2022
DESIGN AND VERIFICATION™
DVCON
CONFERENCE AND EXHIBITION
EUROPE
MUNICH, GERMANY
DECEMBER 6 - 7, 2022

Fault Injection Analysis for Automotive Safety and Security

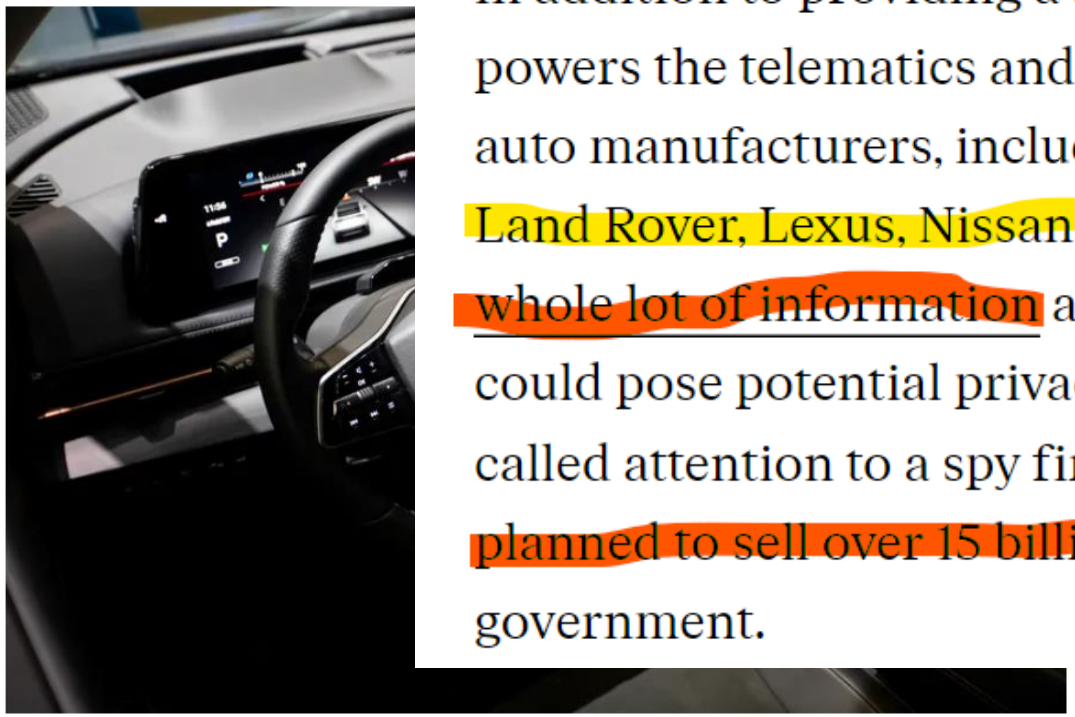
Sesha Sai Kumar C V, Ayman Mouallem, Jamil Mazzawi

Optima Design Automation Ltd.

accellera
SYSTEMS INITIATIVE

Hot from the press…

Sirius XM flaw could've let hackers remotely unlock and start cars



Nissan is just one of the auto manufacturer's that use Sirius XM's connected vehicle services.

In addition to providing a satellite radio subscription, Sirius XM also powers the telematics and infotainment systems used by a number of auto manufacturers, including Acura, BMW, Honda, Infiniti, Jaguar, Land Rover, Lexus, Nissan, Subaru, and Toyota. These systems collect a whole lot of information about your car that's easy to overlook — and could pose potential privacy implications. Last year, a report from *Vice* called attention to a spy firm, called Ulysses, which collected and planned to sell over 15 billion telematics-based car locations to the US government.

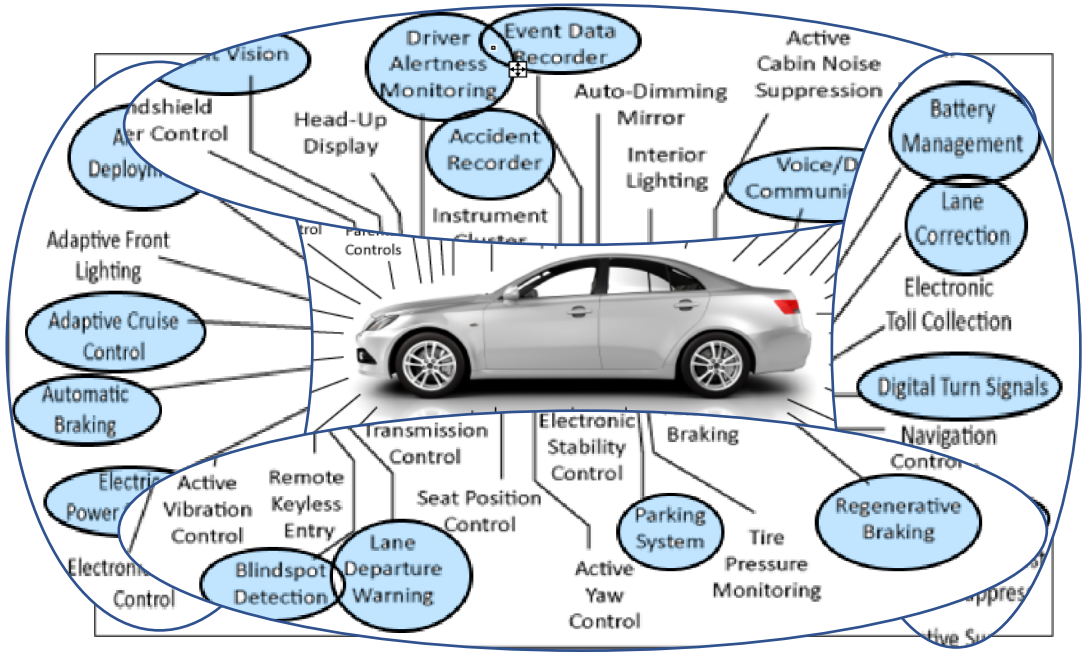
Credits: https://www.theverge.com/2022/12/3/23491259/sirius-xm-hack-remotely-unlock-start-cars

Agenda

- Introduction: Automotive safety and security
- Relevance of Safety and Security for Automotive Use cases
- Optima Safety Platform
- Functional Safety Verification – ISO 26262
- Security Verification

Functional Safety

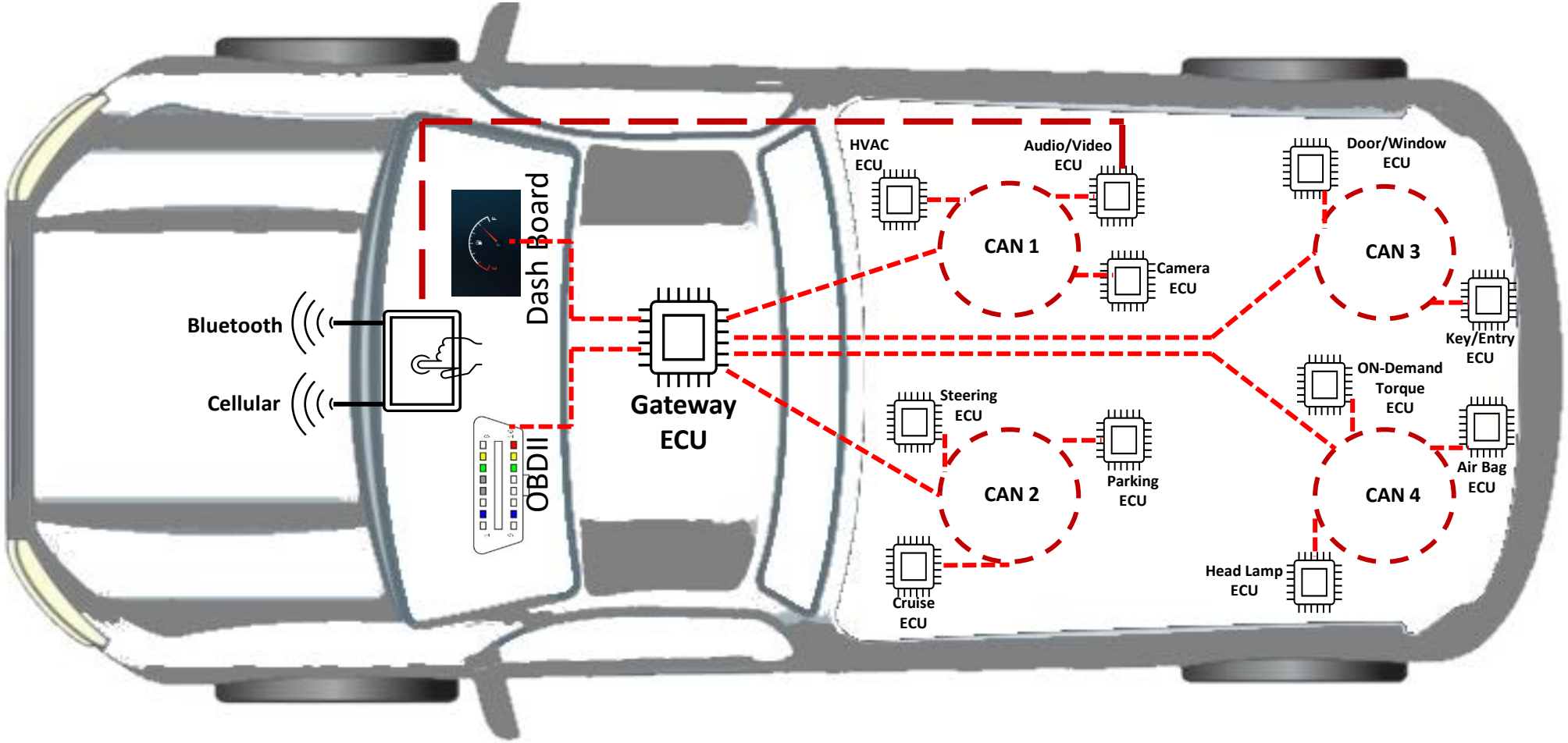
functional safety : ***absence of unreasonable risk due to hazards caused by malfunctioning behavior of E/E systems***



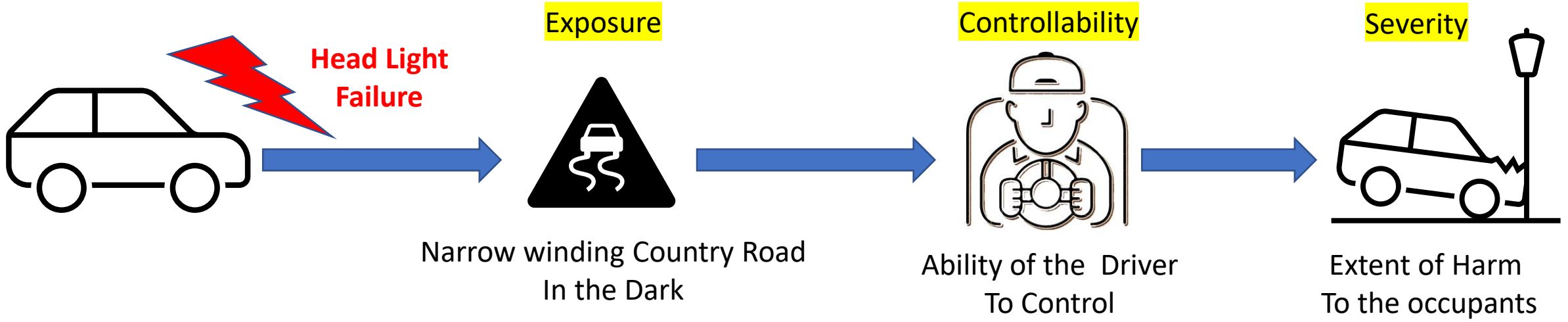
- Average car today uses 300 processors, 50+ complex Electronic Control Units
- ISO 26262 defines automotive safety standards

Functional Safety Standards						
DO-254	IEC 61508					
	ISO 26262	IEC 61511	IEC 62061	IEC 61513 IEC 62138	IEC 62404 ISO 13485	EN 50128
Areospace Defense	Automotive	Industrial Controls	Machine Tooling	Nuclear Power	Medical Devices	Railway Transport

ECUs in the Automobile System



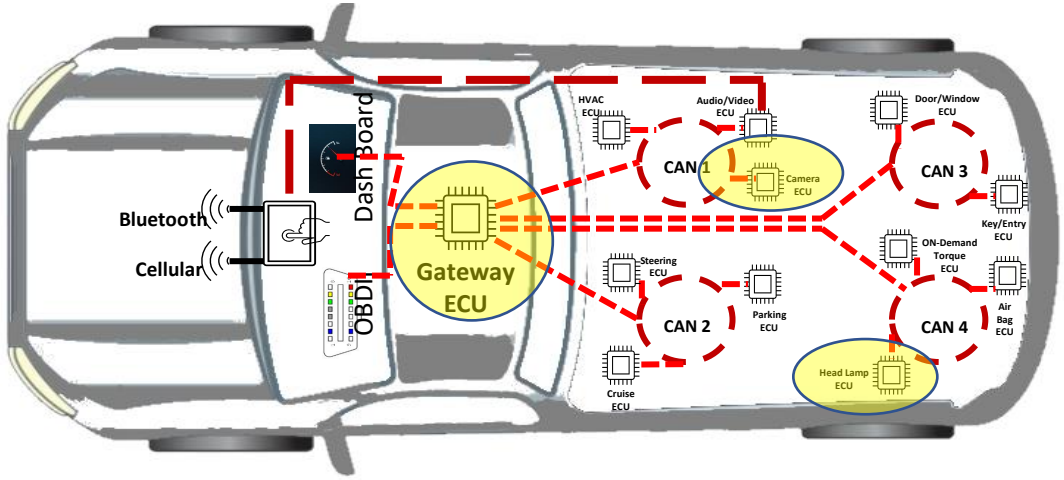
Risk Assessment for Safety: ASIL



Use Case: Head Light Failure while driving Narrow Country Road in the dark

Head Lamp failure: Safety Perspective

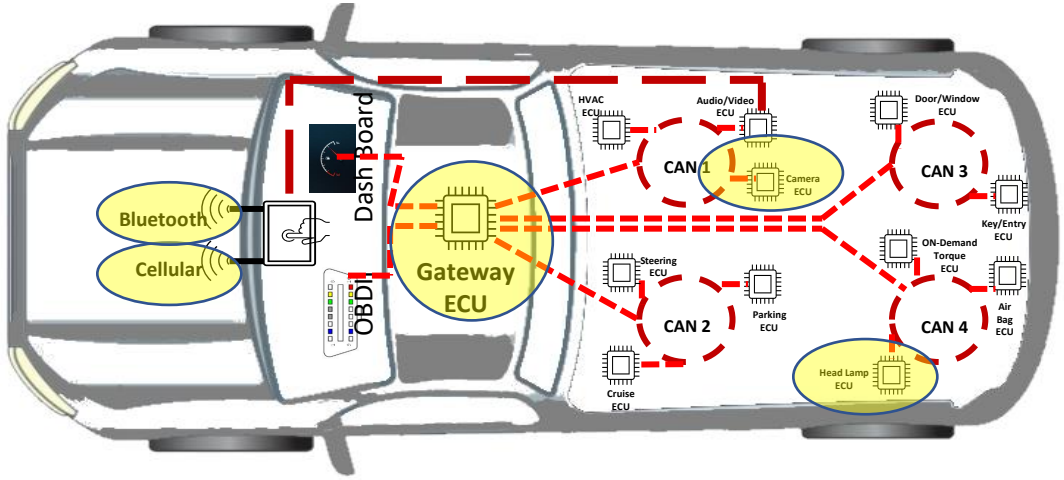
- Driver Requests Head Lamp On/Off
- Head Lamp ECU sends a request via CAN
- Camera Sensor can send Request based on front vehicle Distance.
- Camera ECU sends a request through Gateway ECU
- Safety Consideration
  - ASIL Level for Head Lamp and Camera ECUs based on Exposure X Controllability X Severity
  - Safety Mechanism in MCU, to mitigate loss of Head lamp Switch actuator signal
  - Safety Mechanism in MCU, to mitigate incorrectness of camera sensor signal



Do We need to consider **Security**, in this Use case?

Head Lamp failure: Security Perspective

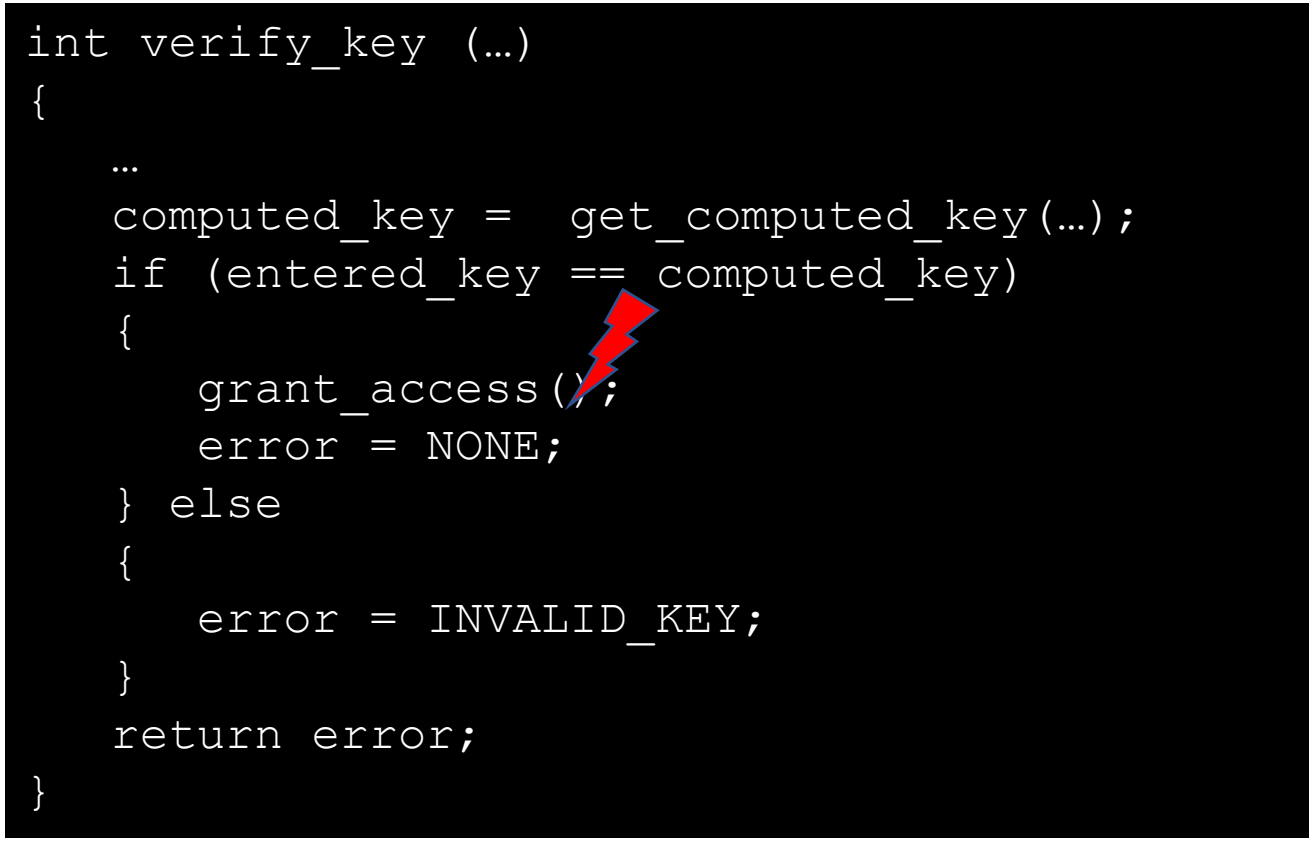
- Gateway ECU
  - Bridge between Camera CAN1 and Headlap CAN4
- Attacker compromises the Navigation ECU
  - Through cellular or Blue tooth
- Cellular
  - Sends malicious CAN signal to turn off Head Lamp ECU via Gateway ECU
- Blue Tooth
  - Flood the CAN bus with High priority CAN messages via Gateway ECU to get Lamp on signal ignored



- Security Consideration
  - Fault Injection Attacks, on the hardware (MCU) has become rampant for unauthorized access
  - Each ECU/MCU to be design with Security in mind with countermeasures
  - Embedded Software need to be tested by injecting faults in
    - ECU authentication and Key calculation logics

Simple Attack to gain Access

- Attacker can directly inject a fault in grant_access();
- Access is granted irrespective of entered key matches or not
- Counter measure
  - Hardened Firmware code
  - Robust security countermeasures in MCU design



Fault Injection Modelling: Safety And Security

- Safety:
  - Hard Errors: Permanent Faults modelled as S@0 and S@1
  - Soft Errors: Transient Faults modelled as Single Event Upset (SEU)
- Verify Safety Mechanisms by injecting Faults
  - ISO 26262 Recommends Fault Injection simulation to verify
- Security
  - Clock and Voltage Glitches
  - EM and Laser beam Attacks: Multiple SEU faults to model attacks
- Verify Security Vulnerabilities by way of FAS (Fault Attack Simulation)

OPTIMA

Enabling the highest Functional Safety and IC-Security levels at fraction of effort

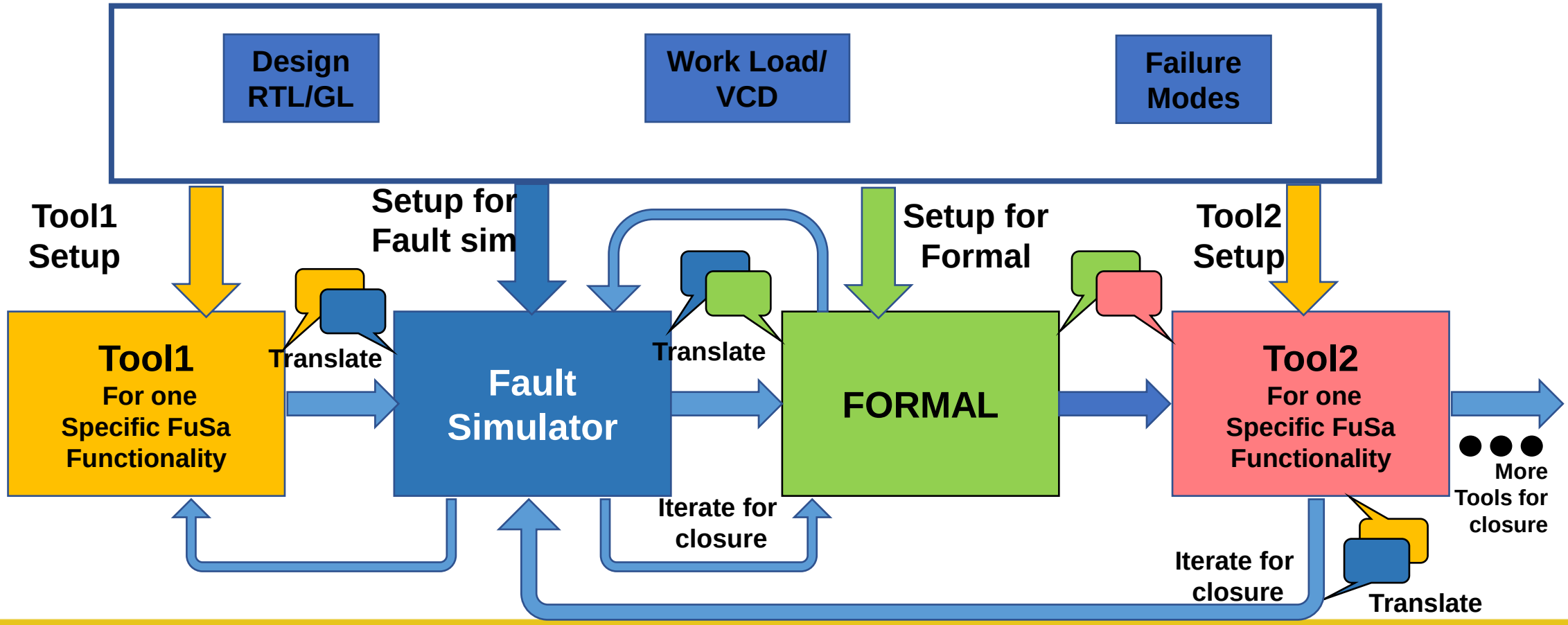
Founded 2014 Nazareth, Israel

Optima-SA™	Optima-HE™	Optima-SE™	Optima-SEC™
Static Analysis Solution	**Hard Error Coverage: Measure & Boost**	**Soft Error / Transient faults mitigation**	**IC Security verification**
Accurate failure-mode-sizing for FMEDA, early identification of Safety issues	Rapid fault coverage measurement with automated coverage boosting	Automated FIT rate reduction to ASIL-D with minimal silicon cost	Automated fault-simulation solution for side-attack protection verification

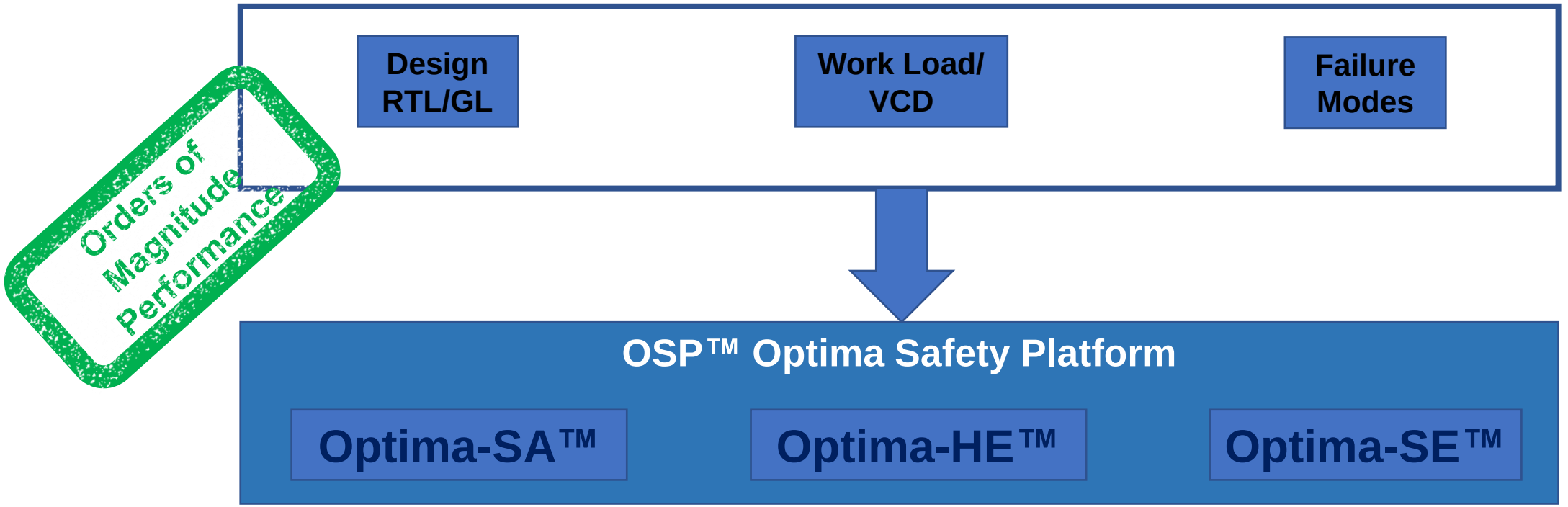
Optima Fault Injection Engine™ (FIE) Technology

Orders of magnitude faster than competing solutions

Safety and Security Verification: Many tools



Unique Fault analysis Platform



Single Platform for Fault Injection Analysis
For Both Safety and Security

2022 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE

MUNICH, GERMANY

DECEMBER 6 - 7, 2022

Functional Safety Verification ISO 26262

accellera SYSTEMS INITIATIVE

ISO26262 Requirements Simplified

- Semiconductor Chips need to have
  - SM :“Safety Mechanisms”
    - Detect and/or correct faults
    - Within the budgeted time interval (e.g., 0.25ms to 100ms) from the time they happen
  - SM needs to be running continually while the device is operating
- SM diagnostic coverage: Quantitative Analysis
  - The SM needs to be able to cover no less than N% of the possible faults
  - Different ASIL levels have different N values
  - For example: for ASIL-D N>99%

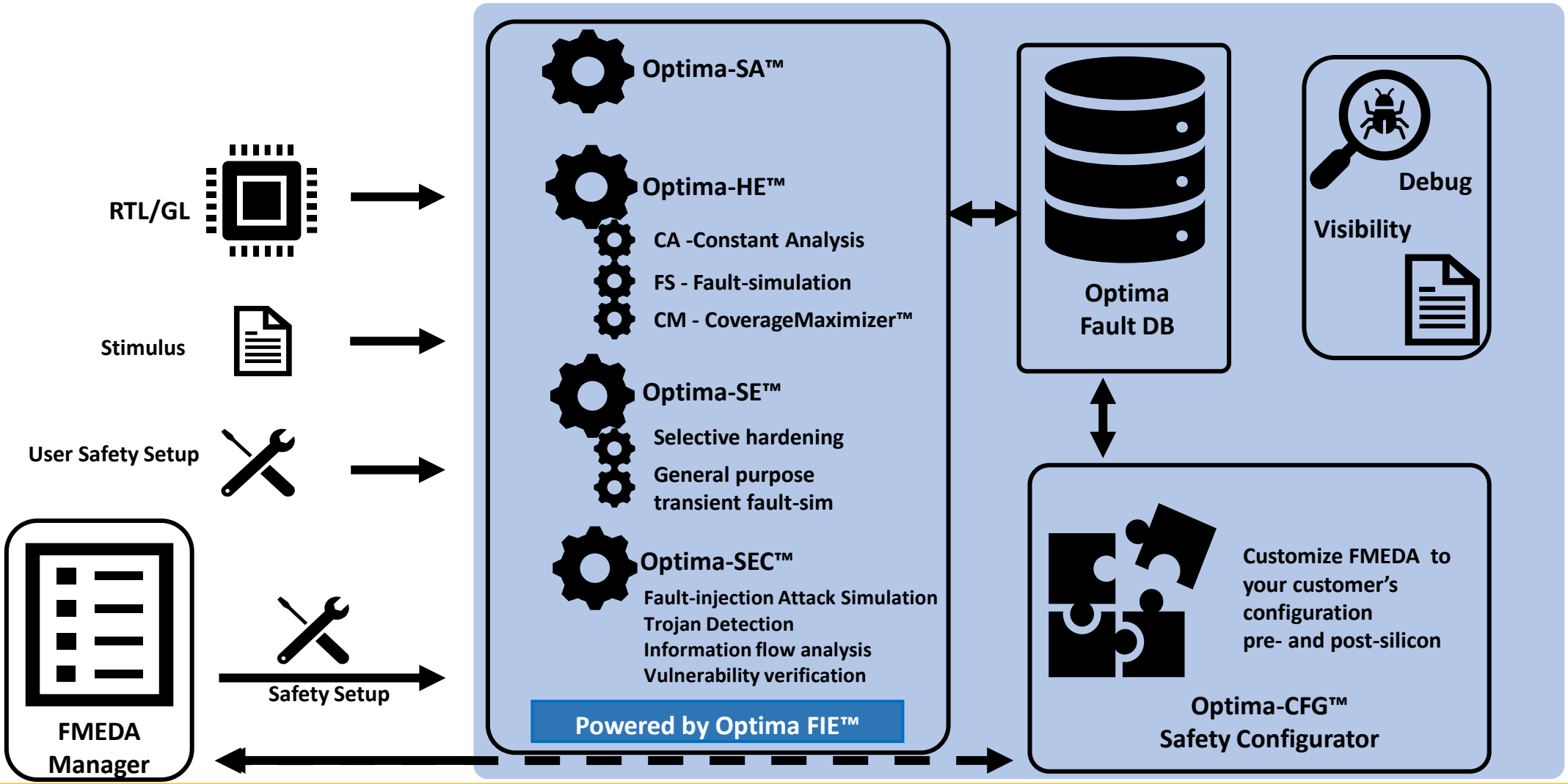
ISO 26262-11:2018 4.8.1

For a Safety Concept with Semiconductor Components, Fault Injection is the Known Methodology for various safety lifecycle activities.

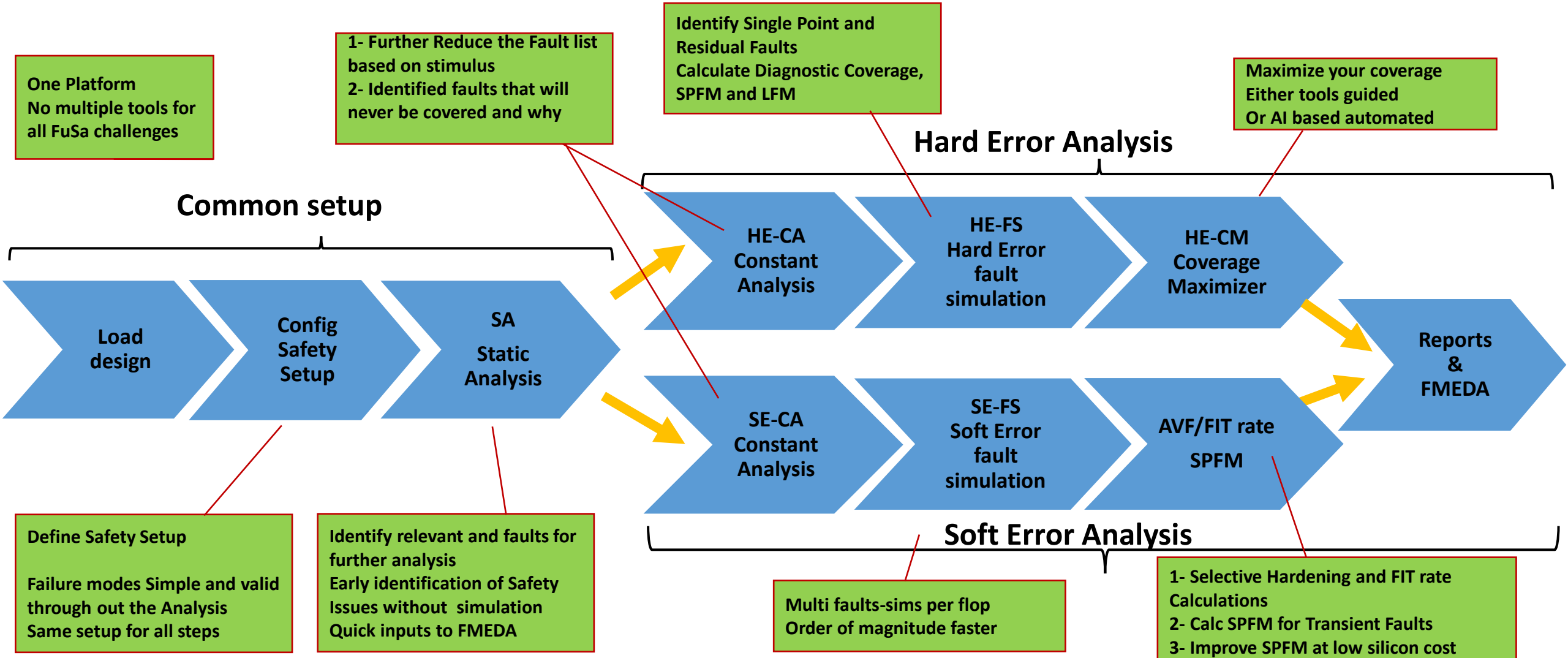
Fault Injection Objectives (ISO 26262-11:2018 4.8)

- Diagnostic Coverage
- Fault Tolerant Time Interval
- Fault Reaction Time Interval
- Architectural Metrics (SPFM, LFM, FIT…)
- Fault Effect and Safety Mechanism Mitigation

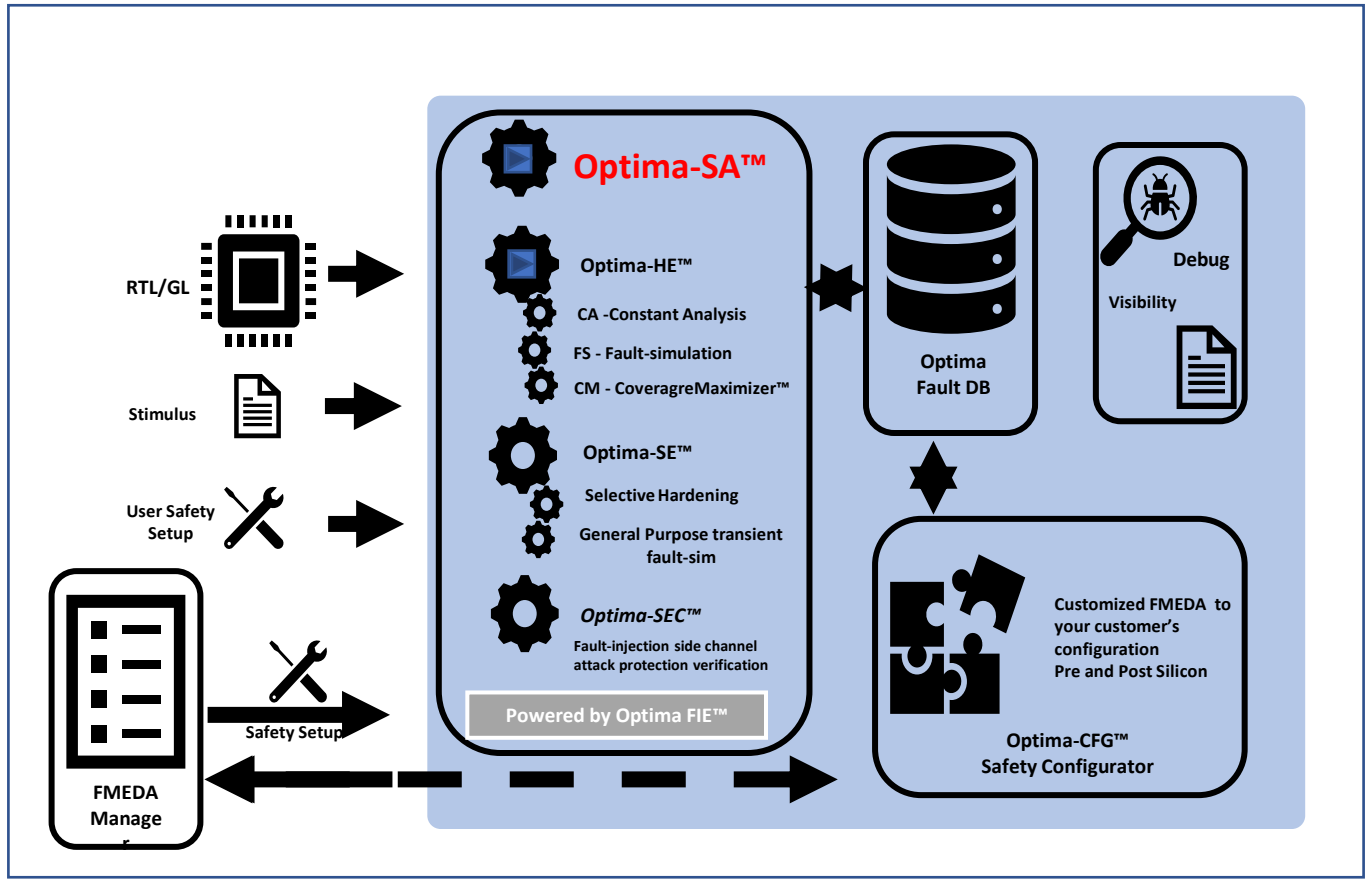
Optima Safety Platform™

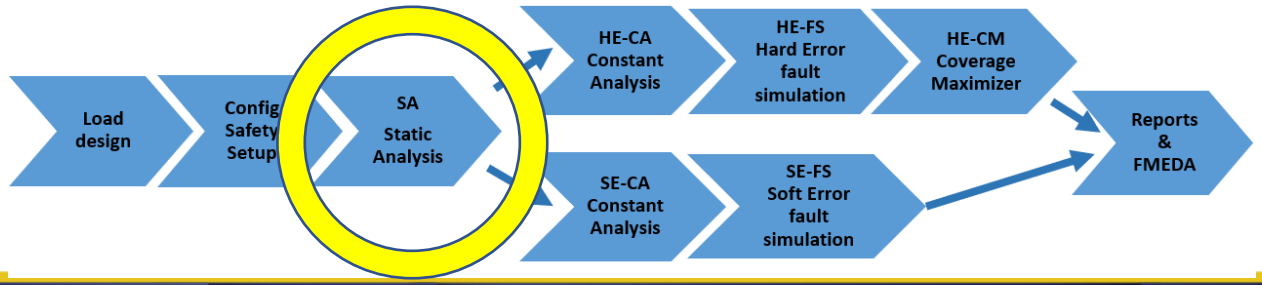


OSP flows for Hard and Soft error analysis



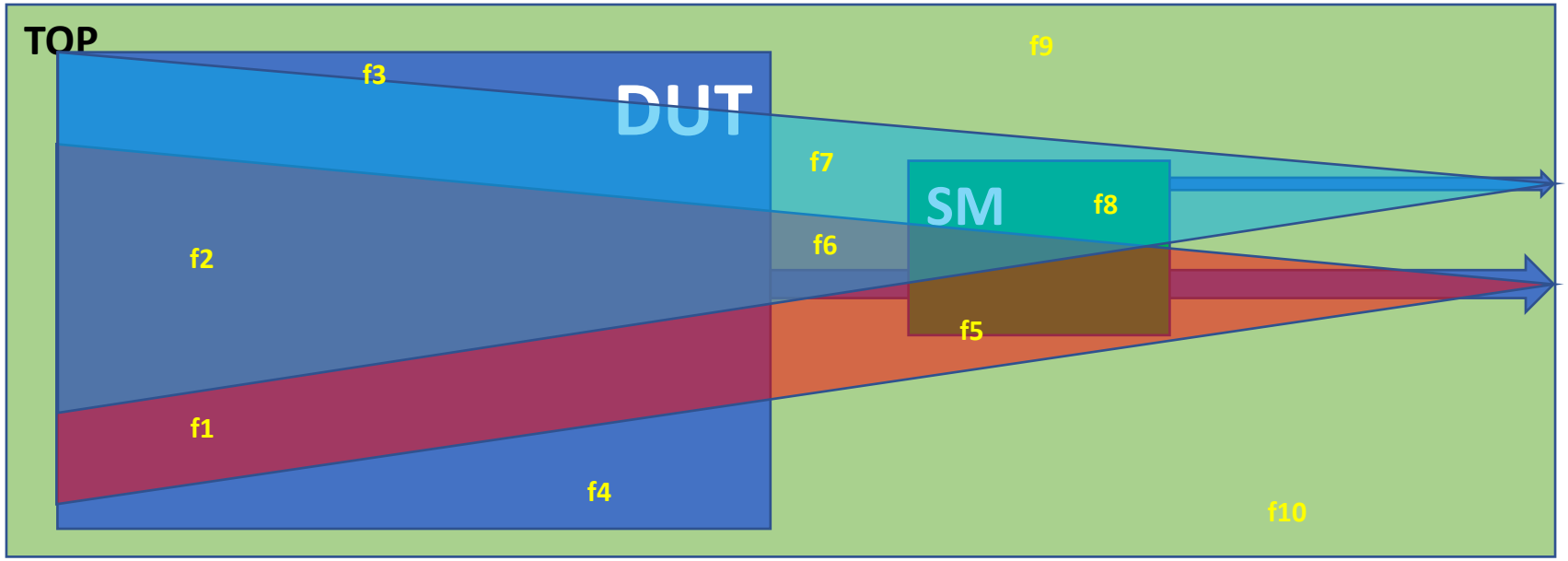
Optima-SA™ Static Analysis





Analyze Only Relevant faults

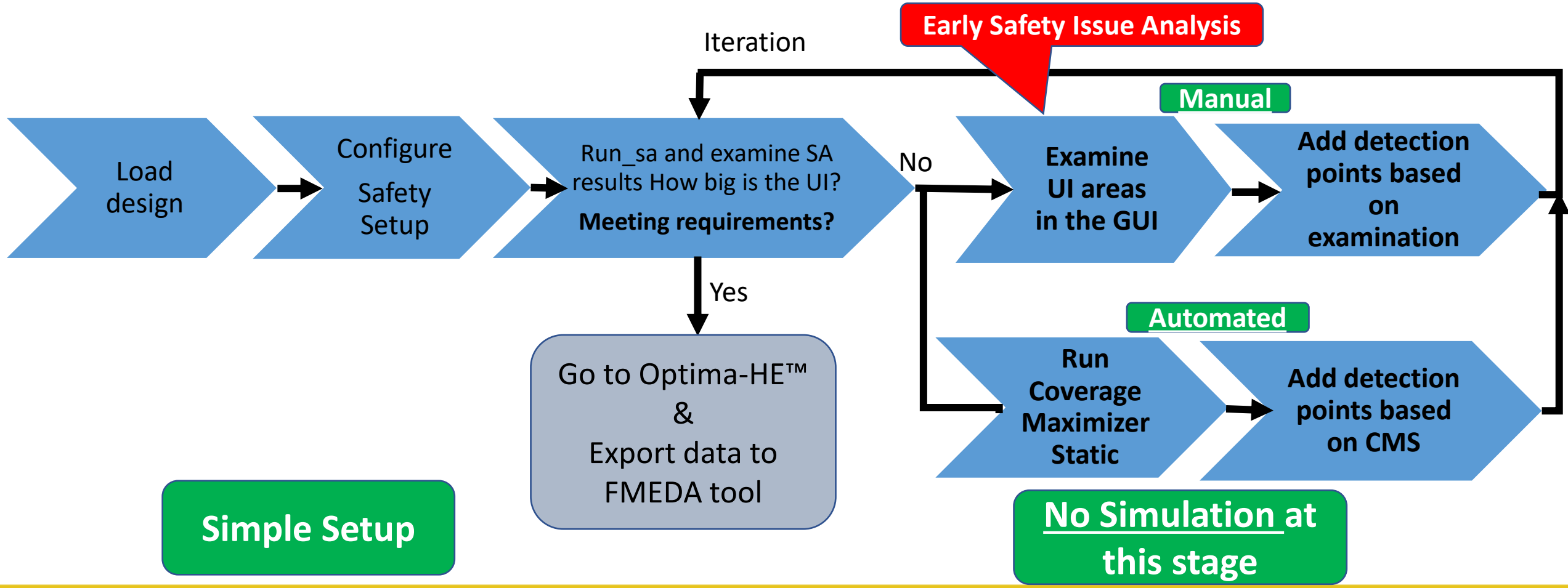
- Static Analysis based on the Failure and Detection Strobes
- COI is established and only relevant faults considered
- For safety critical DUT
  - NO Need to Simulate all f1 to f10 and then decide relevant faults
  - Only f1 and f2 are relevant to analyse further



Save Debug time to identify relevant faults

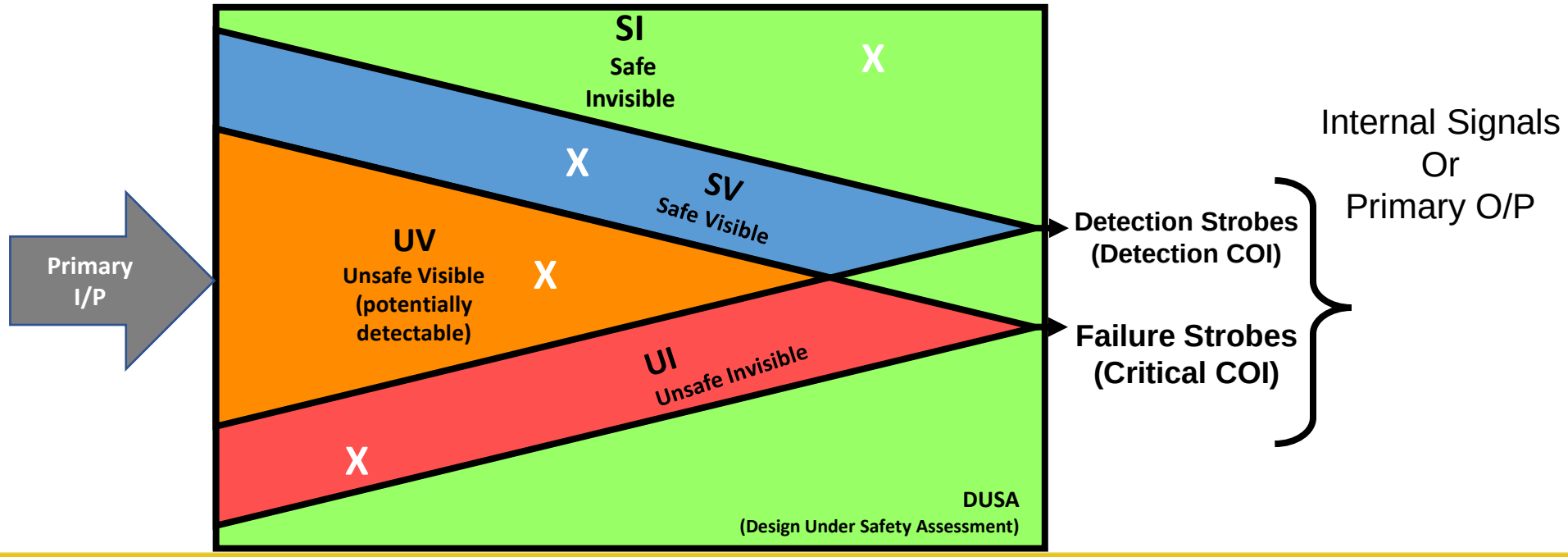
NO Complex setup for Constraining Faults

Optima-SA™ Flow



Safety Setup and Static Analysis

- Safety Setup: Simple way to Capture Safety related information
  - Failure Strobes and Conditional Signals
  - Detection Strobes and Conditional Signals



Static Analysis: Fault Statuses

- SI ➔ SAFE INVISIBLE
  - SAFE: NOT Observed at Failure Strobe
  - INVISIBLE: NOT Observed at Detection Strobe
- SV ➔ SAFE VISIBLE
  - SAFE: NOT Observed at Failure Strobe
  - INVISIBLE: Observed at Detection Strobe

Only Relevant faults to be Analysed

UI ➔ UNSAFE INVISIBLE No Safety Mechanism (SPF)

- UNSAFE: Observed at Failure Strobe
- INVISIBLE: NOT Observed at Detection Strobe

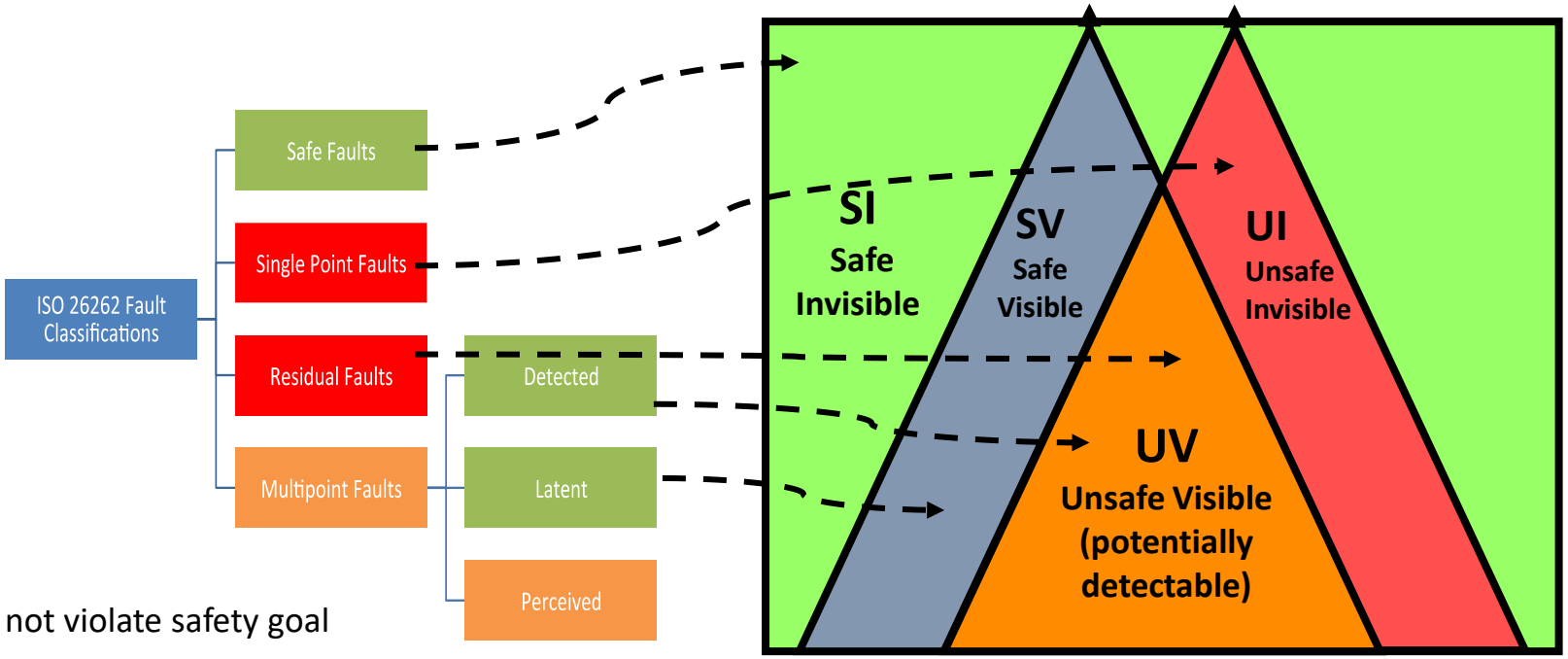
UV ➔ UNSAFE VISIBLE Possible Detects (Diagnostic Coverage)

- UNSAFE: Observed at Failure Strobe
- INVISIBLE: Observed at Detection Strobe

Mapping Static Analysis to ISO2626 Metrics

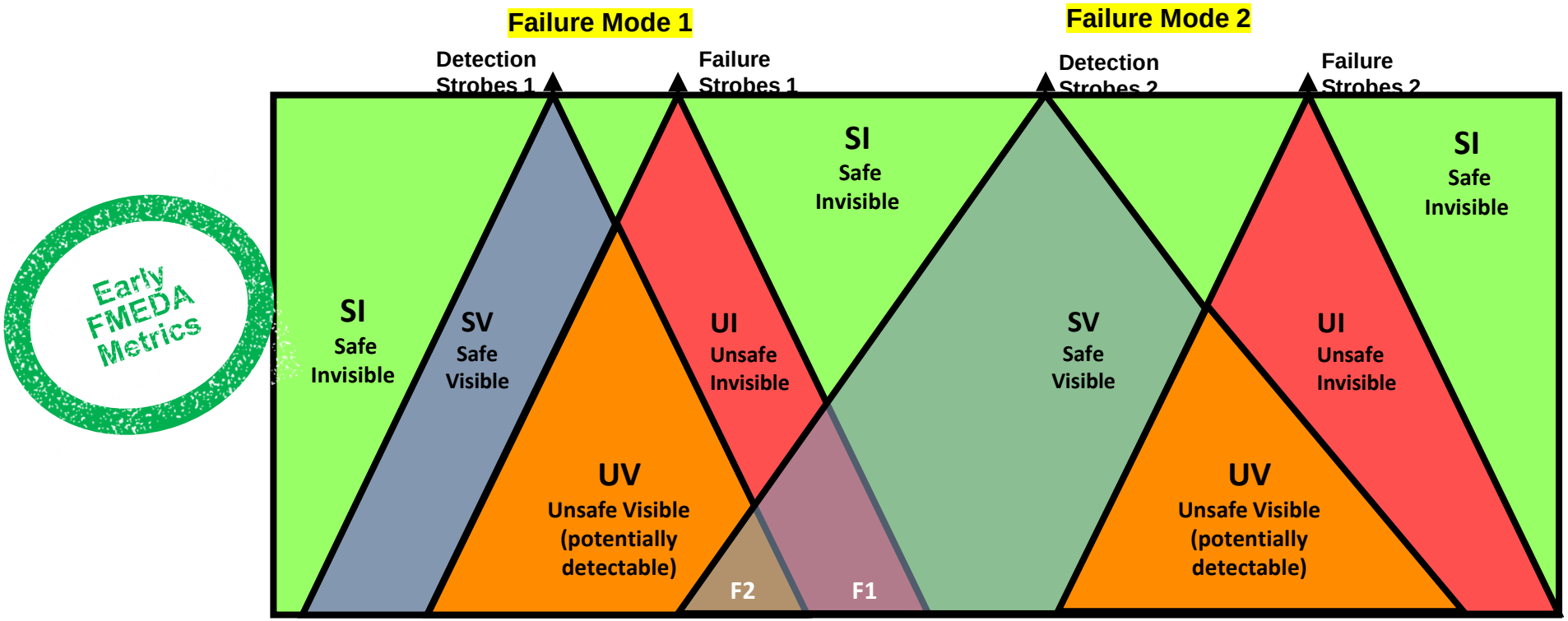
Classified faults are mapped easily and directly

INTUITIVE



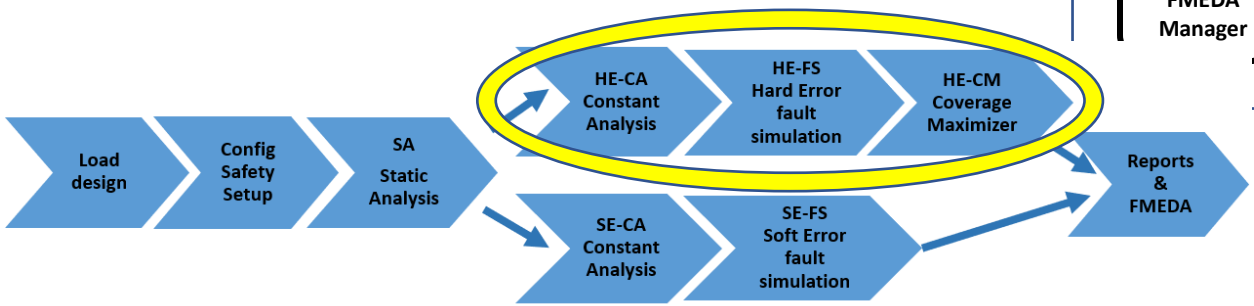
Safe Fault	This will not violate safety goal
Single Point Fault	Fault that violates safety goal and no safety mechanism present
Residual Fault	Fault that violates safety goal and safety mechanism designated is not covering. **Need to simulate UV to find**.
Multipoint Fault - Detected	Combination of more than one fault that violates safety goal. **Need to simulate UV**.
Latent Fault	Alone can not violate safety goal. **Need to simulate SV**.

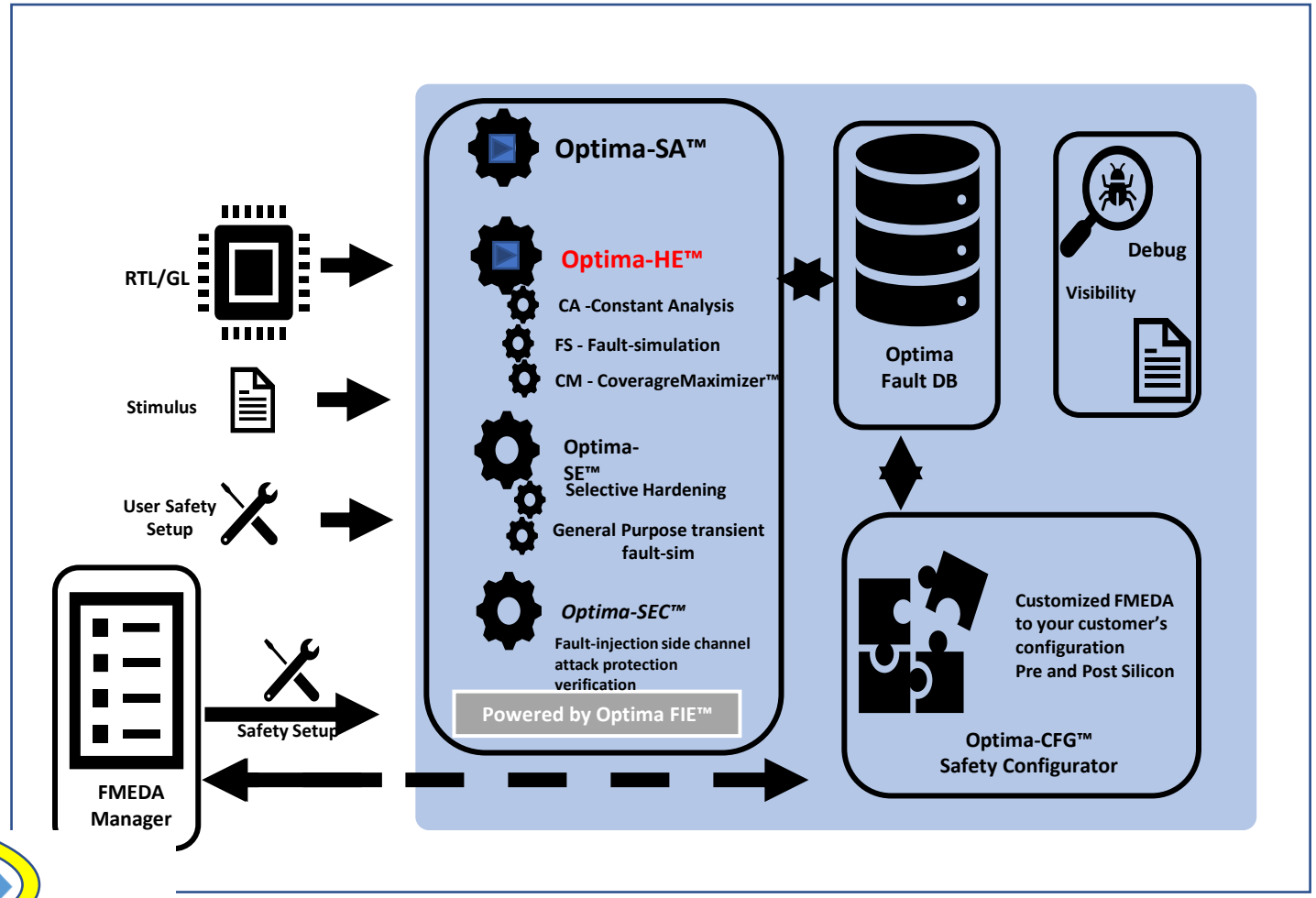
Multiple Failure modes: safety Setup



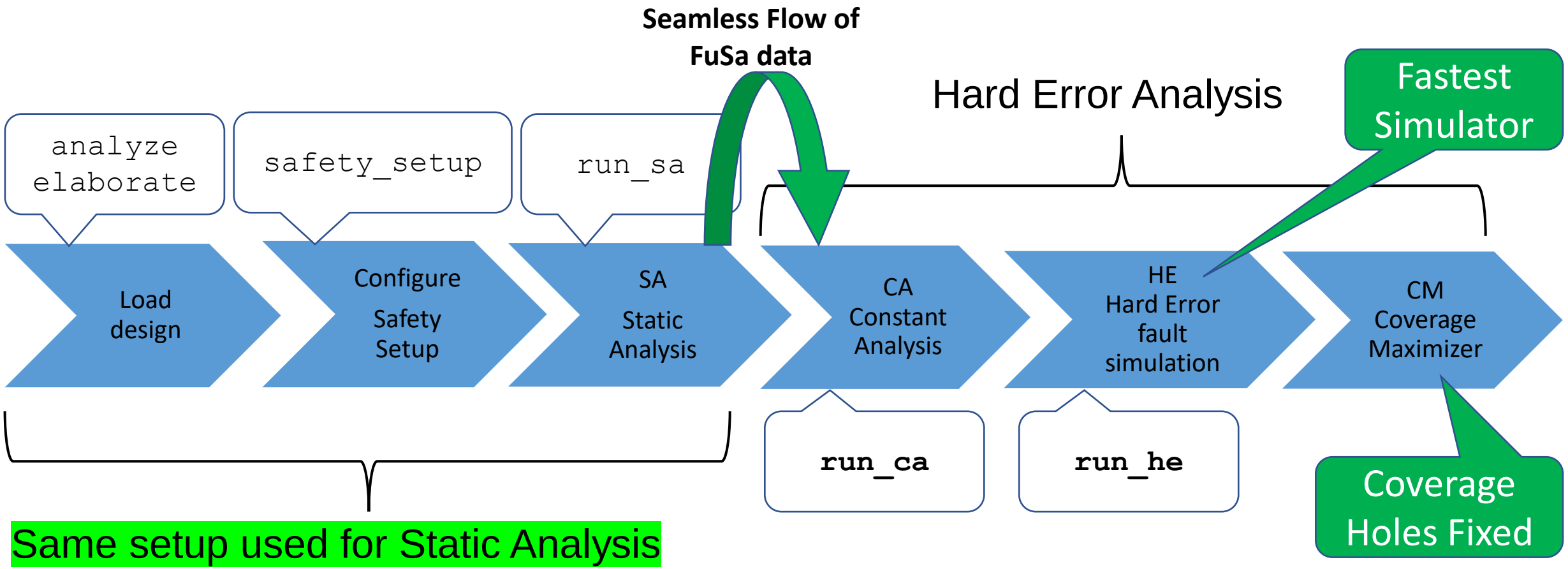
Optima-SA™ Analyzes Faults,
Single, Multiple and Merged Failure Modes

Optima-HE™ Hard Error Fault Simulation

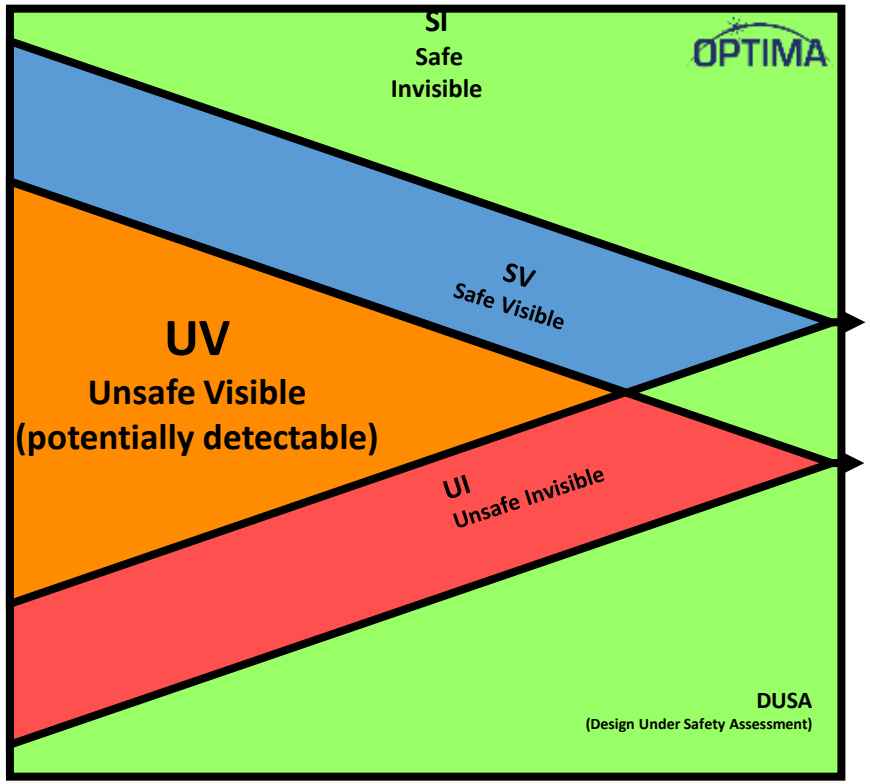




Optima-HE™ Flow

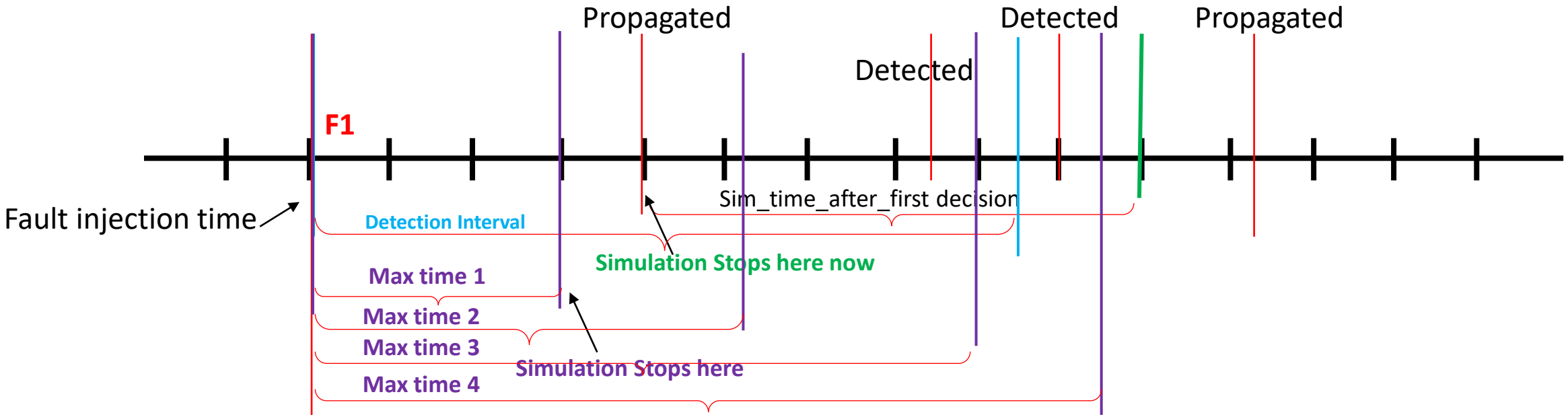


Faults for Optima-HE™



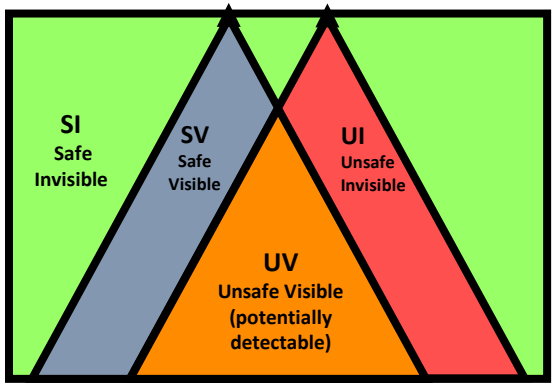
- Relevant Faults
  - Faults after Collapsing
  - Faults Pruned by Optima-SA™ - UV
  - Fault List Optimization by Optima-CA™ Constant Analysis
- Faults analyzed by CA will not be considered
  - Marked BLOCKED/MASKED or SAFE

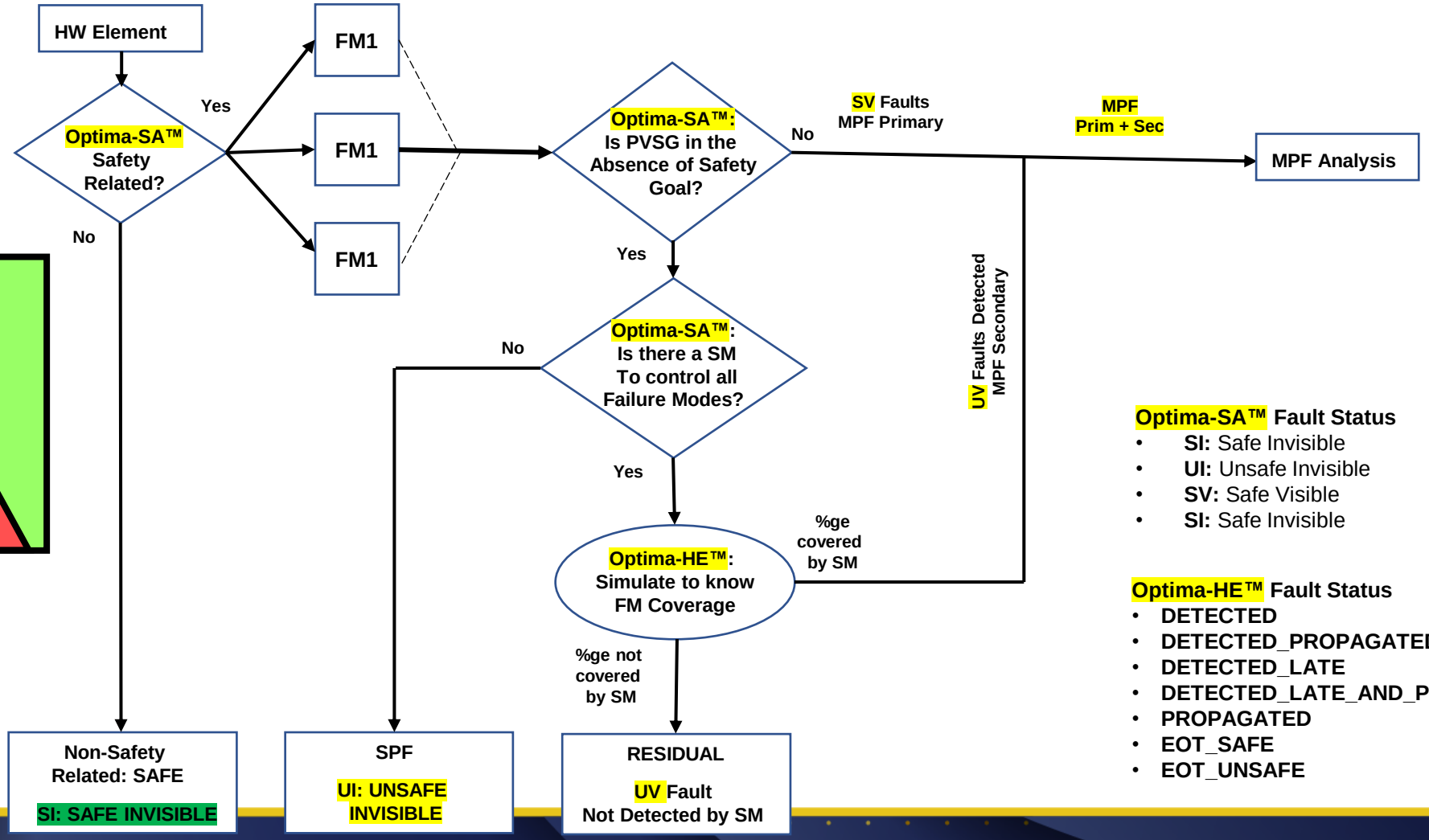
Advanced Fault Simulation Controls



- Max time 1: F1 is EOT_TO
  - no propagation or detection
  - Stops at Max time 1
- Max time 3: F1 is Detected_Propagated
  - Propagated First
  - Simulation continues as set sim_time_after_initial decision
  - Detected Before Detection Interval
  - Stops at Max time 3
- Max time 2: F1 is Propagated
  - Propagated First
  - Simulation continues as set_sim_time_after_initial_decision
  - Stops at max time 2
- Max time 4: F1 is Detected_late_and_Propagated
  - Propagated First
  - Simulation continues as set sim_time_after_initial decision
  - Detected After Detection Interval, late detection
  - Stops at Max time 4

Intuitive Fault Classification: Map to ISO26262





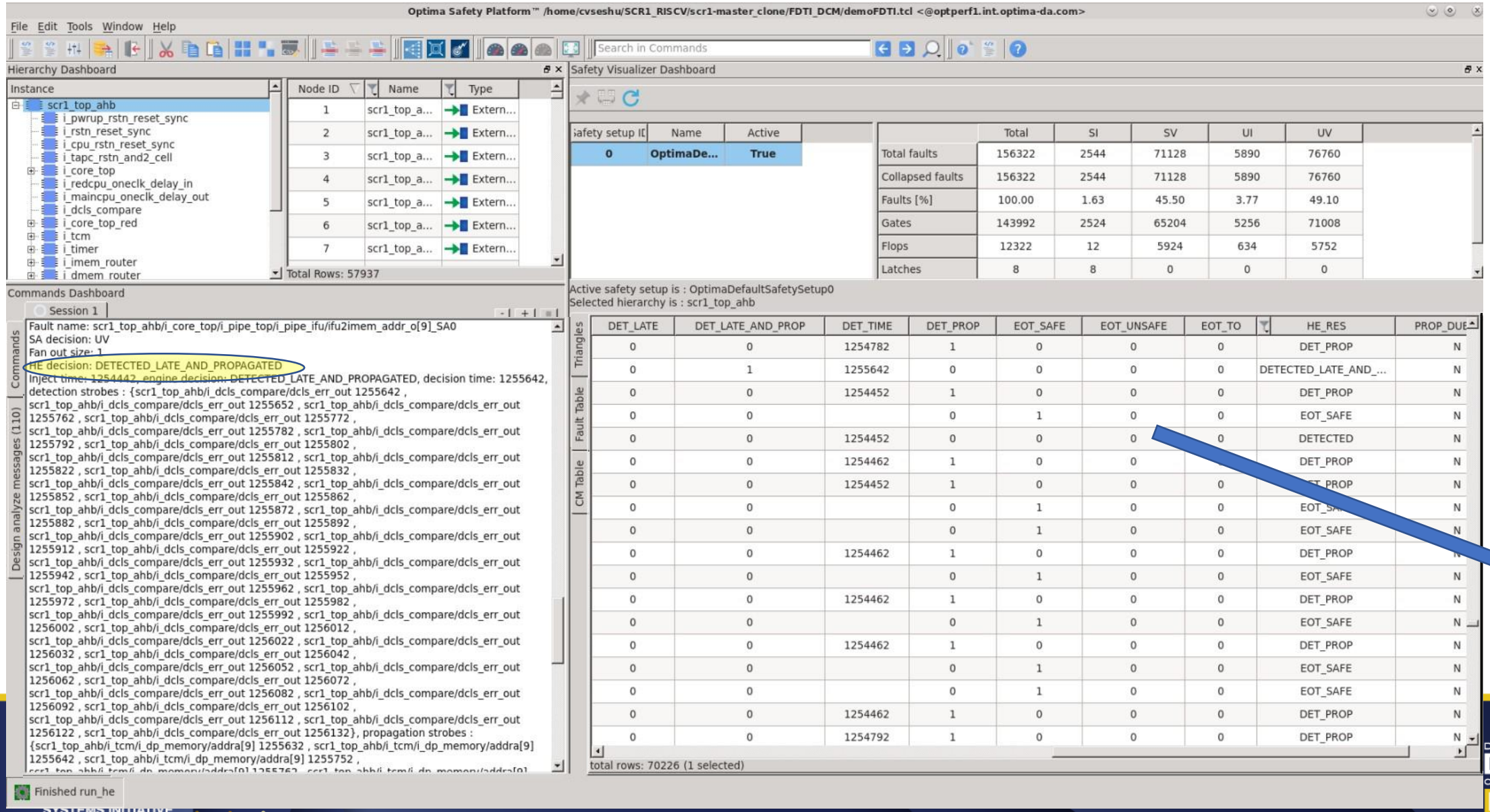
Optima-SA™ Fault Status

- **SI:** Safe Invisible
- **UI:** Unsafe Invisible
- **SV:** Safe Visible
- **SI:** Safe Invisible

Optima-HE™ Fault Status

- **DETECTED**
- **DETECTED_PROPAGATED**
- **DETECTED_LATE**
- **DETECTED_LATE_AND_PROPAGTED**
- **PROPAGATED**
- **EOT_SAFE**
- **EOT_UNSAFE**

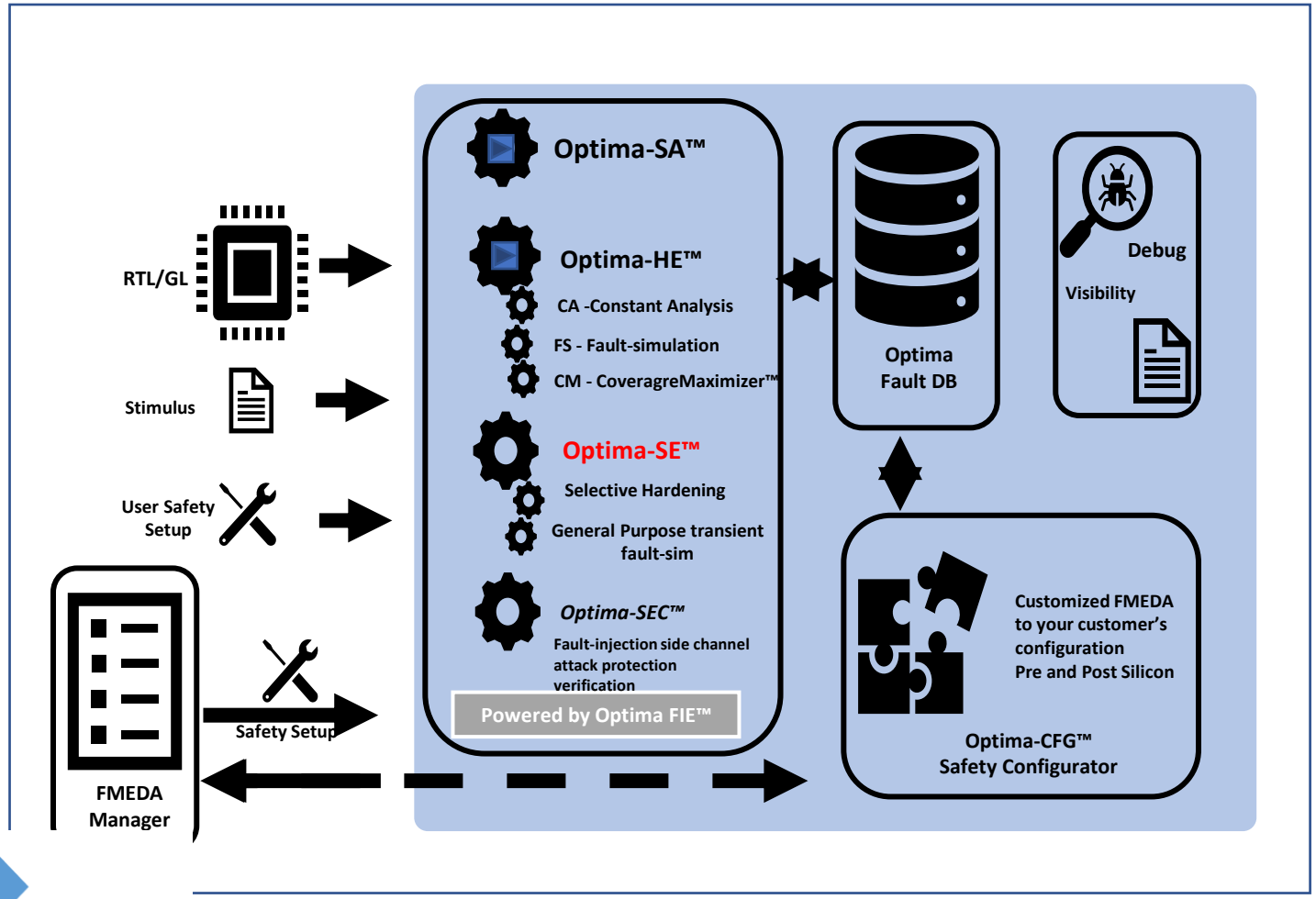
Optima Safety Platform: GUI

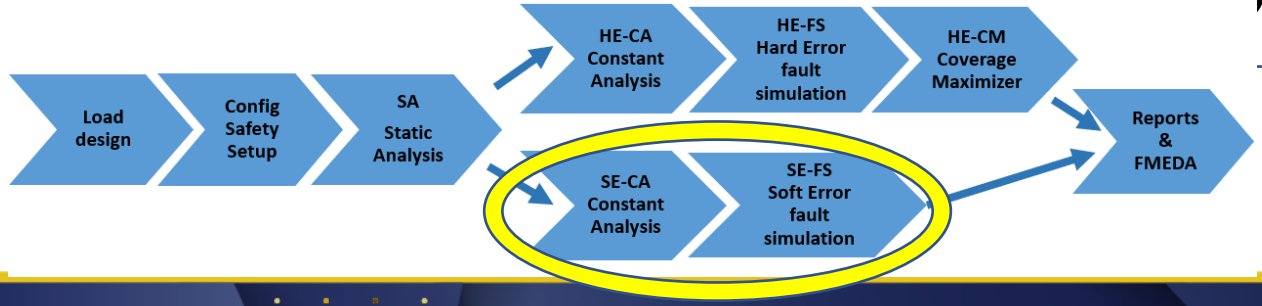


Fault Table Optima-SA™ With all relevant Faults COI Information

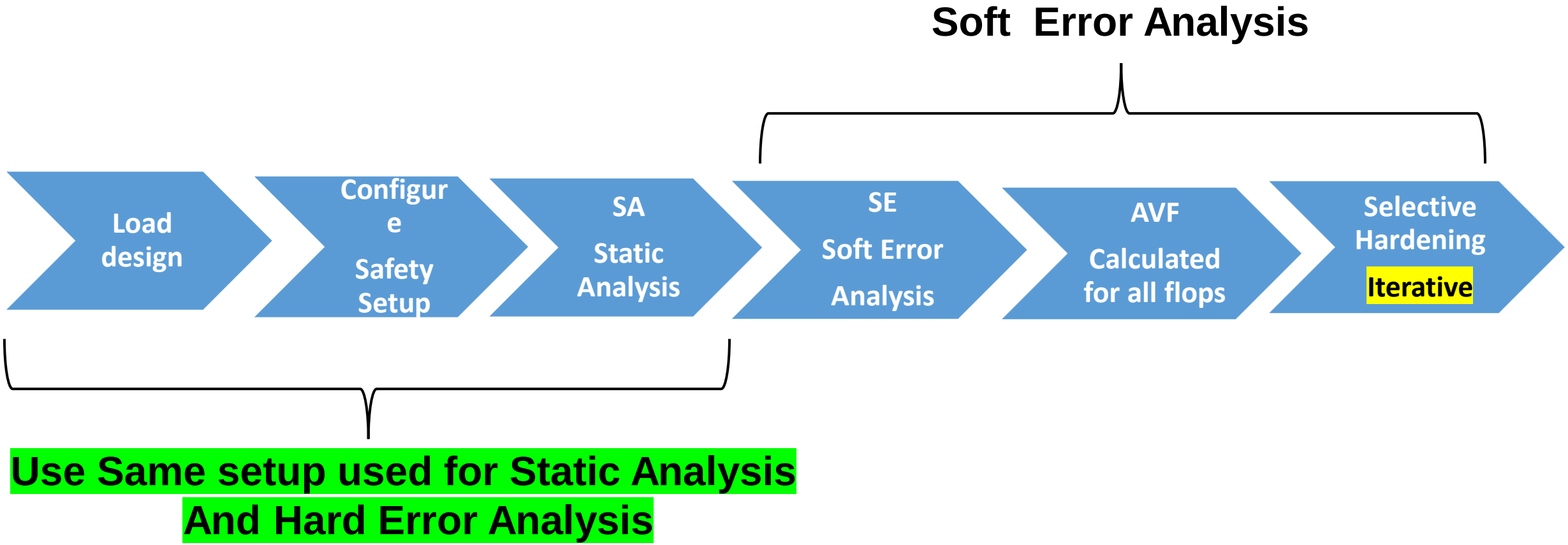
SYSTEMS INITIATIVE
2022 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE
MUNICH, GERMANY DECEMBER 6 - 7, 2022

Optima-SE™ Soft Error Analysis





Soft Error Analysis Flow



Calculating AVF

- Multiple SE fault-simulations are performed for each flop
  - Each SE is injected at different time/cycle in the simulation
  - Optima recommends 50 to 100 faults-per flop
- Each fault-simulation can end with
  - **Propagated** – The fault propagated to a “safety-critical“ node/output
  - **Detected** – The fault has ended at Detection Strobe
  - **Dissipated** – The fault dissipated or disappeared
  - **EOT** – End Of Trace: the fault-sim has reached the end of simulation without any of the results above.
  - Optima marks EOT as SAFE or UNSAFE based on a WIDTH parameter.

$$AVF = \frac{\textit{Number of propagated faults}}{\textit{Total fault simulations executed}}$$

Optima-SE™ Advantage

- Optimal and simple Safety Setup using
  - Find Failure and Detection strobes with Extensive Safety Setup analysis
- User can specify Number of Faults to be injected on a flop
- Fault Placement Times
  - Equal time windows
    - Total stimulus time / number of faults
  - Random Placement Times
    - Stimulus time divided in to equal windows and in each window random time
  - User specified Times
    - User can specify the particular time stamp fault to be placed on the target flop
- Fault Type
  - Flip fault: The value at the target flop output flipped at the placement timestamp
  - Constant: A stuck at fault is placed at the target flop for the specified window
- Fastest Simulation Engine for AVF calculations

FiT Rate with AVF

Without knowing the AVF of each flip-flop

$$FIT\_chip \uparrow = n * fit\_unhard$$

Every Flop Contributes To FIT_Chip

Knowing the AVF of each flop

Without hardening

$$FIT\_chip \downarrow = \sum_{k=0}^{n} (\downarrow AVF(k) * fit\_unhard)$$

Lower AVF flops are safe, Hence FIT_chip is reduced

Knowing the AVF of each flop

With selective hardening of m flops

$$\downarrow FIT\_chip = \sum_{k=0}^{m-1} (\uparrow AVF(k) * \downarrow fit\_hard) + \sum_{i=m}^{n} (\downarrow AVF(i) * fit\_unhard)$$

High AVF flops Hardened, Hard Flops have LOW FIT

Unhardened flops have LOW AVF

n = Number of flops in the chip/IP/unit
AVF(k) = The "personal" AVF of specific flops k
m= number of hardened flops

FIT_chip = FIT Rate for the chip/IP/unit from flop from soft-error
fit_unhard = FIT Rate of a single flop, unhardened regular flop
fit_hard = FIT Rate of a single flop, for hardened flop

2022 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE

MUNICH, GERMANY
DECEMBER 6 - 7, 2022

Security Verification with FAS Fault Attack Simulation

accellera SYSTEMS INITIATIVE

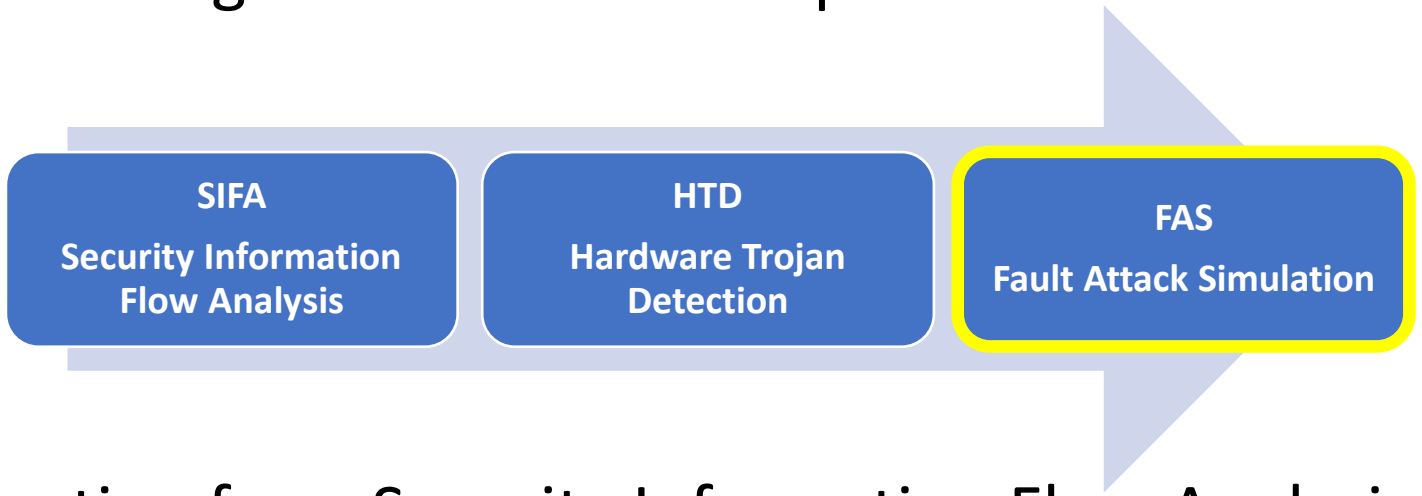
Fault Attacks

- Fault Injection Attacks are mainstream for hackers
- Differential Fault Analysis (DFA) is very well established
- Hackers use DFA to extract the Information and Secret Assets
- Common fault-injection attacks:
  - Laser fault injection
  - EM fault injection
  - Power fault injection
  - Frequency fault injection
  - Spoofing and changing data on a bus or communication channel
  - Changing password-in to try many combinations
  - Etc.

FAS Verifies the attack Countermeasures using fault-simulation, on RTL or Gate Level Designs

FAS: Fault Attack Simulation

- FAS is part of the integrated Solution in Optima-SEC™



- FAS gets information from Security Information Flow Analysis
  - Vulnerable portions of the Design
  - Suspicious Trojan insertion point

Defining FAS

- Complex Attack can be simulated, by defining
  - Target Gate/Node/Flop
  - Type of Fault (SEU, Stuck-at, bridging, etc.)
  - Simulation Time at which attack happens
  - Time span of the attack

Generating FAS Faults

- Auto Generated
  - Millions of Faults Automatically by using `fas_faults –generate` command with various params
- Manual
  - Targeted on specific flops/gates by using command `fas_faults –add` specifying location, fault type and length.

// Directed random attacks generated by the tool:

```
fas_faults -generate
     -total_attacks 100,000
     -each_attack_hits {5 10}
     -hits_window {50,000 51,000}
     -instance_list {aes_128/r7}
     -group round7
fas_faults -generate
     -total_attacks 1,000,000
     -each_attack_hits {5 10}
     -hits_window {150,000 160,000}
     -instance_list {aes_128/r8}
     -group round8
```

// User specified attacks:

```
fas_faults -add
{(SEUF,aes_128_dcls/aes_main/s0[127],0,0)
(SEUF,aes_128_dcls/aes_red/s0[124],50000,0)
(SEUF,aes_128_dcls/aes_red/s0[126],100000,0)
(SEUF,aes_128_dcls/aes_red/s0[123],150000,0)
(SEUF,aes_128_dcls/aes_main/s0[120],200000,0)}
  -group manual
```

FAS Campaign and Results snapshot

RTL/GL

Security Setup

SAI – Secure Assets Instances

Failure Strobes(FS)

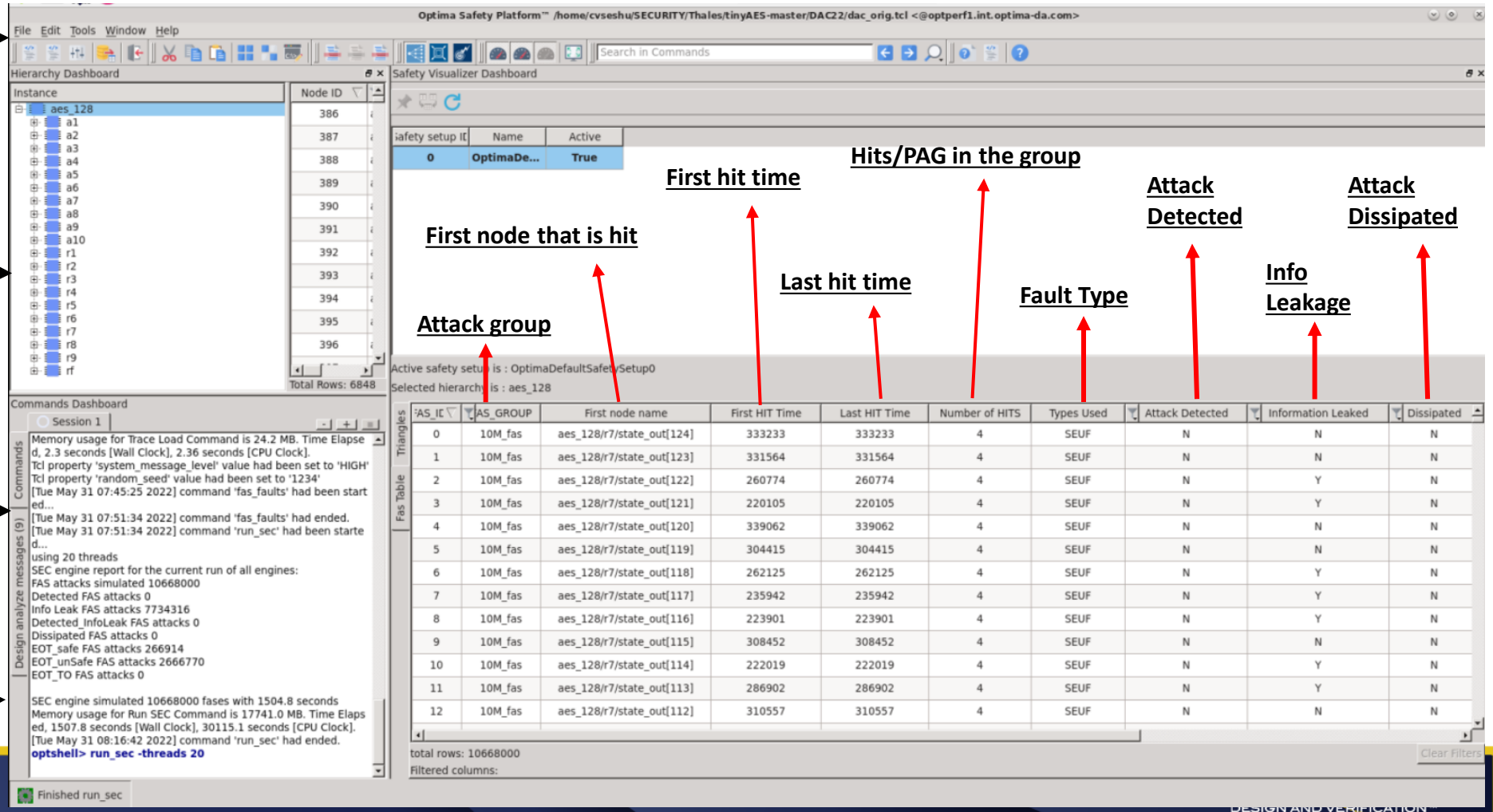
Detection Strobes(DS)

Work Load VCD

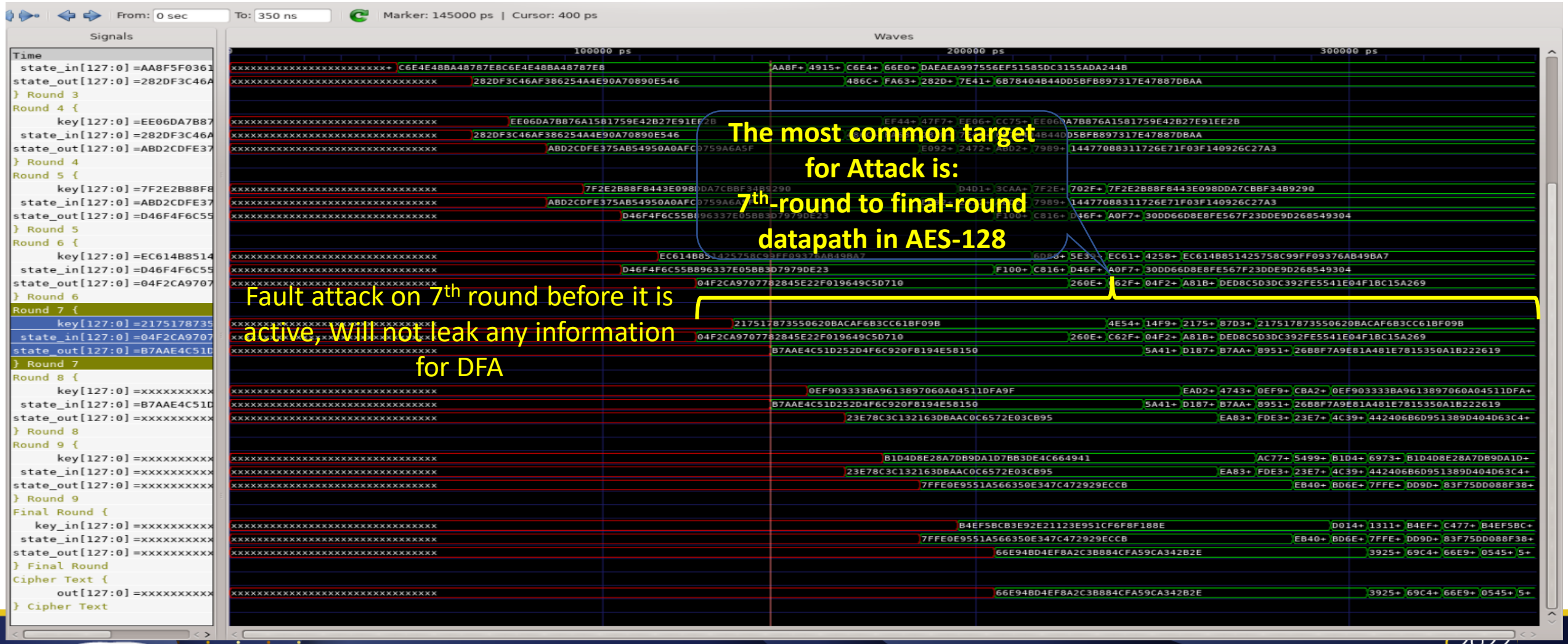
FAS campaign modeling

Attack description or properties

- Random attack
- Laser attack
- EM attack
- Spoofing
- … attack



FAS Attack timing on Block Cypher (AES-128)

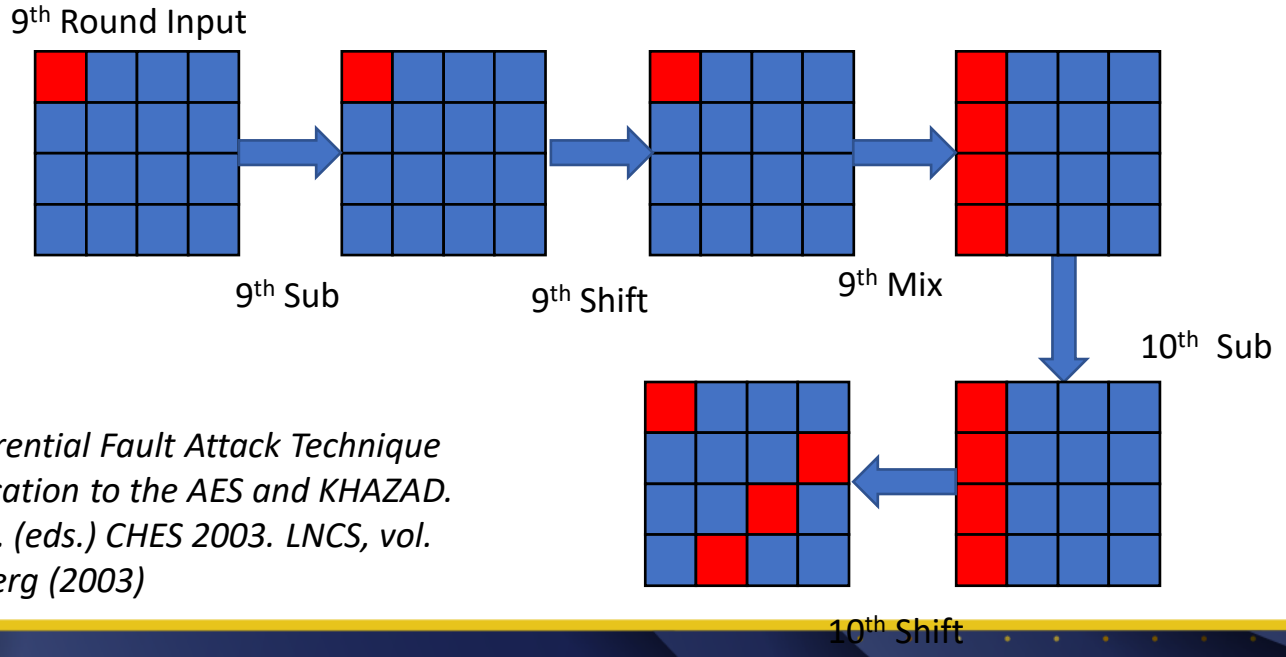


DFA for Block Cypher attacks

- Attack Location
  - AES Data path
  - AES Key –Schedule
  - Single Byte
  - Multiple Byte
- Attacks are based on
  - DFA algorithms
  - Brute Force
- Attacks Discussed in this presentation
  - Attack 1: First round single bit
  - Attack 2: Single bit at 9^{th} round input
  - Attack 3: On Time Redundancy Counter Measure
  - Attack 4: On Parity Bit Counter Measure

Attack 1: Single Bit at 9th round input

- First round attacks are not practical, so attack the 9th round
- Single bit arbitrary fault at input of 9th round
- As demonstrated by Piret et al*, The fault propagates through rounds as below



** Piret, G., Quisquater, J.-J.: A Differential Fault Attack Technique against SPN Structures, with Application to the AES and KHAZAD. In: Walter, C.D., Ko¸c, C¸ .K., Paar, C. (eds.) CHES 2003. LNCS, vol. 2779, pp. 77–88. Springer, Heidelberg (2003)*

Attack 1: modelling

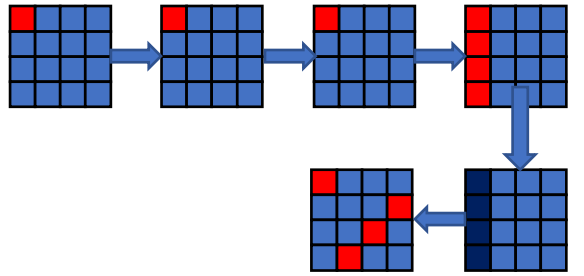
- fas_faults -add {
  (SEUF,
  aes_128/r8/state_out[127],
  265000,
  0)
  }
  -group r9_in_bit

0 127..120	4 95..88	8 63..56	12 31..24
1 119..112	5 87..80	9 55..48	13 23..16
2 111..104	6 79..72	10 47..40	14 15..8
3 103..96	7 71..64	11 39..32	15 7..0

Attack 1: Faults at Final Output

```
fas_faults -add {(SEUF,aes_128/r8/state_out[127],265000,0)} -group r9_in_bit
```

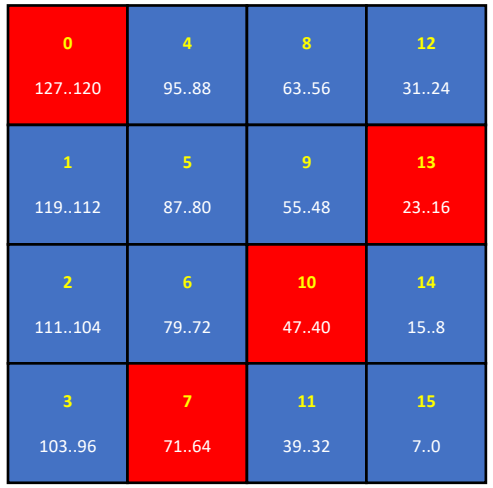
- FAS result: information leaked
- Decision time: 305,000



FAS result:

```
Information leaked (16)times (strobe_name,
observation_time):

(aes_128/out[18], 305000) (aes_128/out[46], 305000)
(aes_128/out[19], 305000) (aes_128/out[21], 305000)
(aes_128/out[42], 305000) (aes_128/out[44], 305000)
(aes_128/out[43], 305000) (aes_128/out[67], 305000)
(aes_128/out[66], 305000) (aes_128/out[17], 305000)
(aes_128/out[16], 305000) (aes_128/out[23], 305000)
(aes_128/out[45], 305000) (aes_128/out[127], 305000)
(aes_128/out[126], 305000) (aes_128/out[122], 305000)
```

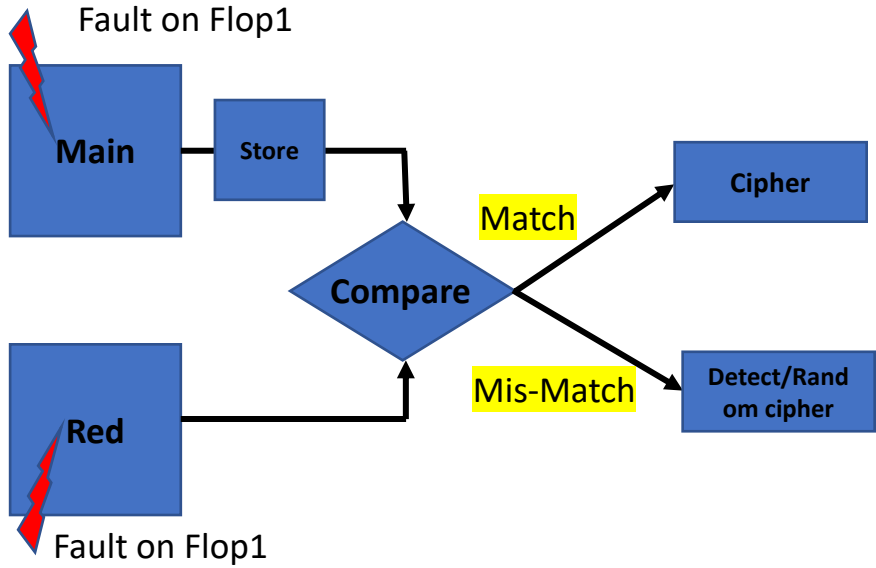


These results match 100% expected behavior according to proposed attack by Piret et al.

Attack 2: on Time Redundancy Counter Measure

- Some attacks beat Hardware redundancy
- Same fault in main AES as well as redundant AES
- Generate FAS attacks to target
  - Flops both in aes_main and aes_red
- Modelling

```
fas_faults -generate
-total_attacks 1000000
-each_attack_hits 2
-hits_window {105000 305000}
-instance_list {aes_128_dcls/aes_main
aes_128_dcls/aes_red} -group oneM_2flops
```

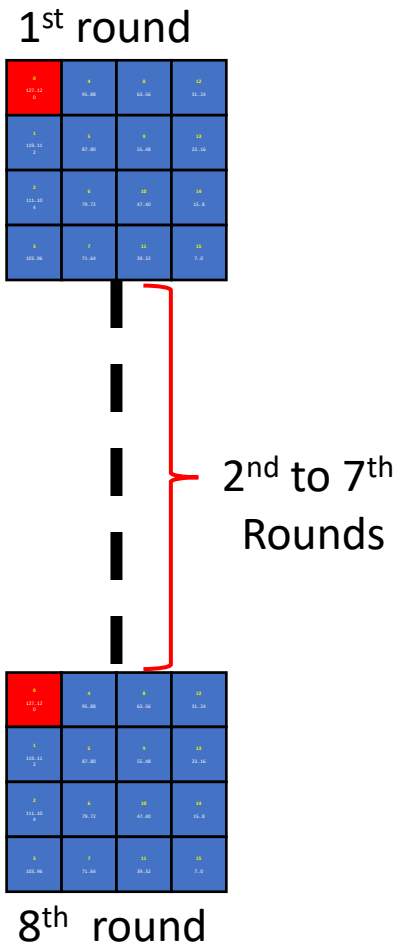


Attack 3: on Parity-bit Countermeasure

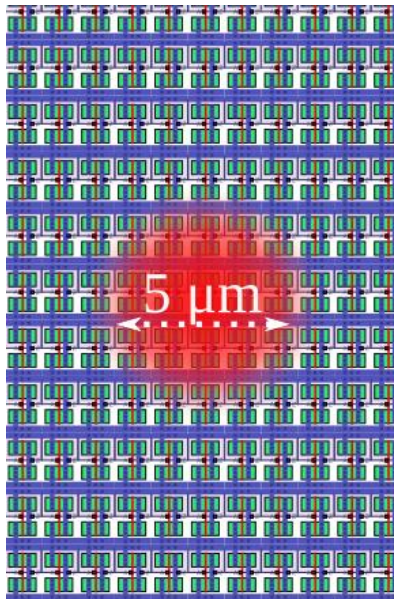
- Two faults hit the same byte in the state
- Hit cycles separated by 8 rounds distance
  - E.g. 1st round 127-bit and 8th round 127-bit
- FAS attack model

```
fas_faults -add {
(SEUF,aes_128/r1/t0/t0/s4/S_/out[0],115000,0) //1st round input
(SEUF,aes_128/r7/state_out[127],245000,0) //8th round input
}
-group r1_r8_parity_attack
```

- Information is leaked from Ineffective Faults



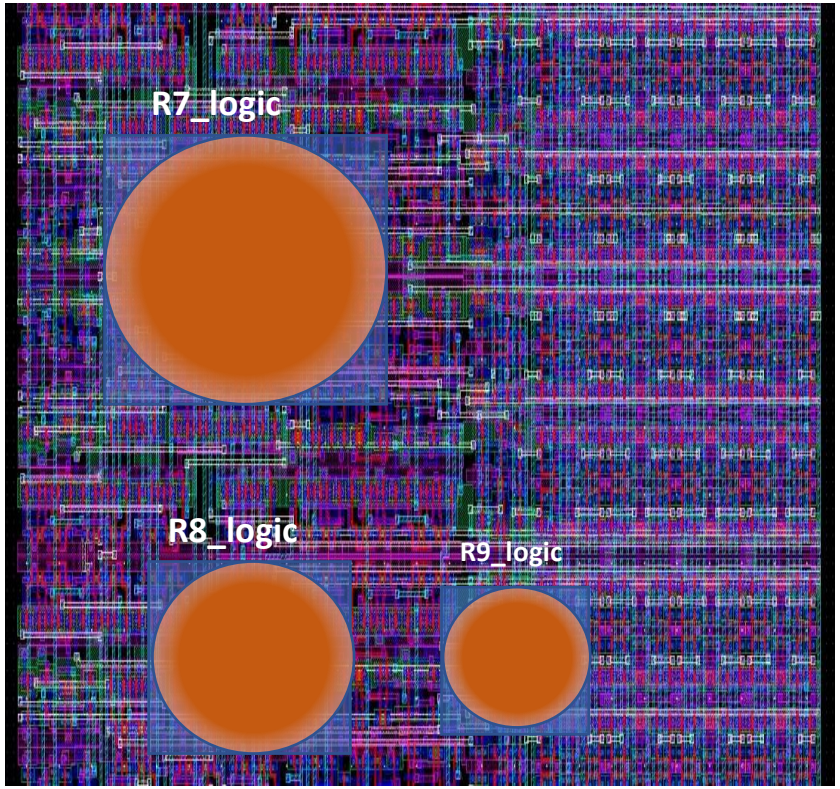
Laser Attack FAS modelling



- Attack Origin: Location of the Attack
- Radius of the attack: Laser focus radius
- PAG: Potentially attacked Gates
  - No Of Gates Under the Laser spot
  - Changes with technology node, more gates in lesser technology node
  - At 22nm assumed to be 100 gates
- Laser Intensity
  - More power more gates influenced
  - Depends on radius, more radius less intensity
- Duration of the Attack: How long IC is irradiated
  - This influences the logic value duration to be modelled
- Each attack done in one pulse
  - In multiple laser pulses, after first pulse, the attack is detected and cleared, so need to model only one pulse at a time

PAG: Potentially Attackable Gates

- R7_logic
  - Laser focused on R7
  - Larger radius
- R8_logic
  - Laser focused on R8
  - Medium radius
- R9_logic
  - Laser focused on R9
  - Smaller radius



Laser FAS attack: PAG8, PAG4, PAG2

- Design Used: AES-128 without protection
- Generated 10,000 FAS's for each attack (laser location)

- R7_logic
  - $PAG_r7_logic is the list of gates affected by Laser on R7 logic

```
fas_faults -generate
-total_attacks 10000
-each_attack_hits 8
-timing_window {225000 235000}
-flops_list {$PAG_R7_logic}
```

Info Leak FAS attacks 7276 out of 10,000 PAG8

- R8_logic
  - $PAG_r8_logic is the list of gates affected by Laser on R7 logic

```
fas_faults -generate
-total_attacks 10000
-each_attack_hits 4
-timing_window {245000 255000}
-flops_list {$PAG_R8_logic}
```

Info Leak FAS attacks 7522 out of 10,000 PAG4

- R9_logic
  - $PAG_r8_logic is the list of gates affected by Laser on R7 logic

```
fas_faults -generate
-total_attacks 10000
-each_attack_hits 2
-timing_window {265000 275000}
-flops_list {aes_128/r9/state_out[127:0]}
```

Info Leak FAS attacks 7523 out of 10,000 PAG2

Performance

- AES-128 is injected with 10 Million FAS
- R7_logic chosen
  - All adjacent 4-bits are injected with faults at random times
  - `$PAG_R7_logic` ➔ `aes_128/r7/state_out[127:0]`
- Attack modelling

```
fas_faults -generate
-all_combinations
-each_attack_hits 4
-timing_window {225000 235000}
-flops_list {$PAG_R7_logic}
-group 10M
```

- Single Thread
  - 10,668,000 faults run in 14,850 sec. (4Hrs 7min.) ➔ 0.00139 sec per FAS attack
- 20 Threads (10times Faster)
  - 10,668,000 FAS's with 1,504 sec. (25 min) ➔ 0.000141 sec per FAS attack

Over 10 million FAS's in 25 min! (Single machine)

Hardware fault Injection: CAPEC-624

- CWE-1247: Improper Protection Against Voltage and Clock Glitches
  - A device needs to guard against fault attacks such as voltage glitches and clock glitches that an attacker may employ in an attempt to compromise the system.
- CWE-1248: Semiconductor Defects in Hardware Logic with Security-Sensitive Implications
  - These defects manifest as faults on chip-internal signals or registers, have the effect of inputs, outputs, or intermediate signals being always 0 or always 1, and do not switch as expected.
- CWE-1332: Improper Handling of Faults that Lead to Instruction Skips
  - Attackers can use fault injection techniques to alter the operating conditions of hardware so that security-critical instructions are skipped more frequently or more reliably than they would in a "natural" setting.

Hardware fault Injection: CAPEC-624 Contd…

- CWE-1256: Improper Restriction of Software Interfaces to Hardware Features
  - Software-controllable mechanisms to dynamically scale device voltage and frequency and monitor power consumption are common features in today's chipsets, but they also enable attackers to mount fault injection and side-channel attacks without having physical access to the device.
- CWE-1261: Improper Handling of Single Event Upsets
  - The hardware logic does not effectively handle when single-event upsets (SEUs) occur.
- CWE-1319: Improper Protection against Electromagnetic Fault Injection (EM-FI)
  - The device is susceptible to electromagnetic fault injection attacks, causing device internal information to be compromised or security mechanisms to be bypassed.
- CWE-1334: Unauthorized Error Injection Can Degrade Hardware Redundancy
  - An unauthorized agent can inject errors into a redundant block to deprive the system of redundancy or put the system in a degraded operating mode.

Blackbox Reverse Engineering CAPEC-189

- CWE-1255: Comparison Logic is Vulnerable to Power Side-Channel Attacks
  - The power consumed by a device may be instrumented and monitored in real time. Attacker can inject faults to alter the power profile and get the information

2022
DESIGN AND VERIFICATION™
DVCON
CONFERENCE AND EXHIBITION
EUROPE
MUNICH, GERMANY
DECEMBER 6 - 7, 2022

Thank you!

accellera
SYSTEMS INITIATIVE