SAFETY-CRITICAL SYSTEMS FROM THE INSIDE

Solw'IT Let's Solve It

MACIEJ GAJDZICA

MACIEJ GAJDZICA

Solw'IT Let's Solve It

- Senior Embedded Developer
- automotive, railway, medical systems
- ucgosu.pl - blog, YouTube
- Gdańsk Embedded Meetup

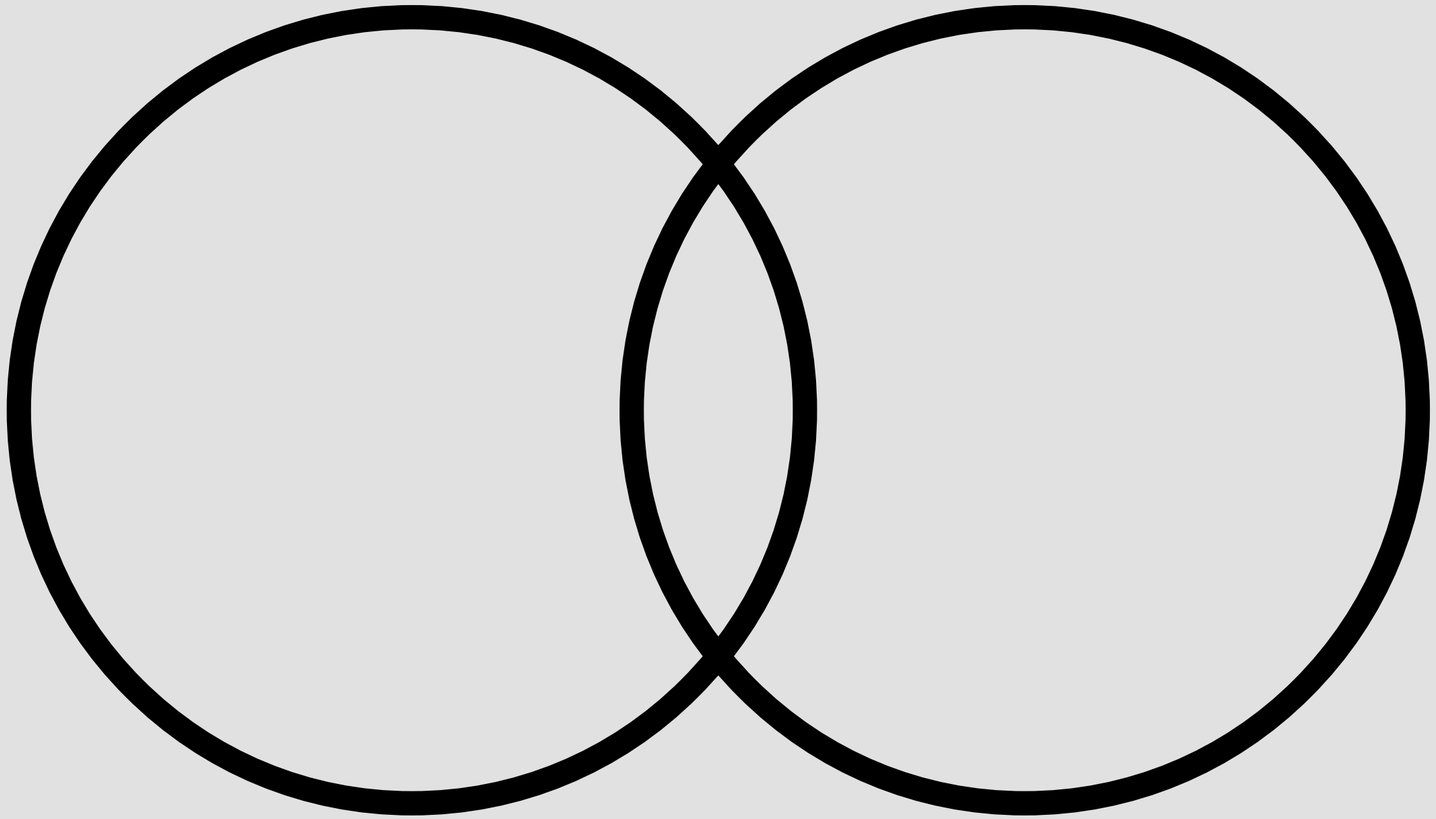
@MaciekGajdzica

WHAT IS SAFETY CRITICAL SYSTEM?

SYSTEM WHOSE MALFUNCTION CAN LEAD TO:

- death or serious injury of people
- enviromental harm
- loss of expensive equipment

SAFETY VS SECURITY



SAFETY VS RELIABILITY

Safe

better to shut down

than to cause accident

Reliable

always works

Solw'IT Let's Solve it

Risk: cutting fingers

Solw'IT Let's Solve it

Risk: cutting fingers

Solution: working only while button is being pressed

Risk: cutting fingers

Solution: working only while button is being pressed

Solw'IT Let's Solve it

Risk: burning everything

Solw'IT Let's Solve it

Risk: burning everything

Solution: unable to light a barbacue

Risk: burning everything

Solution: unable to light a barbacue

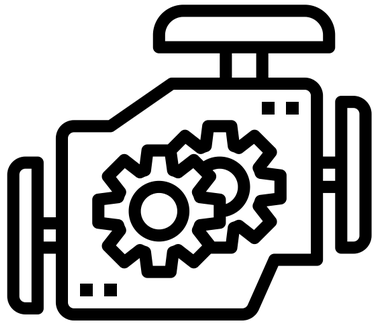

Solw'IT
Let's Solve it


Opening
Mon
Tue-Thur
Fri -Sat
Sunday

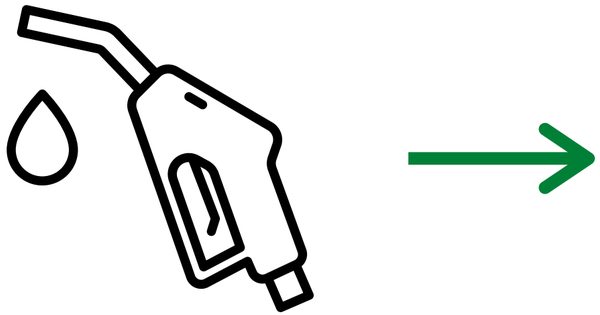
SAFETY INTEGRITY LEVEL (SIL)

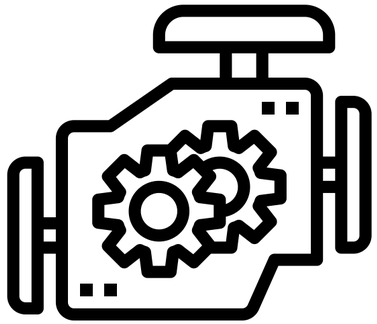
Railway	Automotive	Aerospace	Medical	Malfunction may lead to:
SIL 4	ASIL D	DAL A	-	Death of many people
SIL 3	ASIL C	DAL B	Class C	Death of a single person
SIL 2	ASIL B	DAL C	Class B	Severe injury possible
SIL 1	ASIL A	DAL D	Class A	Minor injury possible
SIL 0	-	DAL E	-	No negative effects

FAIL SAFE

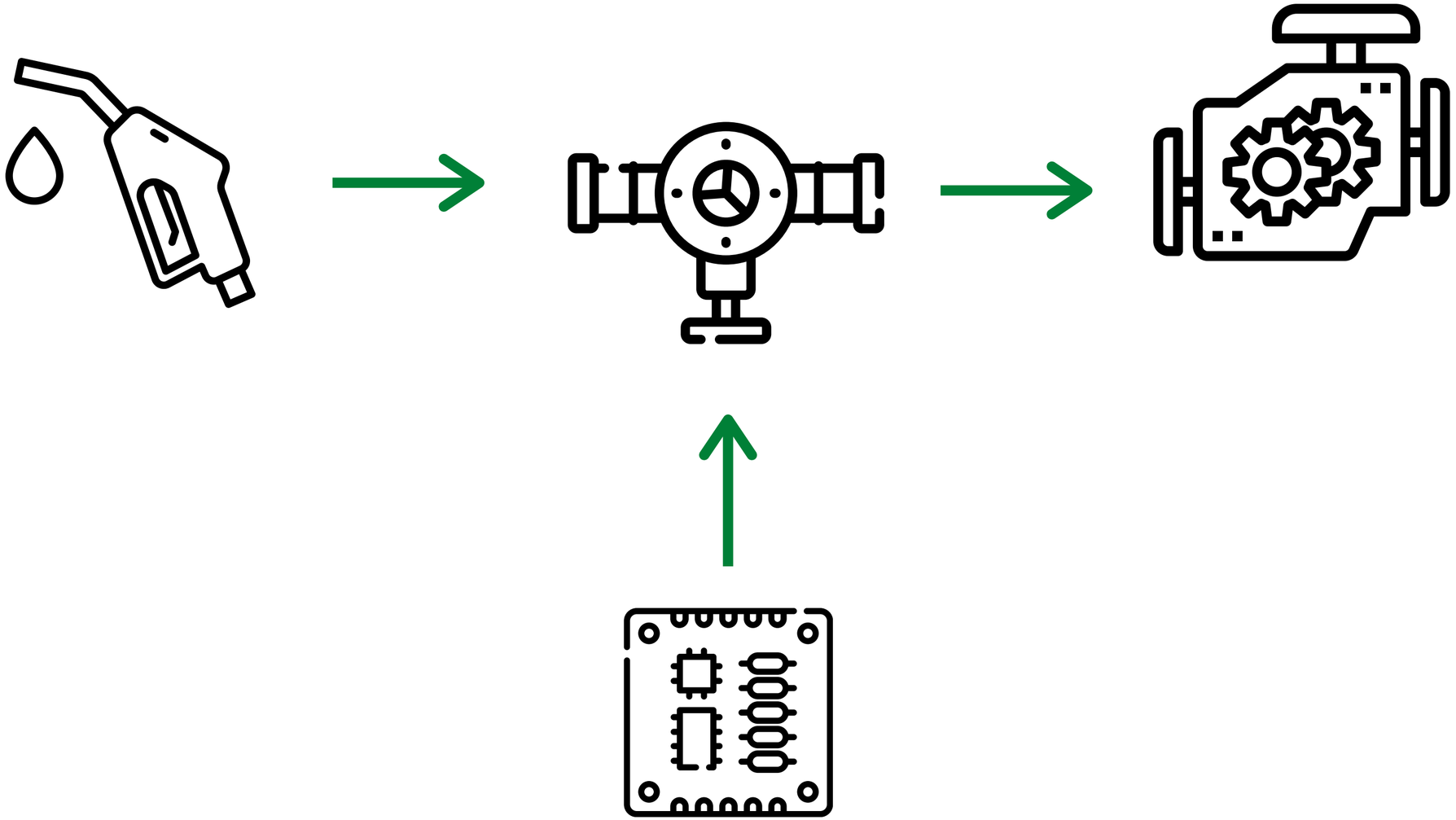


FAIL SAFE

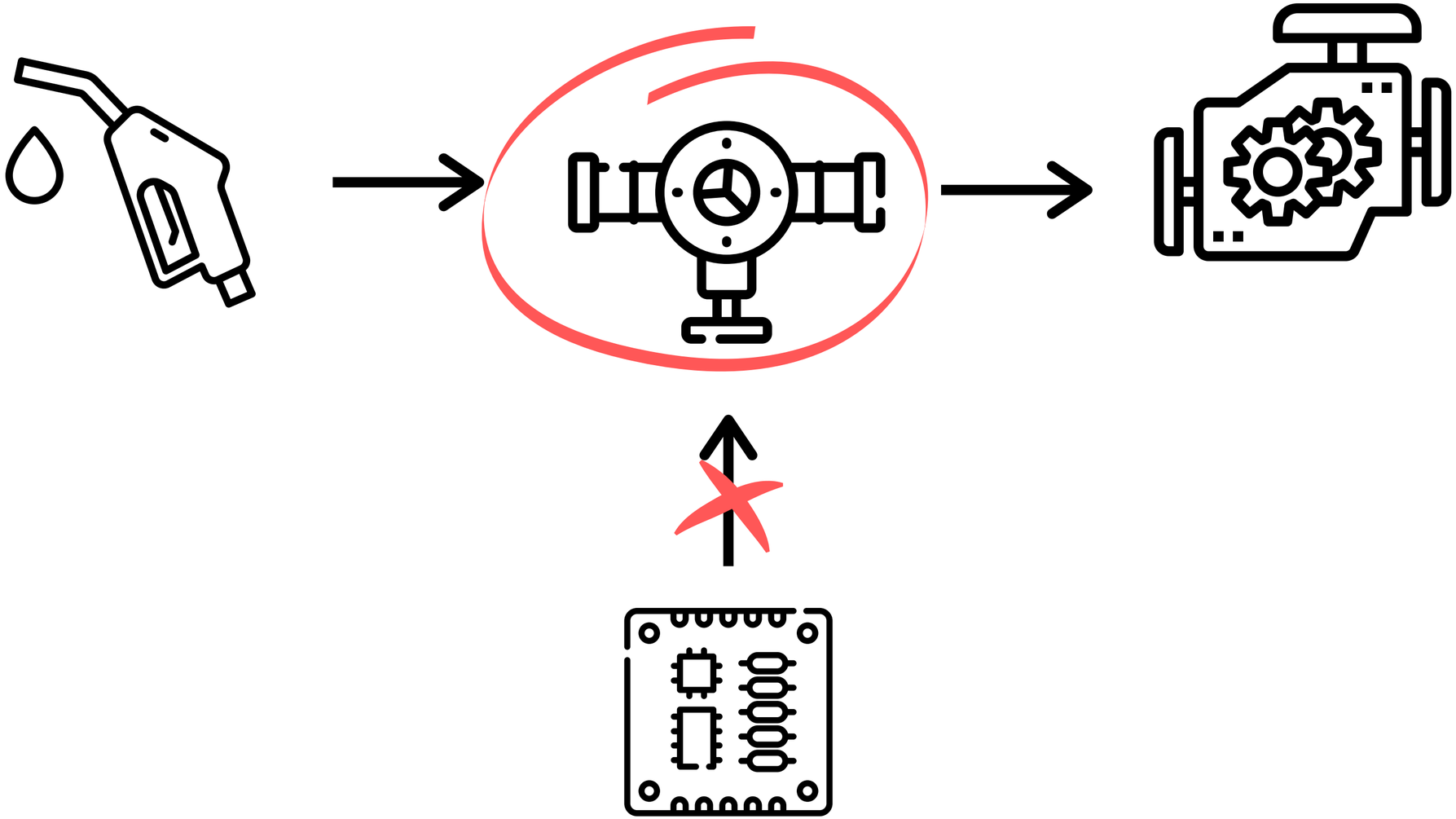




FAIL SAFE



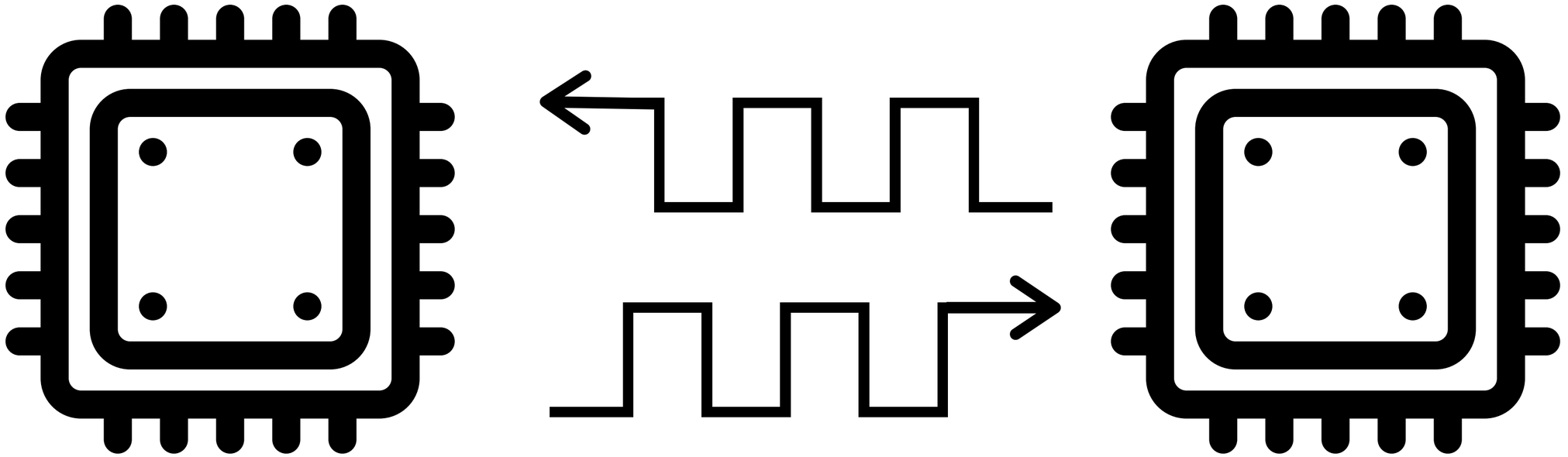
FAIL SAFE



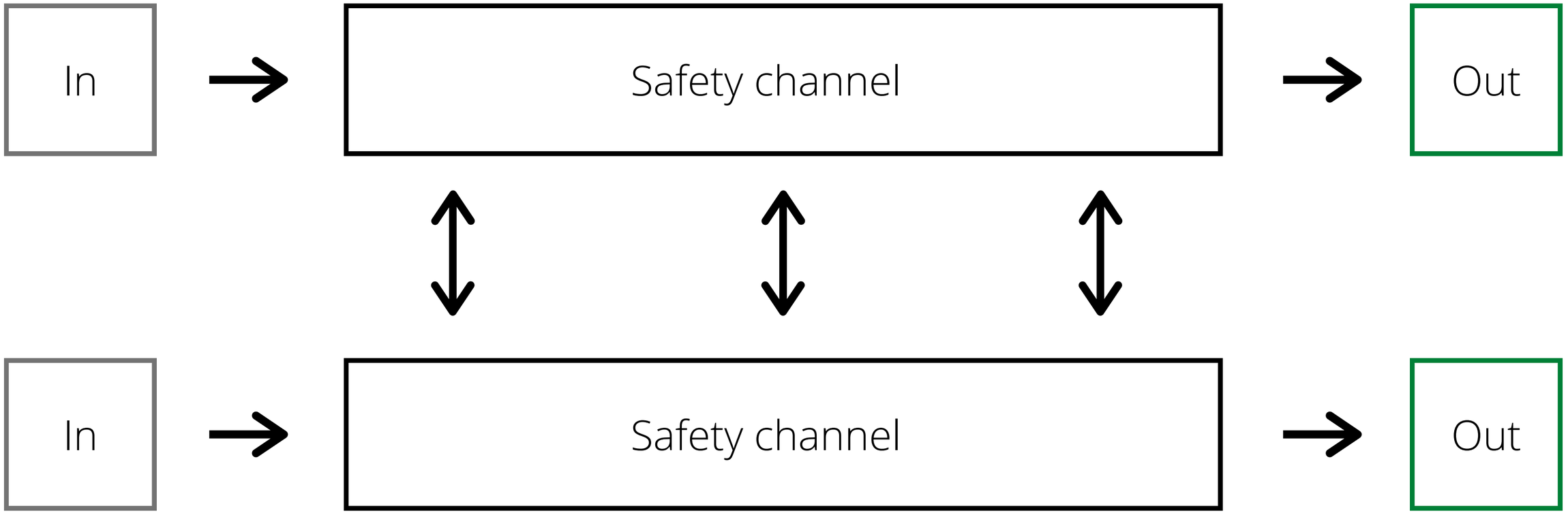
CPU ERRORS

- RAM
- FLASH
- CPU - instructions or registers
- Clock

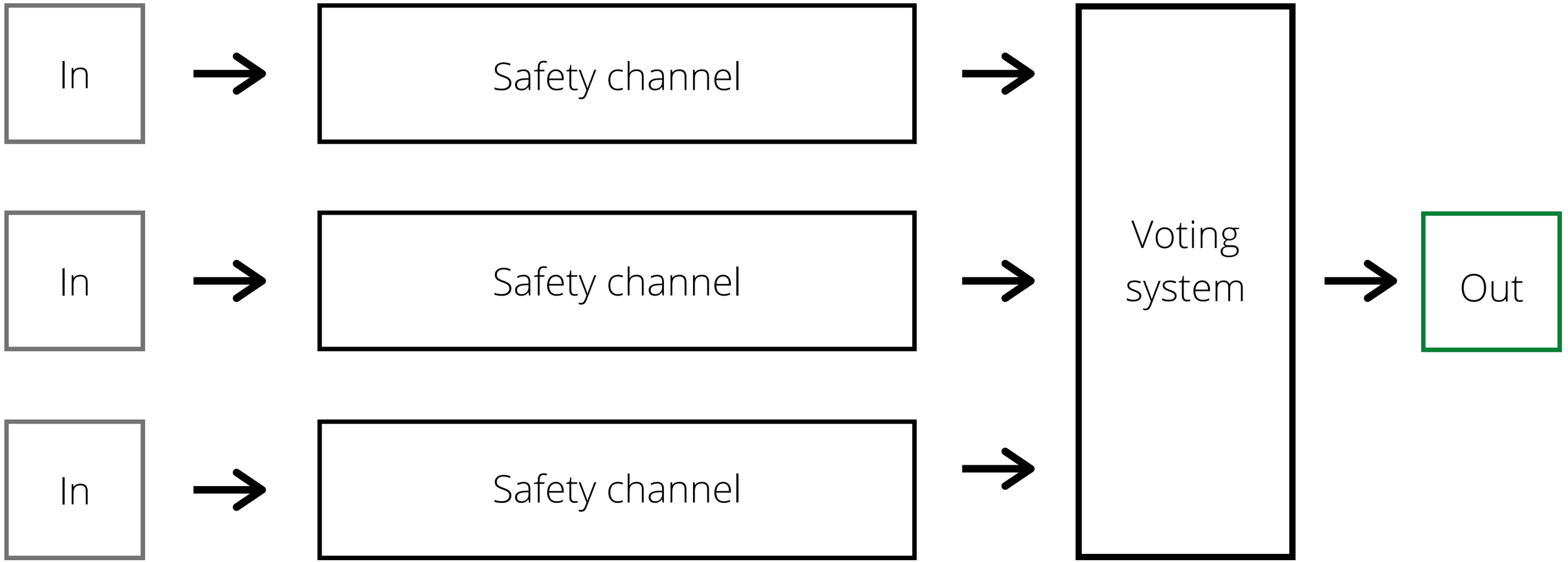
HOW TO DETECT CLOCK FAILURE?



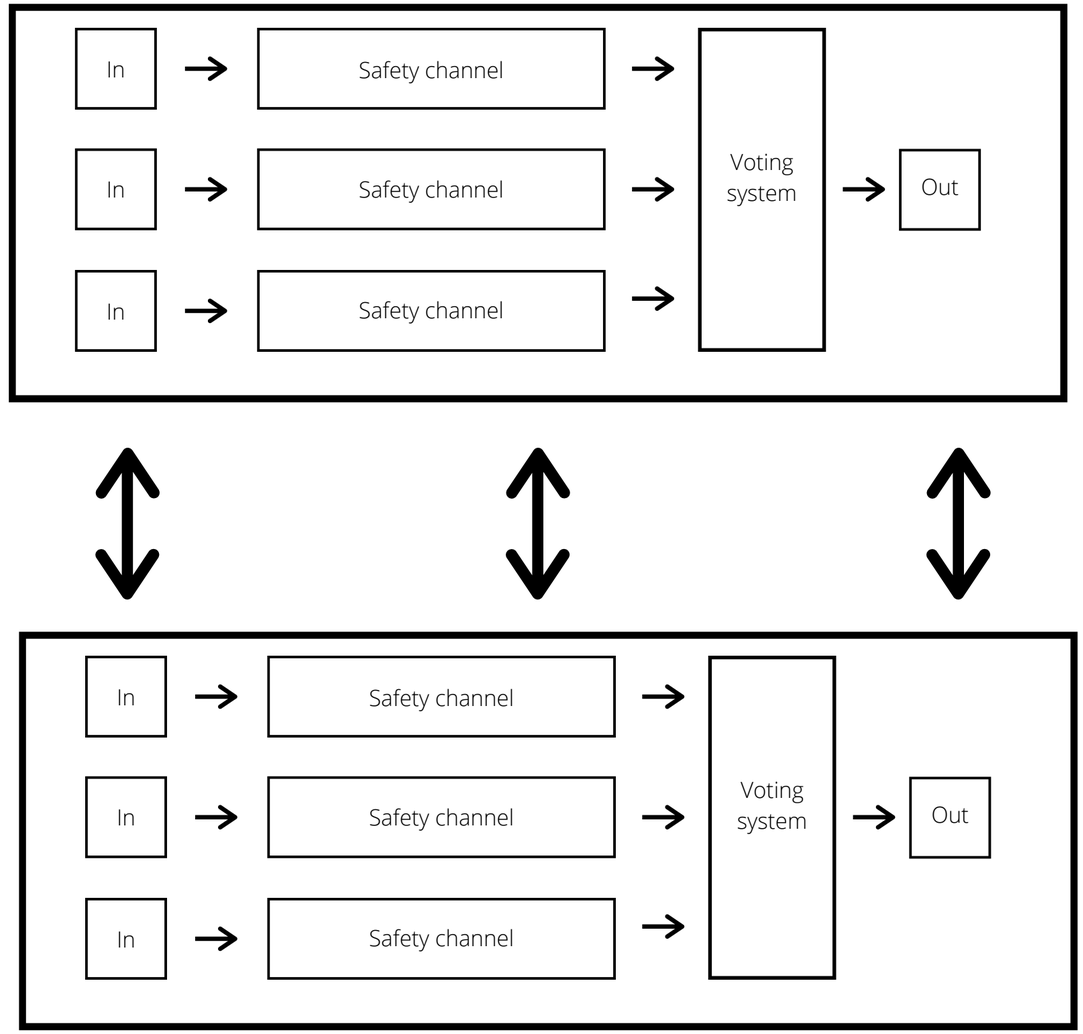
REDUNDANCY



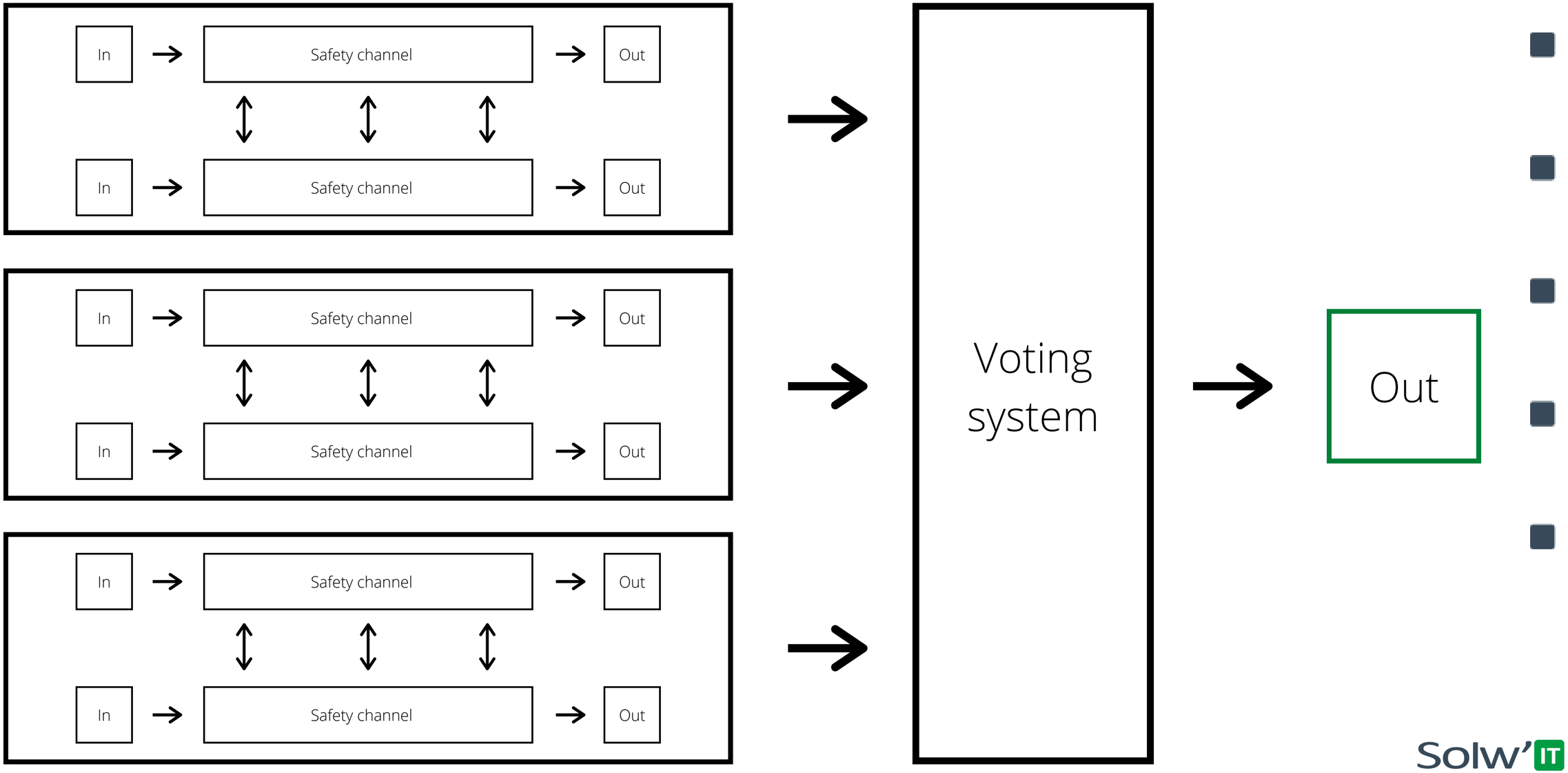
REDUNDANCY



REDUNDANCY


In
Safety channel
In
Safety channel
In
Safety channel
Voting system
Out
In
Safety channel
In
Safety channel
In
Safety channel
Voting system
Out

REDUNDANCY



SAFE COMMUNICATION

	Defences							
	Sequence number	Timestamp	Timeout	Node IDs	Acknowledge	Handshake	Safety code	Encryption
Threats								
Repetition	x	x						
Deletion	x							
Insertion	x			x	x	x		
Resequence	x	x						
Corruption							x	x
Delay		x	x					
Masquerade					x	x		x

SAFE COMMUNICATION

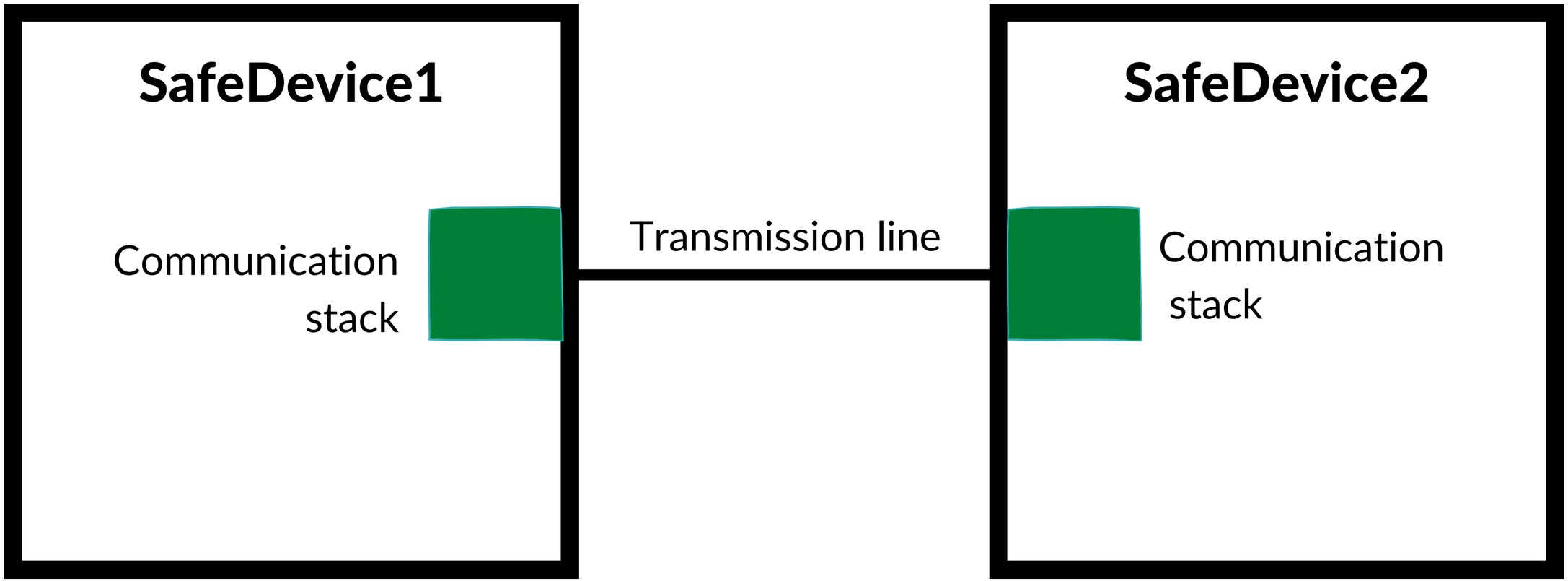
	Threats						
Network category	Repetition	Deletion	Insertion	Resequence	Corruption	Delay	Masquerade
1	+	+	+	+	++	+	-
2	++	++	++	+	++	++	-
3	++	++	++	++	++	++	++

Legend:

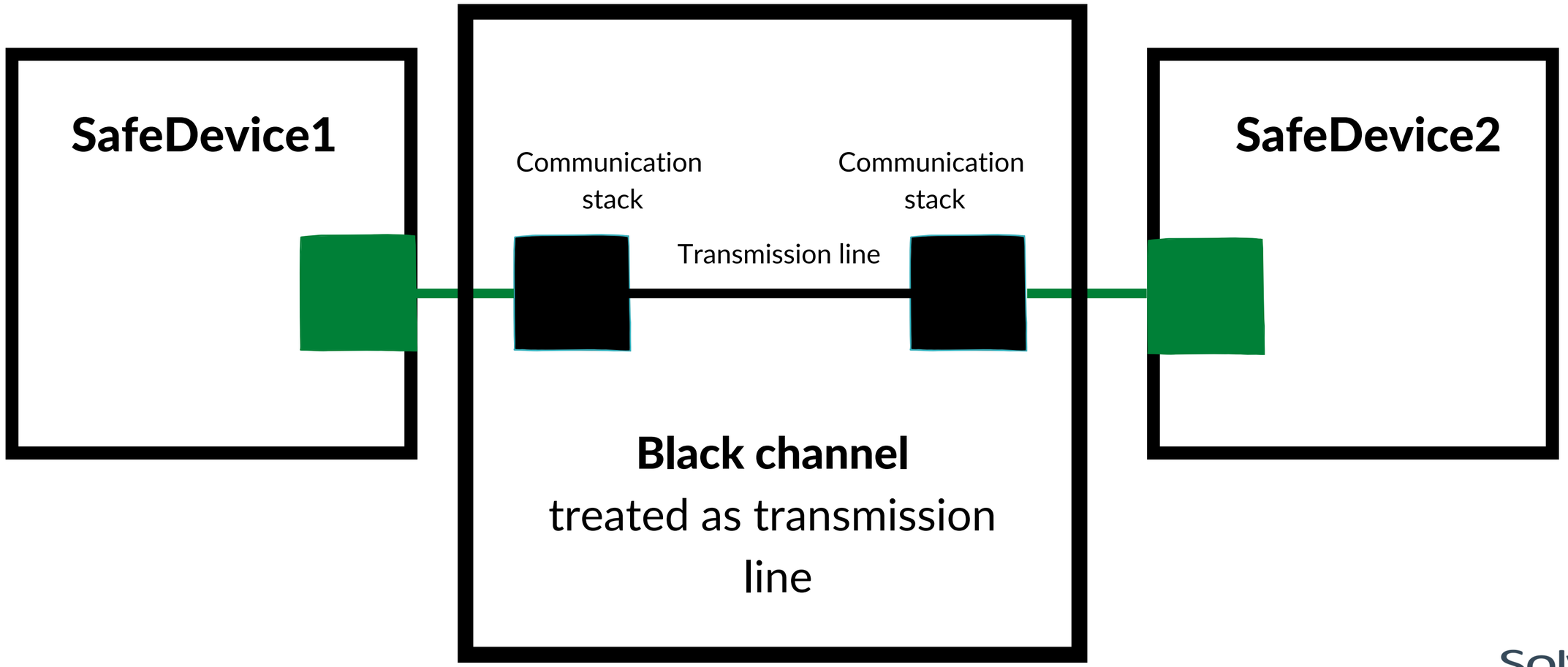
- \- Threat can be neglected
- \+ Rare, weak countermeasures sufficient
- ++ Threat exists, strong countermeasures required

Solw'IT Let's Solve it

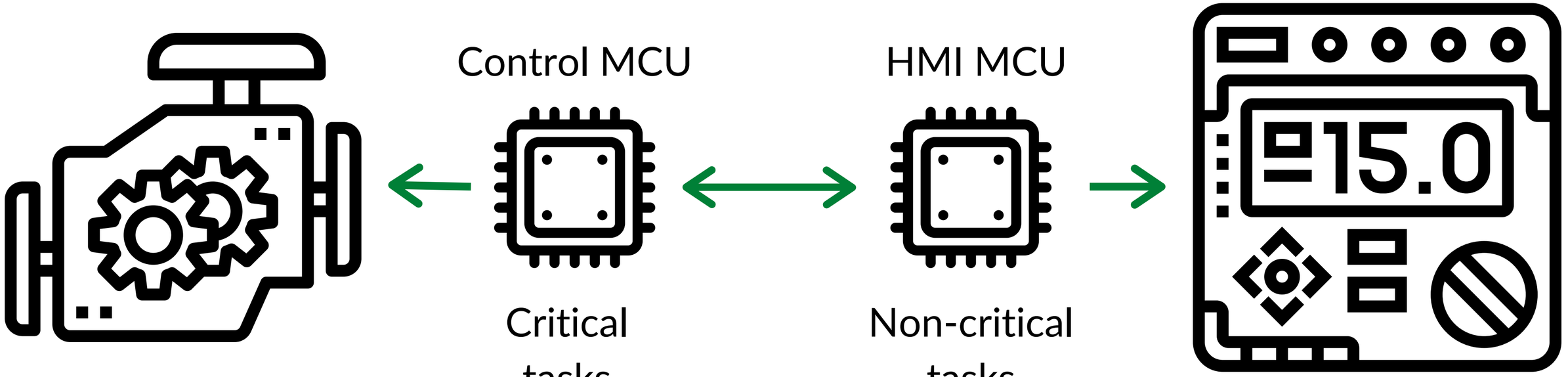
PROBLEM


SafeDevice1
Communication stack
Transmission line
SafeDevice2
Communication stack

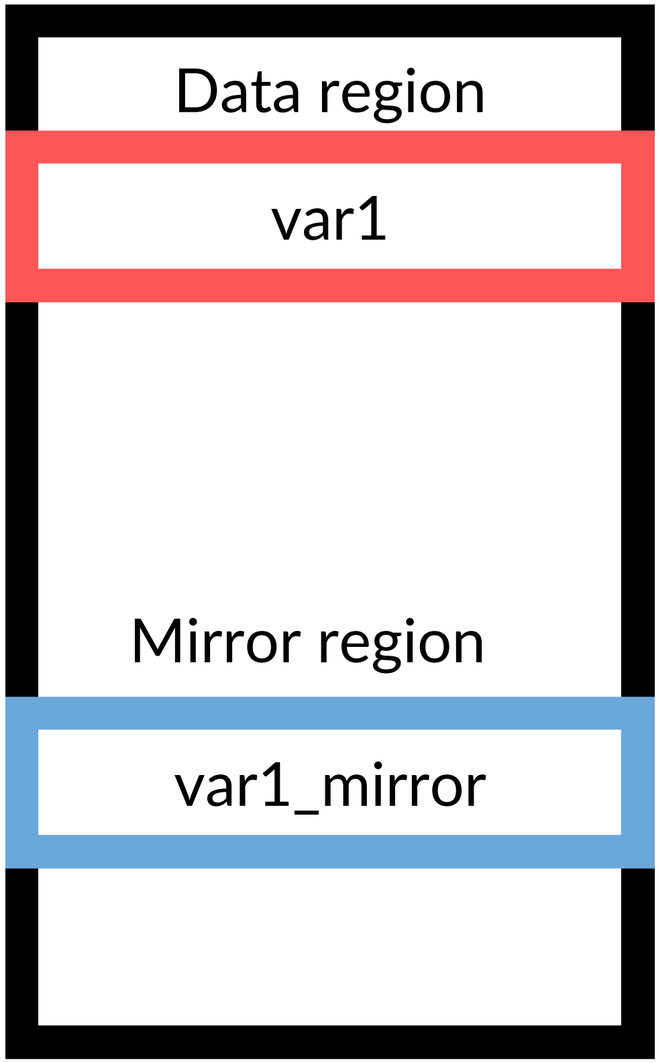
SOLUTION


SafeDevice1
Communication stack
Communication stack
Transmission line
SafeDevice2
Black channel
treated as transmission line

MIXED CRITICALITY



DATA CORRUPTION



Invariant:

var1 ^ var1_mirror = 0xFFFFFFFF

```
uint32_t tick_cnt CRITICAL_DATA;
uint32_t tick_cnt_inv CRITICAL_DATA_MIRROR;

 /* Verify TickCounter integrity */
if ((tick_cnt ^ tick_cnt_inv) == 0xFFFFFFFFuL)
{
    tick_cnt++;
    tick_cnt_inv = ~tick_cnt;

  if (tick_cnt >= SYSTICK_10ms)
  {
      tick_cnt = 0u;
      tick_cnt_inv = 0xFFFFFFFFuL;
  }
}
```

```
struct safe_var
{
    uint32_t * const value;
    uint32_t * const value_inv;
};

void safe_var_init(const struct safe_var *var);
uint32_t safe_var_get(const struct safe_var *var);
void safe_var_set(const struct safe_var *var,
                  uint32_t val);

uint32_t tick_cnt CRITICAL_DATA;
uint32_t tick_cnt_inv CRITICAL_DATA_MIRROR;

const struct safe_var safe_tick_cnt =
    {&tick_cnt, &tick_cnt_inv};

uint32_t tick_val = safe_var_get(&safe_tick_cnt);
safe_var_set(&safe_tick_cnt, ++tick_val);
```

LANGUAGES

Solw'IT Let's Solve It


Solw'IT
Let's
Solve
it

ADA

```
type My_Int is range -1 .. 20;
```

ADA

```
type My_wrapping_int is mod 2 ** 5;
```

ADA

```
type Item is range 0 .. 1000;
type Index is range 0 .. 4;
type My_Array is array (Index) of Item;
```

ADA

```
type Item is range 0 .. 1000;
type Index is range 1 .. 5;
type My_Array is array (Index) of Item;
```

ADA

```
type Item is range 0 .. 1000;
type Index is range 11 .. 15;
type My_Array is array (Index) of Item;
```

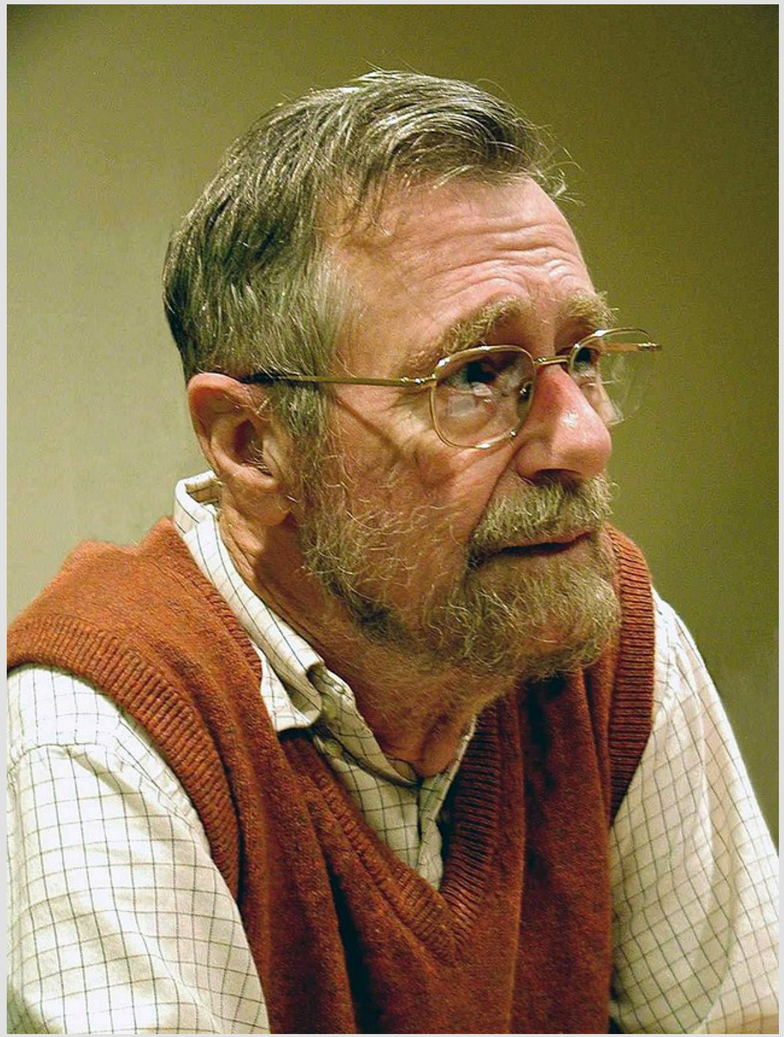
ADA

```
procedure Illegal_Example is
   --  Declare two different floating point types
   type Meters is new Float;
   type Miles is new Float;

   Dist_Imperial : Miles;

   --  Declare a constant
   Dist_Metric : constant Meters := 100.0;
begin
   --  Not correct: types mismatch
   Dist_Imperial := (Dist_Metric * 1609.0) / 1000.0;
   Put_Line (Miles'Image (Dist_Imperial));
end Illegal_Example;
```

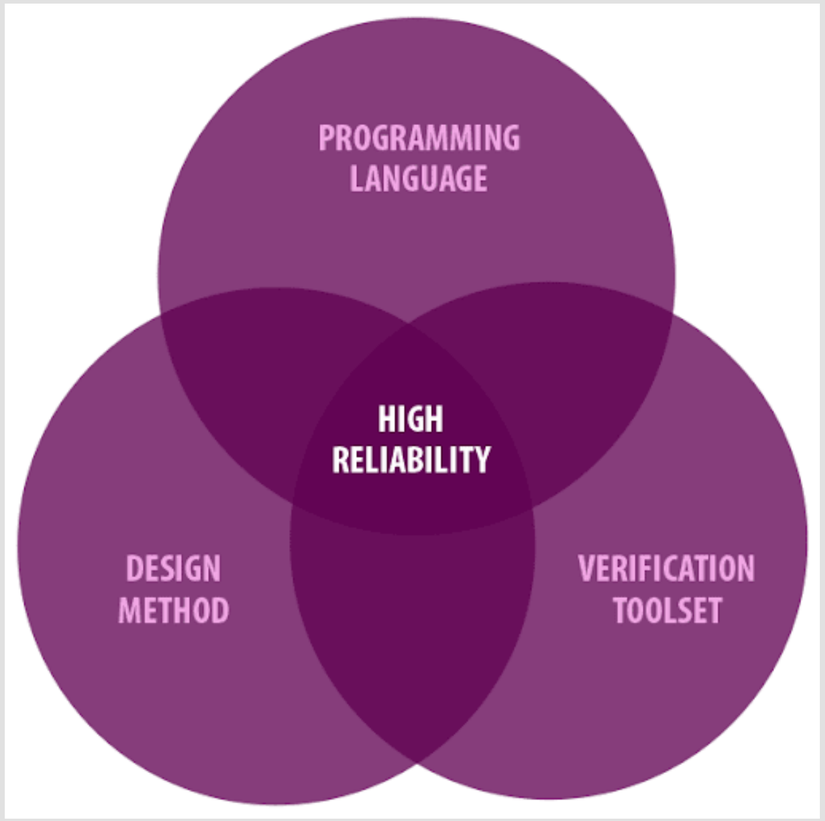
FORMAL PROOF



“Program testing can be used to show the presence of bugs, but never to show their absence!”

EDSGER DIJKSTRA

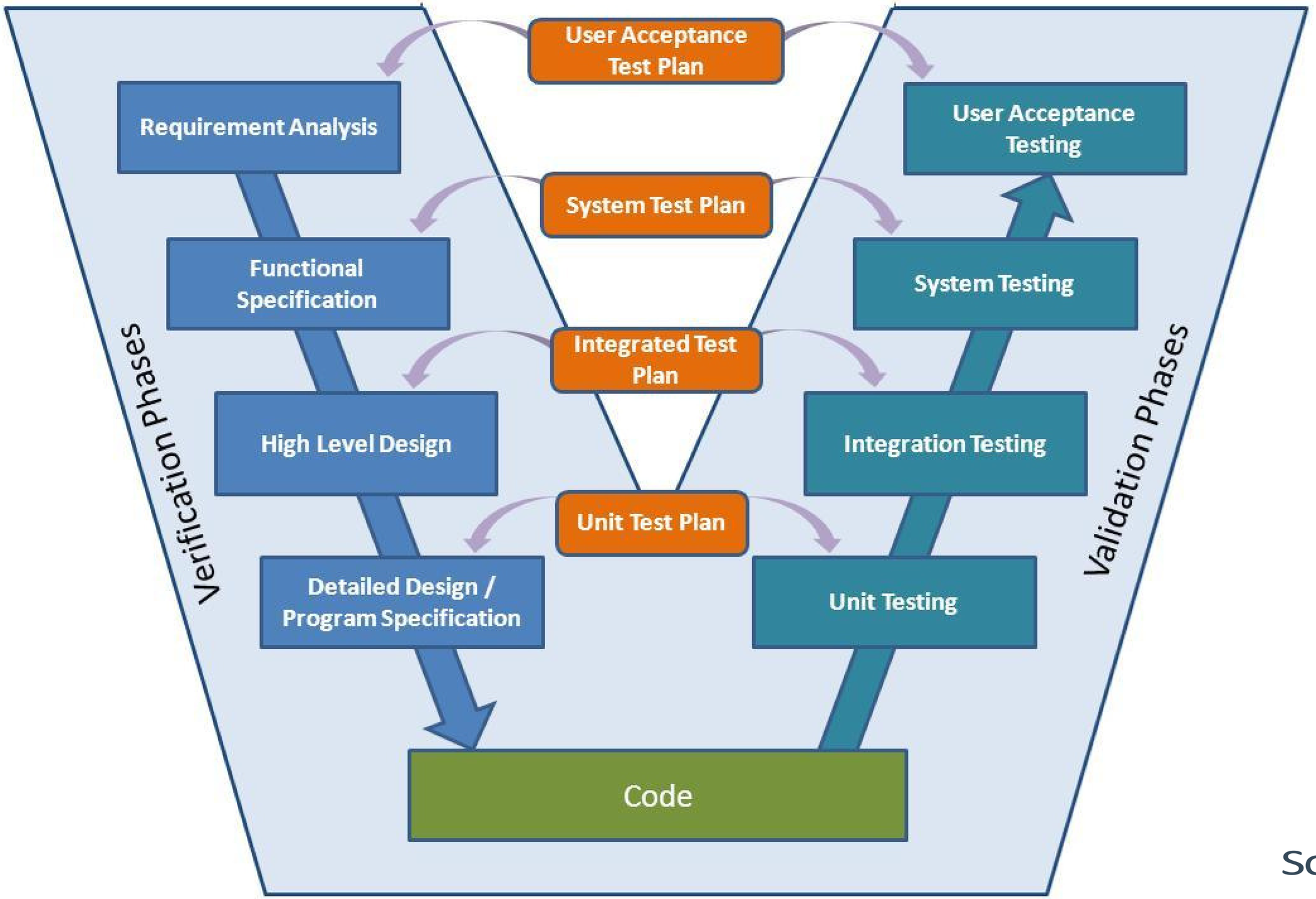
ADA SPARK

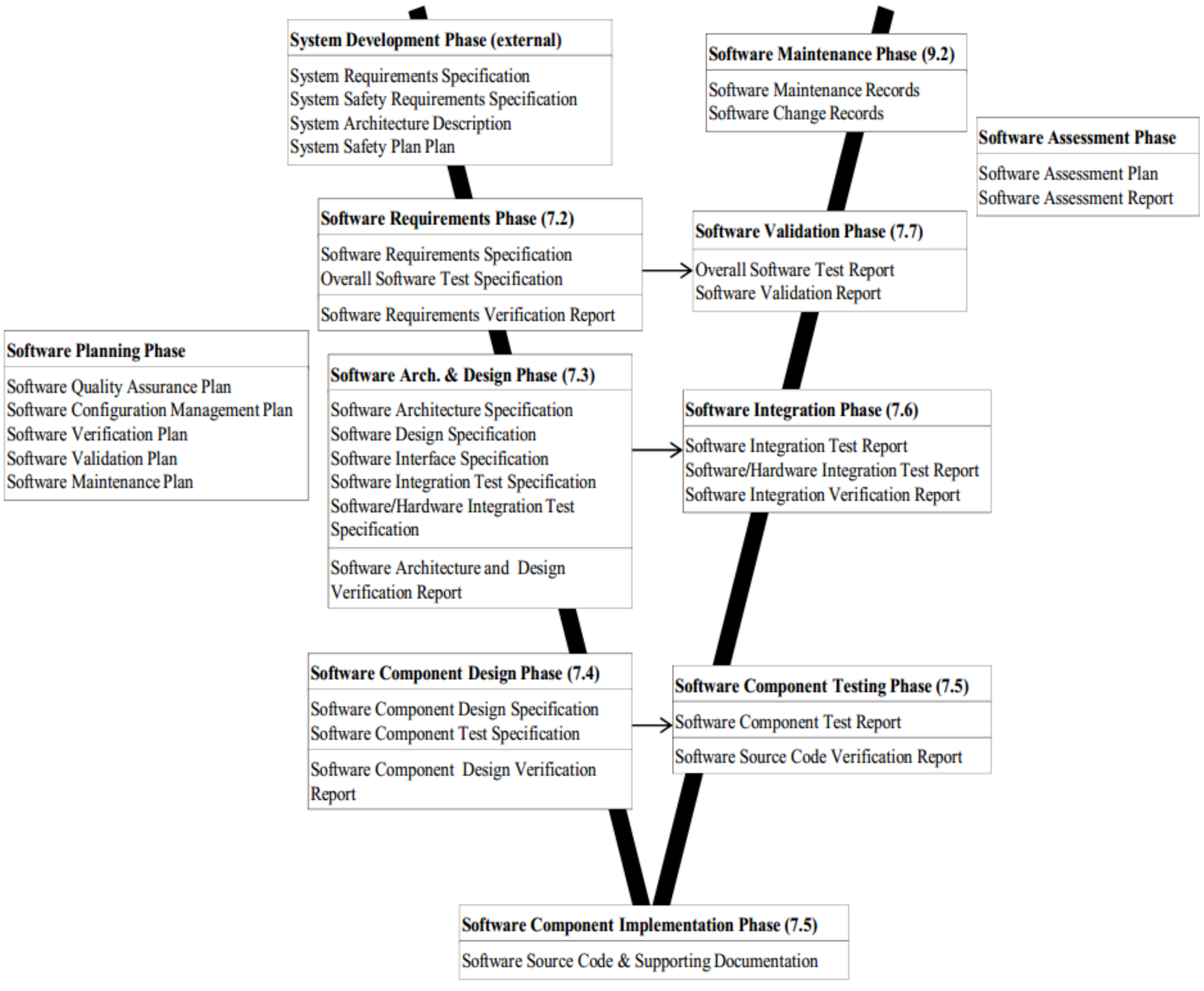


SPARK2014

LANGUAGE SUBSETS

- MISRA C
- AUTOSAR C++





EFFECTIVE DOCUMENTATION

- part of code review
- cannot merge when not updated
- documentation first

VERSION MANAGEMENT

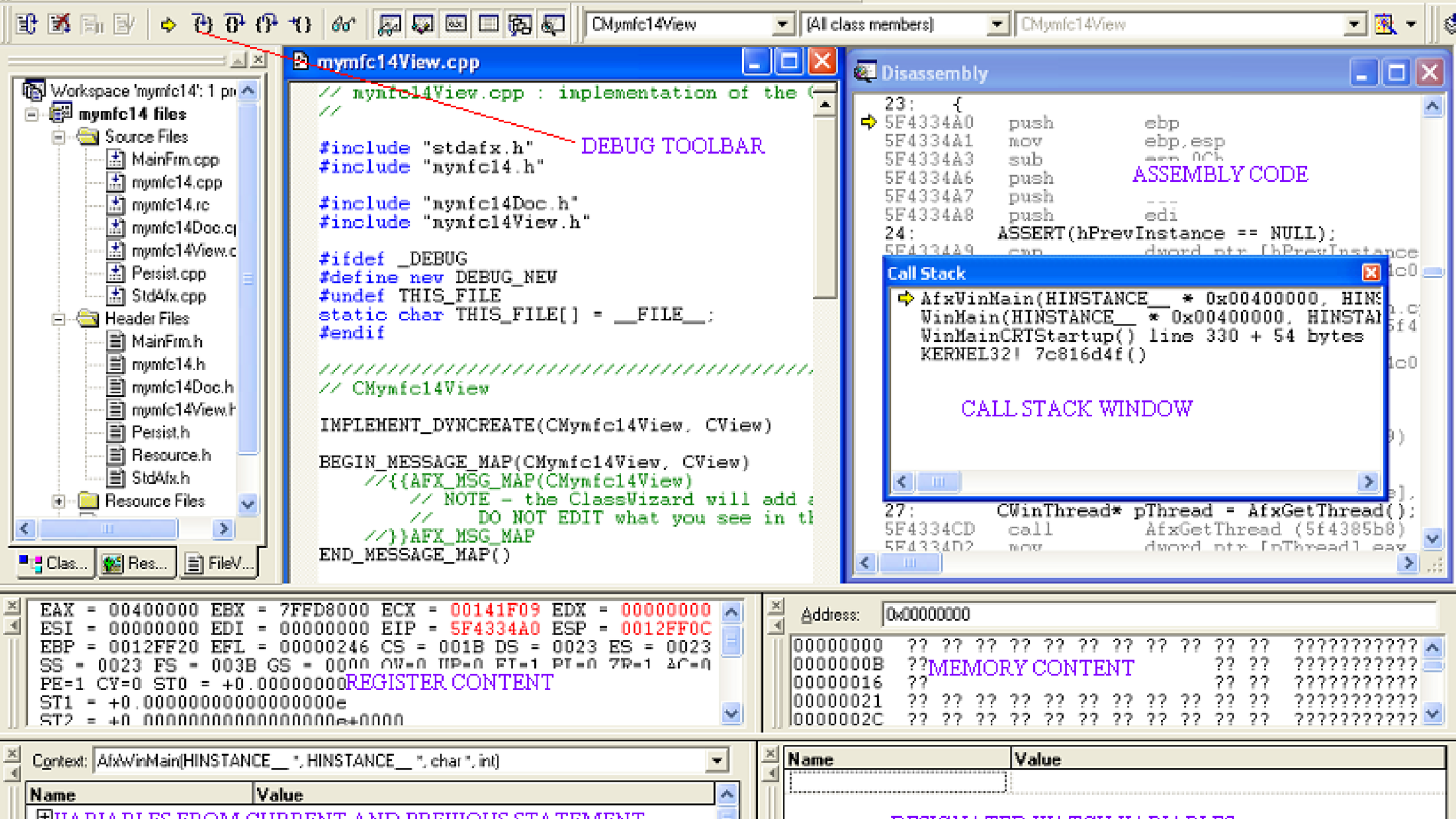
- version for every binary
- version for every PCB
- version for every bundle

BUT ALSO...

- version of compiler
- version of OS
- version of HW debuger
- version of build system
- version of config generator
- version of every tool used

YOU MUST BE ABLE TO REBUILD ORIGINAL BINARY FROM SOURCE FOR THE WHOLE PRODUCT LIFETIME

...EVEN IF PRODUCT LIFETIME IS 20 YEARS


DEBUG TOOLBAR
ASSEMBLY CODE
CALL STACK WINDOW
REGISTER CONTENT
MEMORY CONTENT

PEOPLE AND PROCESSES

“Insisting that operators always follow procedures does not guarantee safety although it does usually guarantee that there is someone to blame-either for following the procedures or for not following them-when things go wrong.”

NANCY LEVESON

ROOT CAUSE ANALYSIS


Why?

ROOT CAUSE ANALYSIS

Root cause → ? → [ ] → Accident

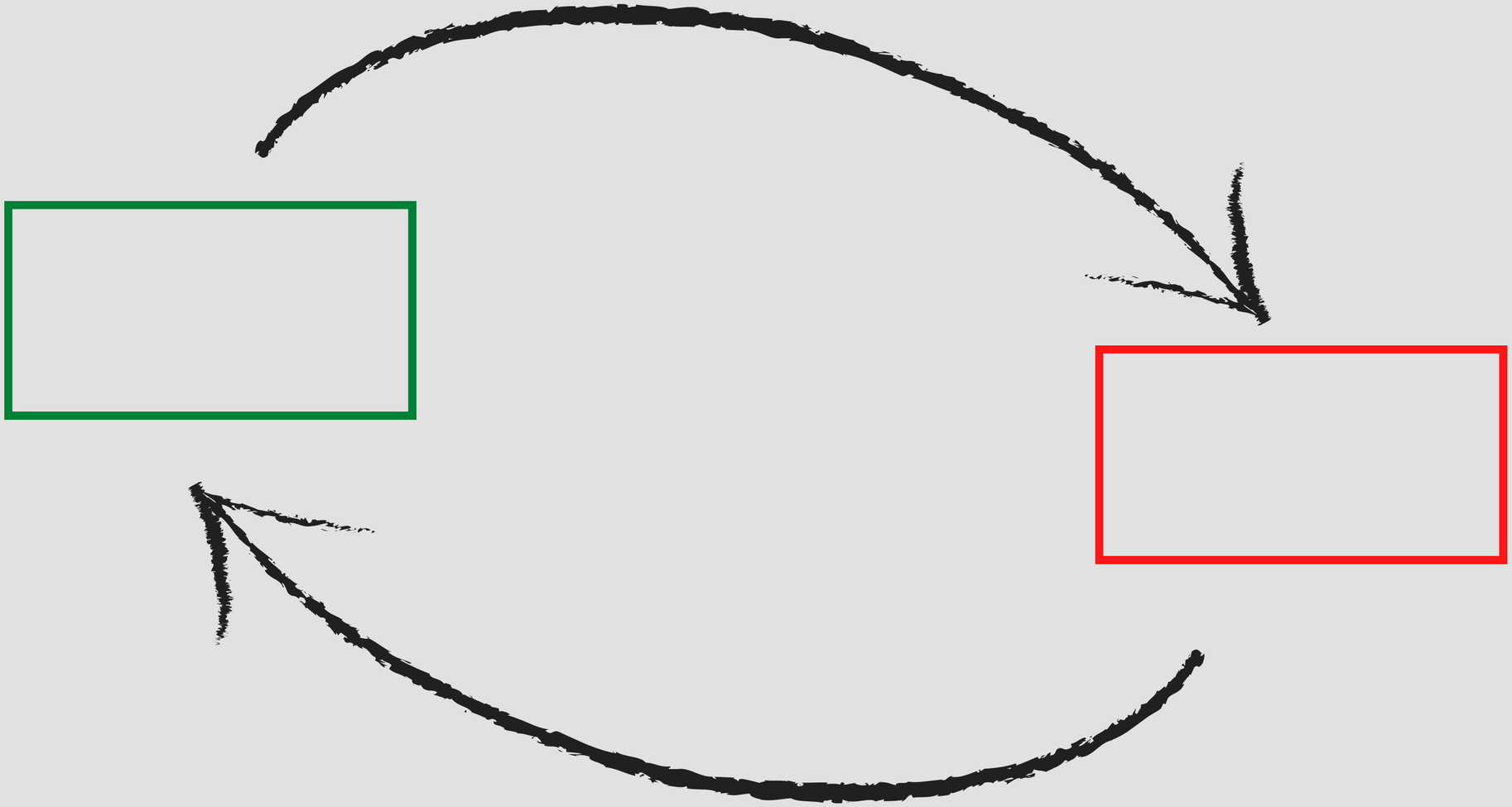
ROOT CAUSE ANALYSIS

Root cause → → → Accident

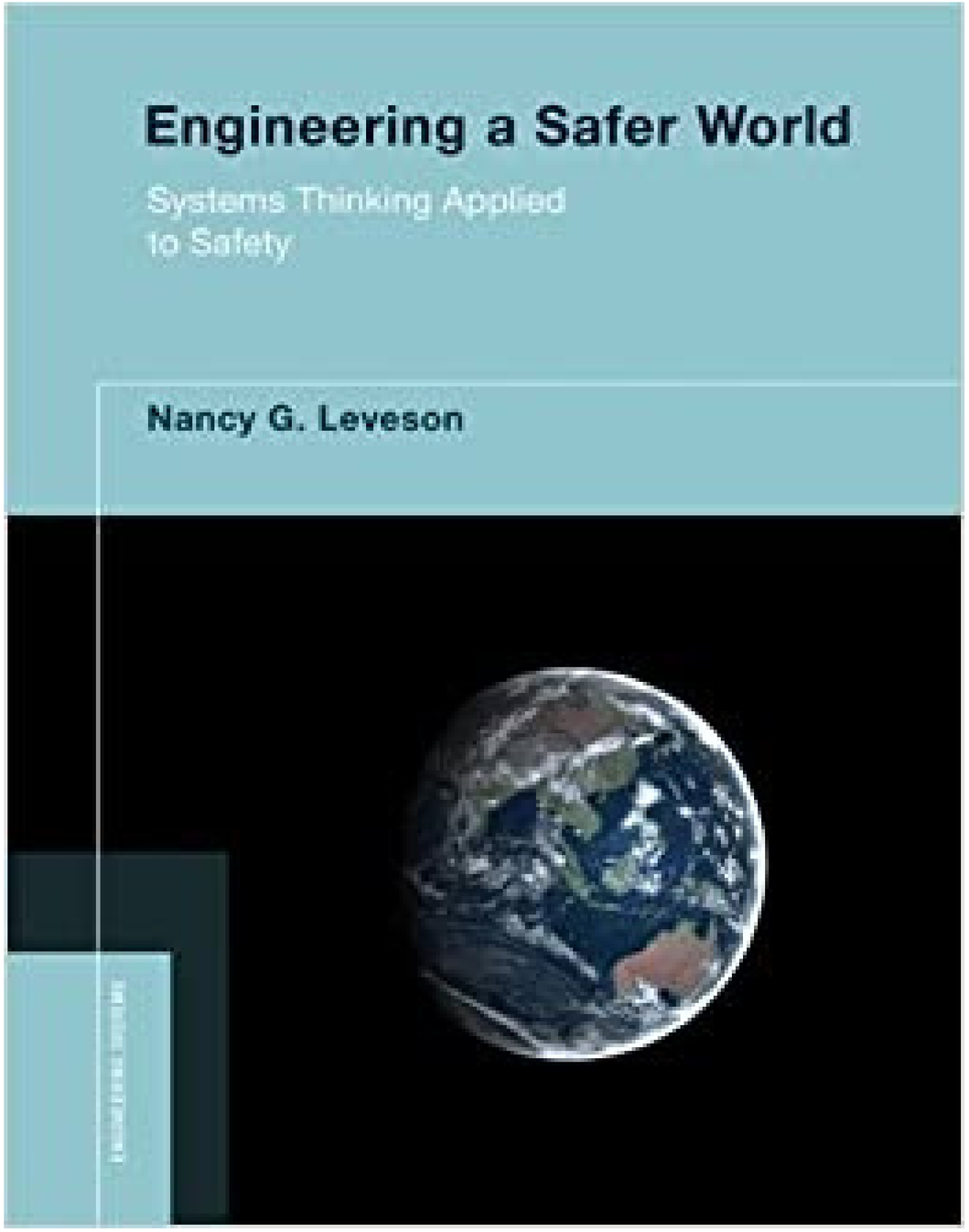
ROOT CAUSE ANALYSIS

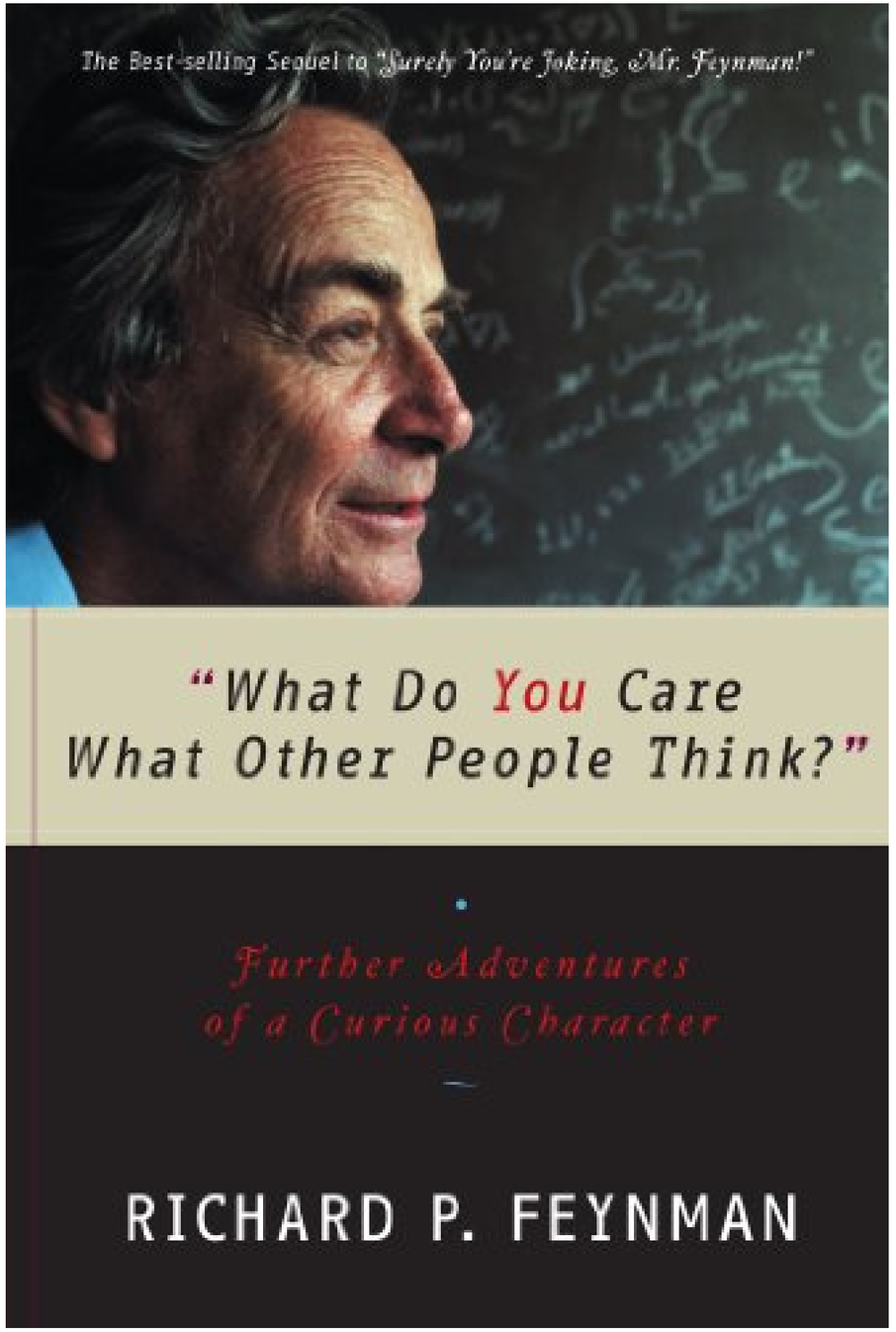
? → Root cause → → → Accident

FEEDBACK LOOP

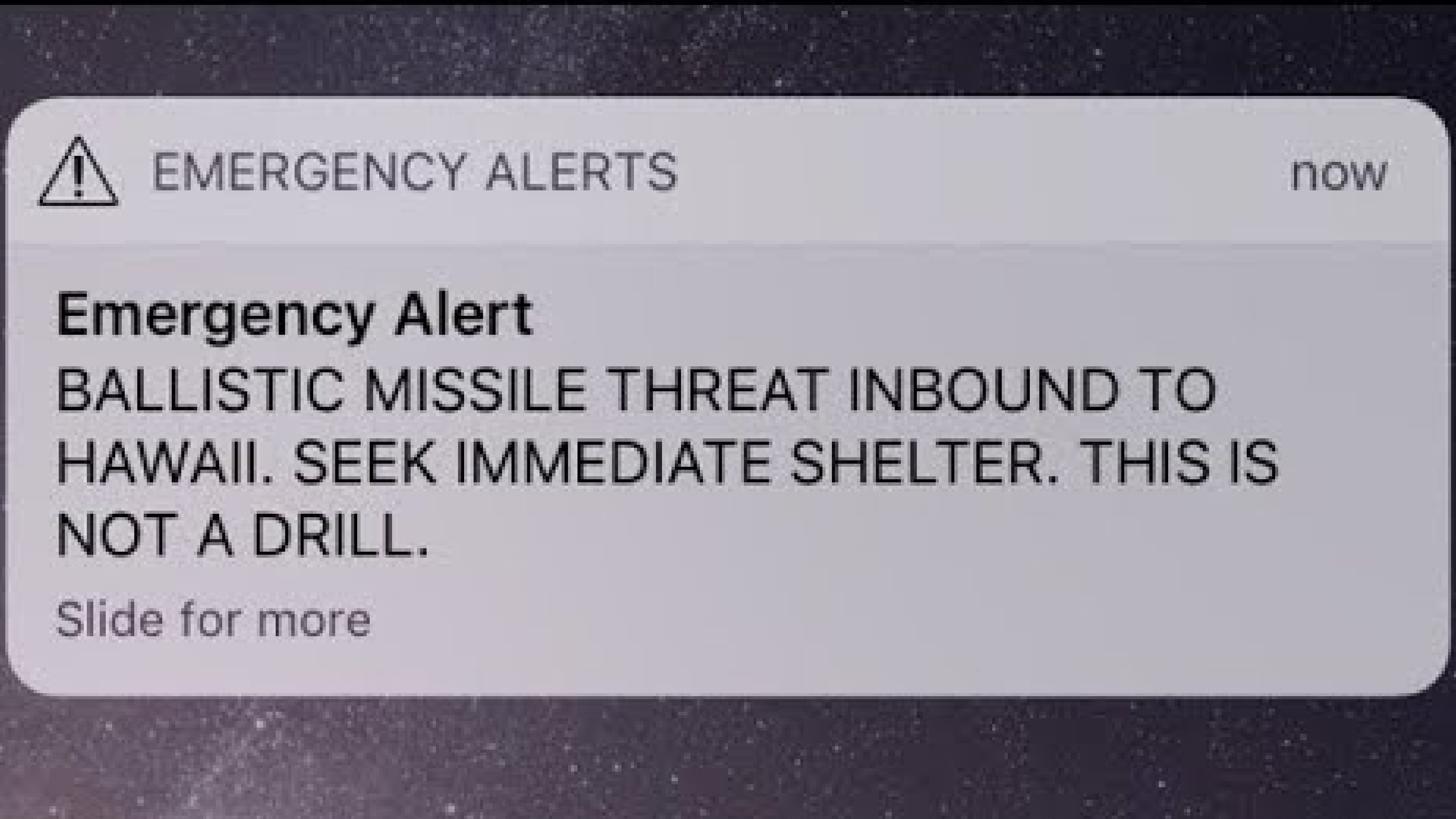


Solw'IT Let's Solve It

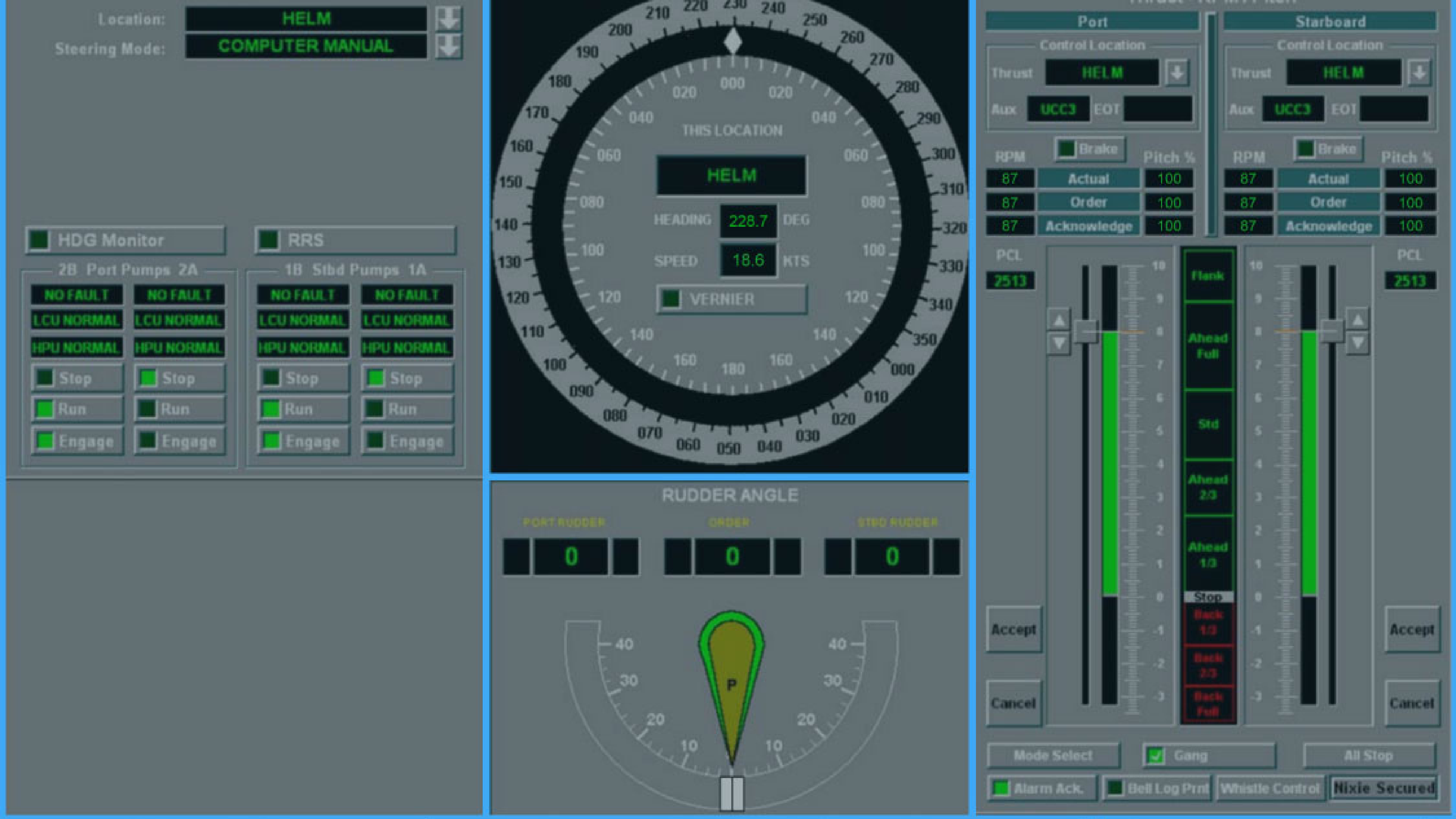

Engineering a Safer World
Systems Thinking Applied to Safety
Nancy G. Leveson


The Best-selling Sequel to "Surely You're Joking, Mr. Feynman!"
"What Do You Care What Other People Think?"
Further Adventures of a Curious Character
RICHARD P. FEYNMAN

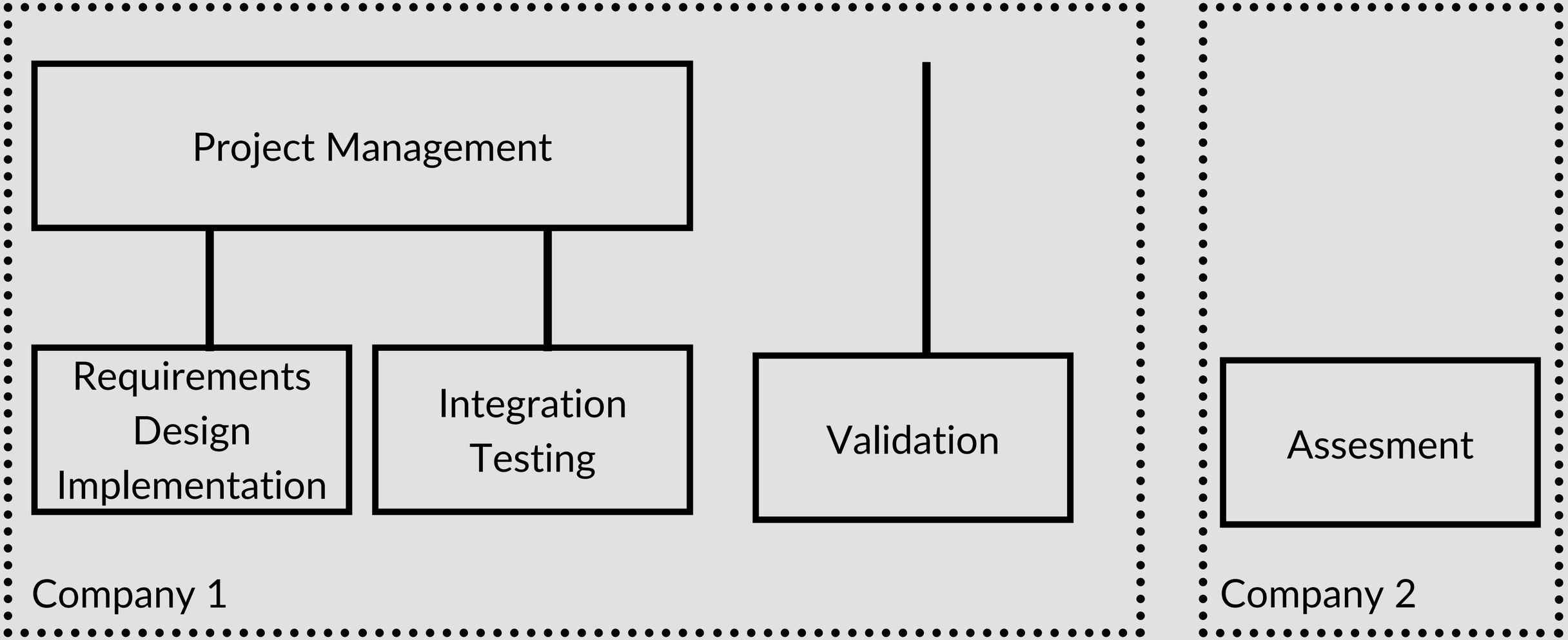

BOEING
737 MAX
39596


EMERGENCY ALERTS
now
Emergency Alert
BALLISTIC MISSILE THREAT INBOUND TO HAWAII. SEEK IMMEDIATE SHELTER. THIS IS NOT A DRILL.
Slide for more


NIGHTLY
NEWS


Location:
HELM
Steering Mode:
COMPUTER MANUAL
HDG Monitor
RRS
2B Port Pumps 2A
1B Stbd Pumps 1A
NO FAULT
LCU NORMAL
HPU NORMAL
Stop
Run
Engage
THIS LOCATION
HELM
HEADING
228.7
DEG
SPEED
18.6
KTS
VERNIER
RUDDER ANGLE
PORT RUDDER
ORDER
STBD RUDDER
0
P
Port
Starboard
Control Location
Thrust
HELM
Aux
UCC3
EOT
RPM
Brake
Pitch %
87
Actual
Order
Acknowledge
100
PCL
2513
Flank
Ahead Full
Std
Ahead 2/3
Ahead 1/3
Stop
Back 1/3
Back 2/3
Back Full
Accept
Cancel
Mode Select
Gang
All Stop
Alarm Ack.
Bell Log Prnt
Whistle Control
Nixie Secured

PROJECT ROLES - SIL4



THANK YOU

https://ucgosu.pl/slides-devoxx-2021

https://solwit.com
https://ucgosu.pl

Twitter: @MaciekGajdzica

Icons from:
https://www.flaticon.com/

Solw'IT | Let's Solve It

ADDITIONAL RESOURCES

Boeing accident preliminary report:
https://transportation.house.gov/imo/media/doc/TI%20Preliminary%20Investigative%20Findings%20Boeing%20737%20MAX%20March%202020.pdf

Ship crash near Singapore analysis:
https://features.propublica.org/navy-uss-mccain-crash/navy-installed-touch-screen-steering-ten-sailors-paid-with-their-lives/

Hawaii false nuclear alert:
https://en.wikipedia.org/wiki/2018_Hawaii_false_missile_alert