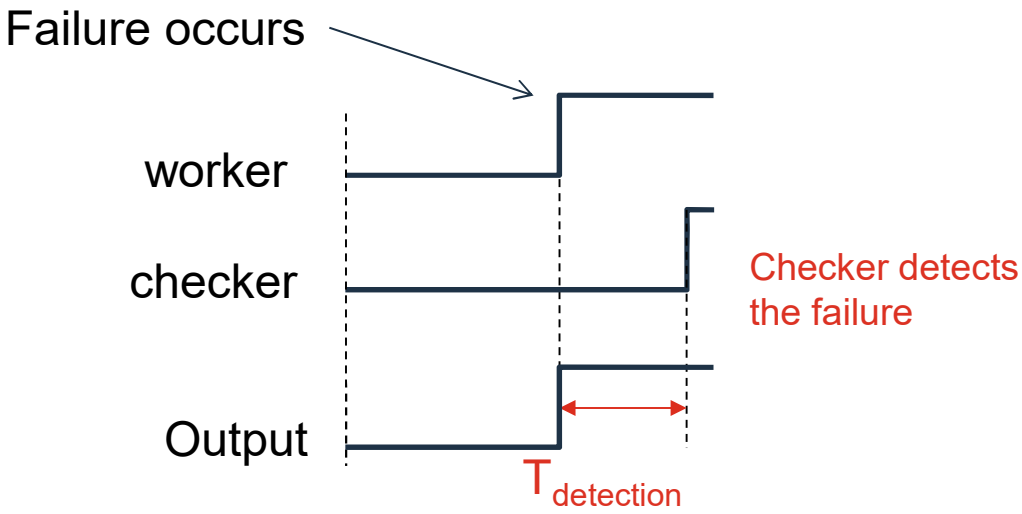
Reactive fail safety: example





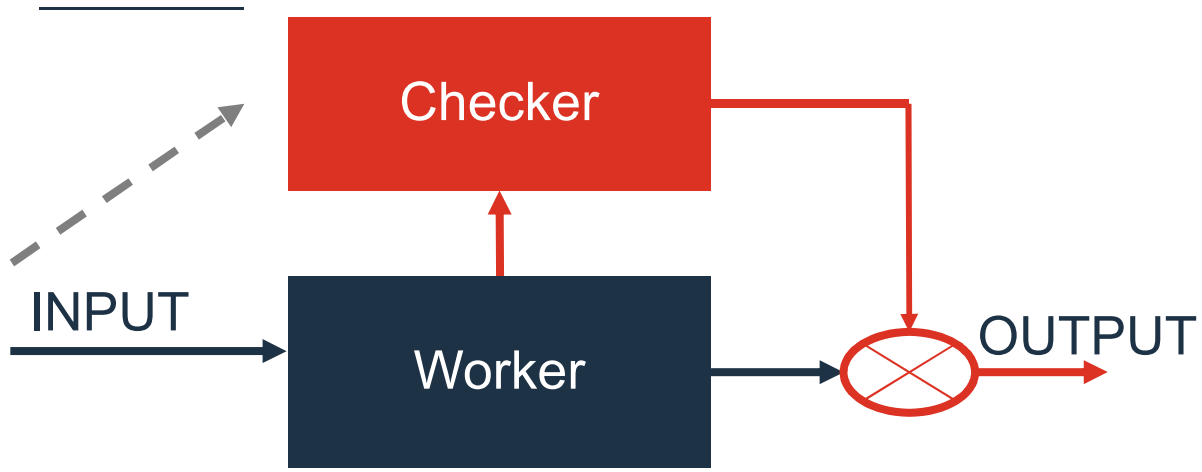
Reactive fail safety: example



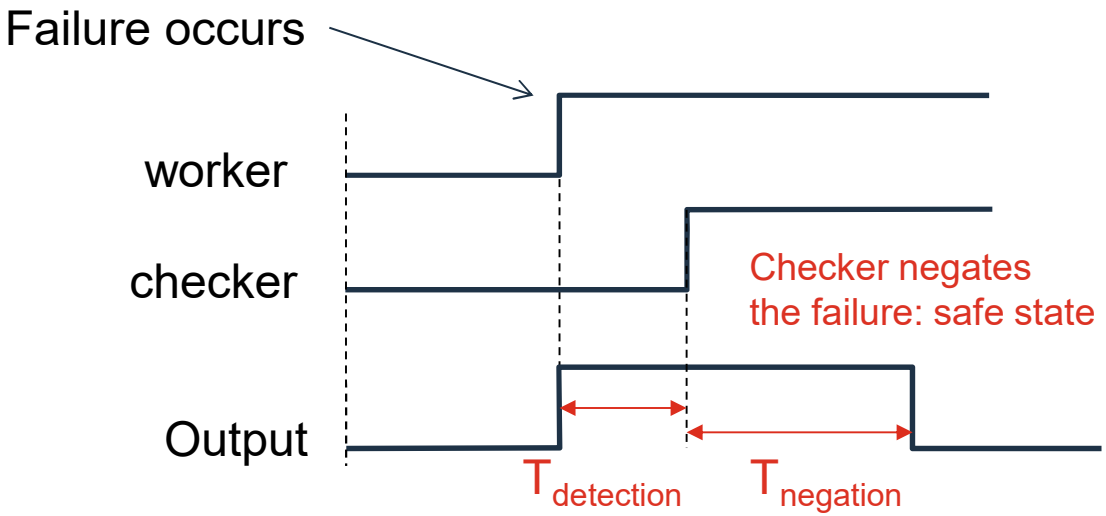


Checker detects the failure

Reactive fail safety: example



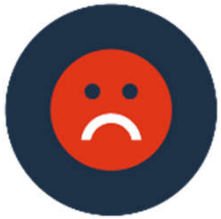
Tdetection + Tnegation must be short enough to avoid the hazardous situation



Reactive fail safety

- Possible to implement complex functions
- Extra cost of safety limited to checker
- Suitable in dissymmetric configurations
  - Complex worker
  - Simple checker

- Erroneous transient shall be tolerated
- Shall react very fast
- Each implementation is specific



Composite fail safety

- **At least 2 channels perform the same function**. By principle, non restrictive activities are realized only if the **2 channels agree***.

- **Principle:**
  - At least 2 independent items realize the same function.
  - Comparison of outputs of the 2 items
  - Disagreement => Outputs are set in a safe state

*EN50129

Composite fail safety: example

- **Use Case:** a student has an exam tomorrow
- **Boundary Hazard:** he does not wake up and misses the exam.

- Good idea! Use composite fail safety
  - 2 alarm clocks

Composite fail safety: example

- **Production of output (vote):** Jamming of the sound of the 2 alarm clocks

- **Detection of fault** is made periodically at each wake up:
  - to check that the 2 alarm clocks have been triggered
  - to check the time (absence of clock drift).

- If at least one clock is failed, **imposition of safe state**:
  - the student doesn't sleep before to maintain the failed item.
  - Ask wake up support to your neighbour?

- **Avoid systematic fault**
  - Make sure that the sound level of each alarm is enough to wake up the student !

Composite fail safety: example

Independence

- Mechanical support: Each side of the bed to avoid simultaneous fall.

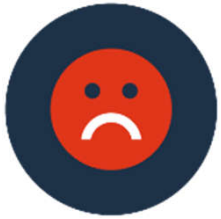
- Energy : 2 sources of energy:
  - Mains/Battery
  - Mechanical

- Diversification:
  - 2 different technologies

Composite fail safety

- Very generic approach

- Cost
- Reliability
- More complex that it looks



Safety architectures

- Inherent
- Reactive
- Composite

… other concepts exist

Concepts are almost always mixed

Part 1 conclusion

Safety architectures

Guillaume Vibert

CPE – part 2

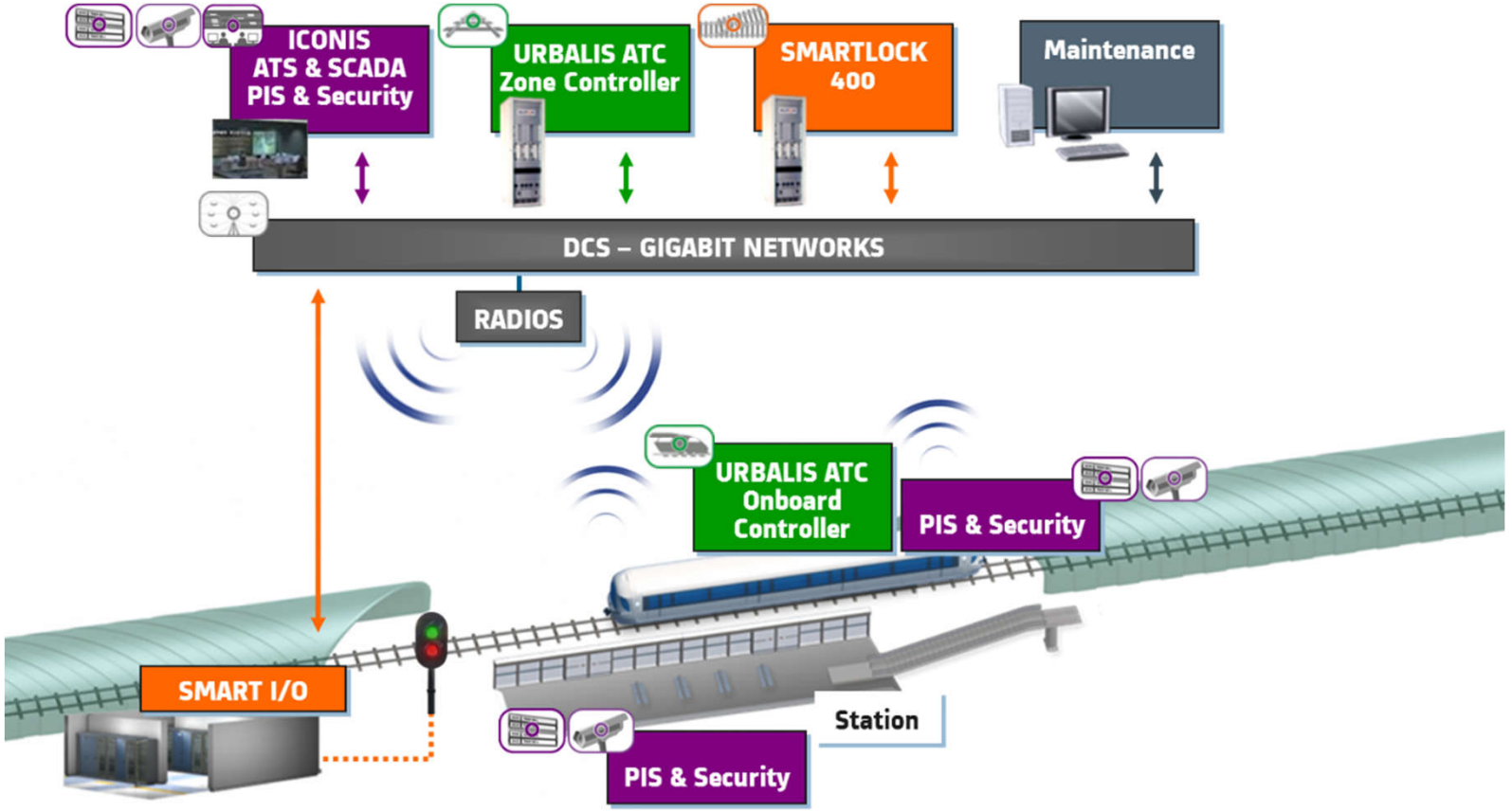
ALSTOM
• mobility by nature •

Wake-up!

Please connect!

Safety communication

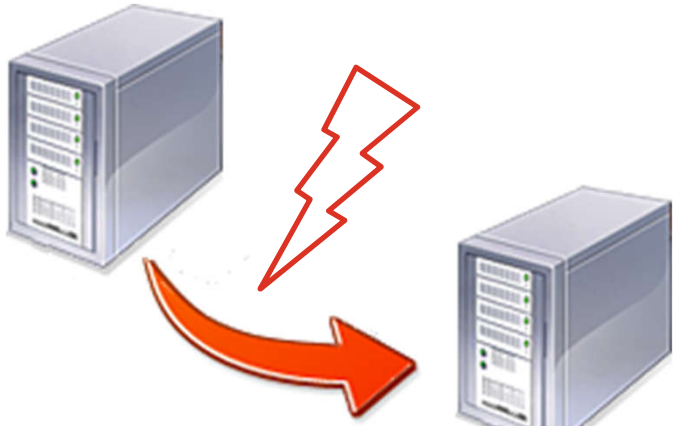
Example of CBTC application: ***Communication Based*** *Train Control*



Safety communication: threats

Basic communication model

- One data producer
- One data receiver
- One digital communication system



Identify threats:

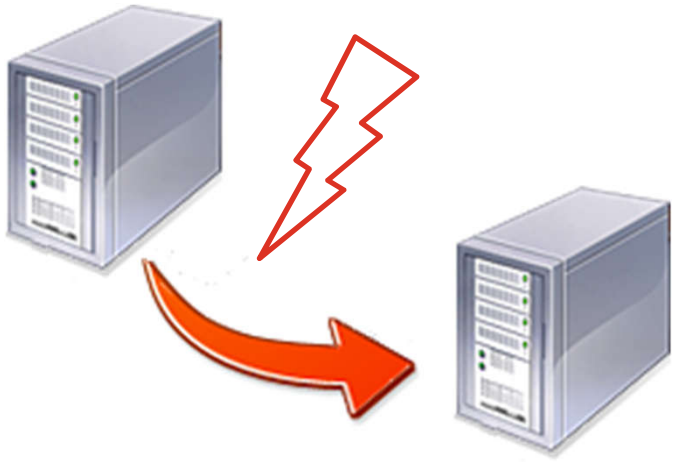
- How can data be altered by the communication system ?

Communication threats

Please connect!

Safety communication: threats

Threats defined by EN50159 : 2010



REPETITION	Hello world world !
DELETION	world !
CORRUPTION	H*llo w%rld !
DELAY	Hello … … … … … … … world !
RE-SEQUENCE	world ! Hello
INSERTION	Hello everyone in the world !
MASQUERADE	*Ciao world !*

Communication defences

Please connect!

Safety communication: defences

Table 1 – Threats/Defences matrix

Threats	Defences							
	Sequence number	Time stamp	Time-out	Source and destination identifiers	Feed-back message	Identification procedure	Safety code	Cryptographic techniques
Repetition	X	X						
Deletion	X							
Insertion	X			X [a]	X [b]	X [b]		
Re-sequence	X	X						
Corruption							X [c]	X
Delay		X	X					
Masquerade					X [b]	X [b]		X [c]

[a] Only applicable for source identifier.
Will only detect insertion from invalid source.
If unique identifiers cannot be determined because of unknown users, a cryptographic technique shall be used, see 7.3.8.

[b] Application dependent.

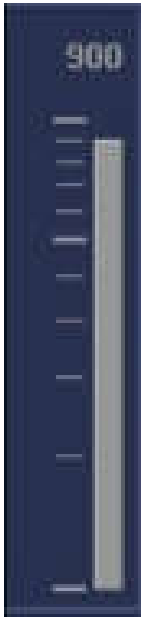
[c] See 7.4.3 and Clause C.2.

Reactive fail-safe architecture

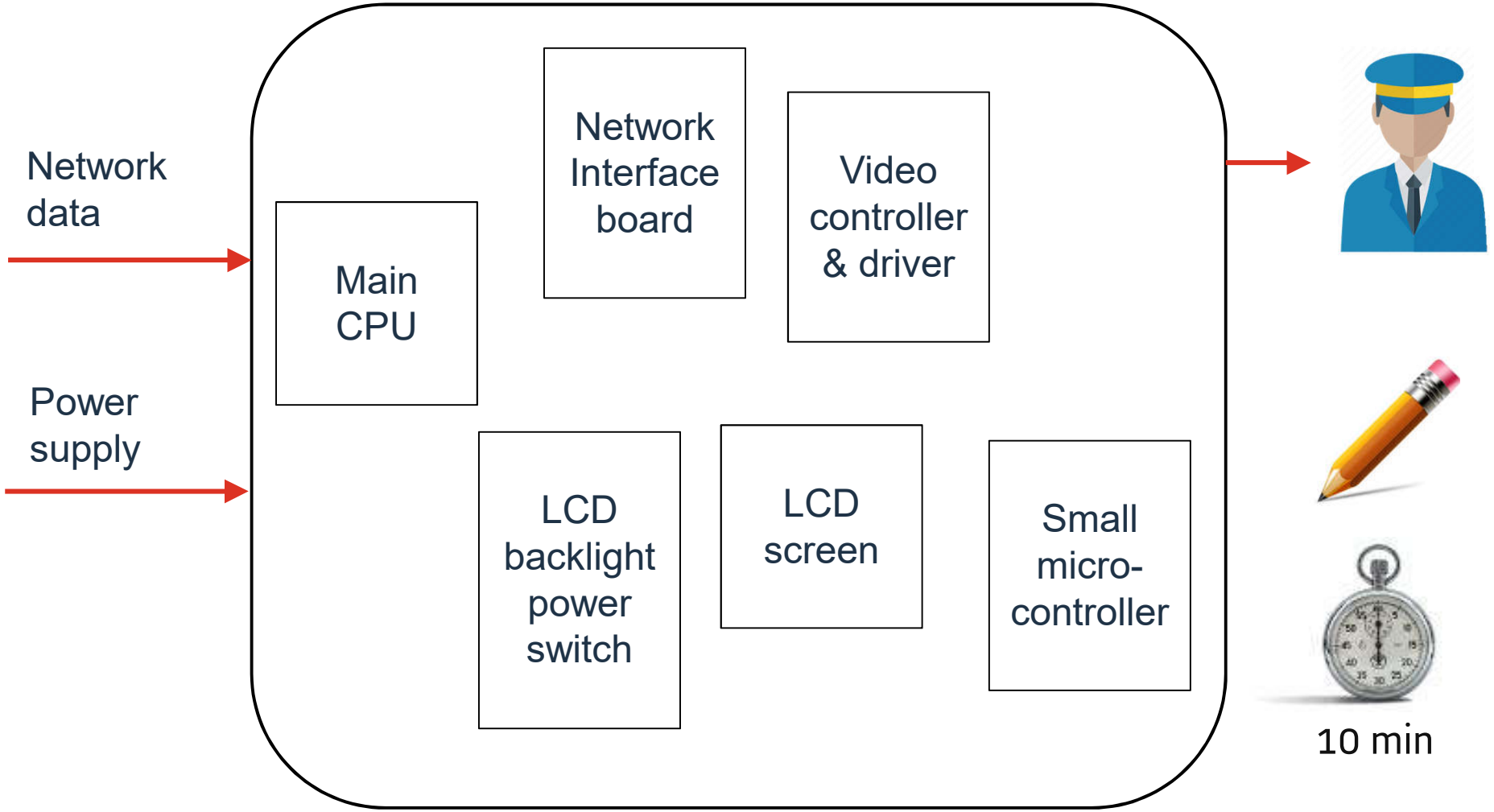
Design a reactive fail-safe system

Safe display for Driver-Machine Interface

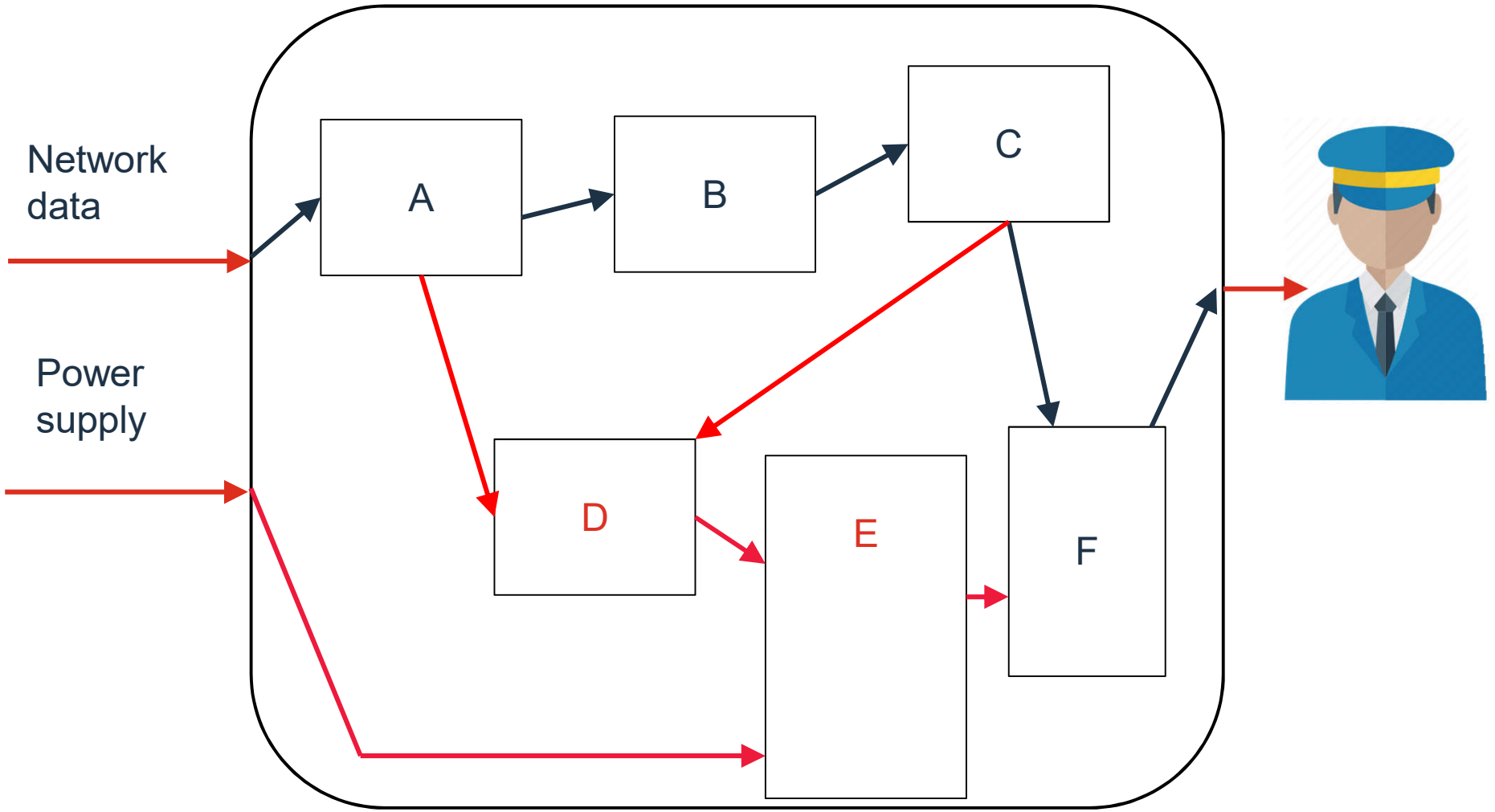
- **Need:** Display safety-related graphical objects to the driver
- **Boundary Hazard:** Display of a frozen or corrupted safety graphical objects
- **THR:** 2*10-7/h
- **Required SIL**: SIL2
- Examples of objects:



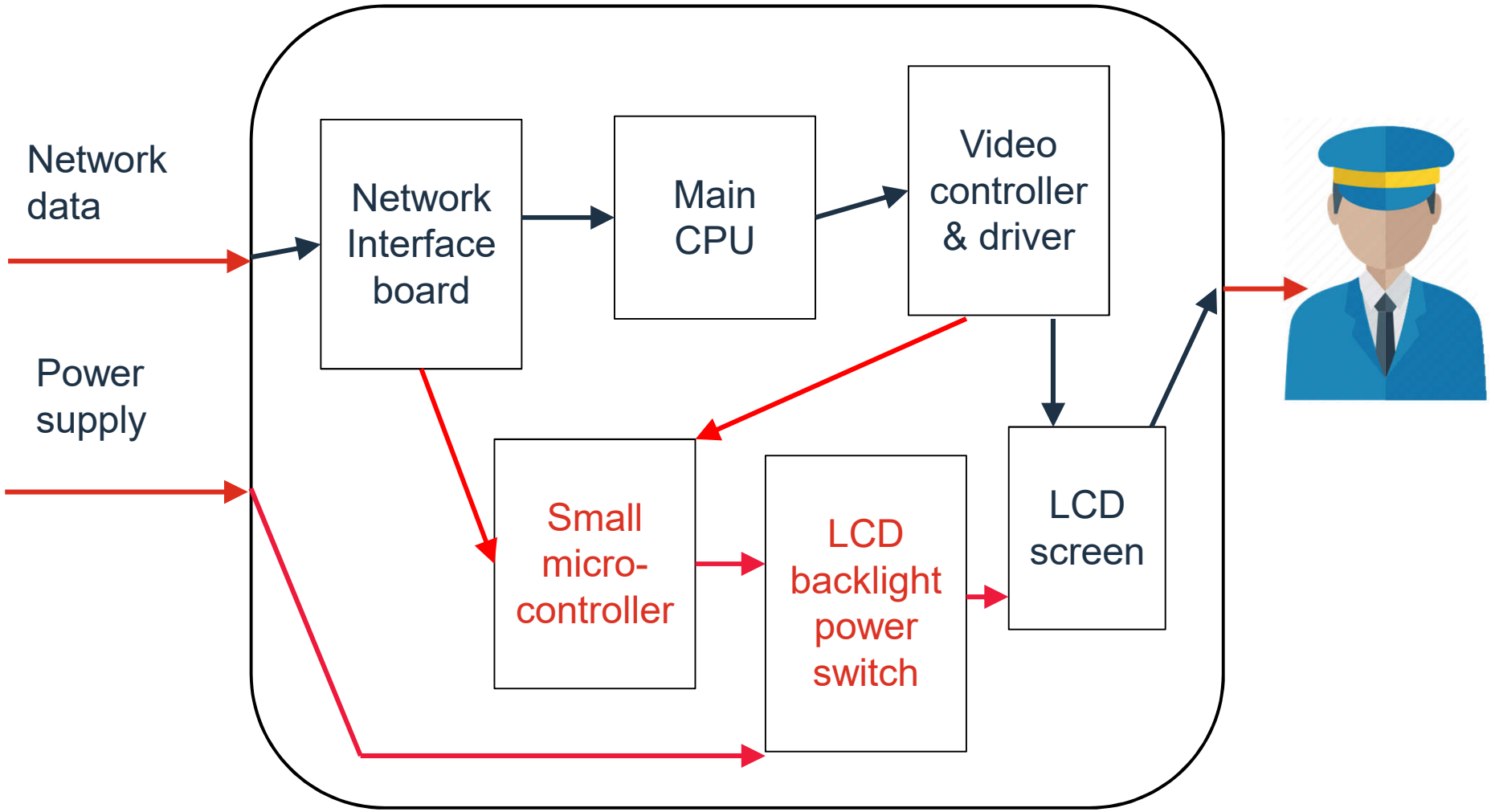
Design a reactive fail-safe system



Design a reactive fail-safe system



Design a reactive fail-safe system



Reactive fail-safe system

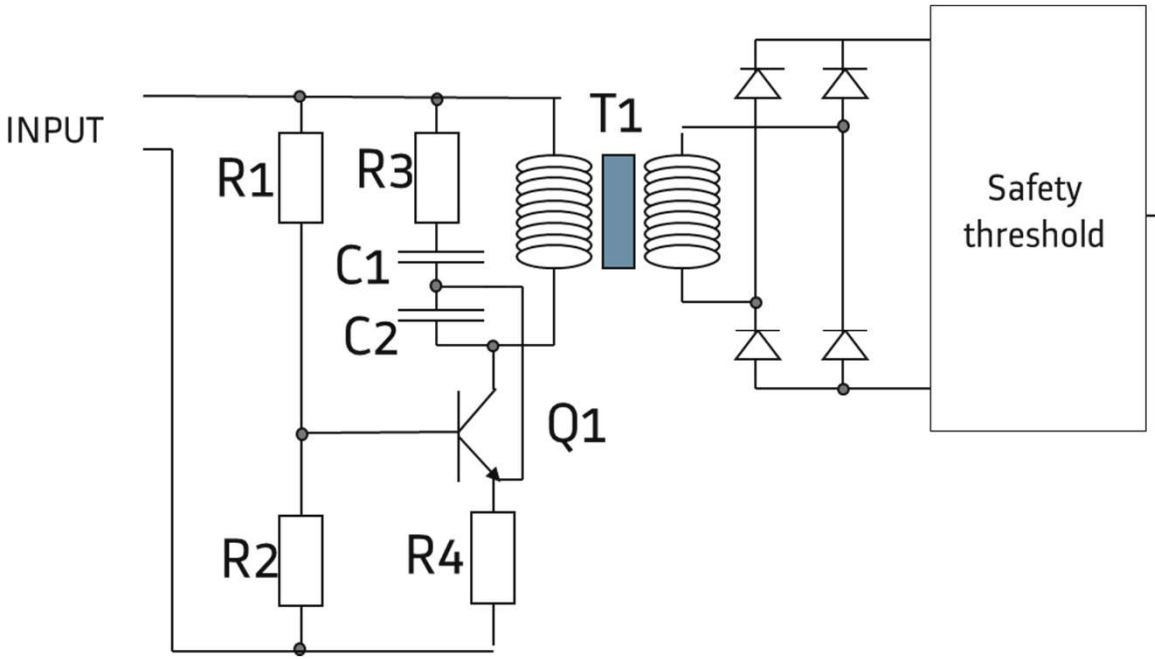
- Synchronization is not an option!
- Illustration of “Safety vs. Availability” trade-off

Break

ALSTOM

Inherent fail-safe architecture

Inherent fail-safe architecture



Inherent fail-safe architecture

“Mono Processeur codé”

Immune to:

- Arithmetical error
- Operator error
- Operand error
- Data refresh error
- Branch error

Inherent fail-safe architecture

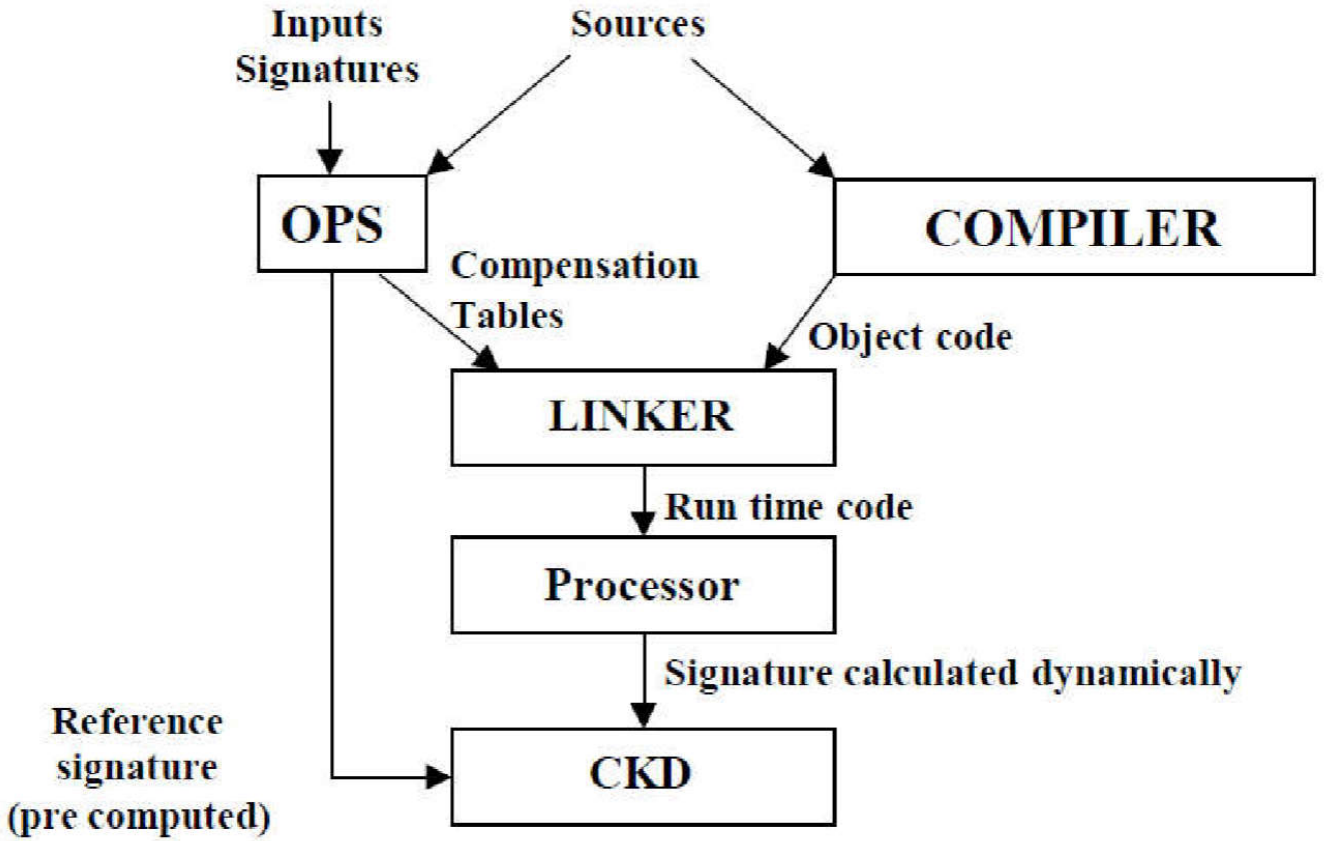
X.C = Control part = -Rk(X) +Bx(X) + D

- -Rk(X) = (2^k * X.F) mod A
  - A primary, 2^k > A
- Bx(X) = data signature, constant
- D = processing cycle number

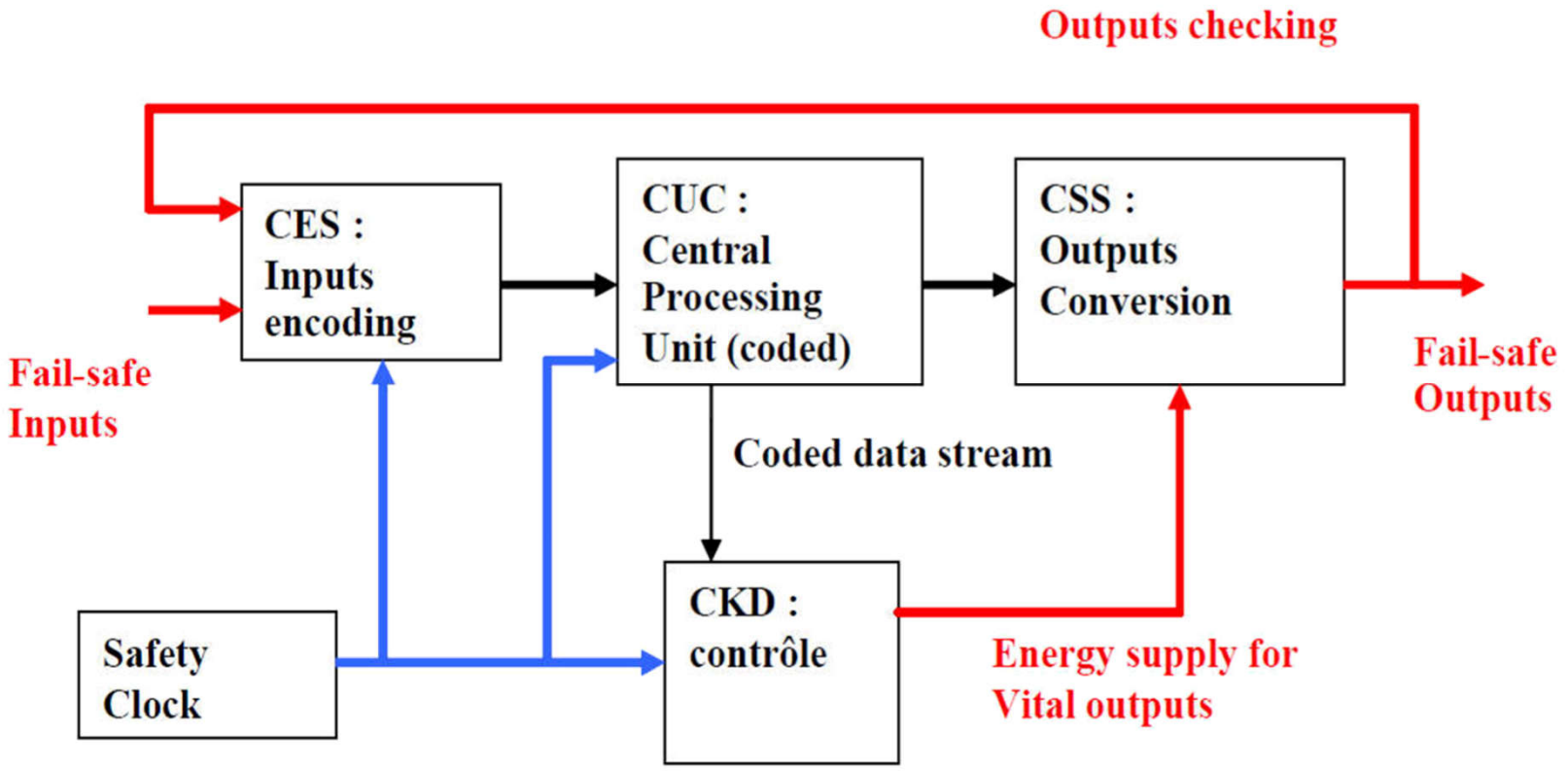
All operations on data are done with specific operators

- (Z = X +~ Y)

Inherent fail-safe architecture

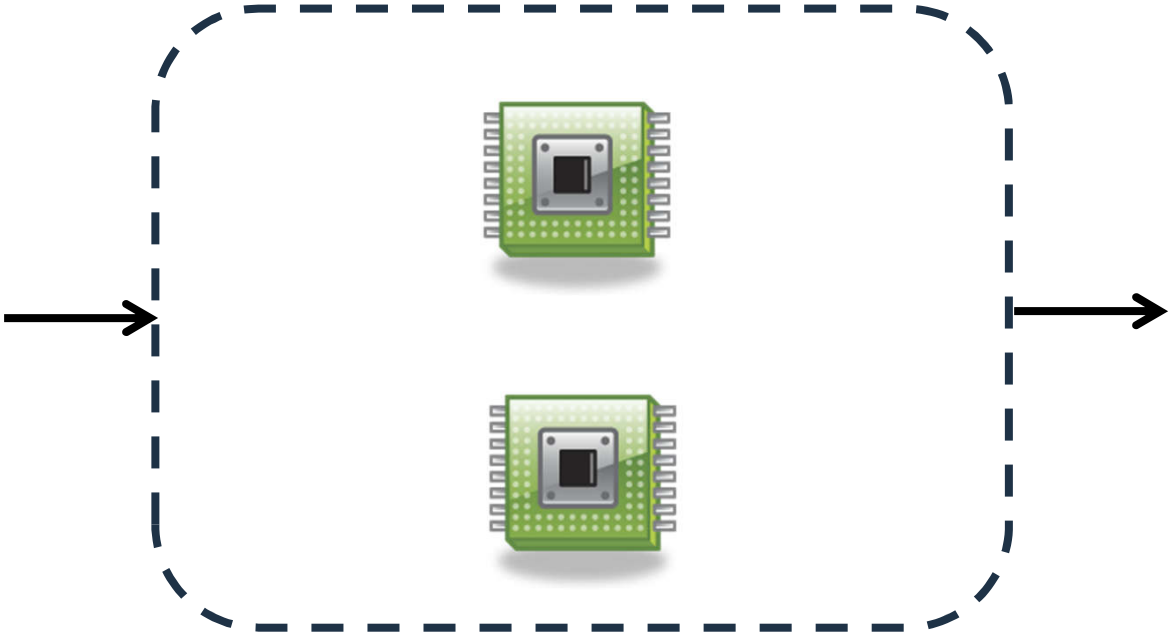


Inherent fail-safe architecture



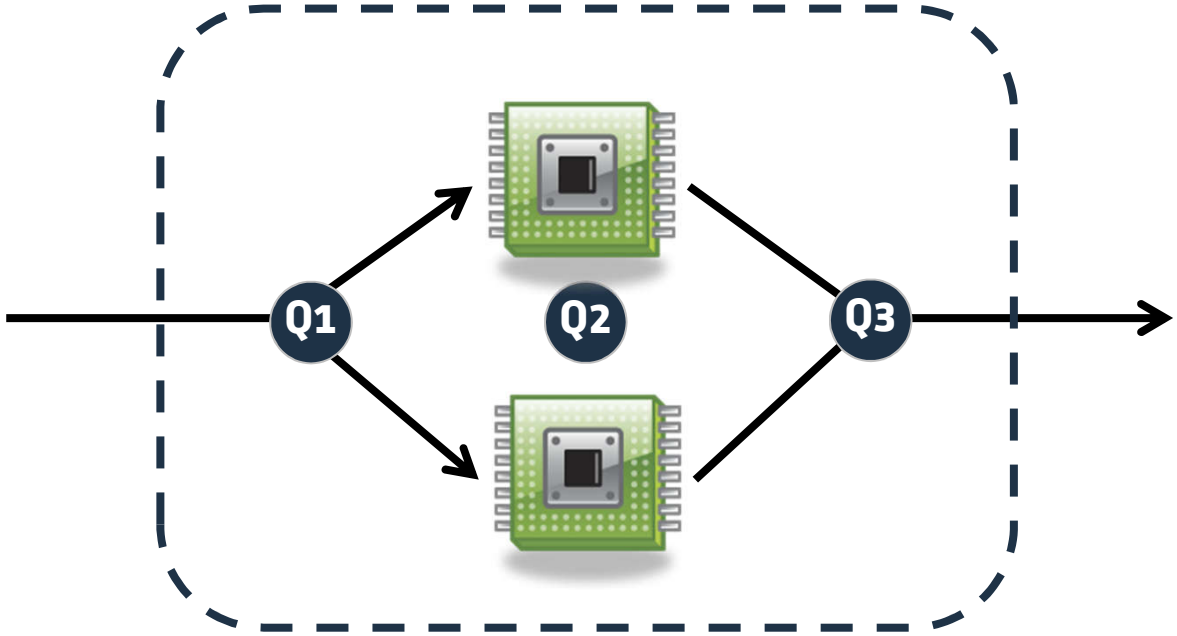
Composite fail-safe architecture

Composite fail-safe architecture



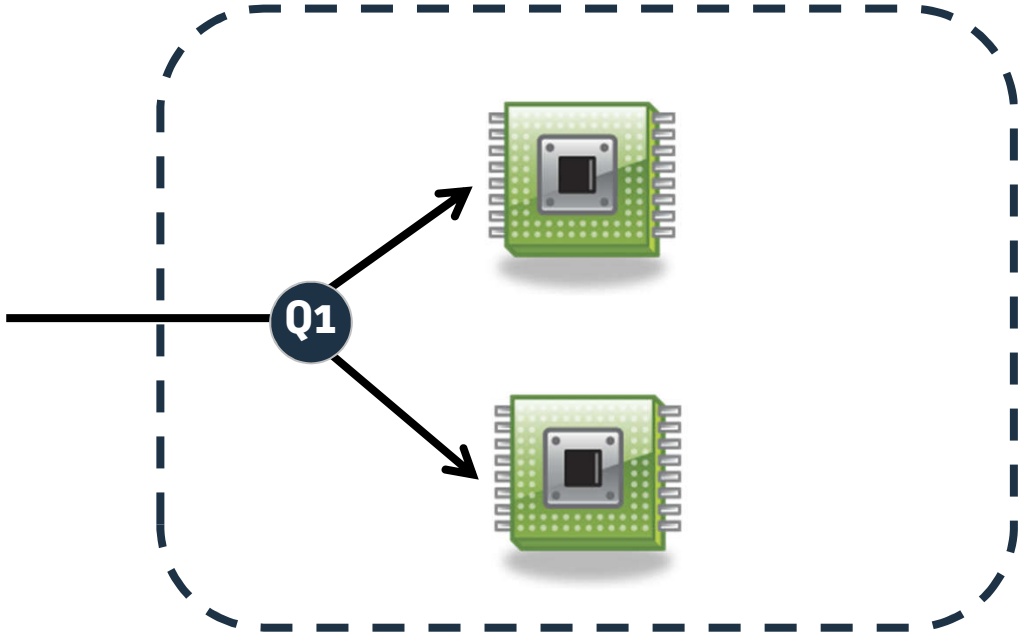
- Basic structure: 2-out-of-2 (2oo2)

Composite fail-safe architecture



- Q1: synchronisation of inputs?
- Q2: independence of items?
- Q3: production of outputs (« vote »)?

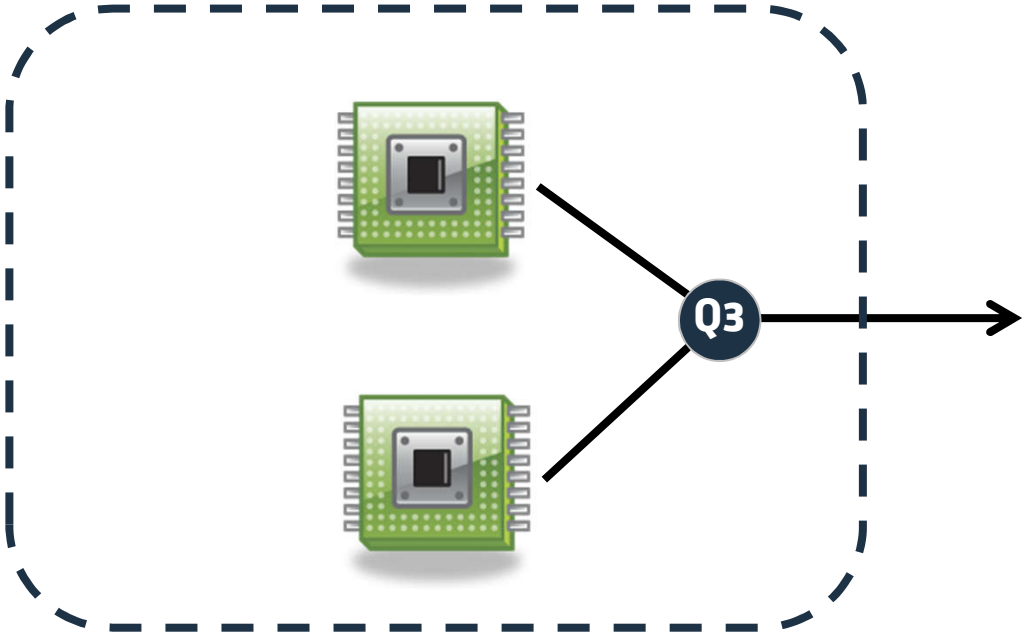
Composite fail-safe architecture



- Temporal / logical synchronisation
- Need to define a common time reference

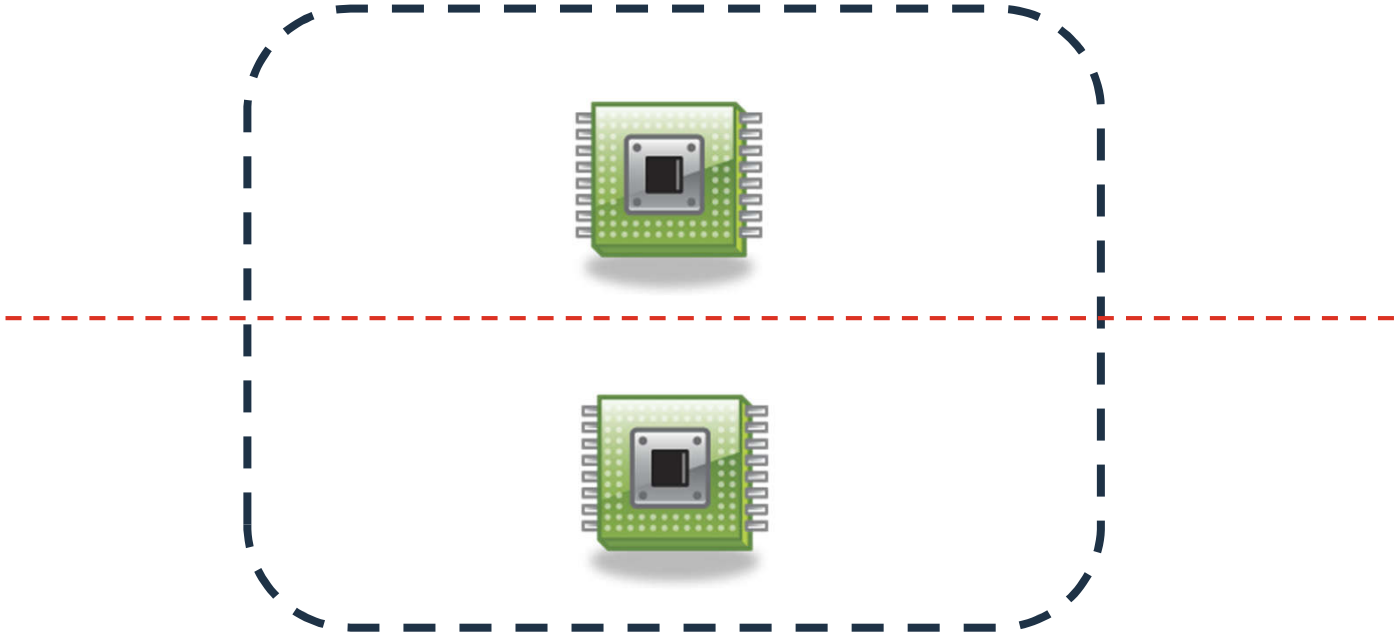
- Synchronisation: a key to stability

Composite fail-safe architecture



- What if the “voter” fails?

Composite fail-safe architecture

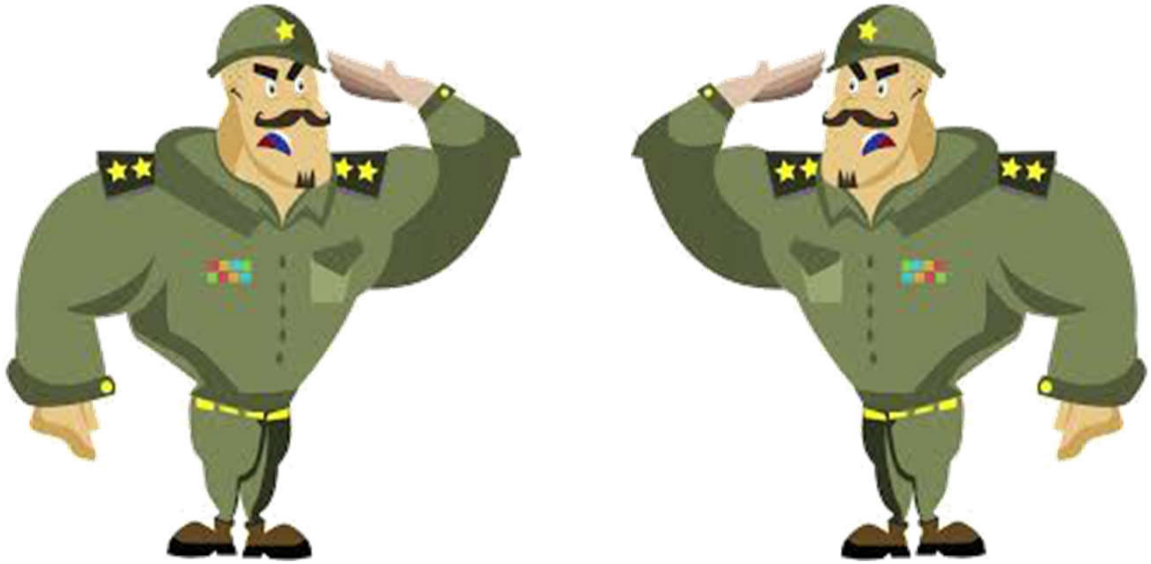


- Independence needed: avoid common modes
- But… in contradiction with Q1, Q3!

Composite fail-safe architecture: consensus problem

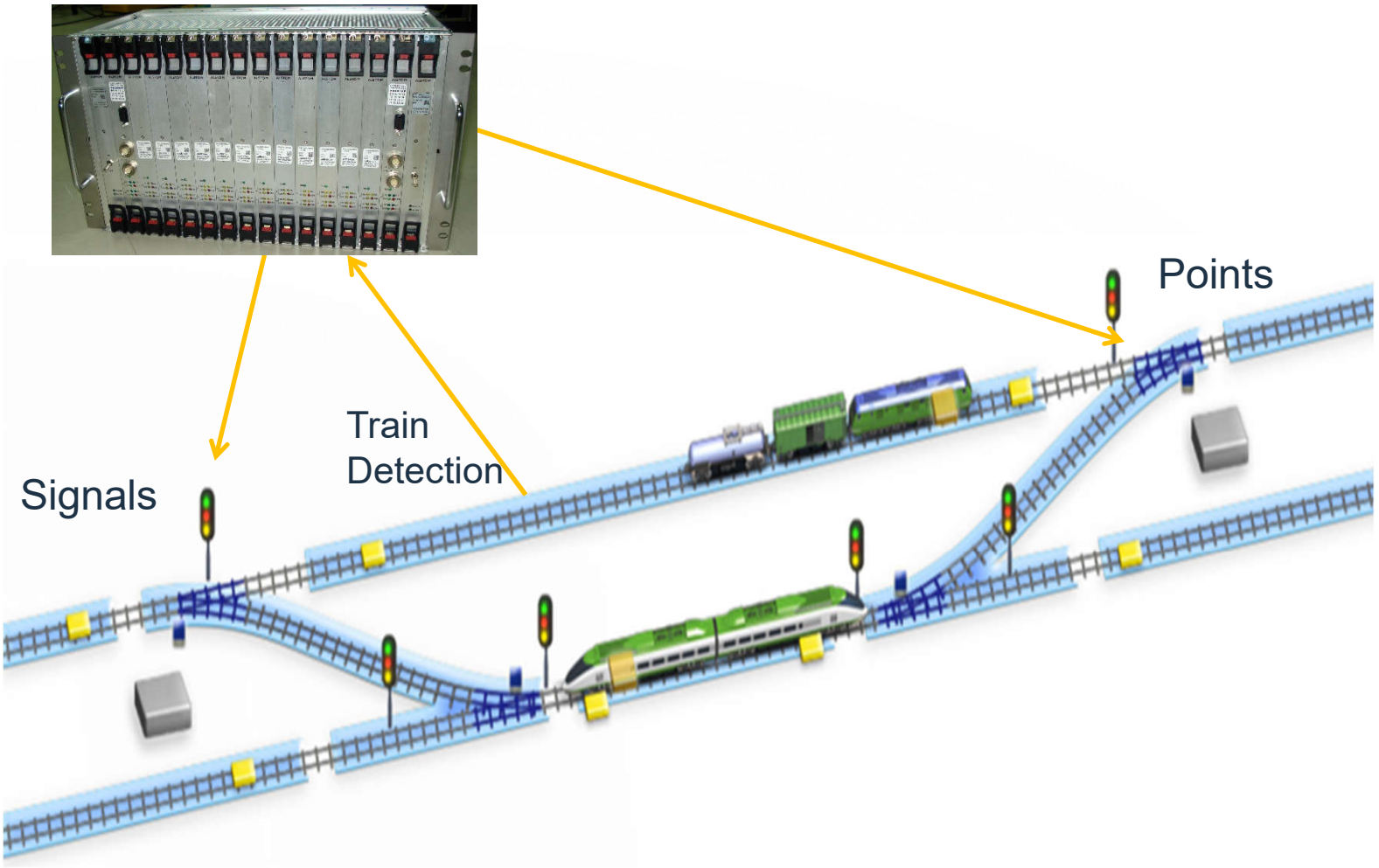
Please connect!

Composite fail-safe architecture: consensus problem



“The Two Generals Problem was the first computer communication problem to be proved to be unsolvable”

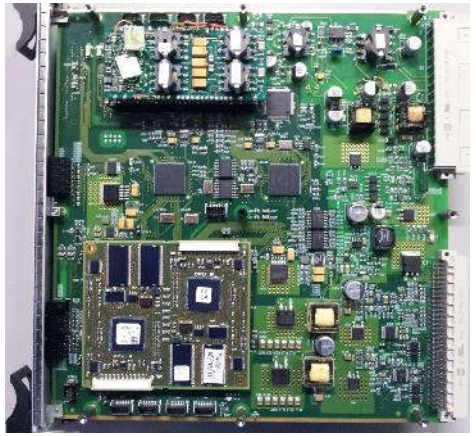
Composite fail-safe architecture: example



Composite fail-safe architecture: example

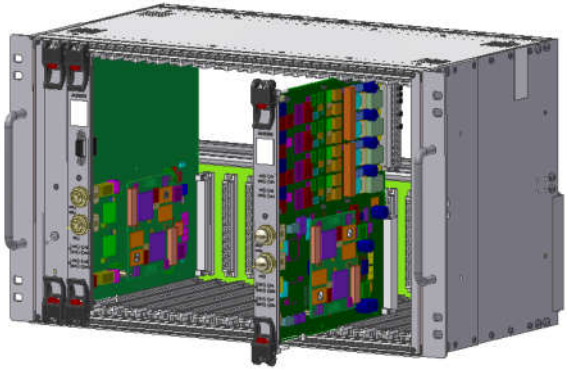
Function

- Drive 1…8 digital outputs (0/1) from network data

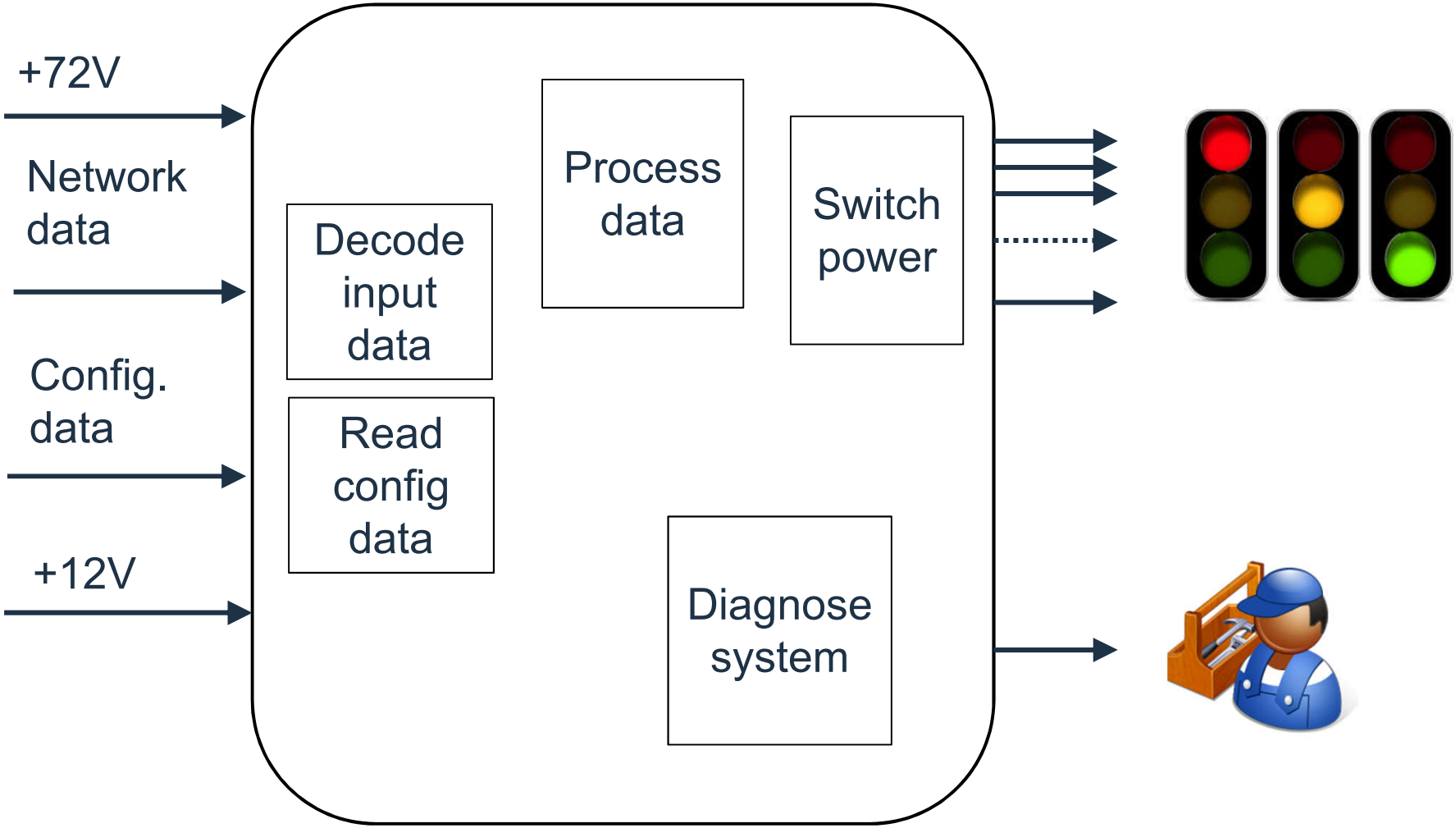


Hazard

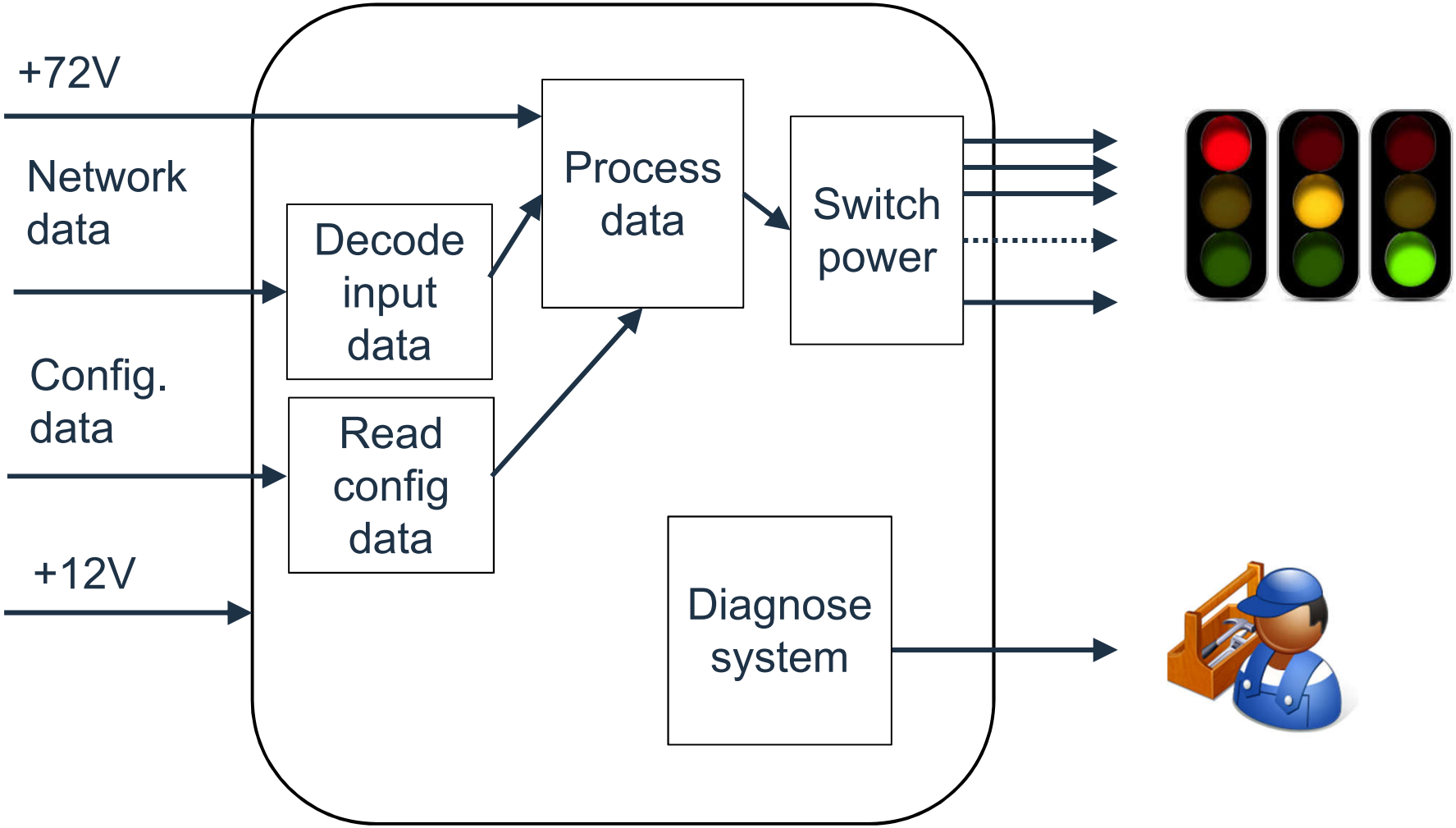
- Drive output to 1 erroneously
- THR = 10^{-12} h^{-1}



Functional analysis



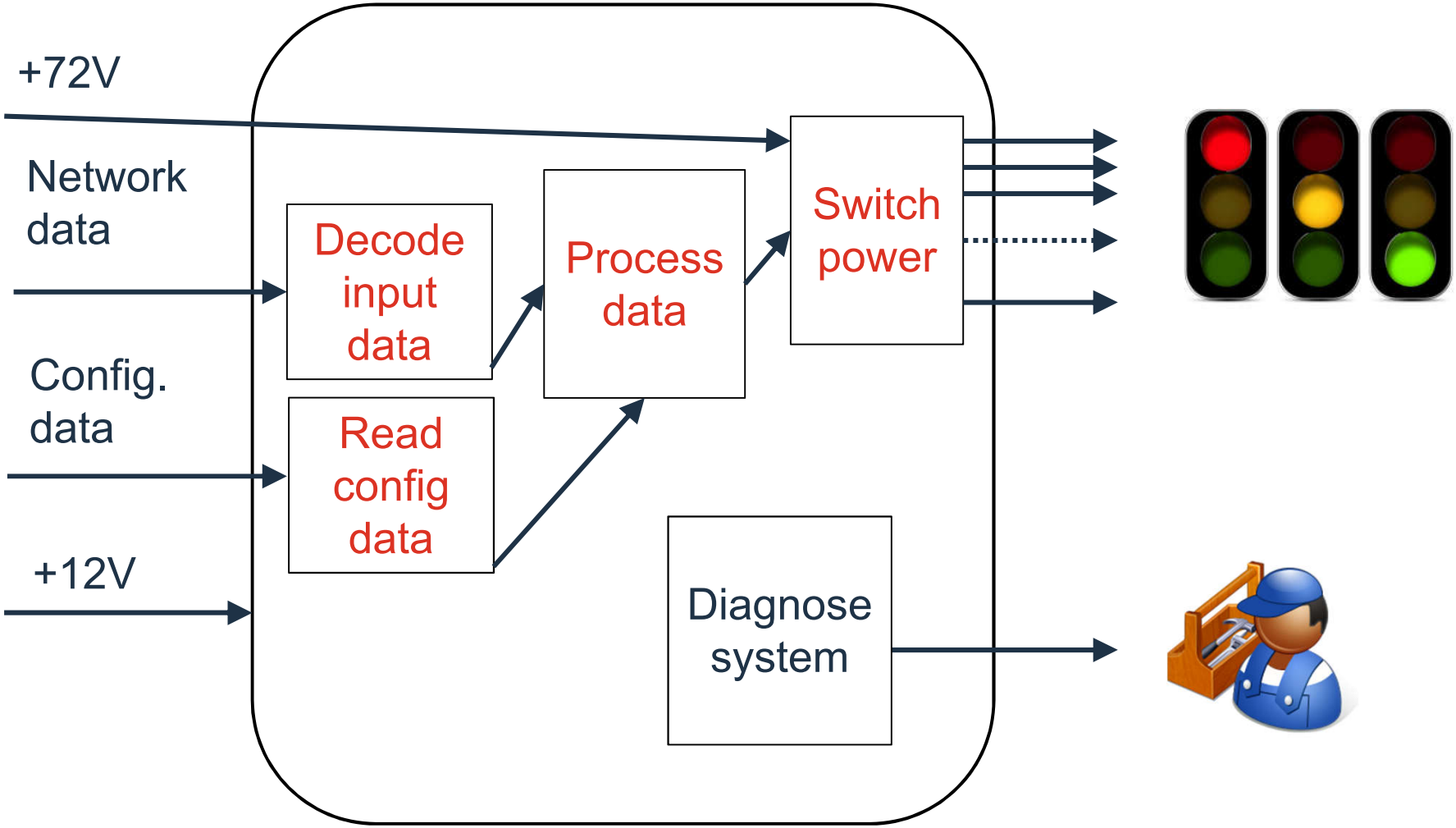
Functional analysis … Dysfunctional analysis



Functional analysis … Dysfunctional analysis

Please connect!

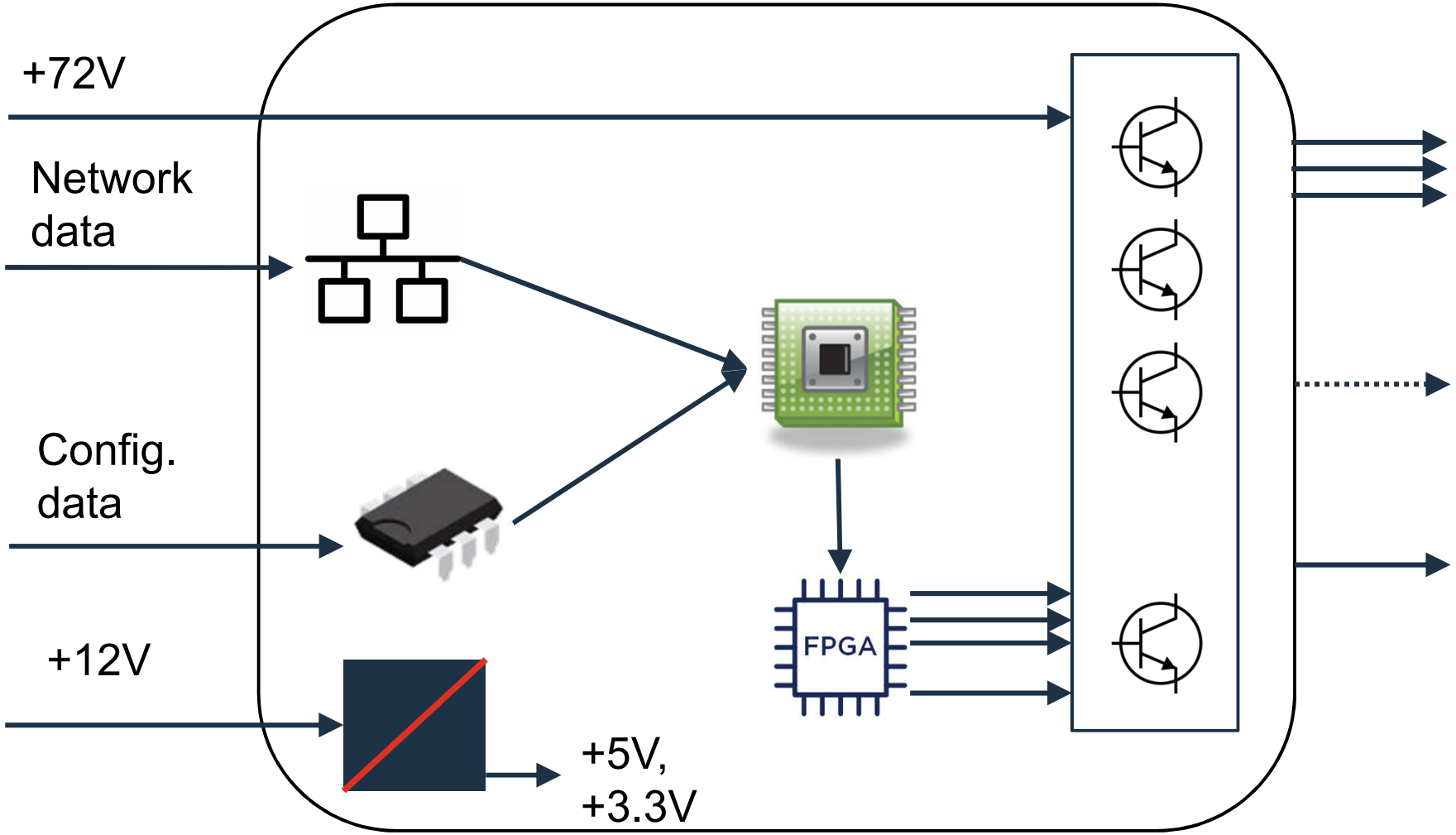
Functional analysis … Dysfunctional analysis



Architecture patterns

Please connect!

Constructional analysis

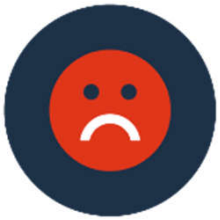


Duplicate components?

Please connect!

Safety analysis

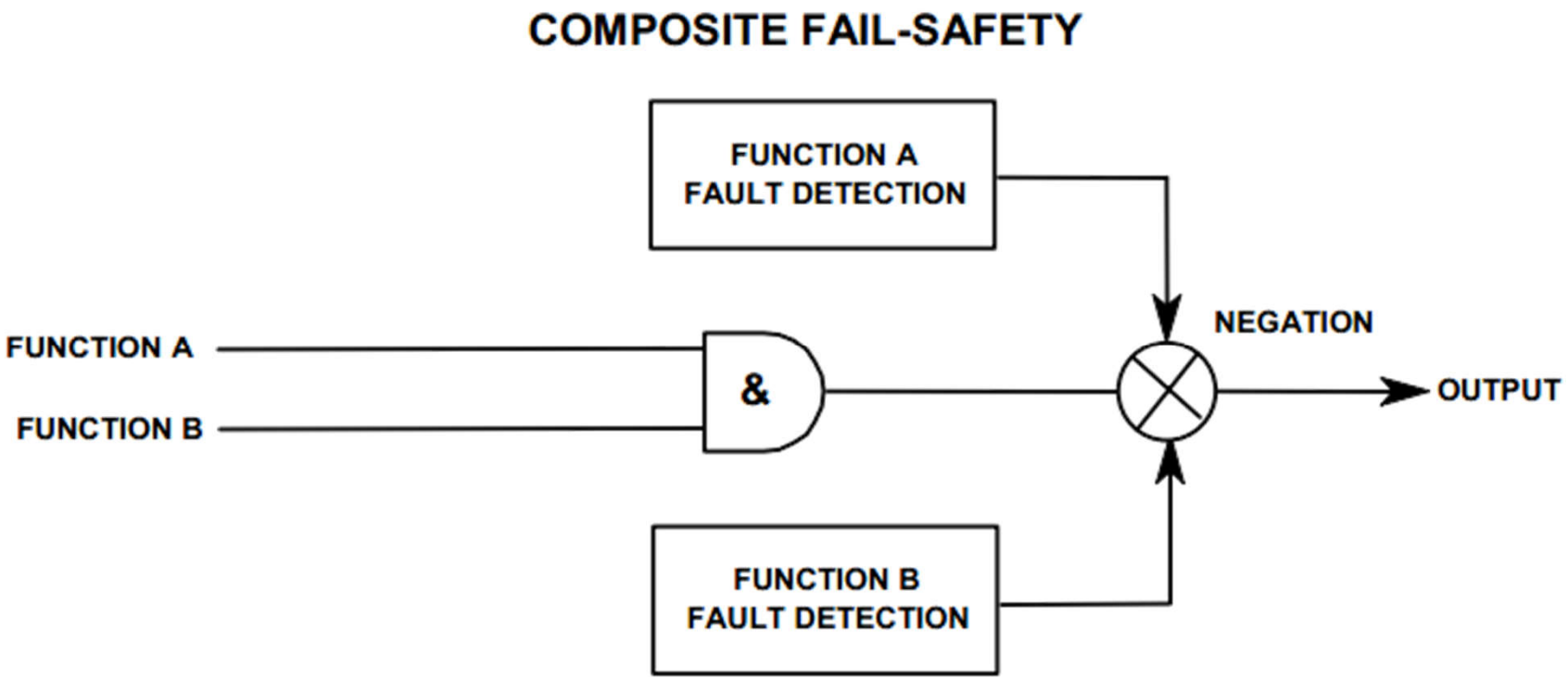
- **Define system model**
- Analyze possible failure modes (FMEA)
- Compute statistically the residual risk
  “Wrong side failure rate”
- Exercise: Wrong side failure rate is too high…



Safety analysis

Please connect!

Safety analysis



Conclusion

Conclusion

- Don’t forget dysfunctional approach

"Anything that can go wrong will go wrong“

… sooner or later

- Robustness is a key

Your feedback

Please connect!