Solw'IT Let's Solve It

SAFETY-CRITICAL SYSTEMS FROM THE INSIDE

MACIEJ GAJDZICA

MACIEJ GAJDZICA

Solw'IT Let's Solve It

- senior embedded developer at Solwit
- automotive, railway, medical systems
- ucgosu.pl - blog, YouTube
- Gdańsk Embedded Meetup

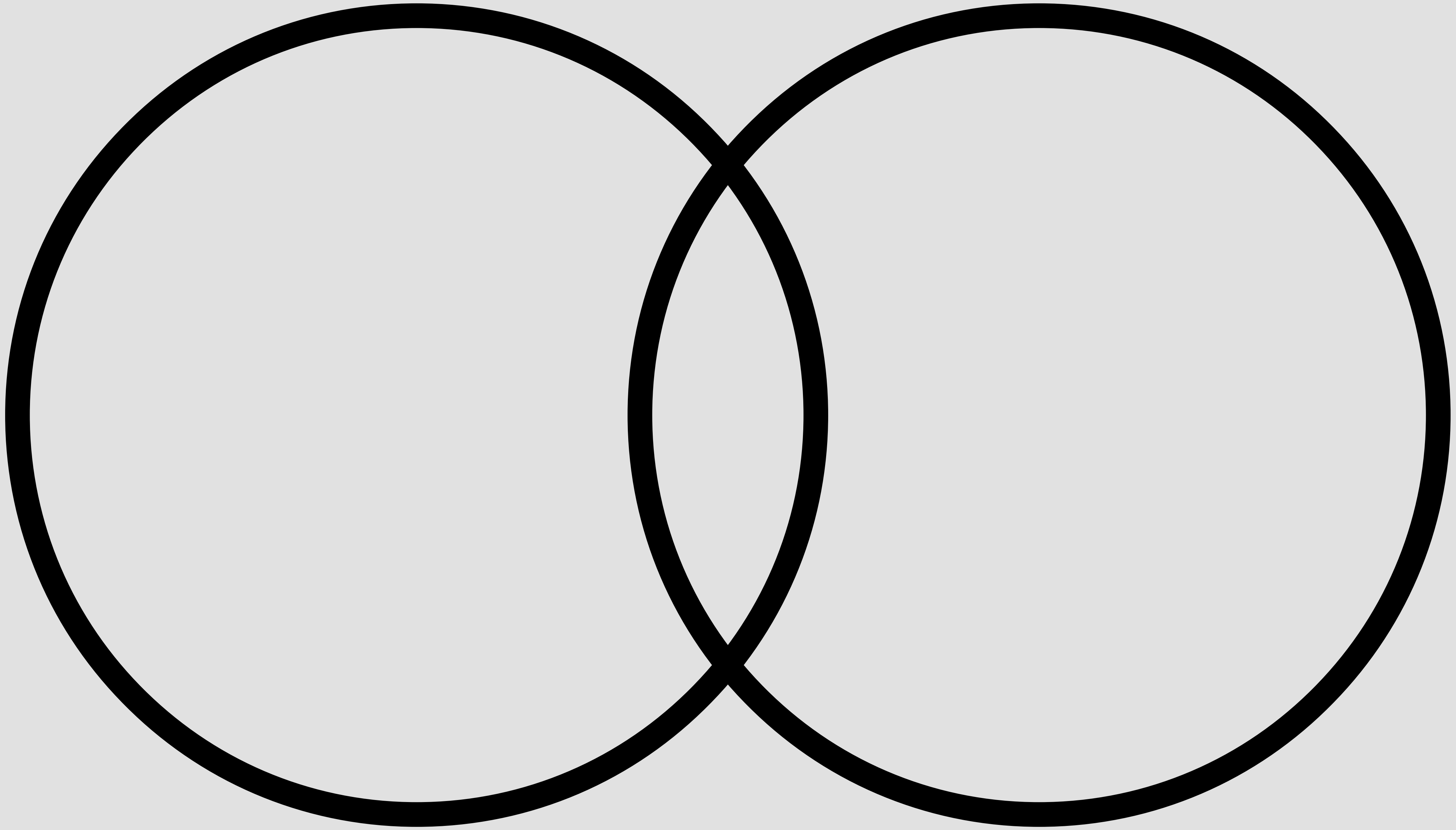
@MaciekGajdzica

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

Risk: cutting fingers

Risk: cutting fingers

Solution: working only while button is being pressed

Risk: cutting fingers

Solution: working only while button is being pressed

Risk: burning everything

Risk: burning everything

Solution: unable to light
a barbacue

Risk: burning everything

Solution: unable to light a barbacue

Opening
Mon
Tue-Thur
Fri-Sat
Sunday

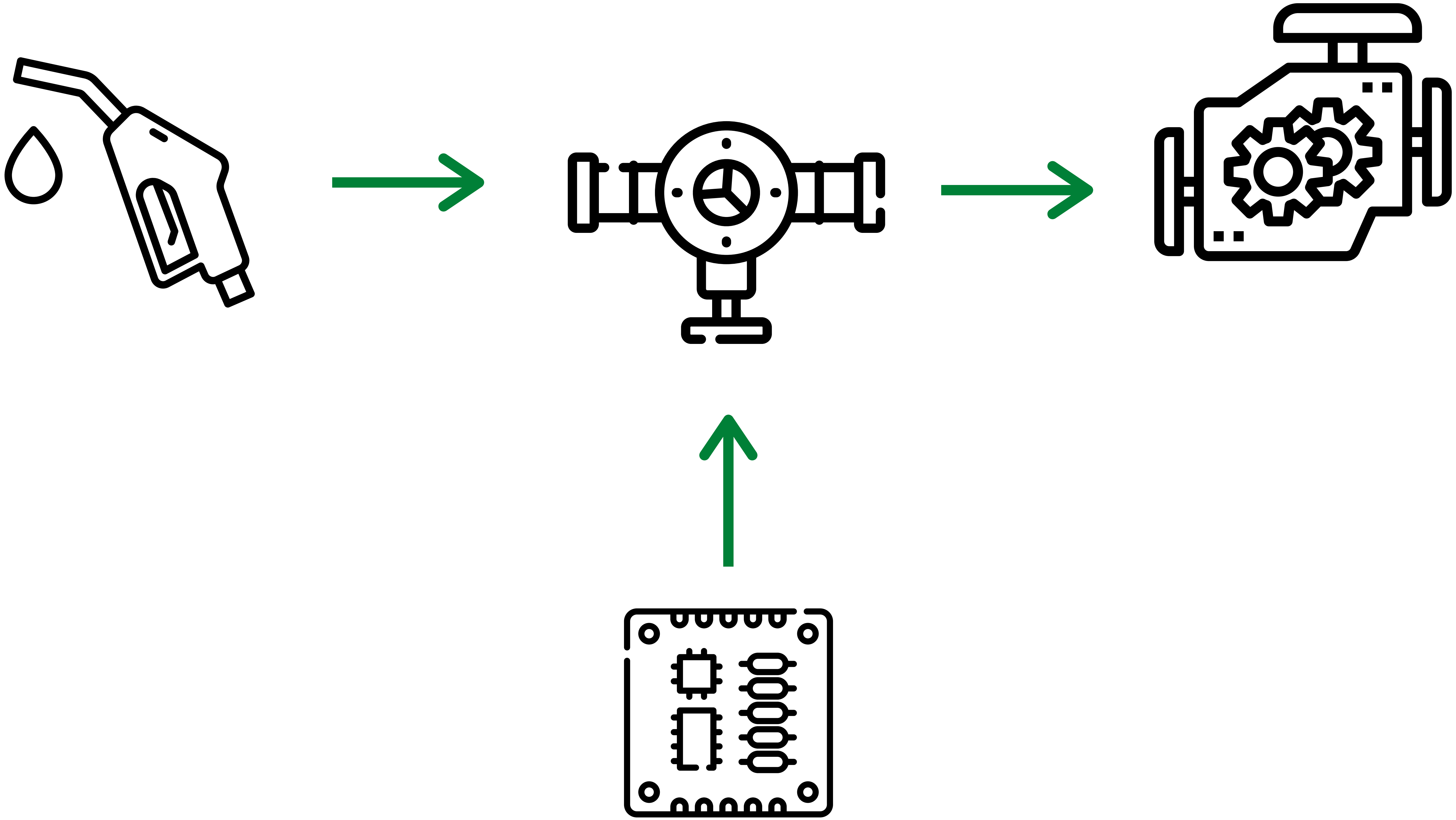
SAFETY INTEGRITY LEVEL (SIL)

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

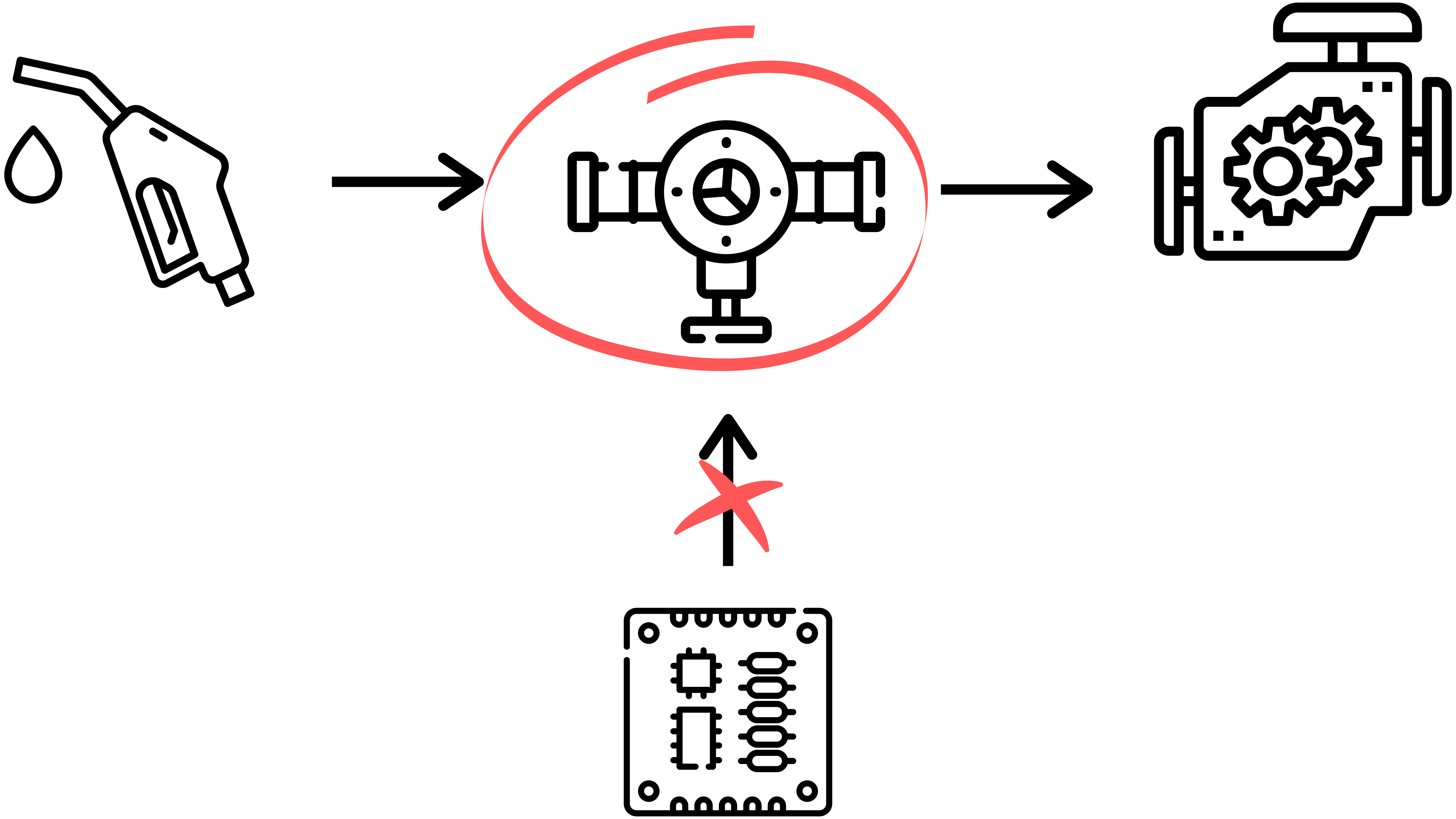
FAIL SAFE

FAIL SAFE

FAIL SAFE



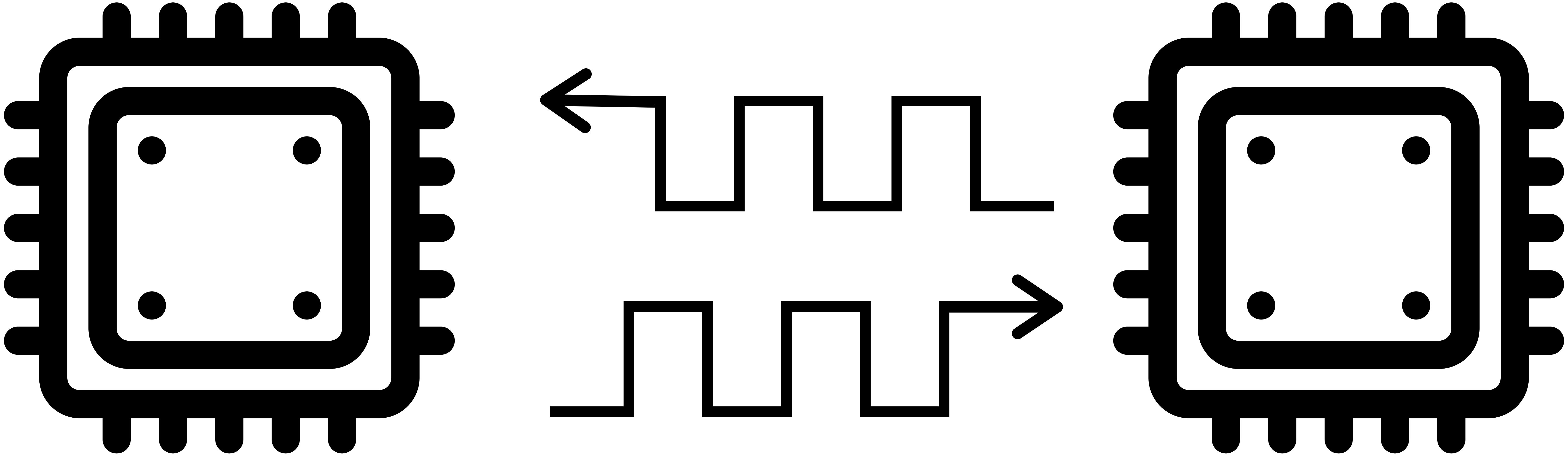
FAIL SAFE



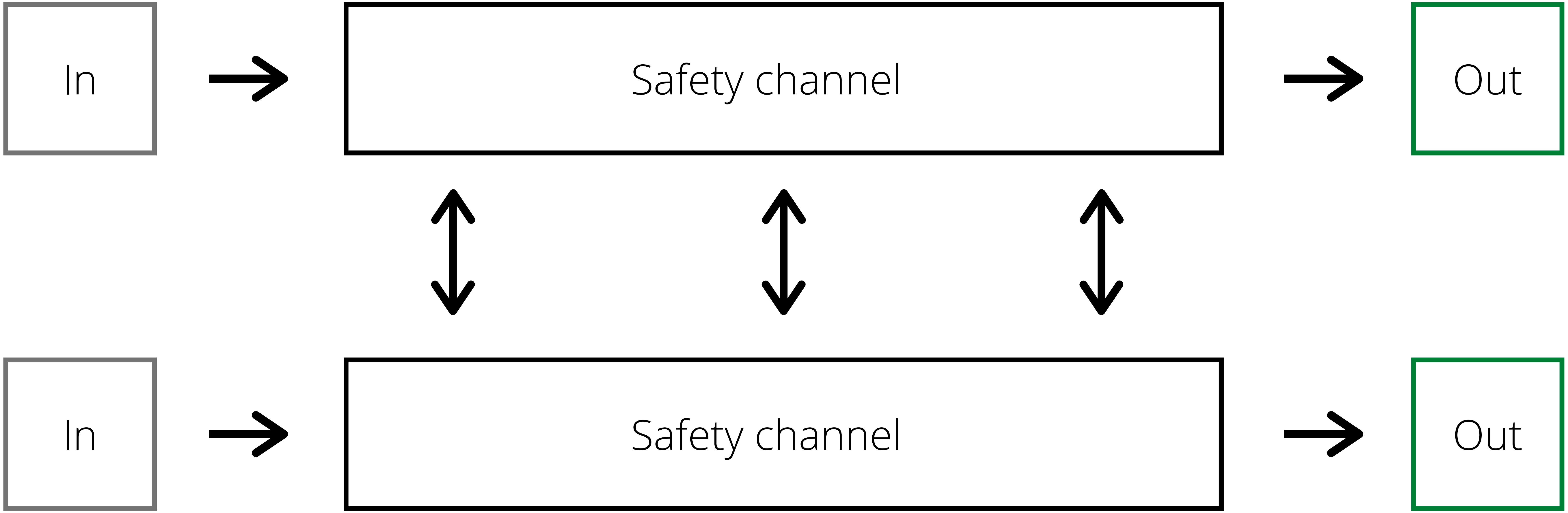
CPU ERRORS

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

HOW TO DETECT CLOCK FAILURE?



REDUNDANCY

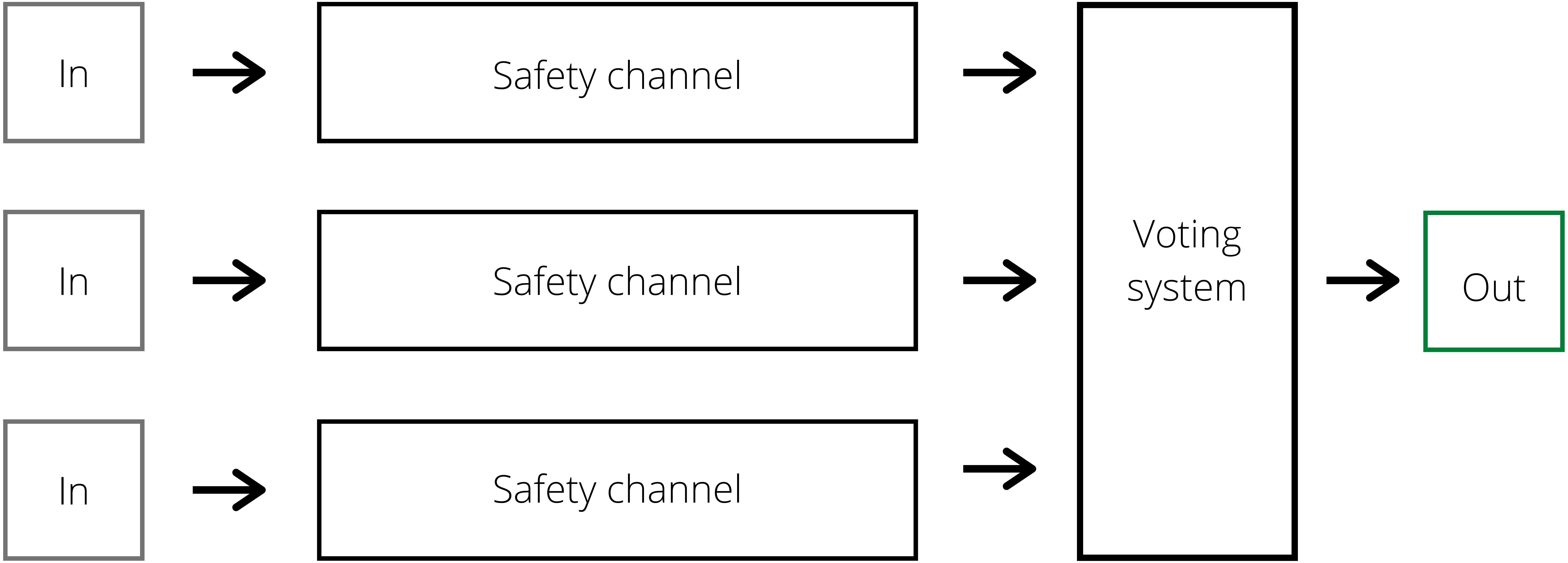


In → Safety channel → Out

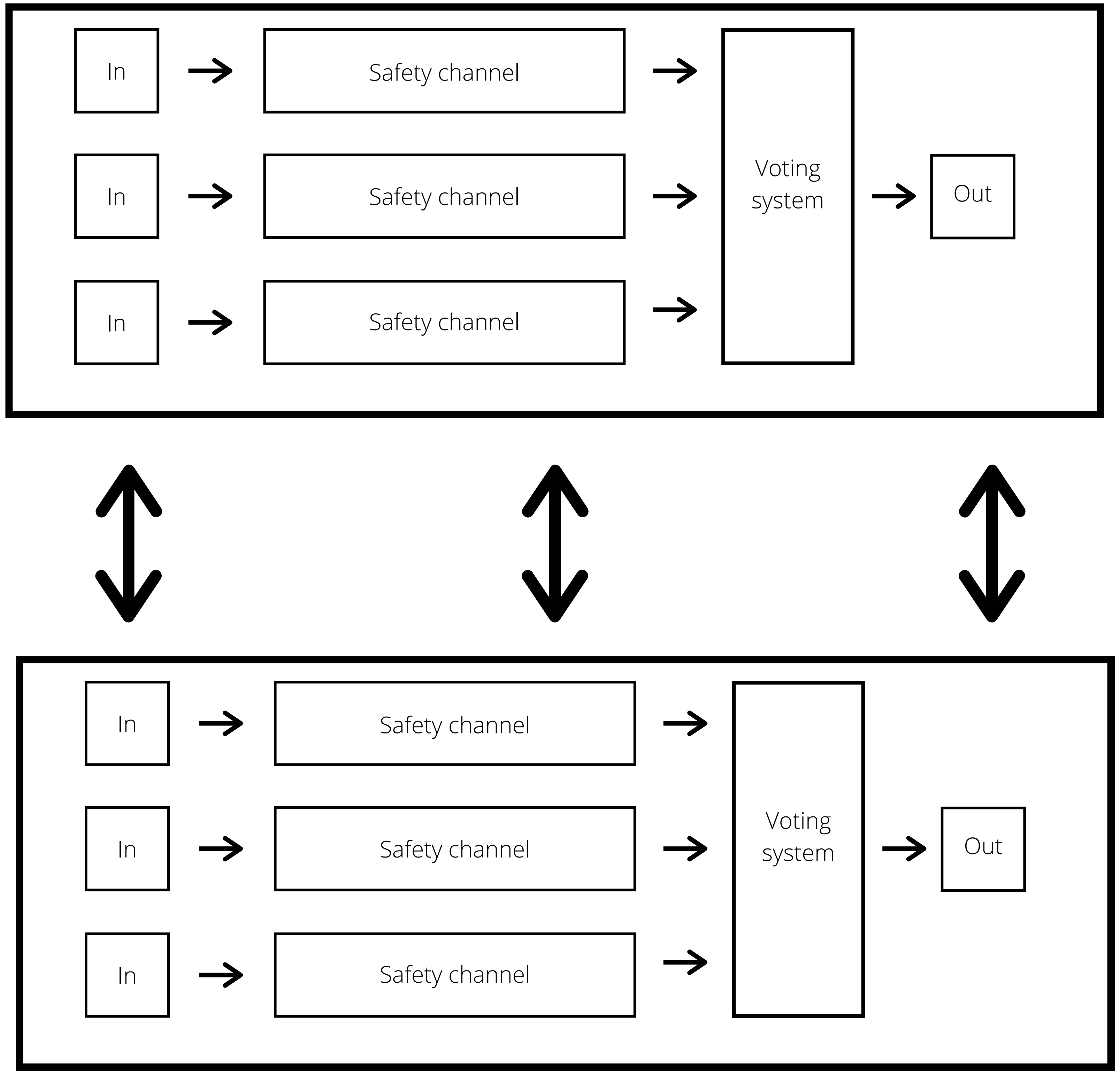
In → Safety channel → Out

Solw'IT Let's Solve It

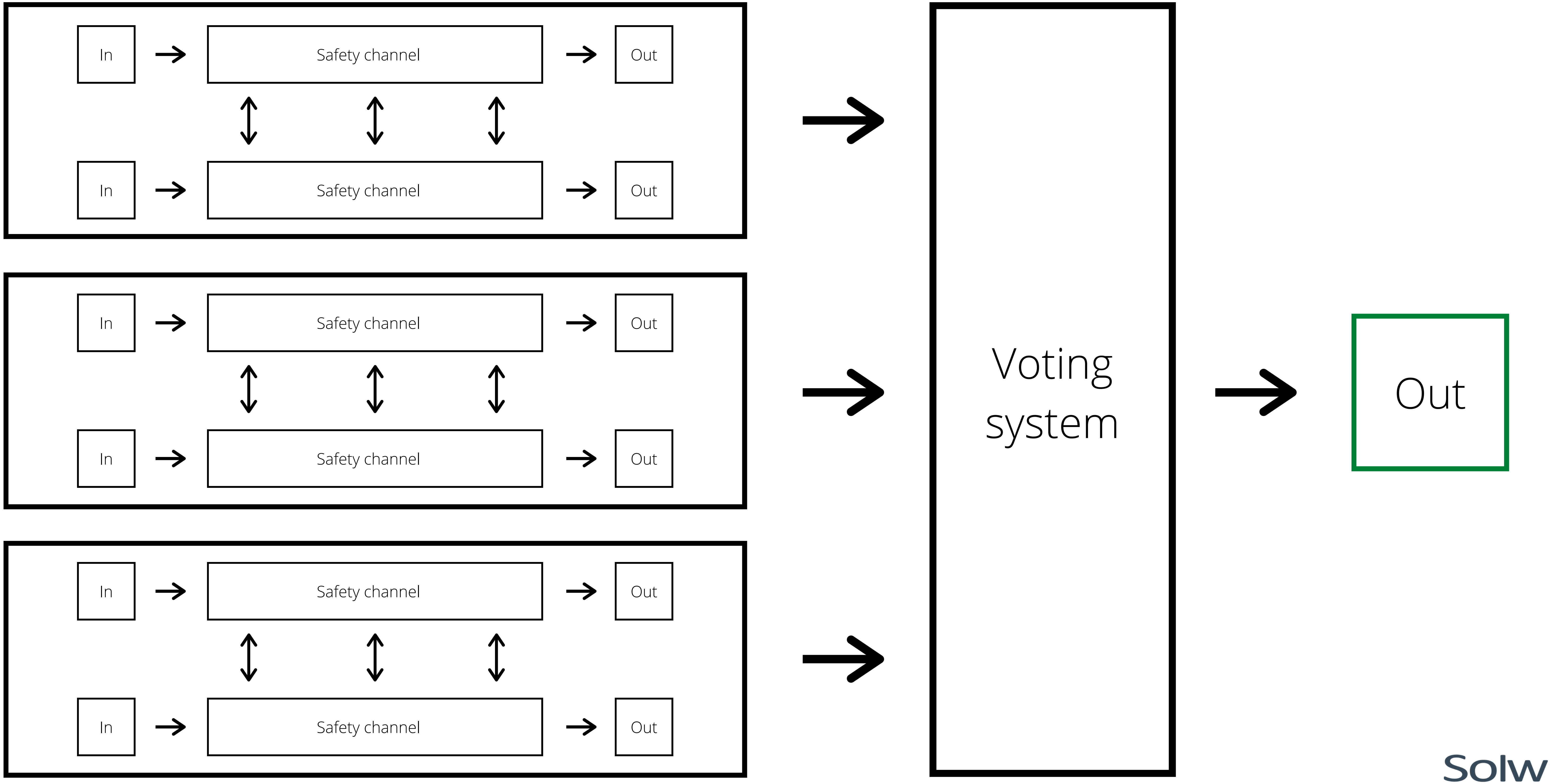
REDUNDANCY


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

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


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

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	++	++	++	++	++	++	++

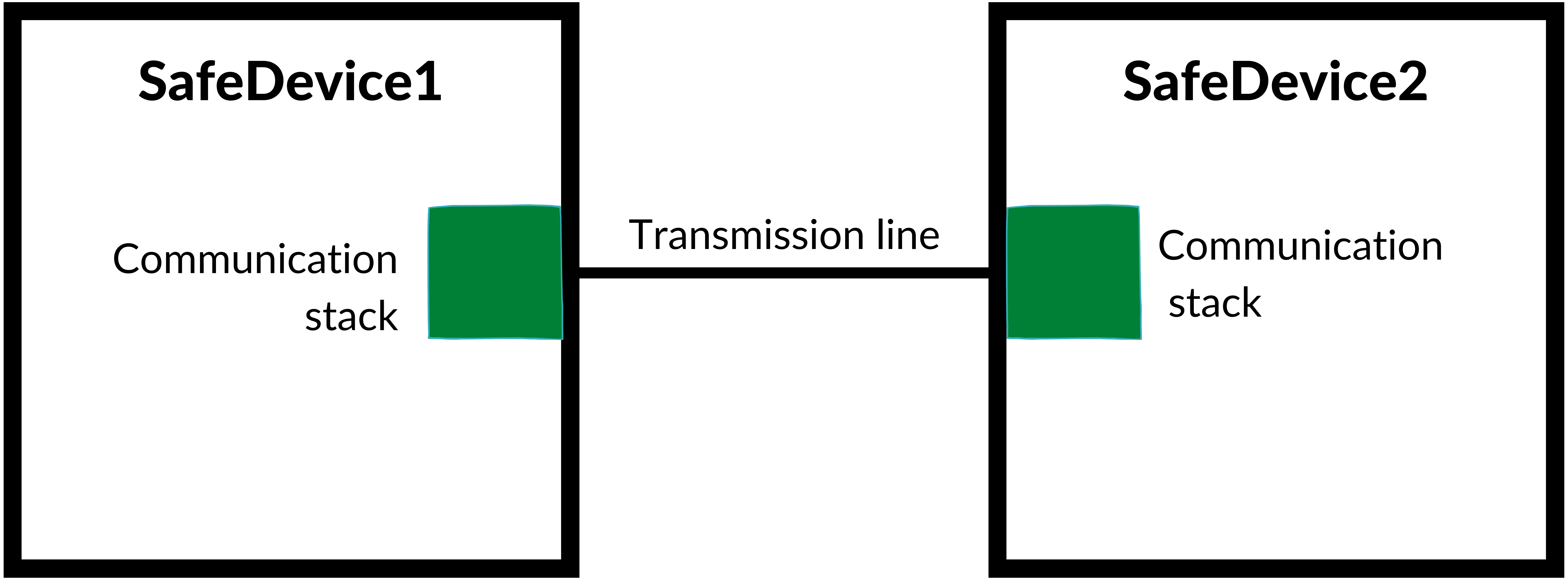
Legend:

- Threat can be neglected

+ Rare, weak countermeasures sufficient

++ Threat exists, strong countermeasures required

PROBLEM



SafeDevice1

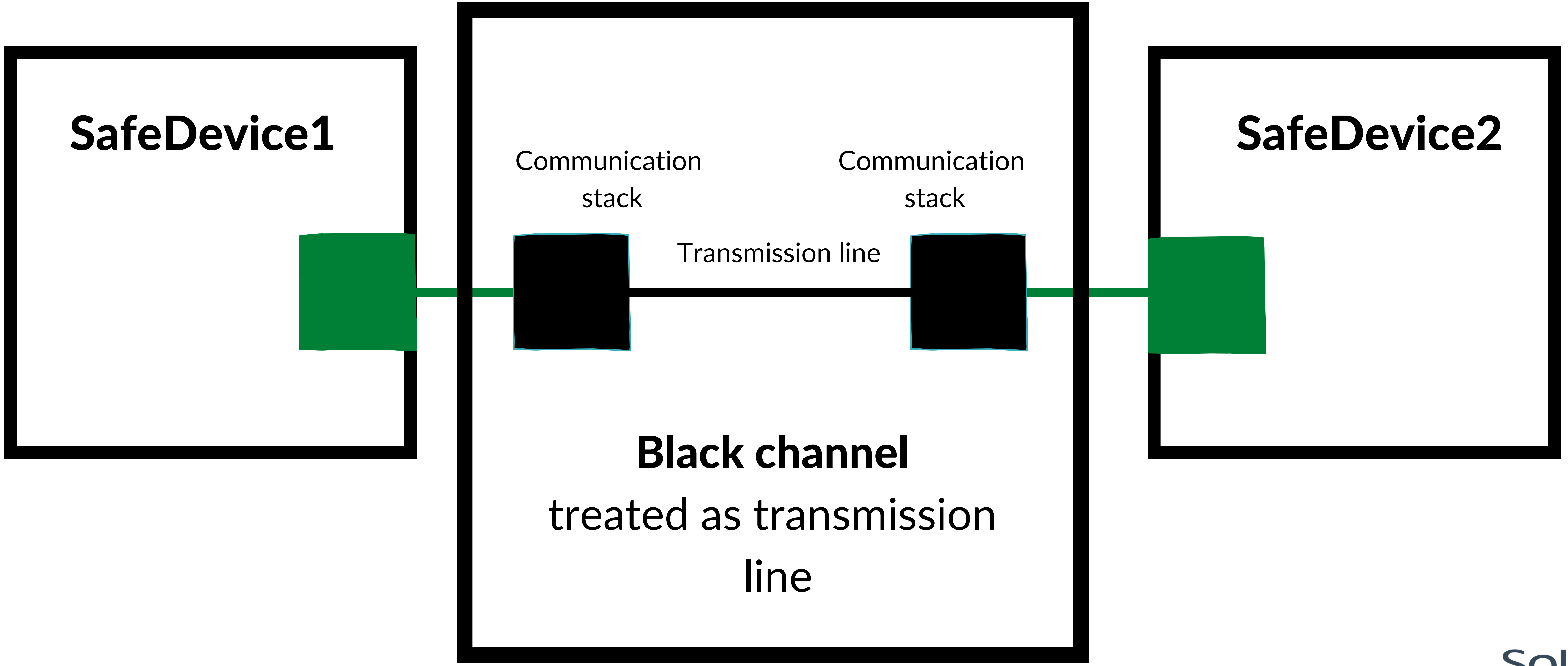
Communication stack

Transmission line

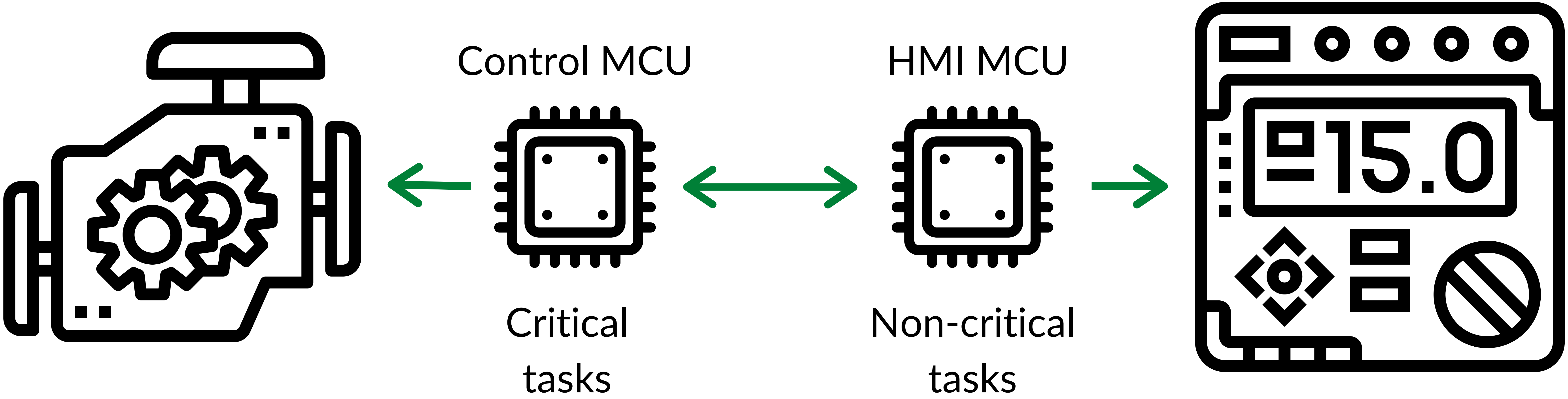
SafeDevice2

Communication stack

SOLUTION


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

MIXED CRITICALITY



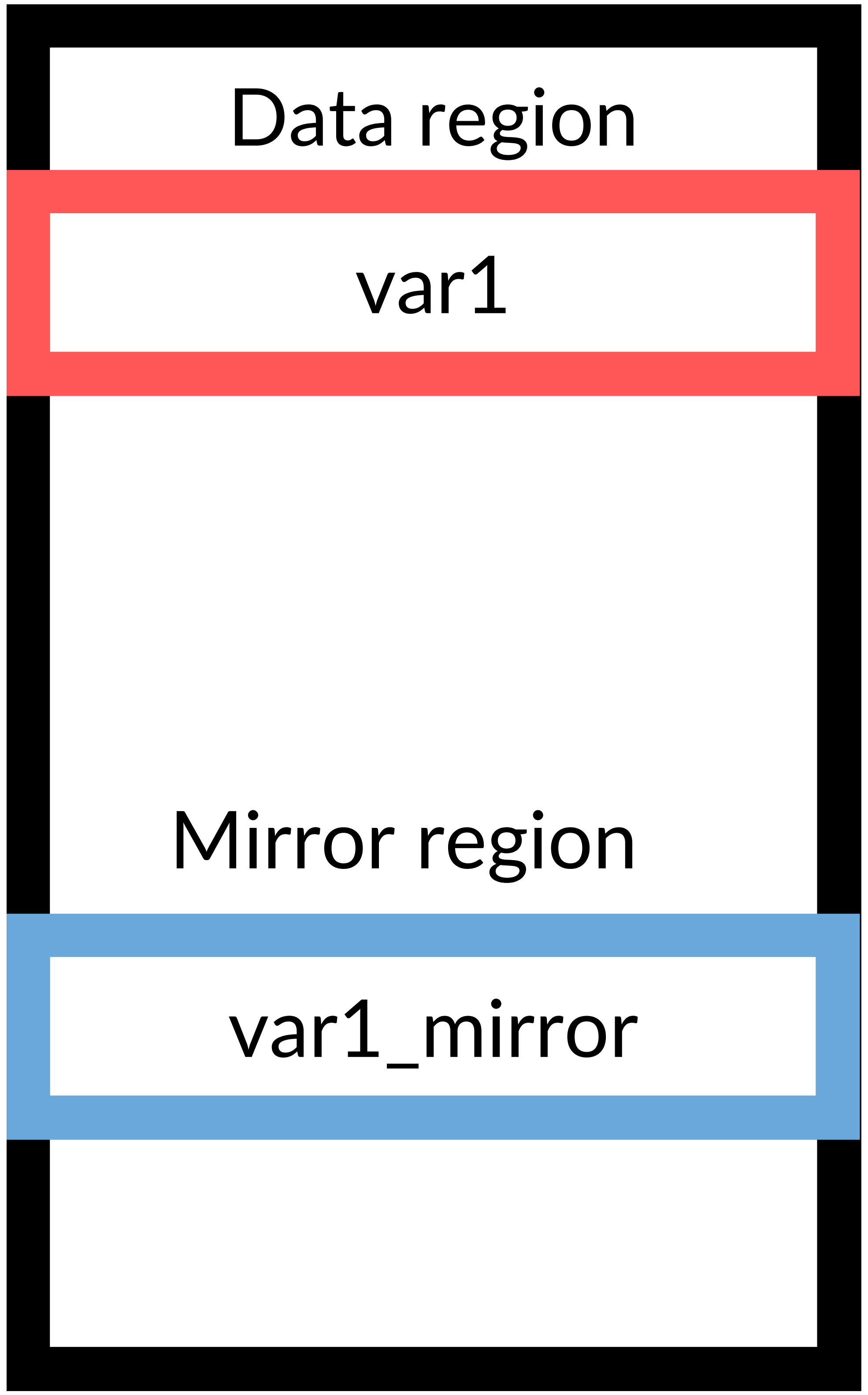
Control MCU

Critical tasks

HMI MCU

Non-critical tasks

DATA CORRUPTION

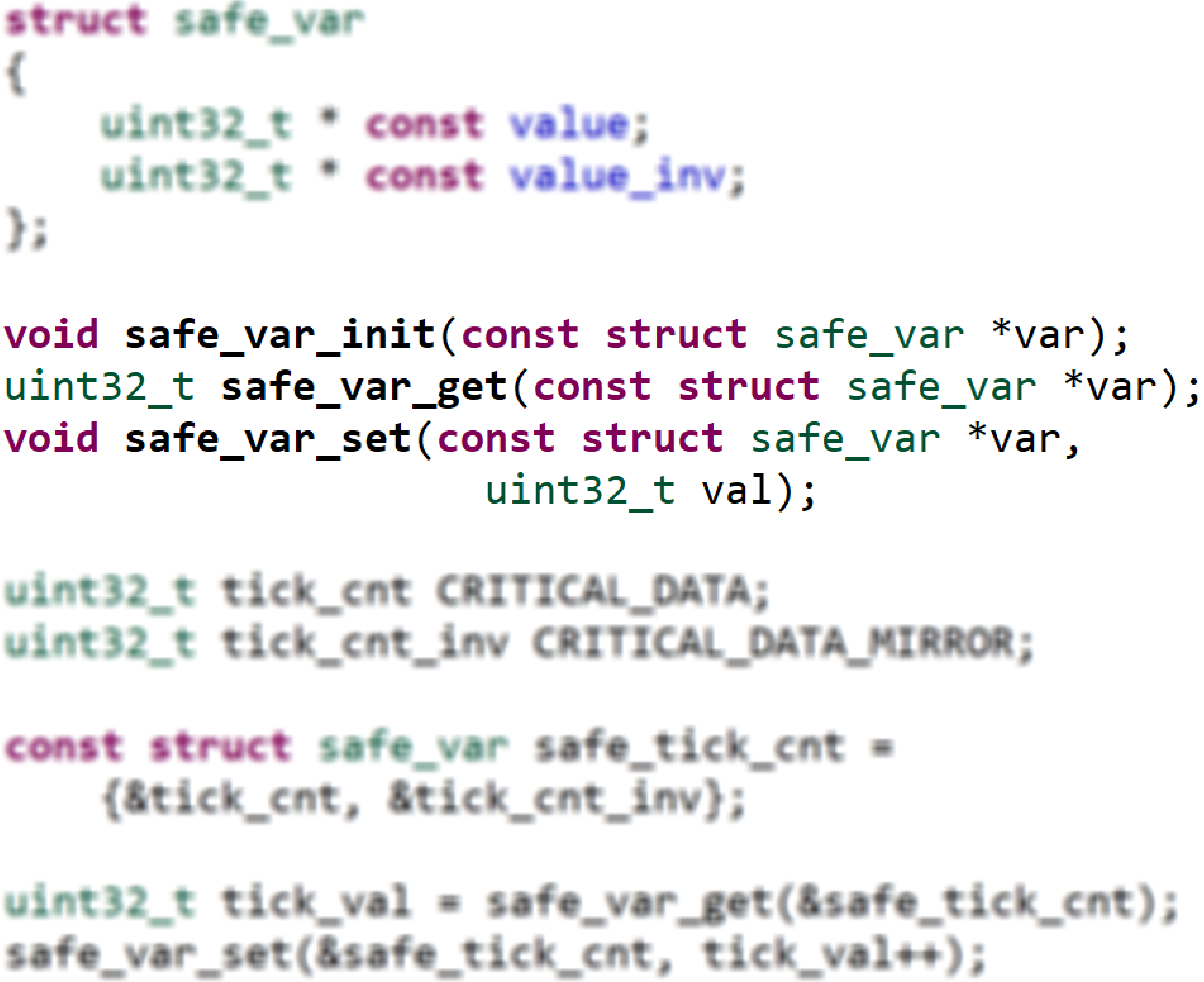


Invariant:

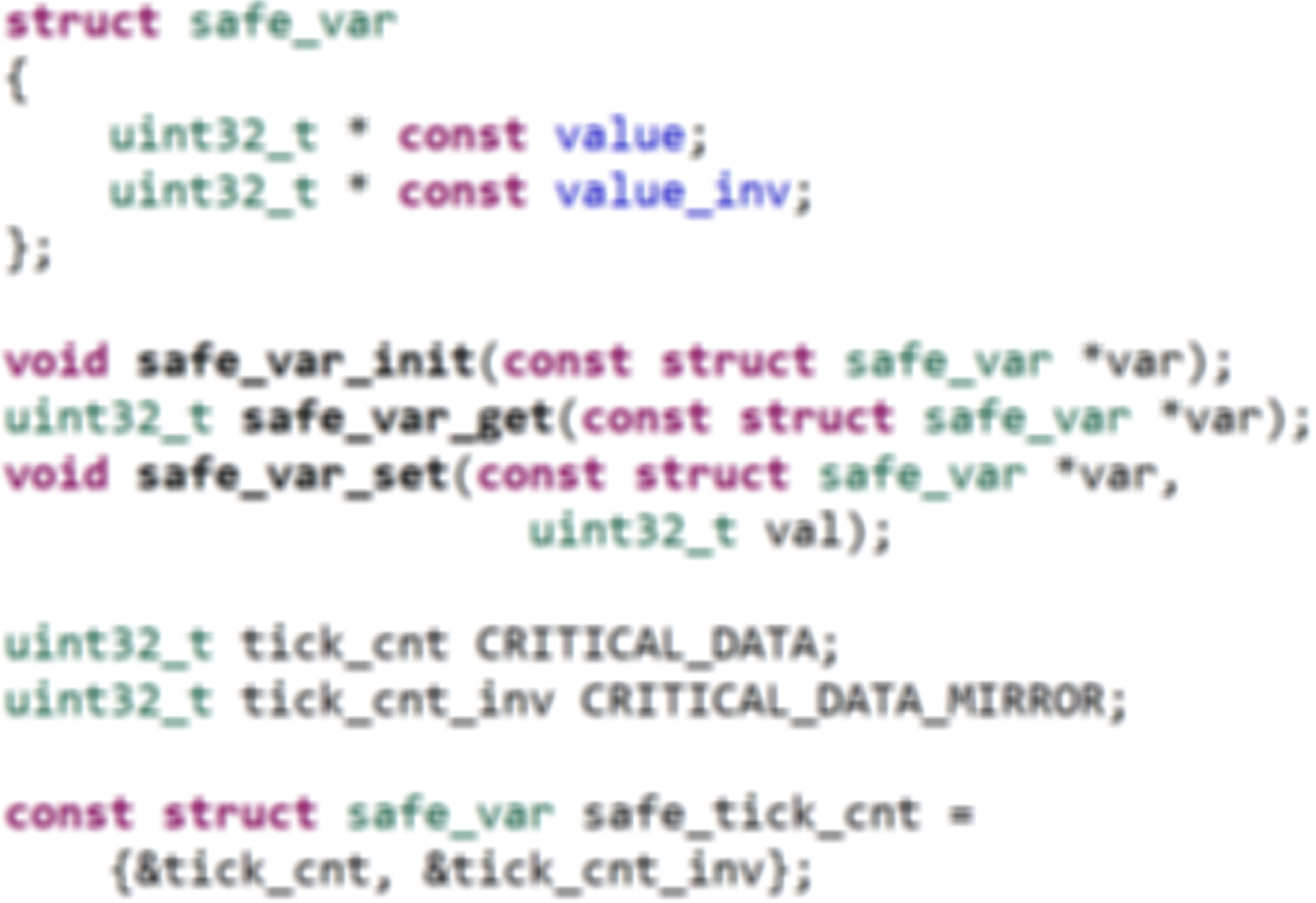
var1 ^ var1_mirror = 0xFFFFFFFF

```c
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++);
```

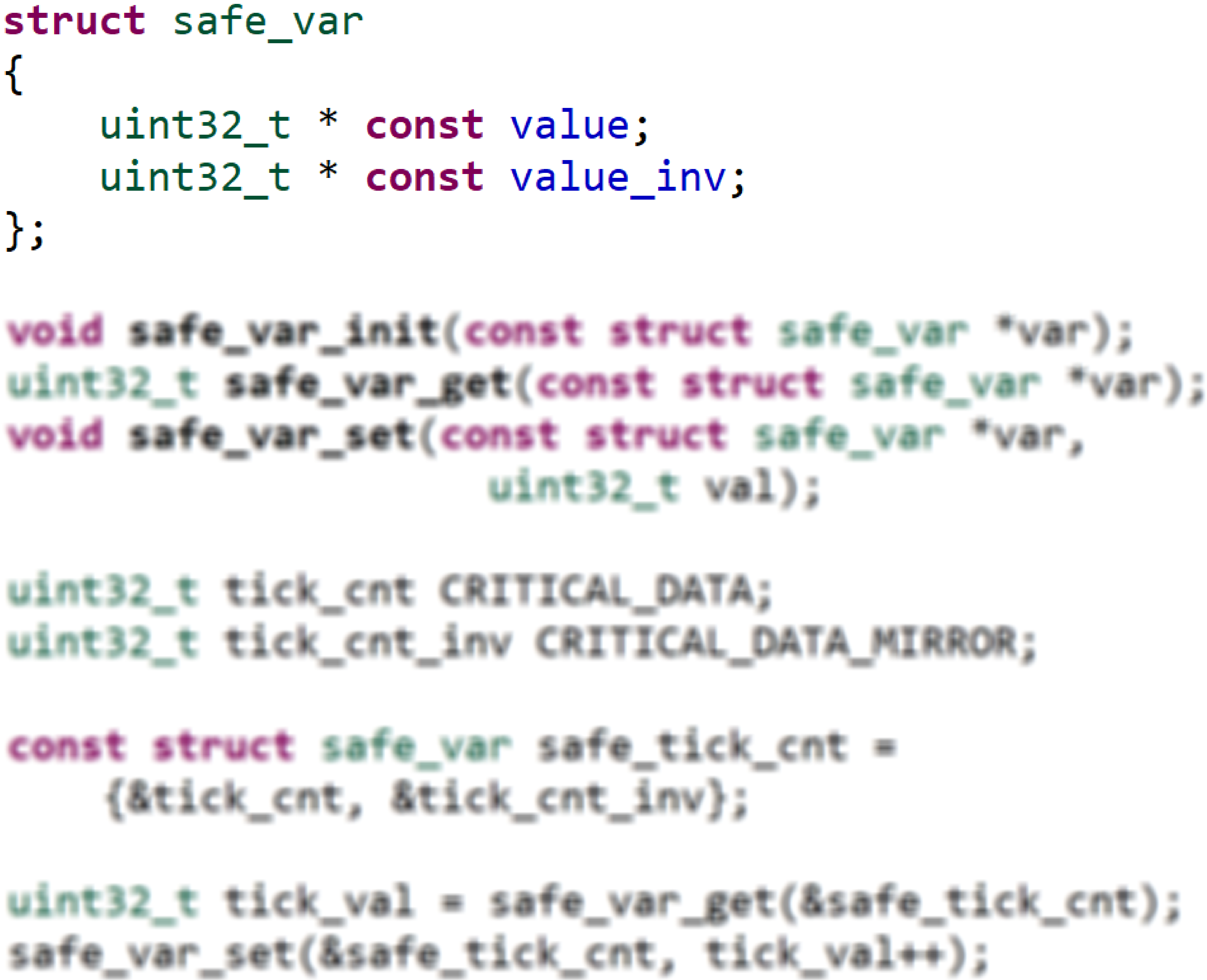


```
struct safe_var
{
    uint32_t * const value;
    uint32_t * const value_inv;
};

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++);
```

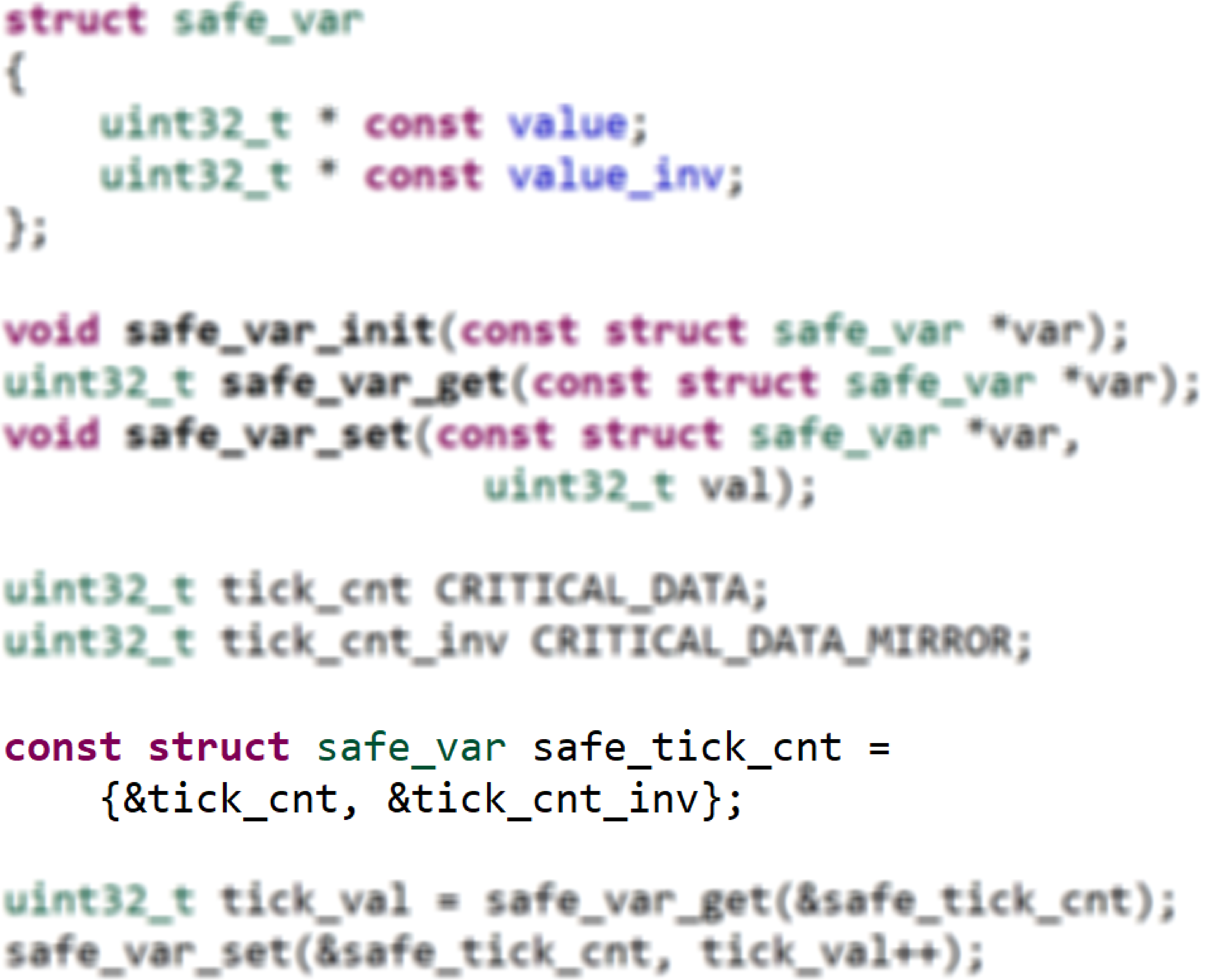


```
struct safe_var
{
    uint32_t * const value;
    uint32_t * const value_inv;
};

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++);
```



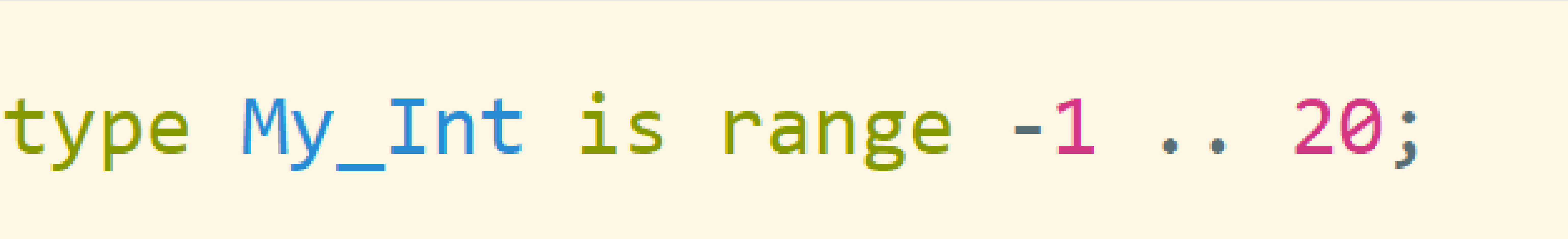
```c
struct safe_var
{
    uint32_t * const value;
    uint32_t * const value_inv;
};

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++);
```



```
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++);
```

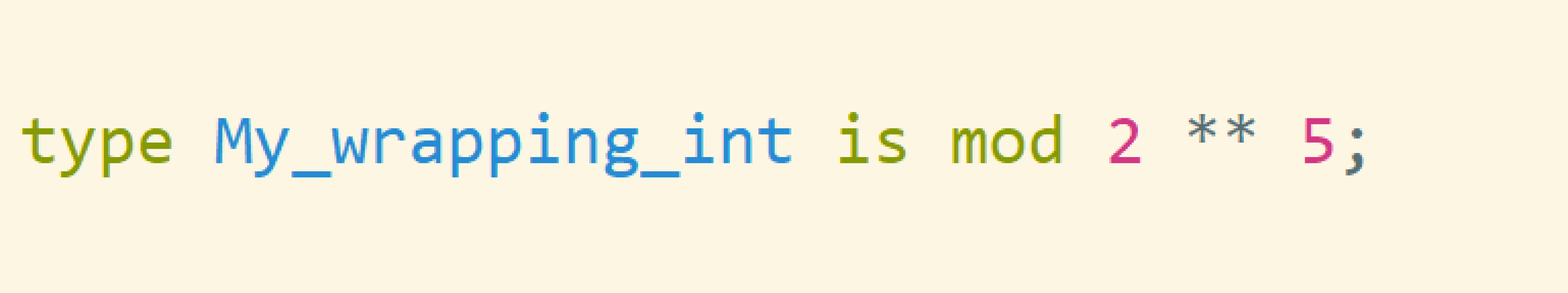
LANGUAGES

ADA



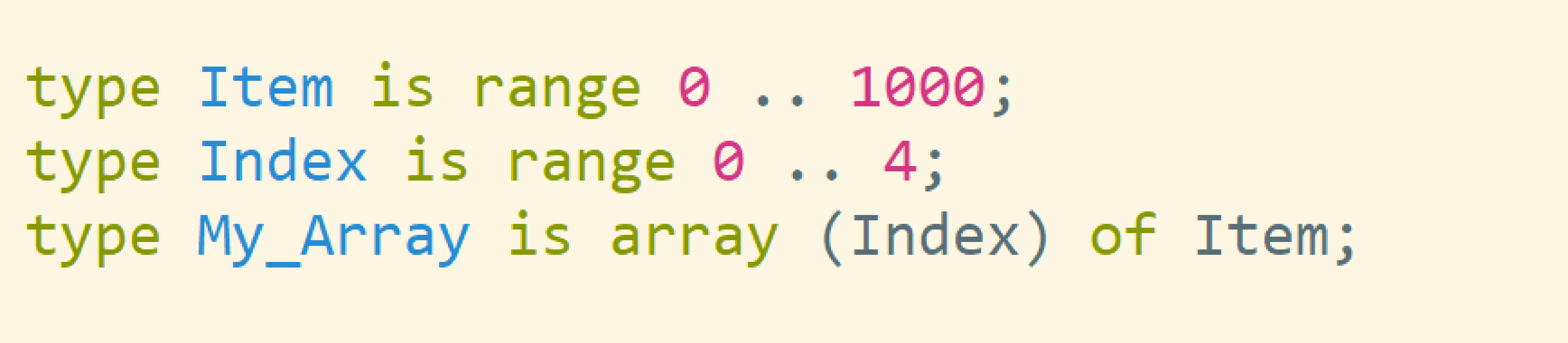
```
type My_Int is range -1 .. 20;
```

ADA



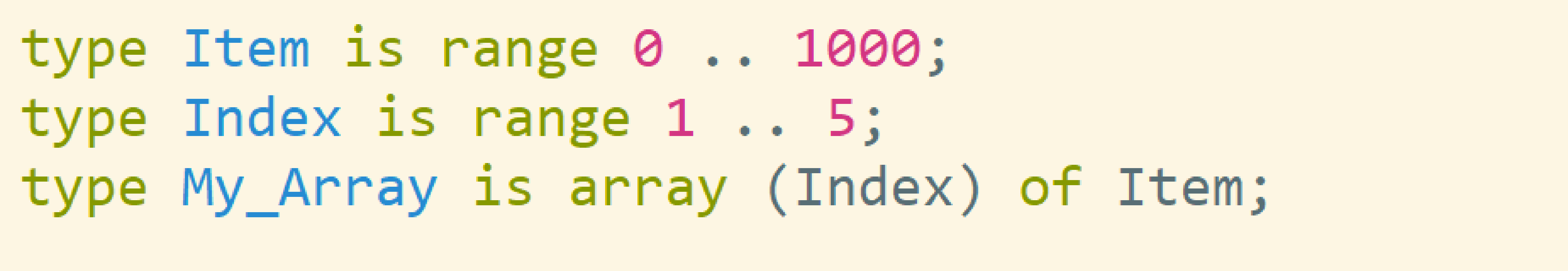
```
type My_wrapping_int is mod 2 ** 5;
```

ADA



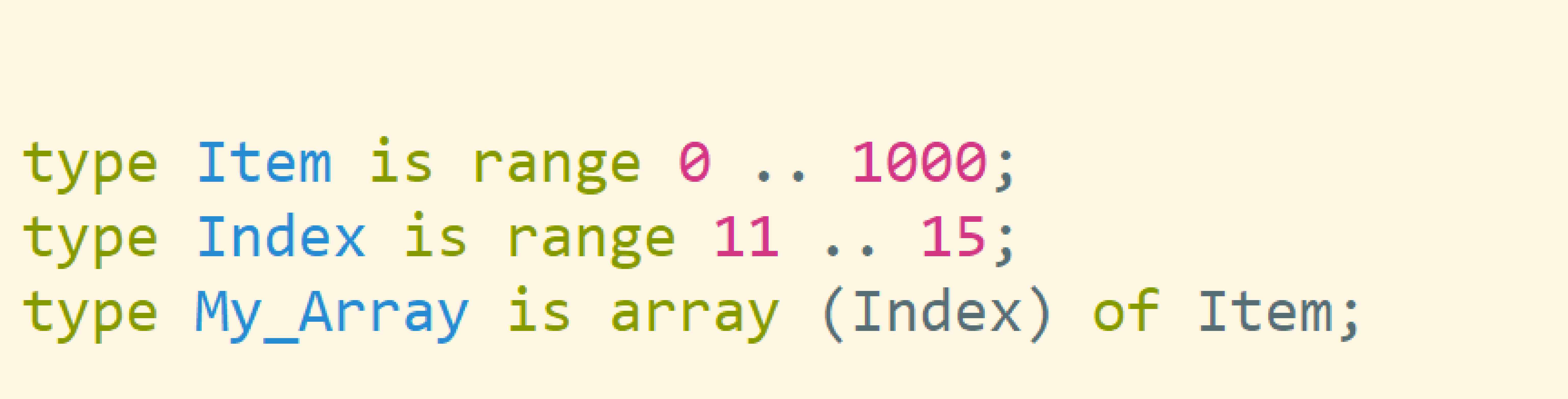
```ada
type Item is range 0 .. 1000;
type Index is range 0 .. 4;
type My_Array is array (Index) of Item;
```

ADA



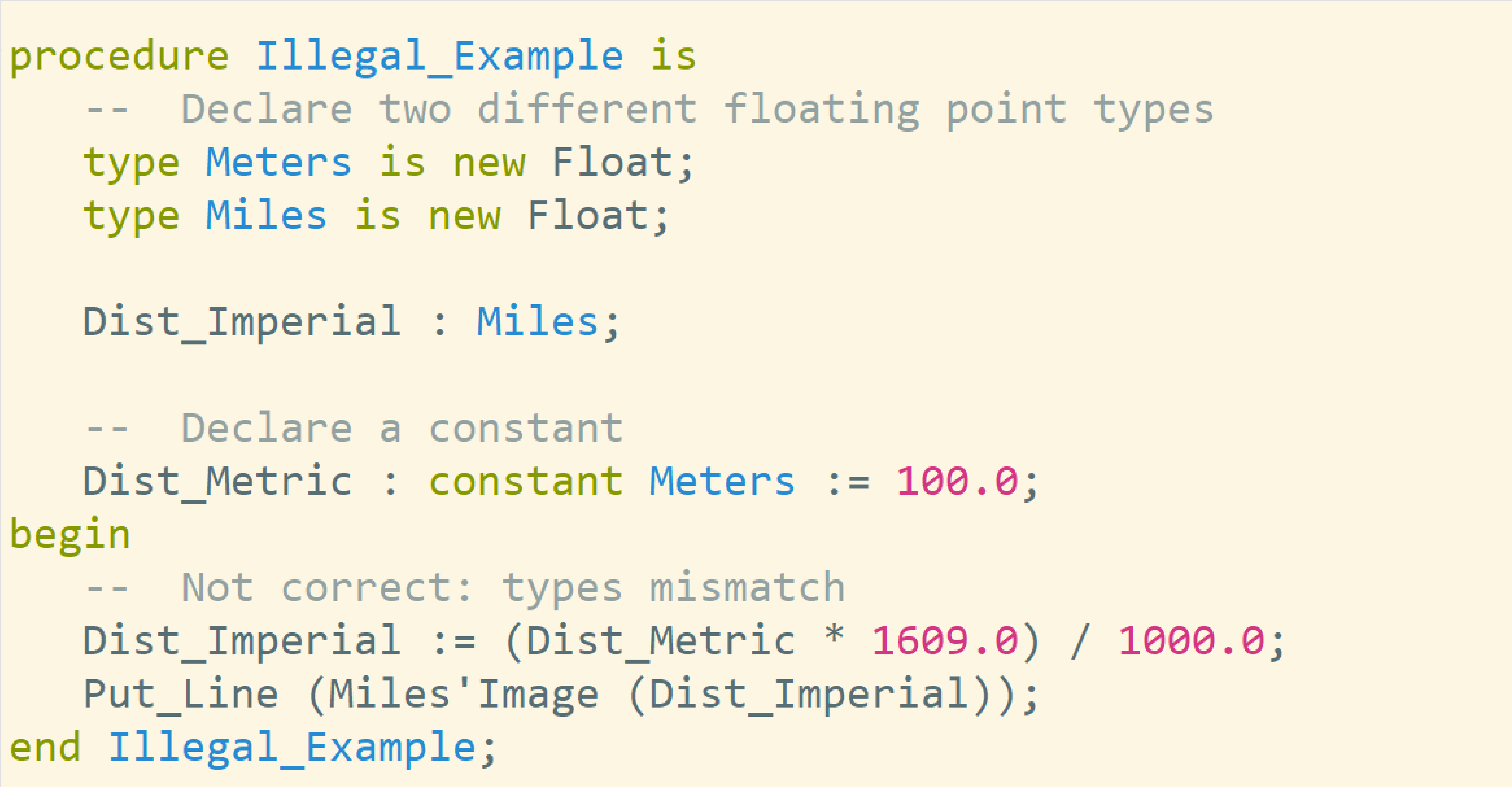
```
type Item is range 0 .. 1000;
type Index is range 1 .. 5;
type My_Array is array (Index) of Item;
```

ADA



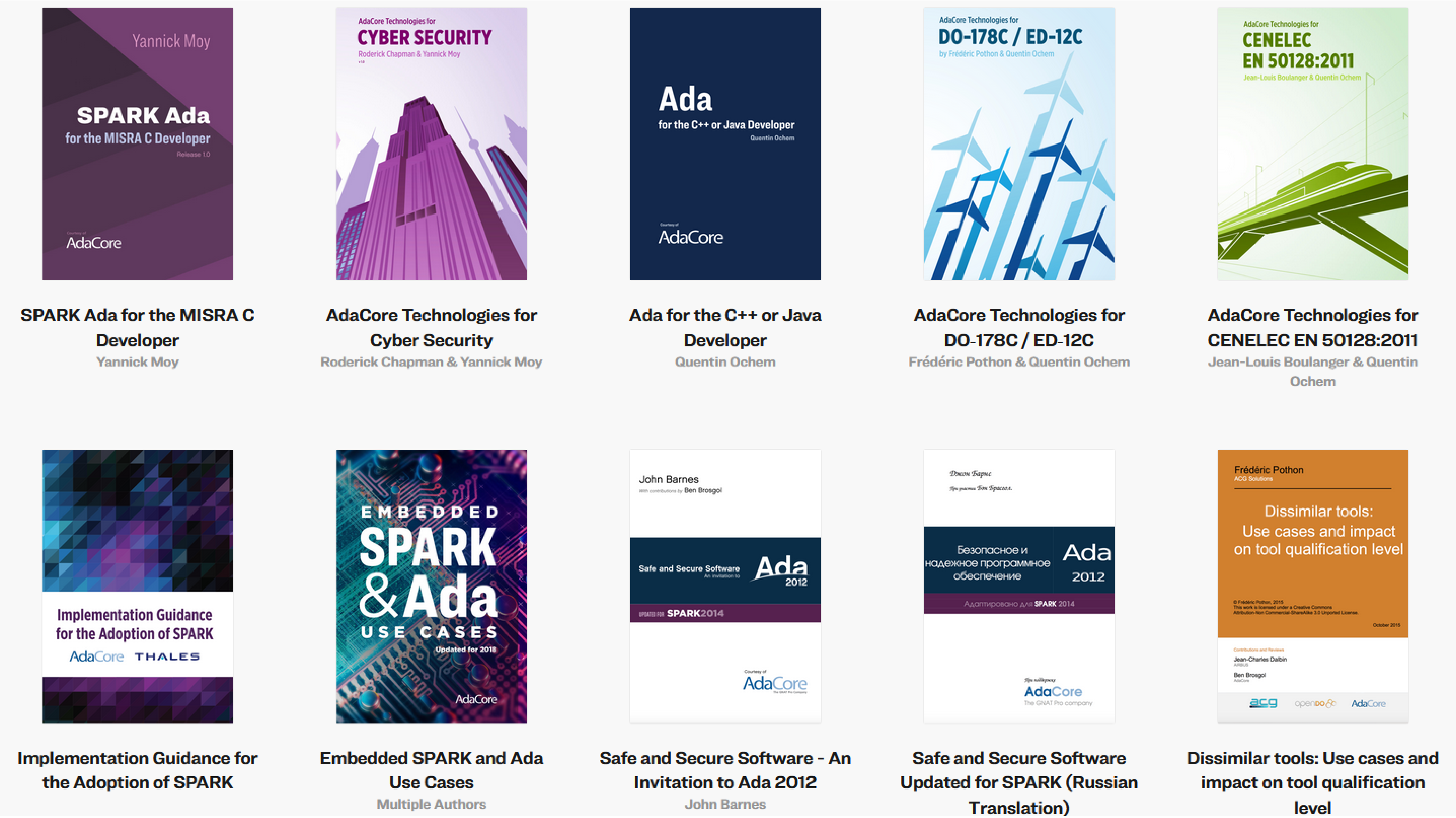
```
type Item is range 0 .. 1000;
type Index is range 11 .. 15;
type My_Array is array (Index) of Item;
```

ADA



```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;
```

ADACORE



SPARK Ada for the MISRA C Developer
Yannick Moy

AdaCore Technologies for Cyber Security
Roderick Chapman & Yannick Moy

Ada for the C++ or Java Developer
Quentin Ochem

AdaCore Technologies for DO-178C / ED-12C
Frédéric Pothon & Quentin Ochem

AdaCore Technologies for CENELEC EN 50128:2011
Jean-Louis Boulanger & Quentin Ochem

Implementation Guidance for the Adoption of SPARK

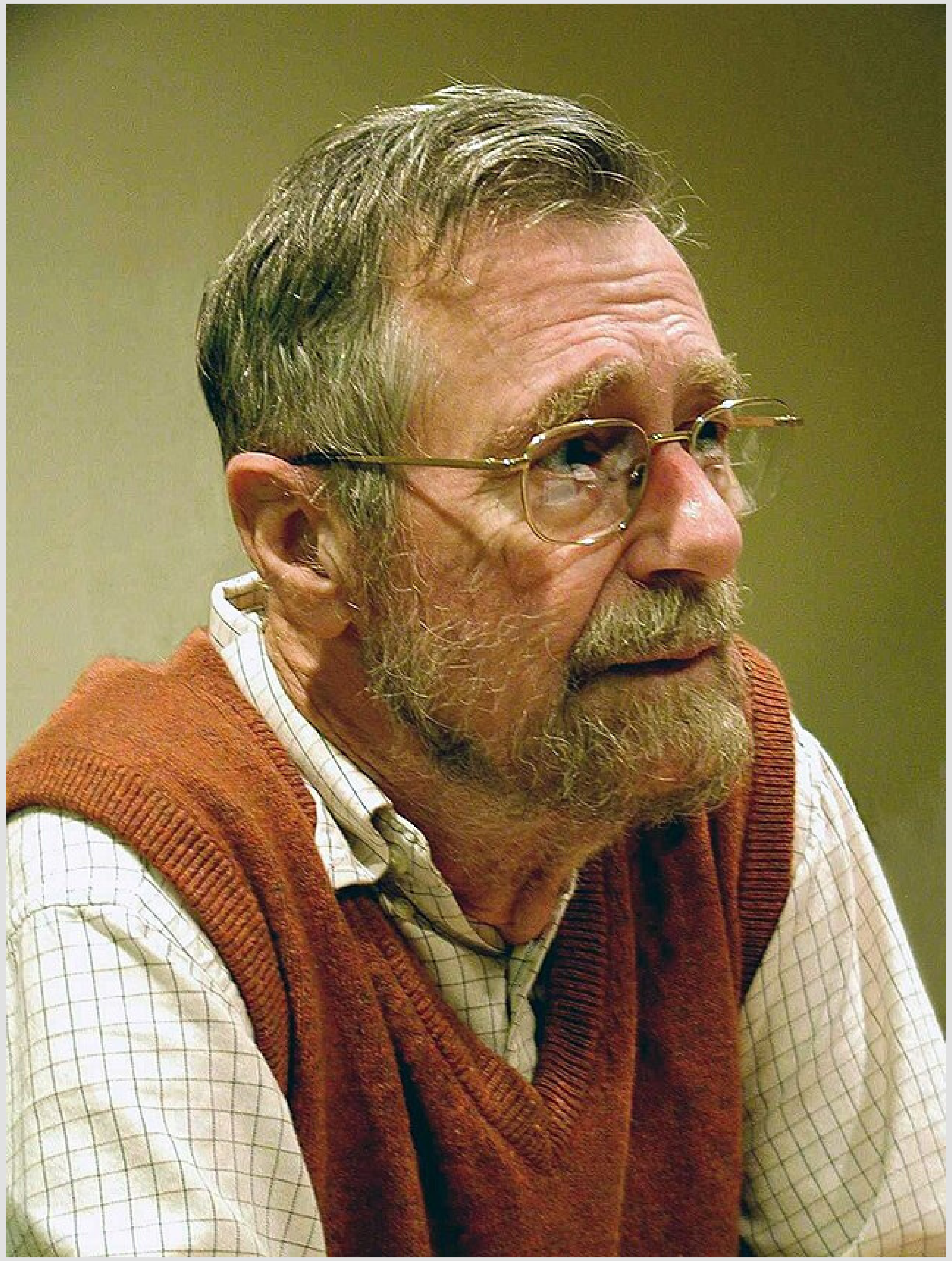
Embedded SPARK and Ada Use Cases
Multiple Authors

Safe and Secure Software - An Invitation to Ada 2012
John Barnes

Safe and Secure Software Updated for SPARK (Russian Translation)

Dissimilar tools: Use cases and impact on tool qualification level

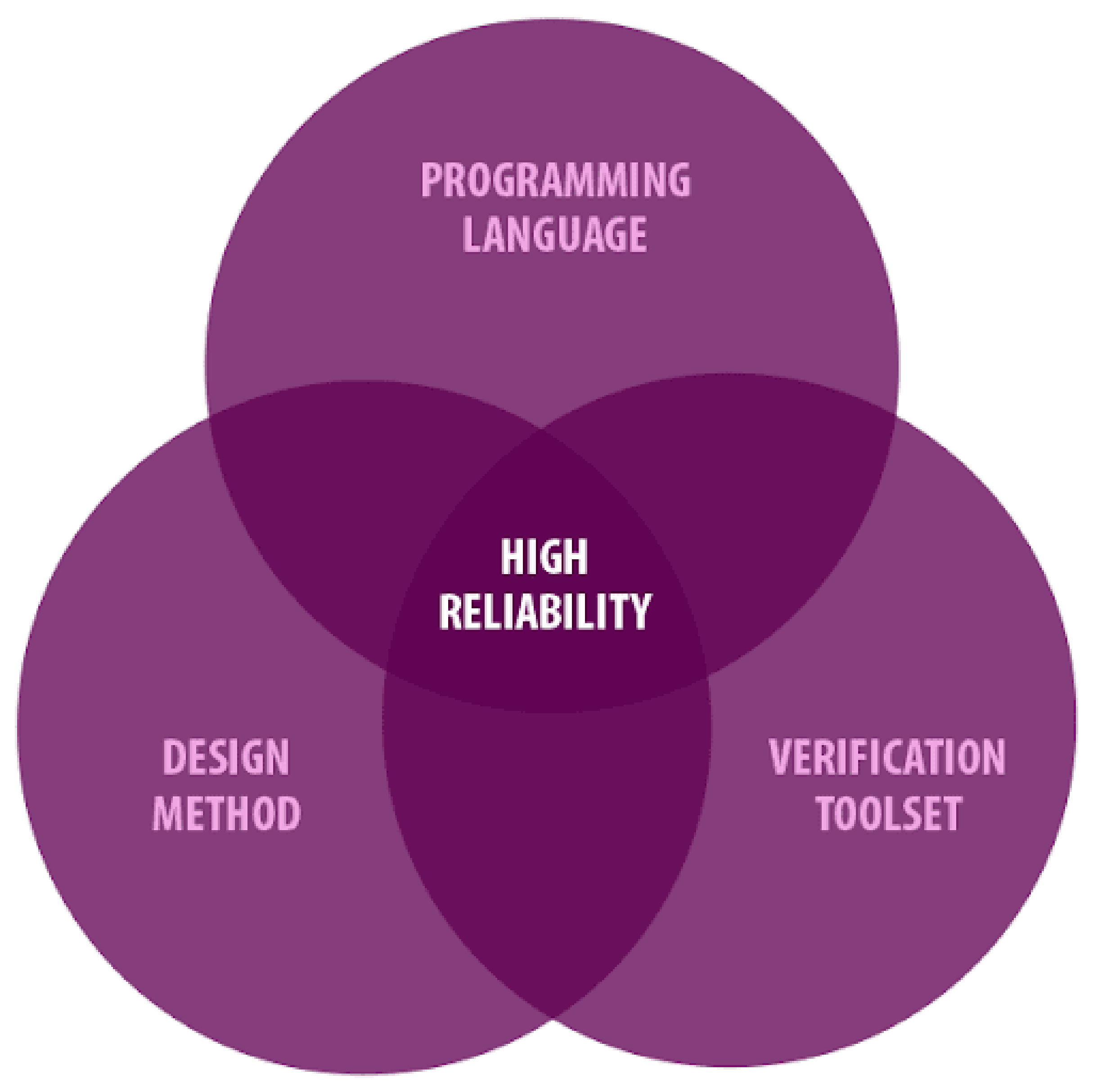
FORMAL PROOF



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

EDSGER DIJKSTRA

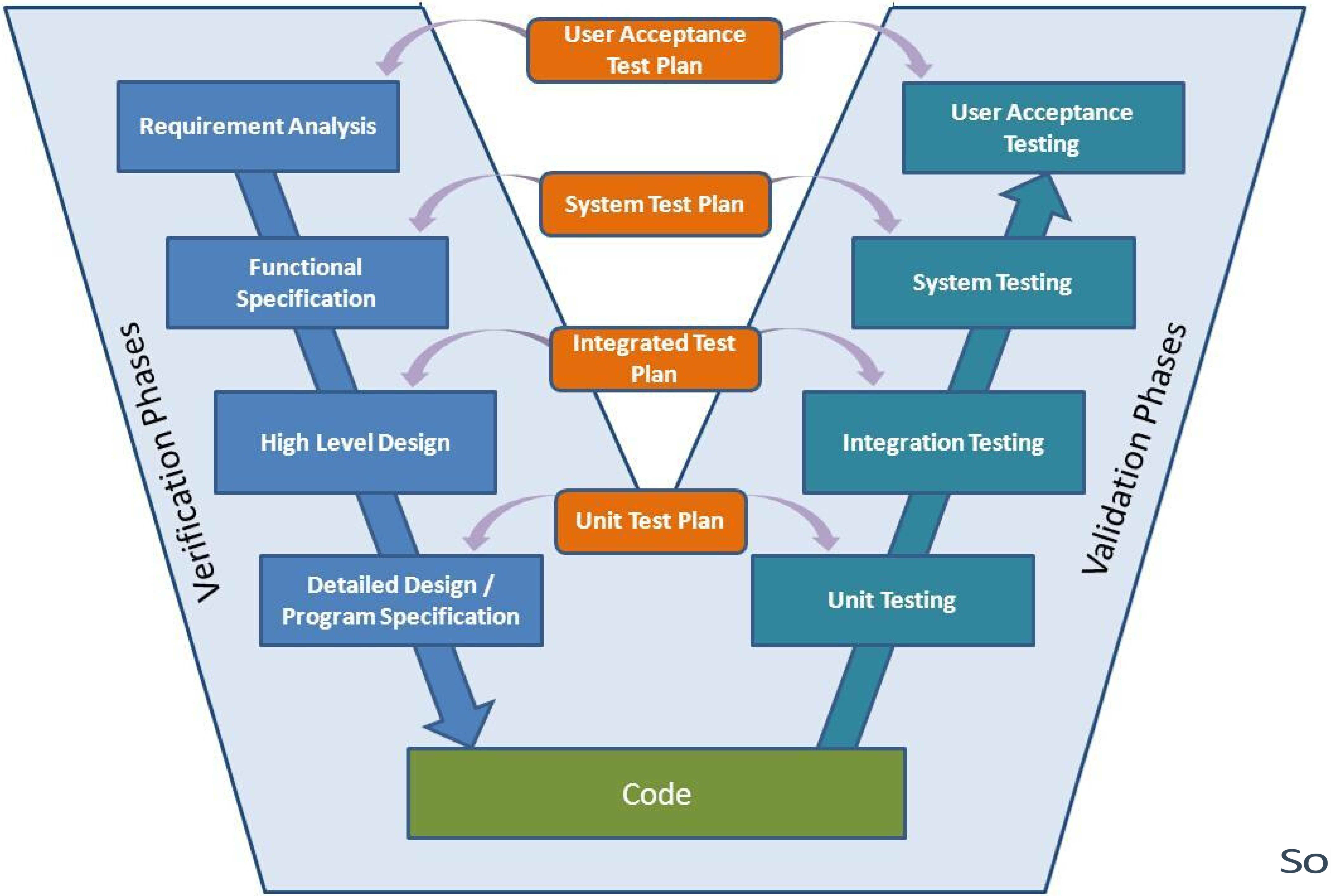
ADA SPARK


PROGRAMMING LANGUAGE
HIGH RELIABILITY
DESIGN METHOD
VERIFICATION TOOLSET

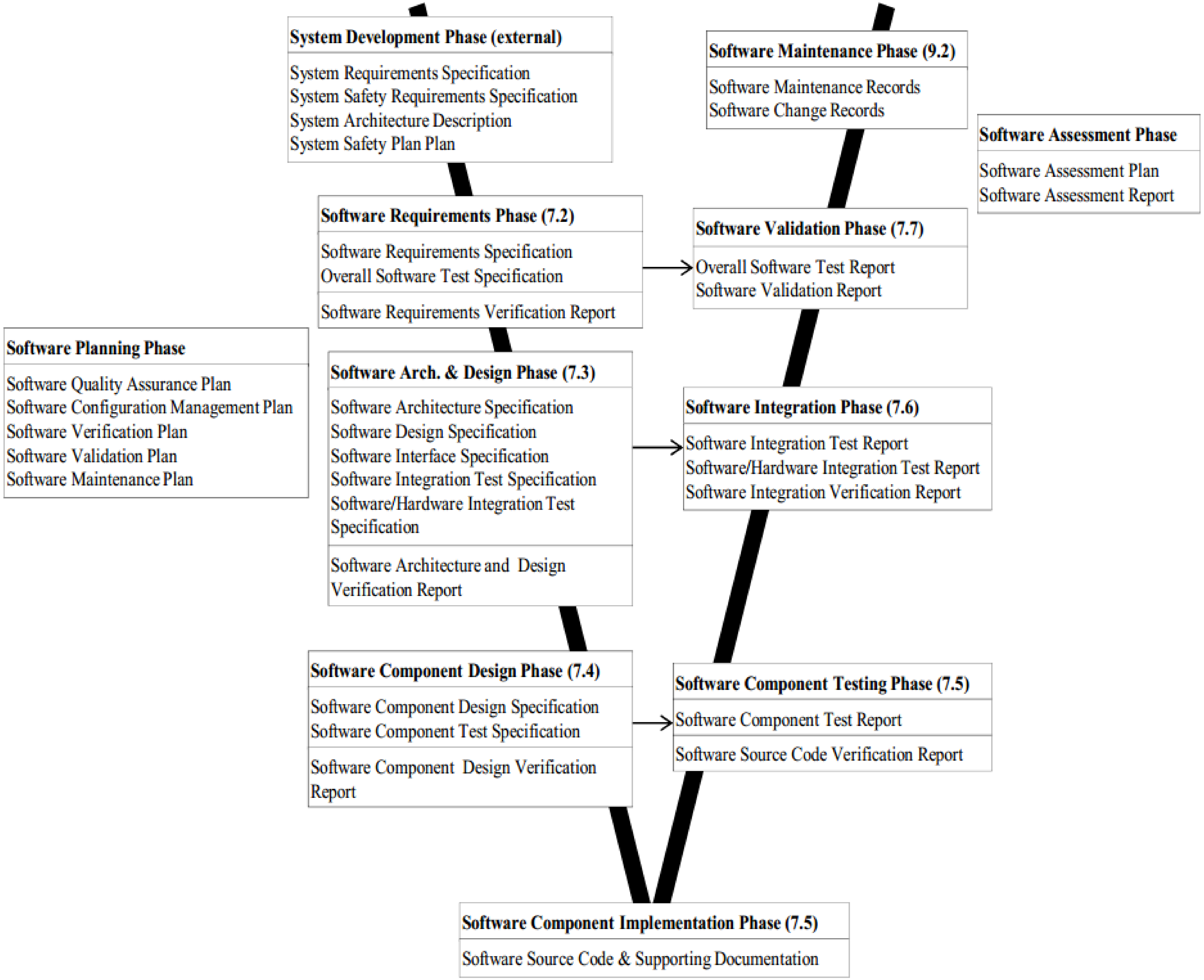
SPARK2014

LANGUAGE SUBSETS

- MISRA C
- AUTOSAR C++



Solw'IT Let's Solve It



Software Planning Phase

Software Quality Assurance Plan
Software Configuration Management Plan
Software Verification Plan
Software Validation Plan
Software Maintenance Plan

System Development Phase (external)

System Requirements Specification
System Safety Requirements Specification
System Architecture Description
System Safety Plan Plan

Software Requirements Phase (7.2)

Software Requirements Specification
Overall Software Test Specification

Software Requirements Verification Report

Software Arch. & Design Phase (7.3)

Software Architecture Specification
Software Design Specification
Software Interface Specification
Software Integration Test Specification
Software/Hardware Integration Test Specification

Software Architecture and Design Verification Report

Software Component Design Phase (7.4)

Software Component Design Specification
Software Component Test Specification

Software Component Design Verification Report

Software Component Implementation Phase (7.5)

Software Source Code & Supporting Documentation

Software Component Testing Phase (7.5)

Software Component Test Report

Software Source Code Verification Report

Software Integration Phase (7.6)

Software Integration Test Report
Software/Hardware Integration Test Report
Software Integration Verification Report

Software Validation Phase (7.7)

Overall Software Test Report
Software Validation Report

Software Maintenance Phase (9.2)

Software Maintenance Records
Software Change Records

Software Assessment Phase

Software Assessment Plan
Software Assessment Report

EFFECTIVE DOCUMENTATION

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

VERSION MANAGEMENT

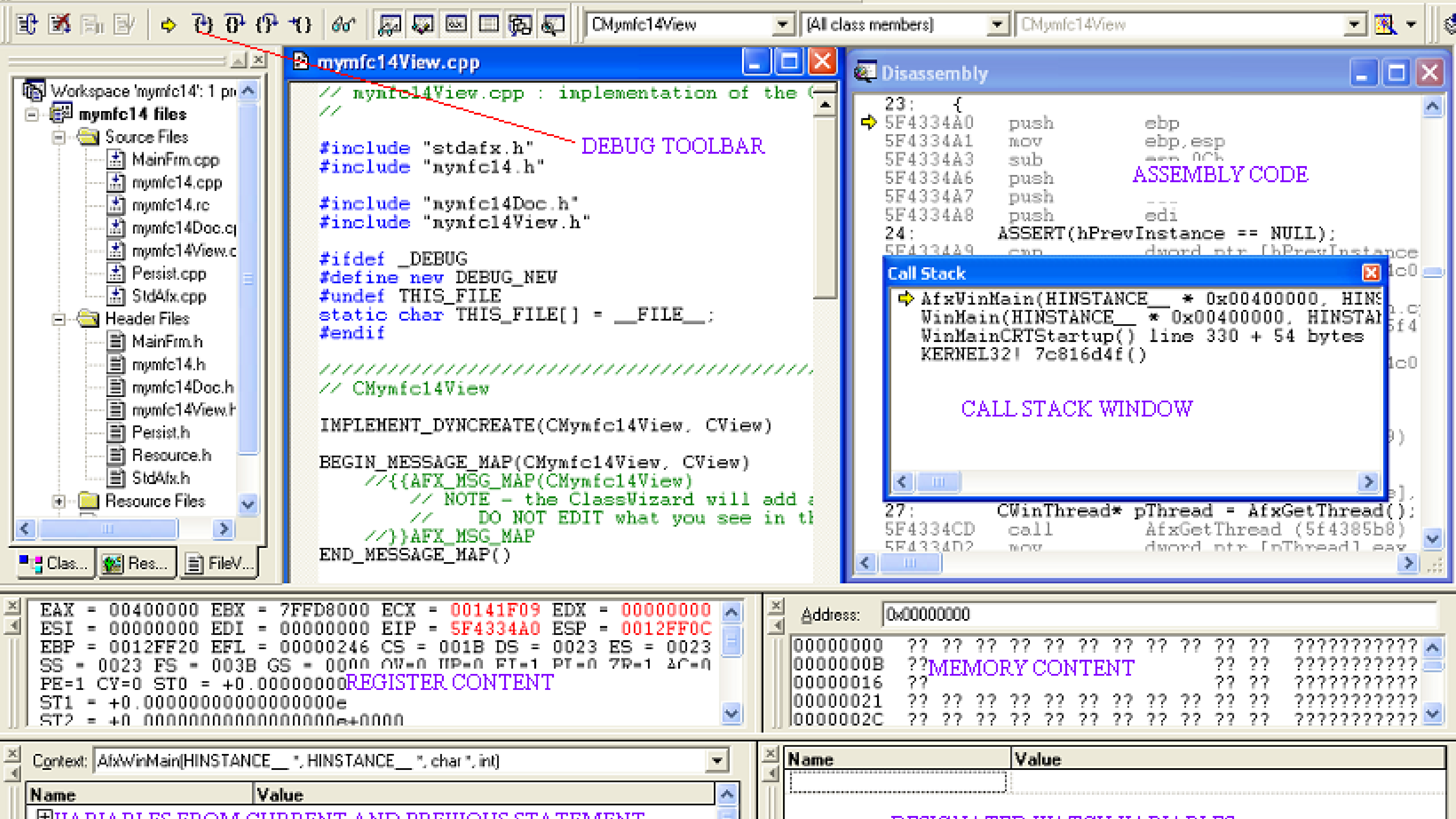
- version for every binary
- version for every PCB
- version for every bundle

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



```
// mymfc14View.cpp : implementation of the C
//

#include "stdafx.h"
#include "mymfc14.h"

#include "mymfc14Doc.h"
#include "mymfc14View.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

/////////////////////////////////////////////////////////
// CMymfc14View

IMPLEMENT_DYNCREATE(CMymfc14View, CView)

BEGIN_MESSAGE_MAP(CMymfc14View, CView)
    //{{AFX_MSG_MAP(CMymfc14View)
        // NOTE - the ClassWizard will add a
        //    DO NOT EDIT what you see in th
    //}}AFX_MSG_MAP
END_MESSAGE_MAP()
```

```
23:     {
5F4334A0   push        ebp
5F4334A1   mov         ebp,esp
5F4334A3   sub         esp,0Ch
5F4334A6   push
5F4334A7   push        ...
5F4334A8   push        edi
24:        ASSERT(hPrevInstance == NULL);
5F4334A9   cmp         dword ptr [hPrevInstance
27:        CWinThread* pThread = AfxGetThread();
5F4334CD   call        AfxGetThread (5f4385b8)
5F4334D2   mov         dword ptr [pThread],eax
```

```
AfxWinMain(HINSTANCE__ * 0x00400000, HINS
WinMain(HINSTANCE__ * 0x00400000, HINSTAN
WinMainCRTStartup() line 330 + 54 bytes
KERNEL32! 7c816d4f()
```

```
EAX = 00400000 EBX = 7FFD8000 ECX = 00141F09 EDX = 00000000
ESI = 00000000 EDI = 00000000 EIP = 5F4334A0 ESP = 0012FF0C
EBP = 0012FF20 EFL = 00000246 CS = 001B DS = 0023 ES = 0023
SS = 0023 FS = 003B GS = 0000 OV=0 UP=0 EI=1 PL=0 ZR=1 AC=0
PE=1 CY=0 ST0 = +0.00000000
ST1 = +0.000000000000000000e
ST2 = +0.000000000000000000e+0000
```

Address: 0x00000000

```
00000000  ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??  ???????????
0000000B  ??                            ?? ??  ???????????
00000016  ??                            ?? ??  ???????????
00000021  ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??  ???????????
0000002C  ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??  ???????????
```

Context: AfxWinMain(HINSTANCE__ *, HINSTANCE__ *, char *, int)

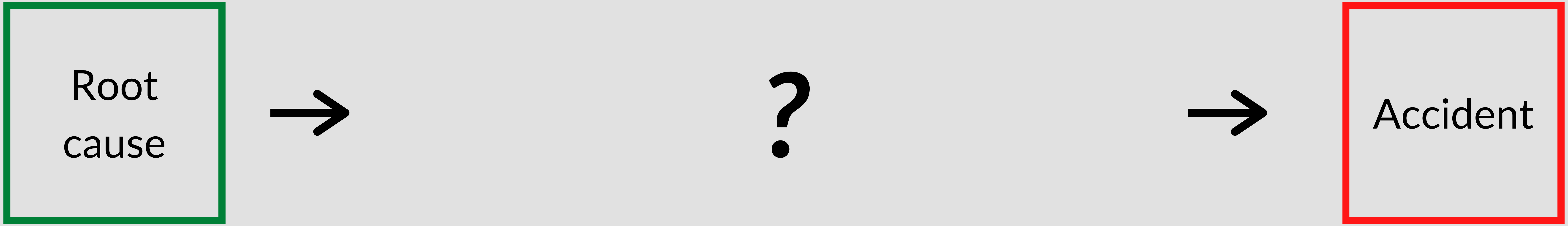
PROOVE THAT EVERY TOOL CAN BETRUSTED

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

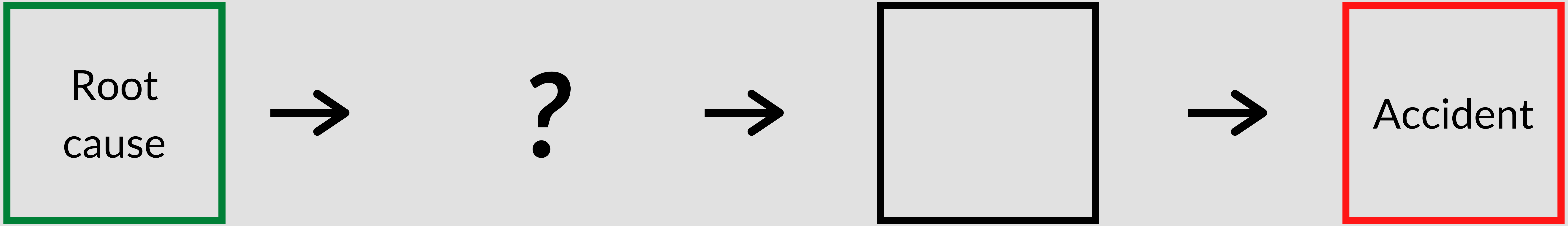
ROOT CAUSE ANALYSIS



Root cause → ? → Accident

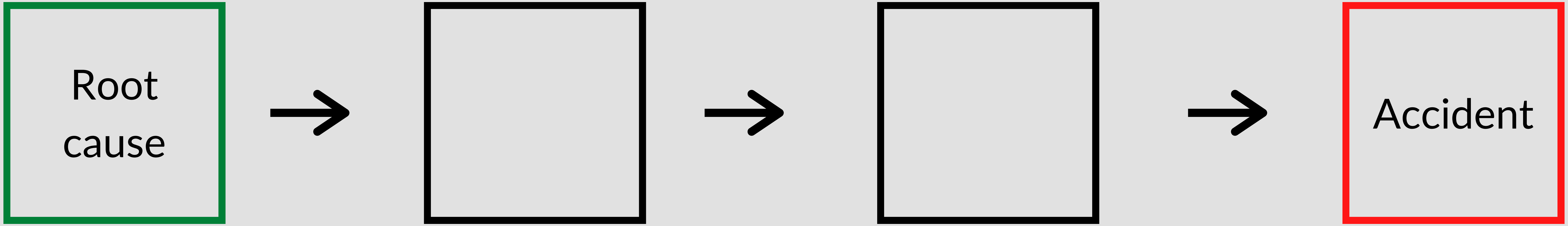
Why?

ROOT CAUSE ANALYSIS



Root cause → ? → → Accident

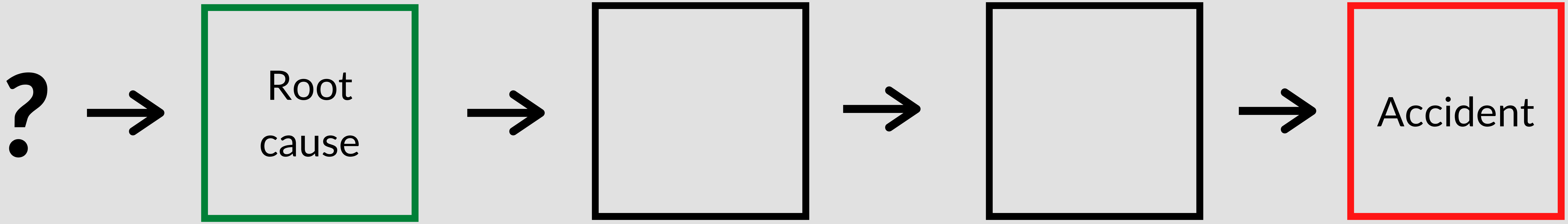
ROOT CAUSE ANALYSIS



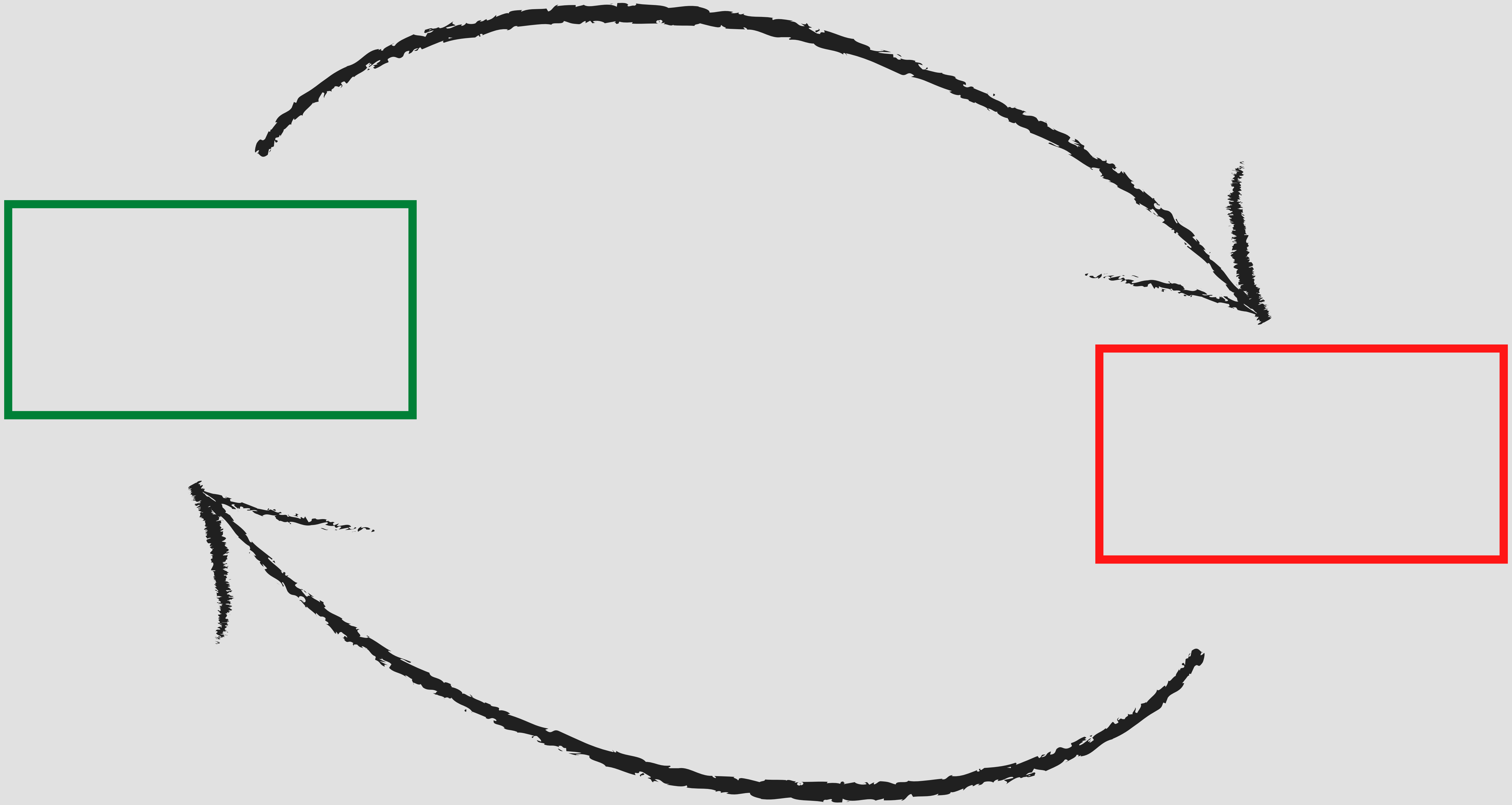
Root cause

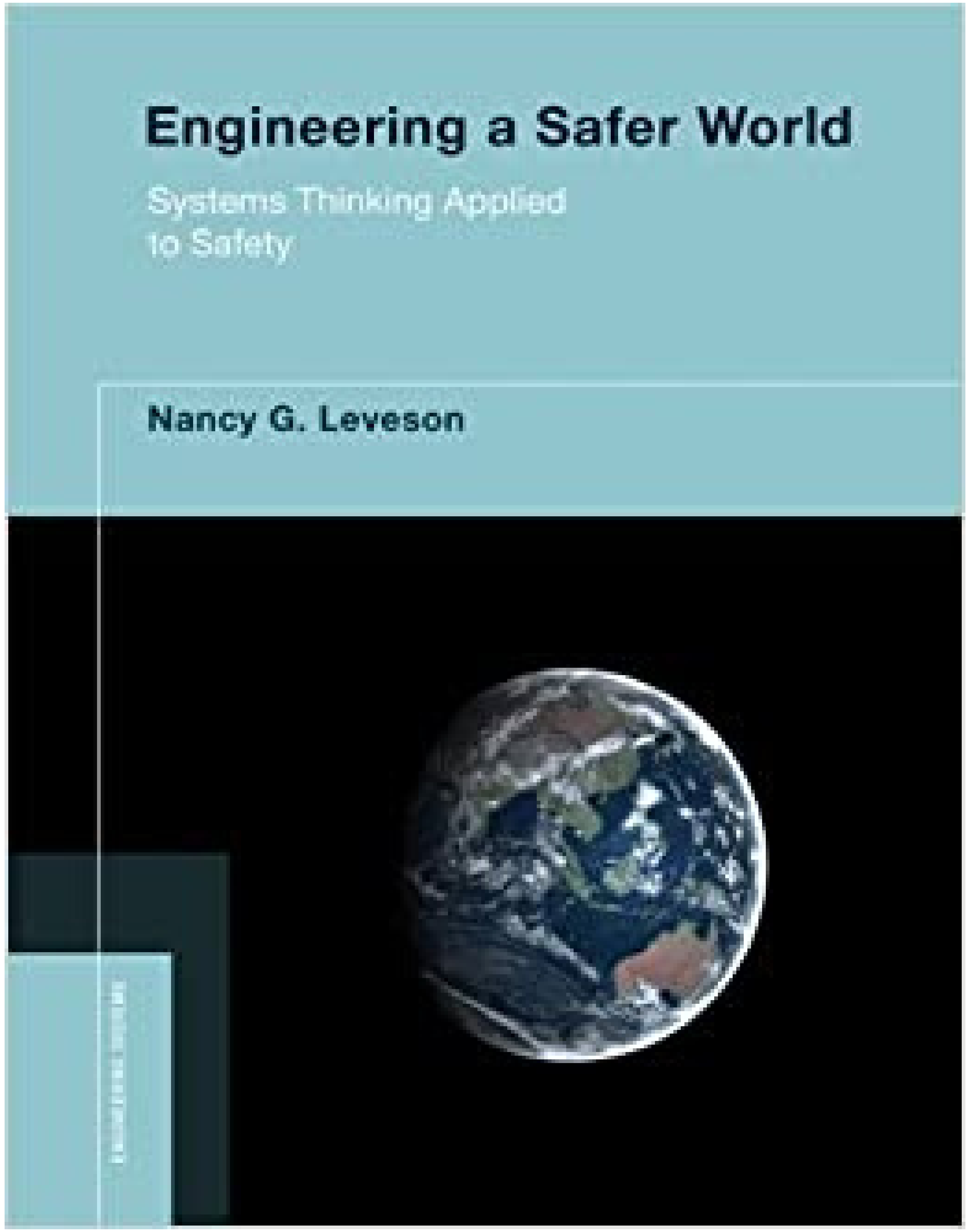
Accident

ROOT CAUSE ANALYSIS



FEEDBACK LOOP

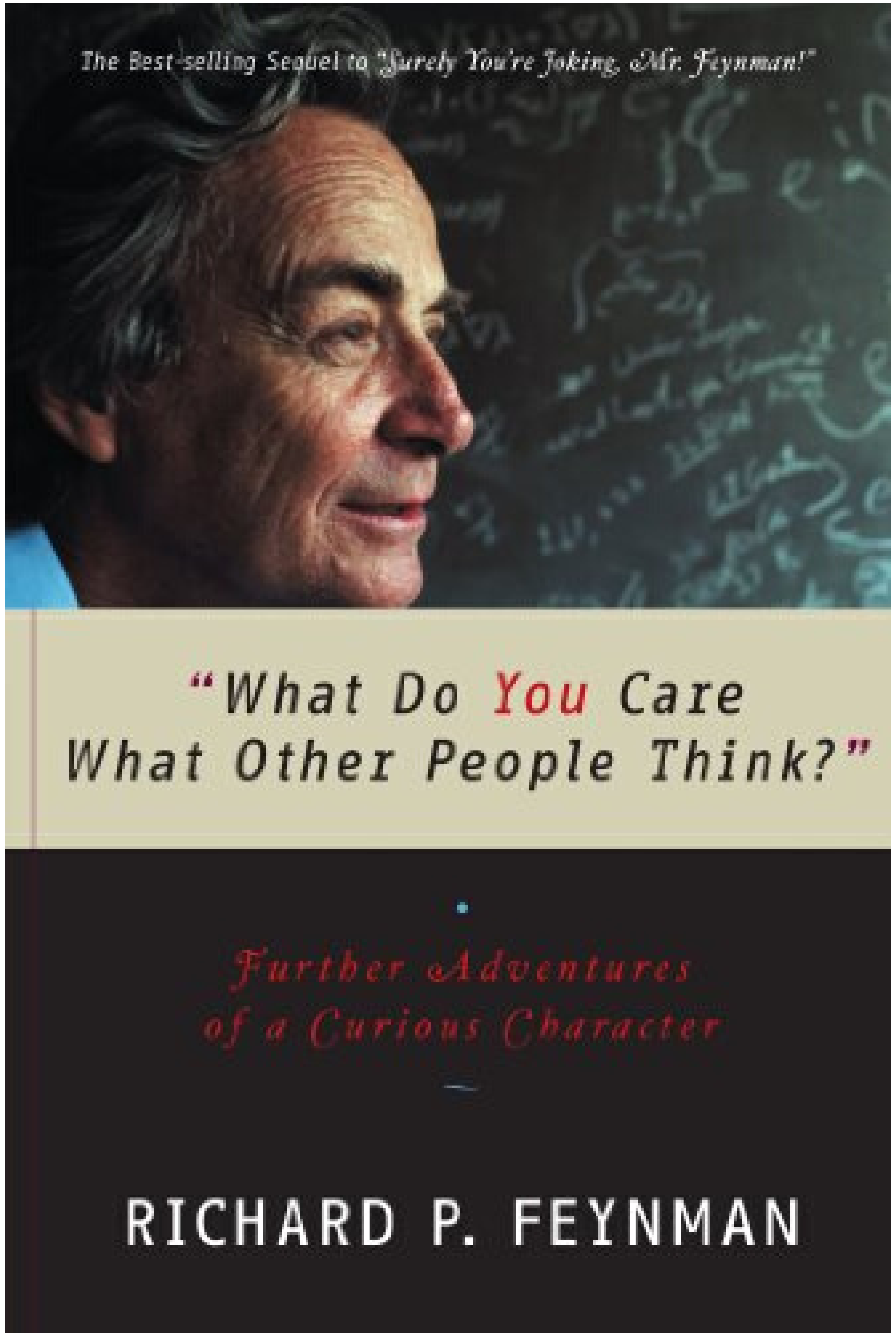




Engineering a Safer World

Systems Thinking Applied to Safety

Nancy G. Leveson



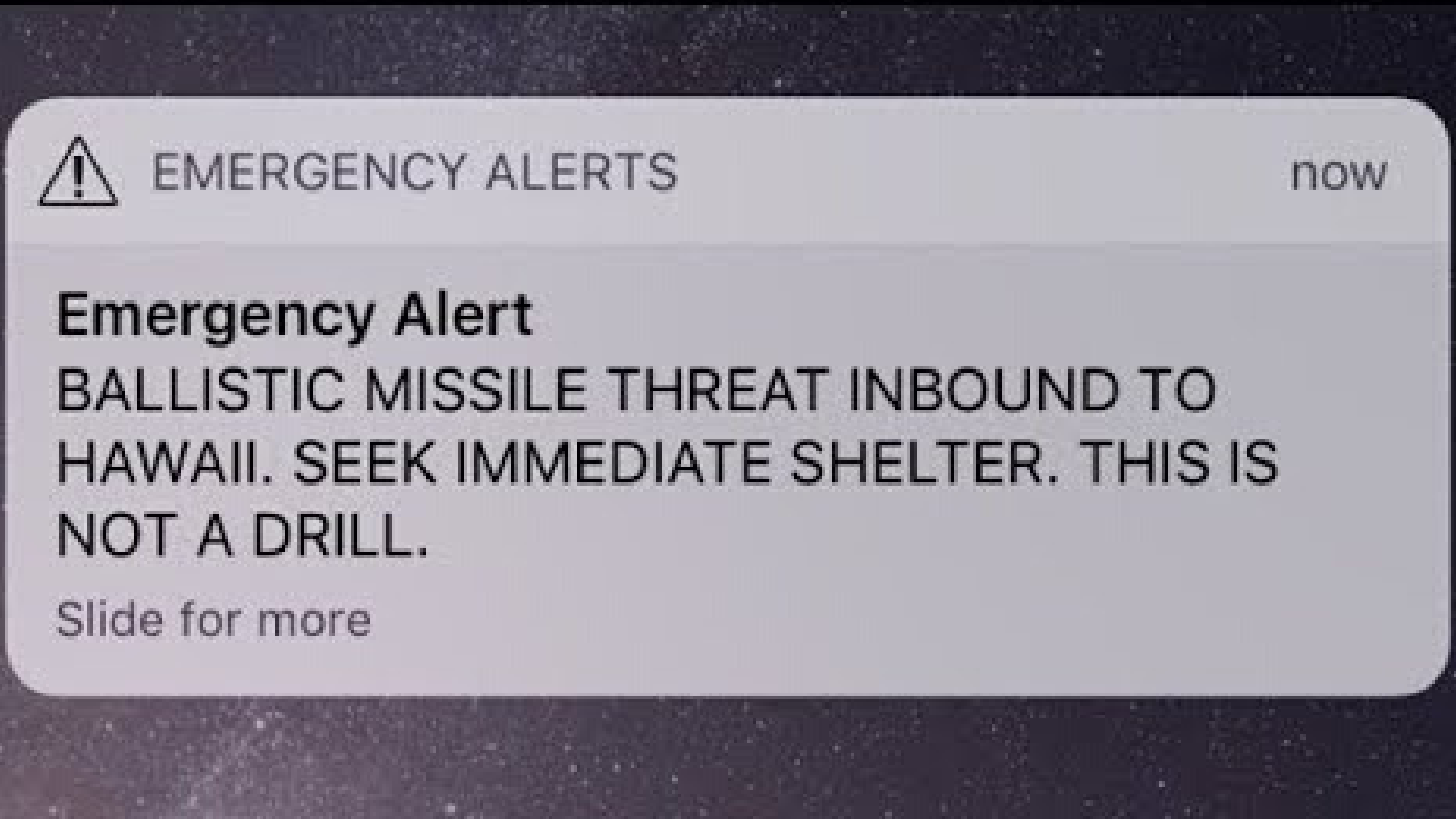
The Best-selling Sequel to "Surely You're Joking, Mr. Feynman!"

"What Do You Care What Other People Think?"

Further Adventures of a Curious Character

RICHARD P. FEYNMAN

737
737 MAX
BOEING
737 MAX
39596



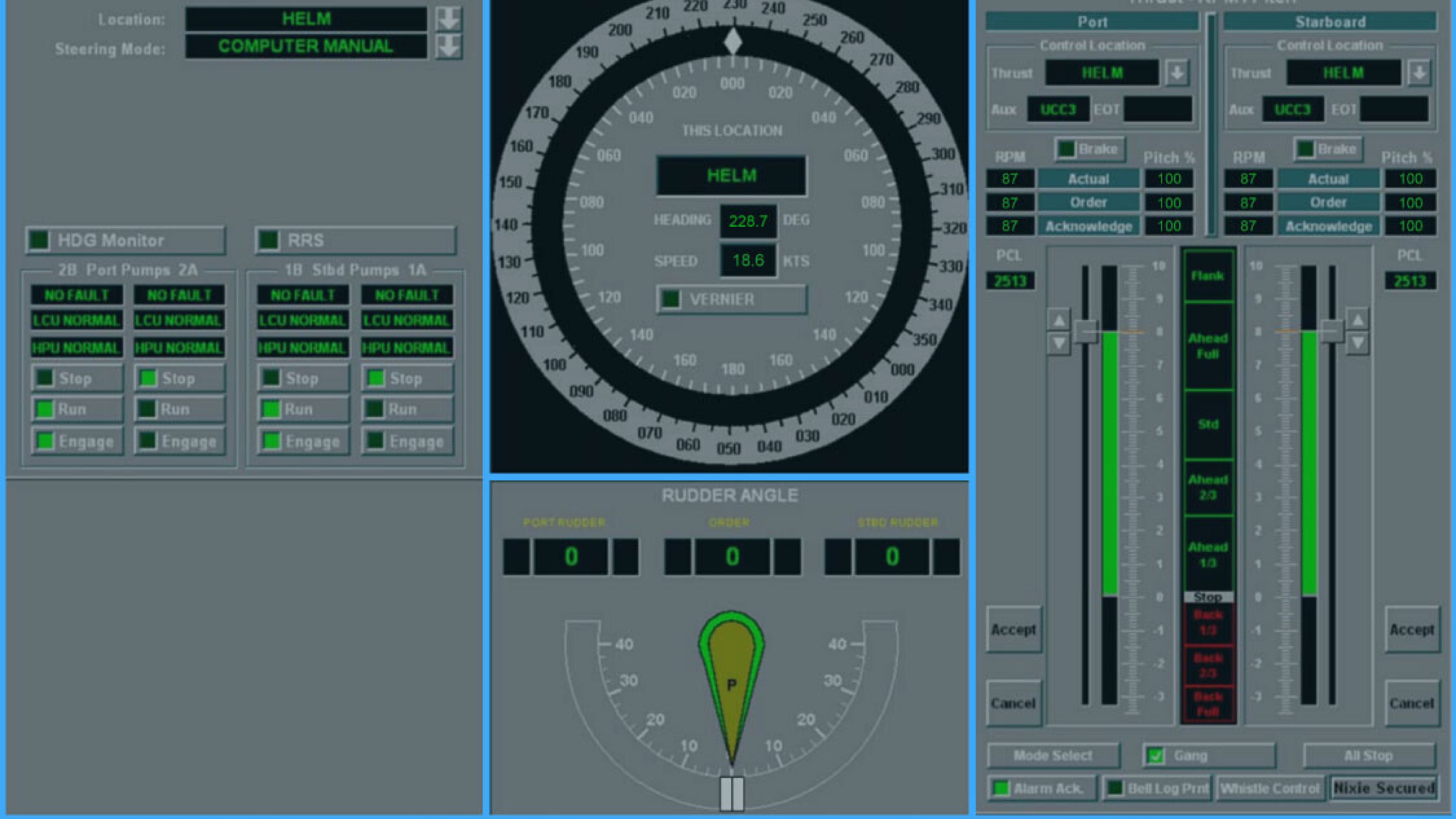
⚠ EMERGENCY ALERTS now

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

2B Port Pumps	2A	1B Stbd Pumps	1A
NO FAULT	NO FAULT	NO FAULT	NO FAULT
LCU NORMAL	LCU NORMAL	LCU NORMAL	LCU NORMAL
HPU NORMAL	HPU NORMAL	HPU NORMAL	HPU NORMAL

HELM

HEADING 228.7 DEG

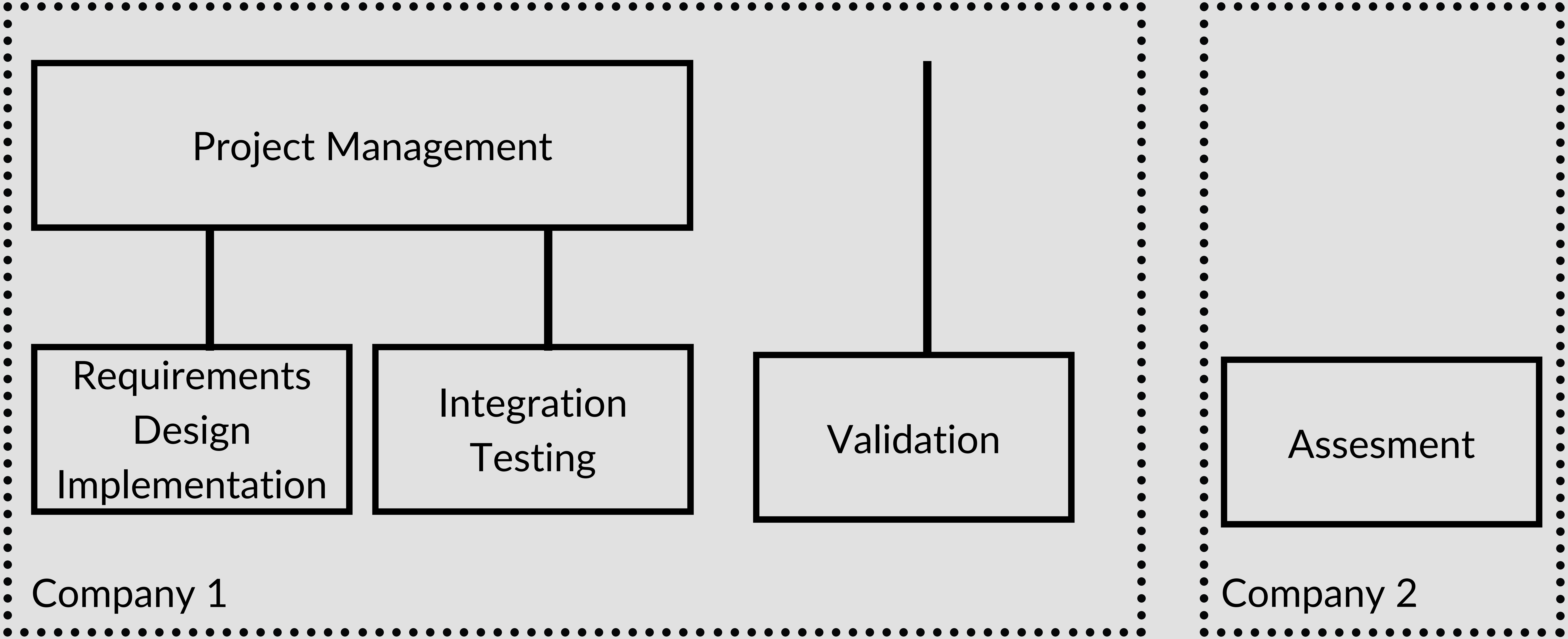
SPEED 18.6 KTS

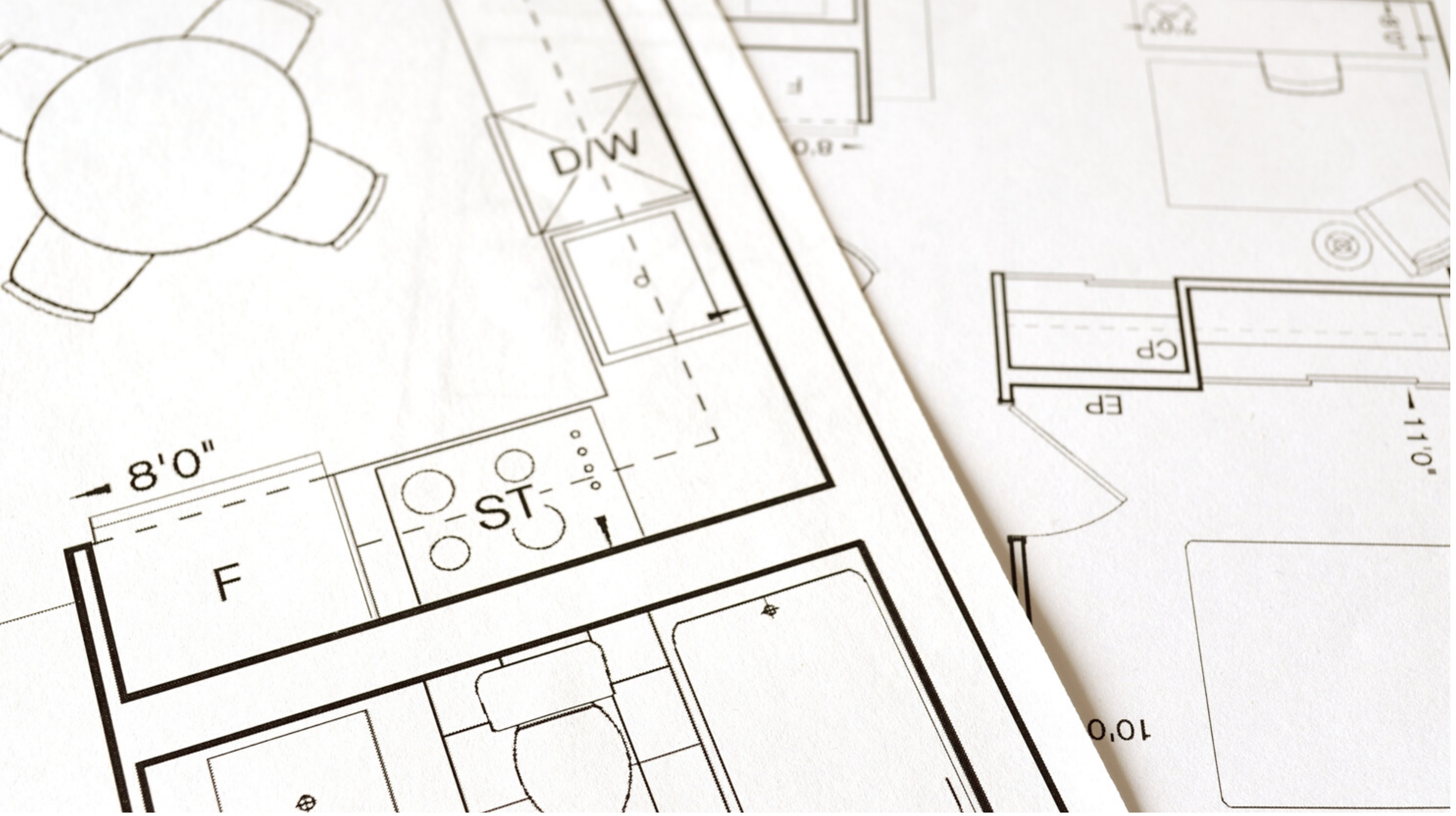
PORT RUDDER	ORDER	STBD RUDDER
0	0	0

	Port	Starboard
Control Location Thrust	HELM	HELM
Aux	UCC3	UCC3
EOT		
PCL	2513	2513

Port RPM		Port Pitch %	Starboard RPM		Starboard Pitch %
87	Actual	100	87	Actual	100
87	Order	100	87	Order	100
87	Acknowledge	100	87	Acknowledge	100

PROJECT ROLES - SIL4





Solw'IT Let's Solve It

THANK YOU

https://ucgosu.pl/slides-ndc-oslo-2020

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