argosim

SYSTEM VALIDATION

STIMULUS

Debug Your Requirements

RIL: Requirements in the Loop

An End-to-End Requirement-based System Engineering and Validation Process

argosim
SYSTEM VALIDATION

System Engineering: State of the Art

Advanced Practice

System ***analysis***: MBSE (SysML) tools

➔ Use cases, functions, architecture… but:

- Complex, no simulation
- Ignore the requirements

Language ***formalization***: in-house methods/tools, “**boiler plates**”

➔ Avoid natural language ambiguities

Common Practice

Requirements ***definition***: **Natural language**

➔ Reference for safety-critical standards

Requirements ***management***: Doors, Reqtify…

➔ Versioning & traceability features

R&D Practice

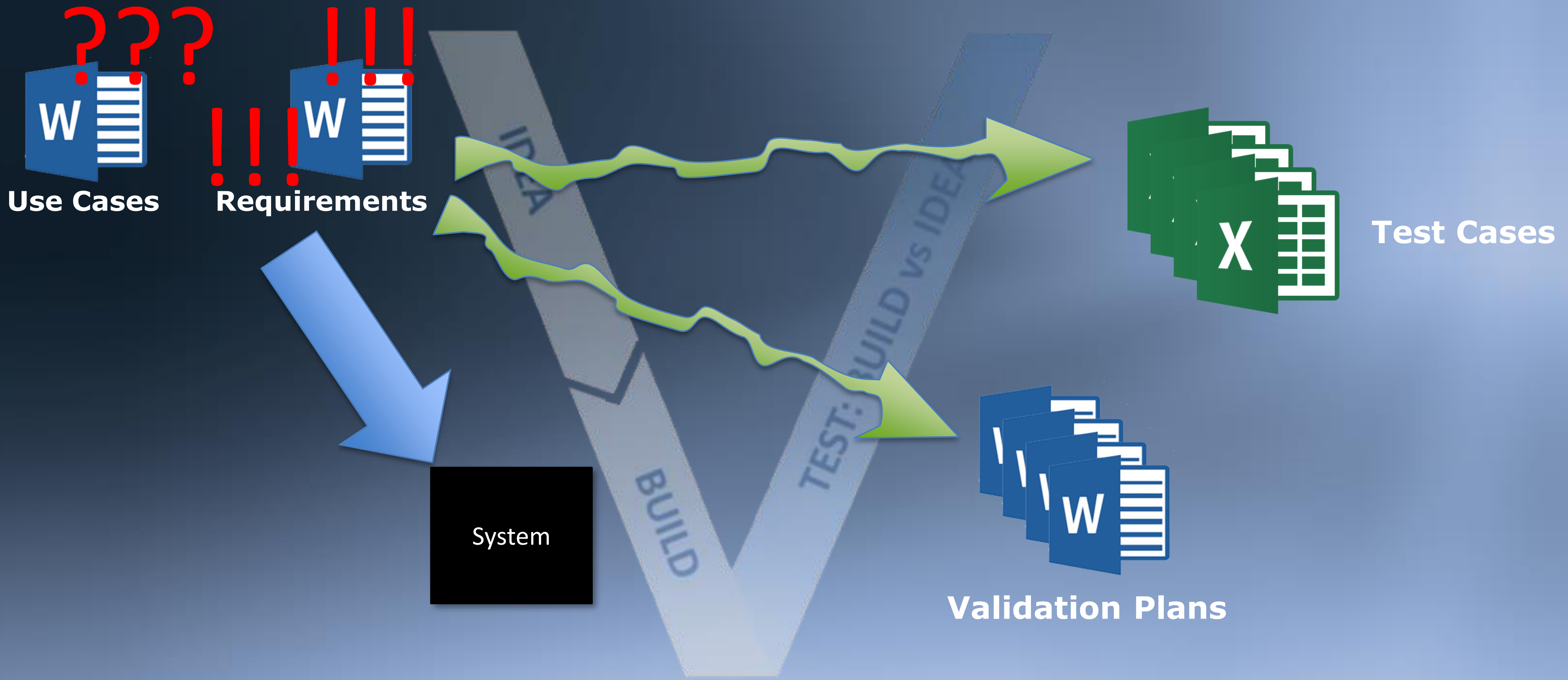
Formal ***verification***: proof tools, MBSA

➔ Expertise required & scalability issues

Language ***verification***: ontological tools

➔ Very heavy to deploy

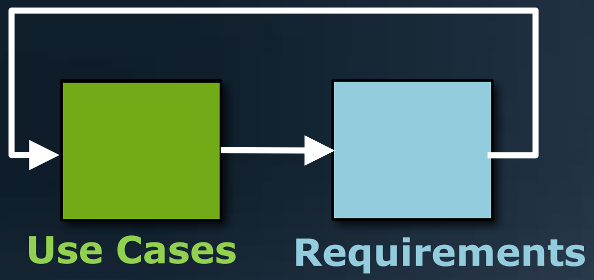
Today's Industry Practice

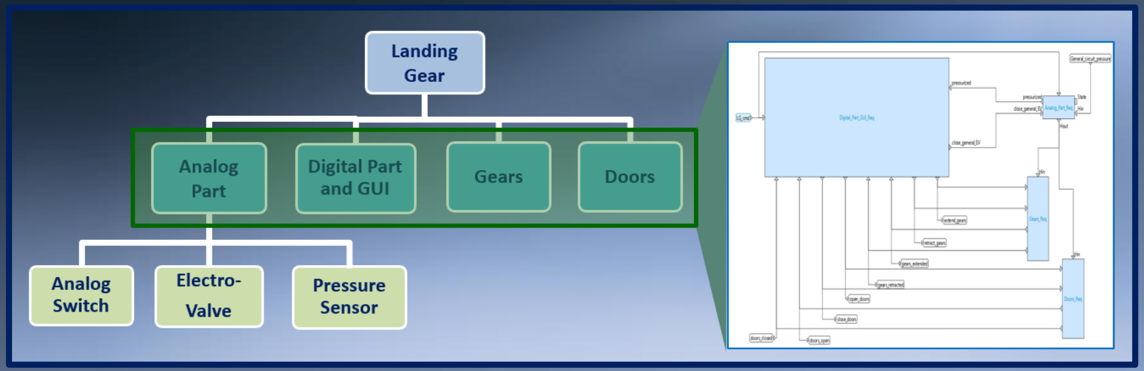


- All transformations are manual
- 70% of the errors introduced in SW projects are introduced during the specification phase, only 4% are detected in that phase

RIL: Requirements In the Loop

STIMULUS Value Proposition





System Architecture and Interfaces

argosim
SYSTEM VALIDATION

STIMULUS Value Proposition

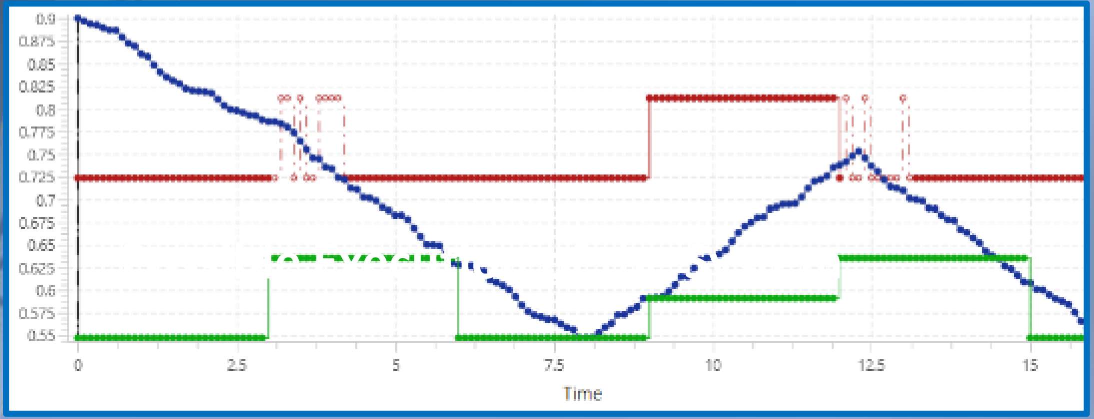
When lightIntensity > 60 % *and* headLight *was* 'OFF ,
headLight *shall be* 'OFF

Use Cases

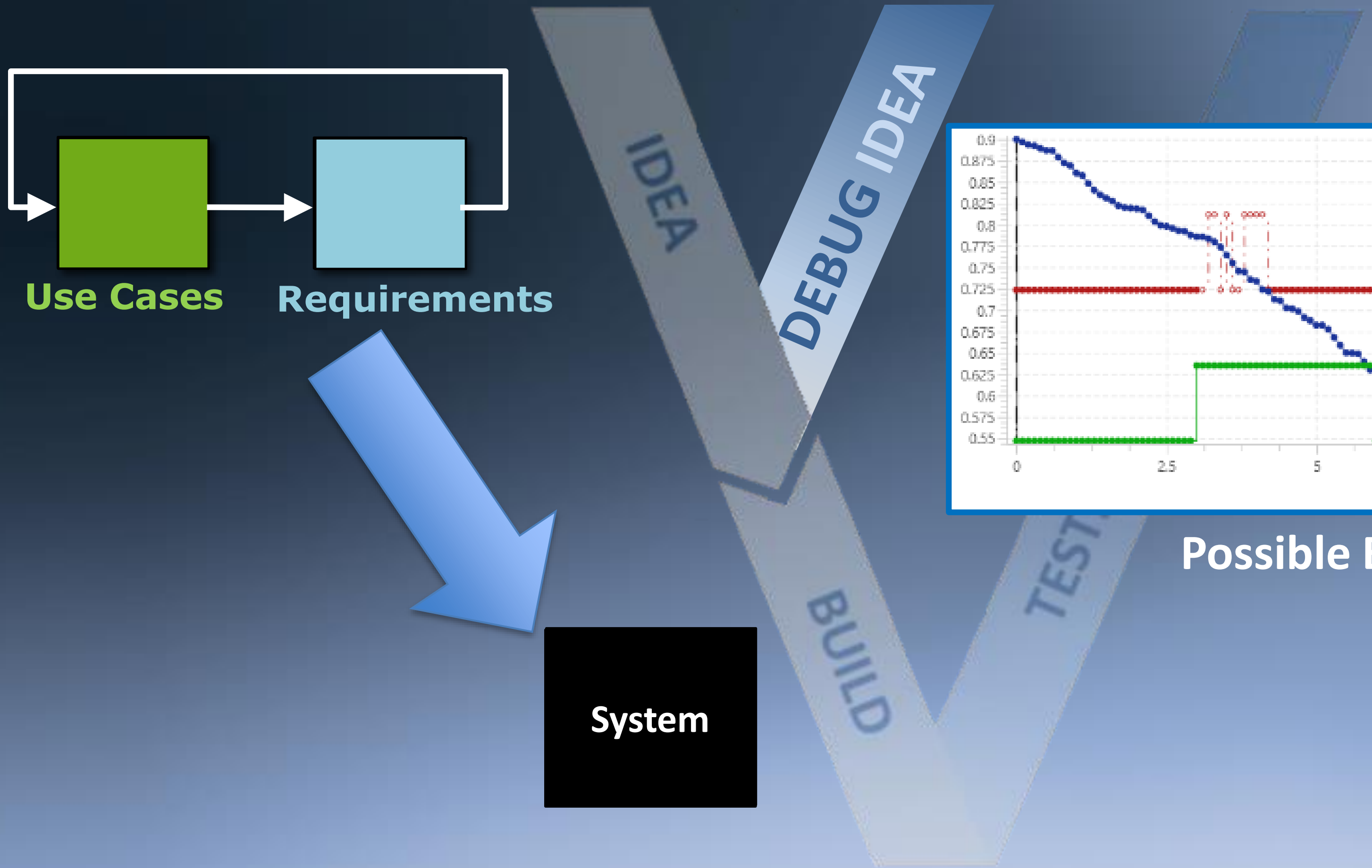
Requirements

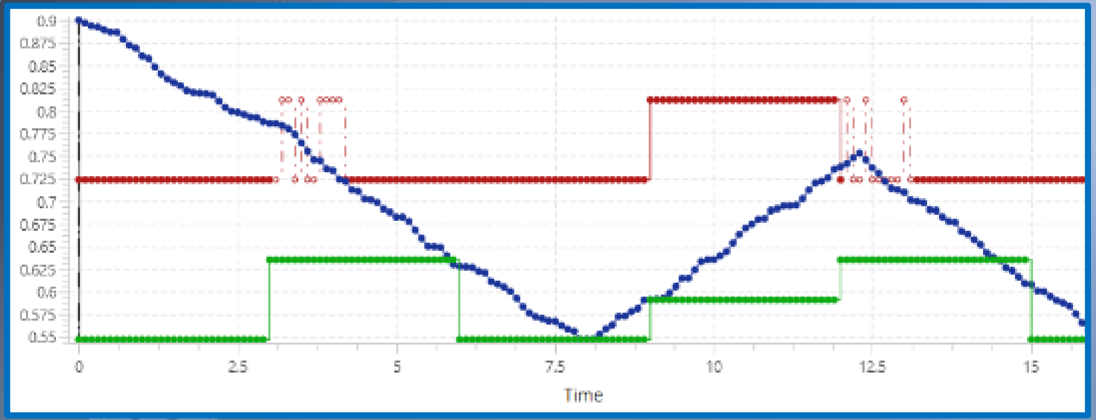
switch *is stable during* 3 [second]

lightIntensity *goes up and down respectively to* 75 % *and* 55 %



STIMULUS Value Proposition





Possible Execution of the System

Basic Concepts

Requirements:

- All functional Requirements can be written using a set of templates
- Example:
  - Natural language requirement: “When the light intensity is less than 60% for more than one second, headlight shall be set to ON”
  - Equivalent template-based requirement:

condition

When lightIntensity *is less then* 60 % *for more than* 1 second ,
then headlight *shall be* ‘ON

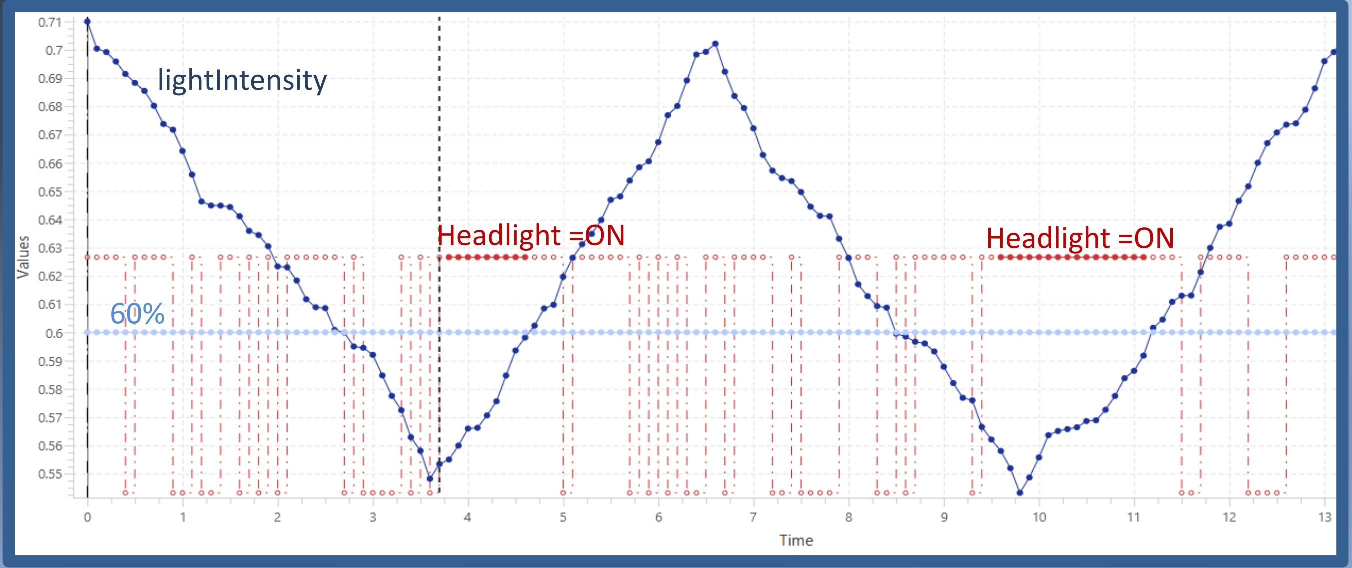
action

Or, in Korean:

lightIntensity 가 60 % 미만 인 상태가 1 [second] 이상 유지될 때,
headLight 는 'ON 이어야한다

Each template has an executable semantics:

When lightIntensity *is less then* 60 % *for more than* 1 second ,
then headlight *shall be* ‘ON



Possible Execution of the System

Benefits: Debug the Requirements as soon as your write them

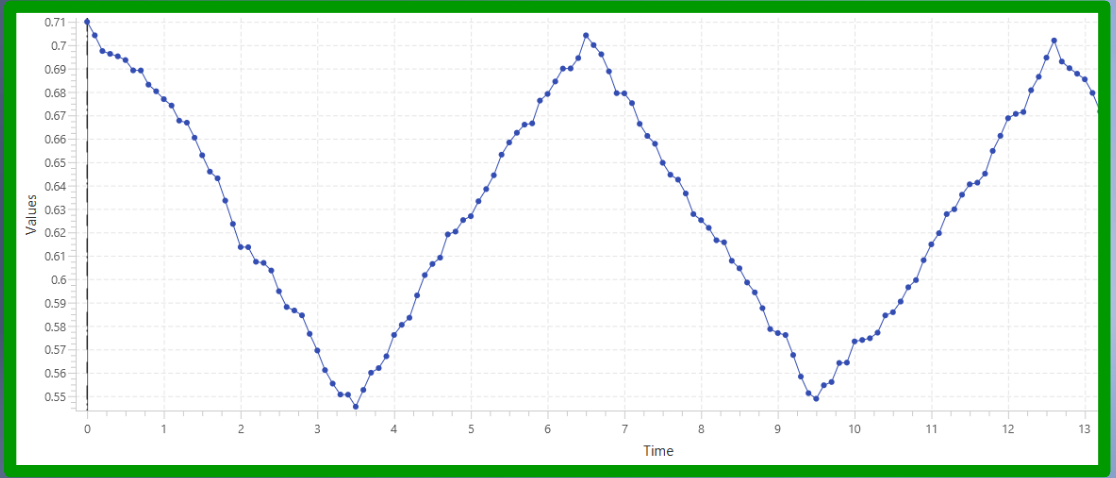
Use Case: a set of constraints on the inputs and between the inputs

((lightIntensity) ∈ [0 , 1])

Initially | lightIntensity *is* (71 %)

lightIntensity *goes up and down between* 70 % *and* 55 %

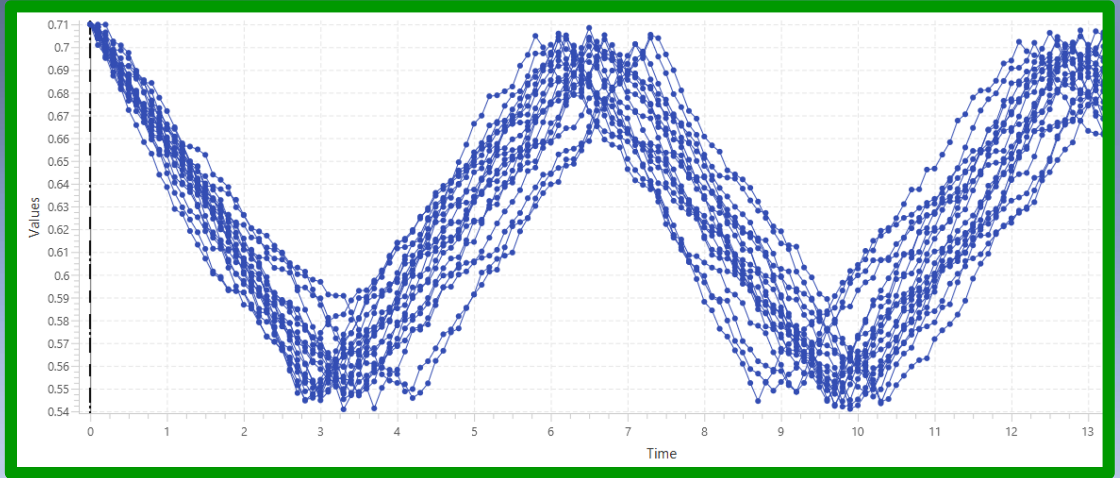
(*derivative of* lightIntensity) ∈ [-0.1 [1/second] , 0.1 [1/second]]



One possible Test Vector

One Use Case can be turned into **as many Test Vectors as desired**

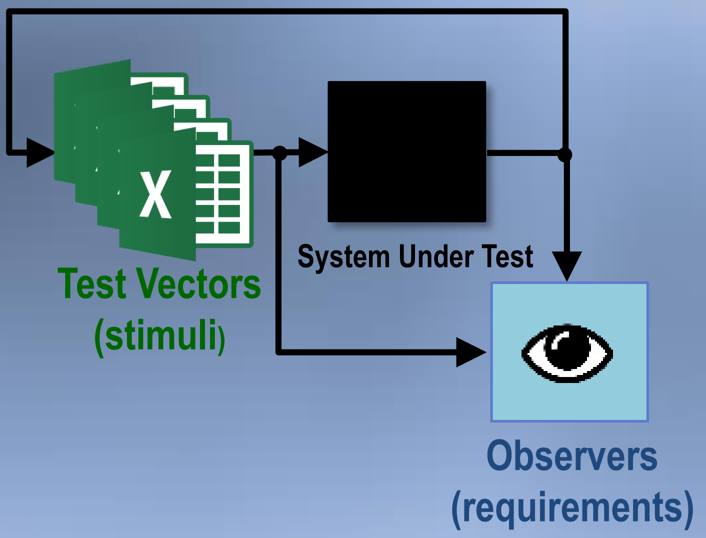
(Automatic Test Vectors Generation)



20 possible Test Vectors

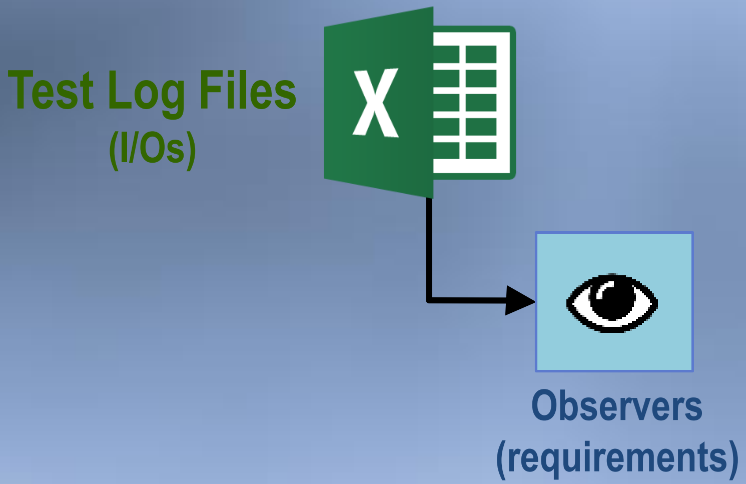
argosim
SYSTEM VALIDATION

STIMULUS Value Proposition

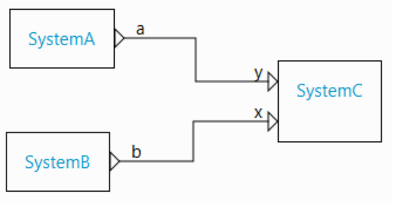


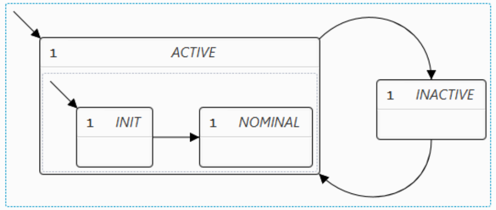
argosim
SYSTEM VALIDATION

STIMULUS Value Proposition

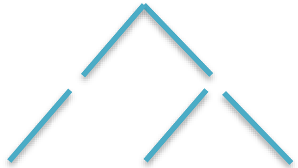


Theoretical and Technical Backgrounds

After error , Maintain alarm During 10 [second] + 

\+ 

Compiler

Constraints over variables 

Data constraints:
- Logico-numerical solver processing the relationships among data

- Control graph
- Backtrack mechanism

Simulator

BDD (Binary Decision Diagrams) + convex polyhedra

Argosim Company Profile

- Company created in 2013. STIMULUS released in early 2015.
- STIMULUS users in avionics, automotive, transportation, energy.
- International presence: USA, UK, Germany, Spain, Israel, Japan, China, Korea, India

STIMULUS Users

AIRBUS HELICOPTERS

SAFRAN AEROSPACE · DEFENCE · SECURITY

THALES

DASSAULT AVIATION

SAFRAN LANDING SYSTEMS

ZODIAC AEROSPACE

AIRBUS DEFENCE & SPACE

Honeywell

AIRBUS

Rockwell Collins

ALSTOM

CS

eDF

TechnicAtome

RATP

SIEMENS

SOGETI High Tech

sopra steria

AREVA

SNCF

AnsaldoSTS

altran

assystem

RENAULT

DENSO

EASY MILE

MAGNETI MARELLI

PSA PEUGEOT CITROËN

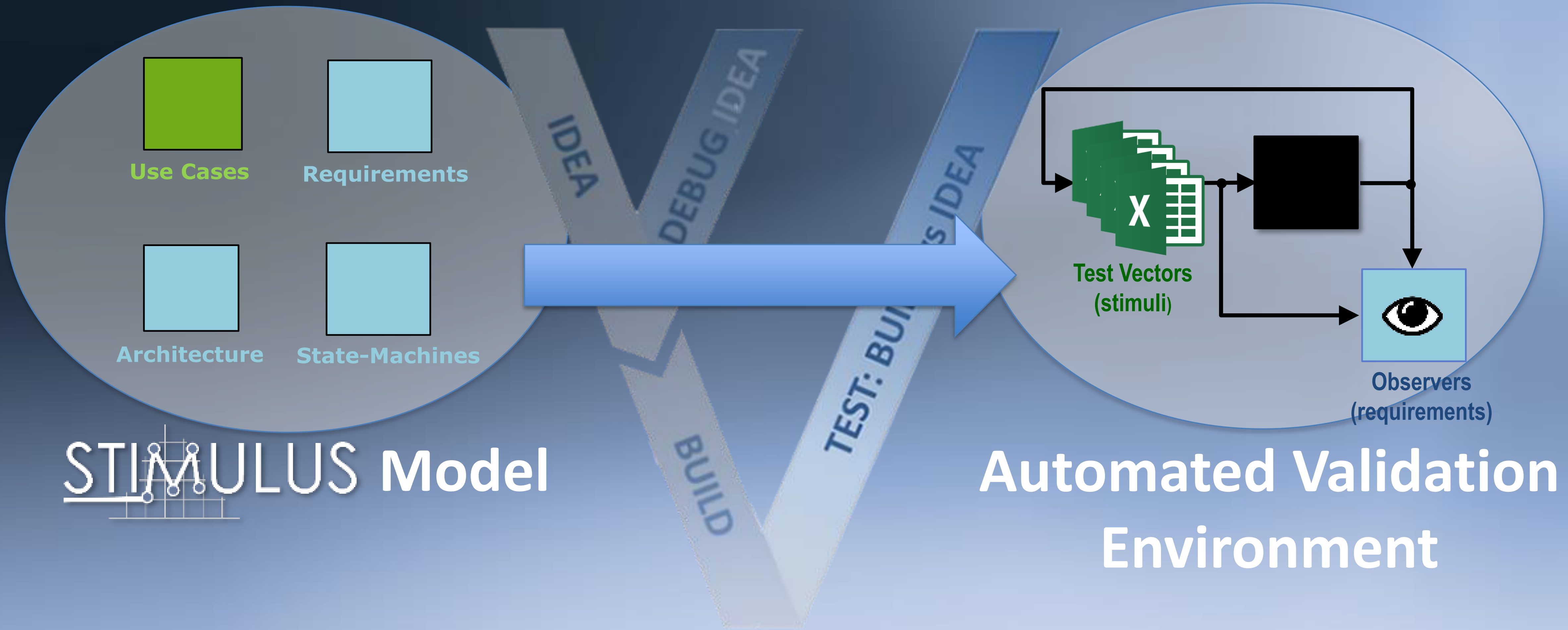
TOYOTA

Valeo

ZF

akebono BRAKE EXPERTS

Requirements In the Loop



STIMULUS

Debug Your Requirements

Demo

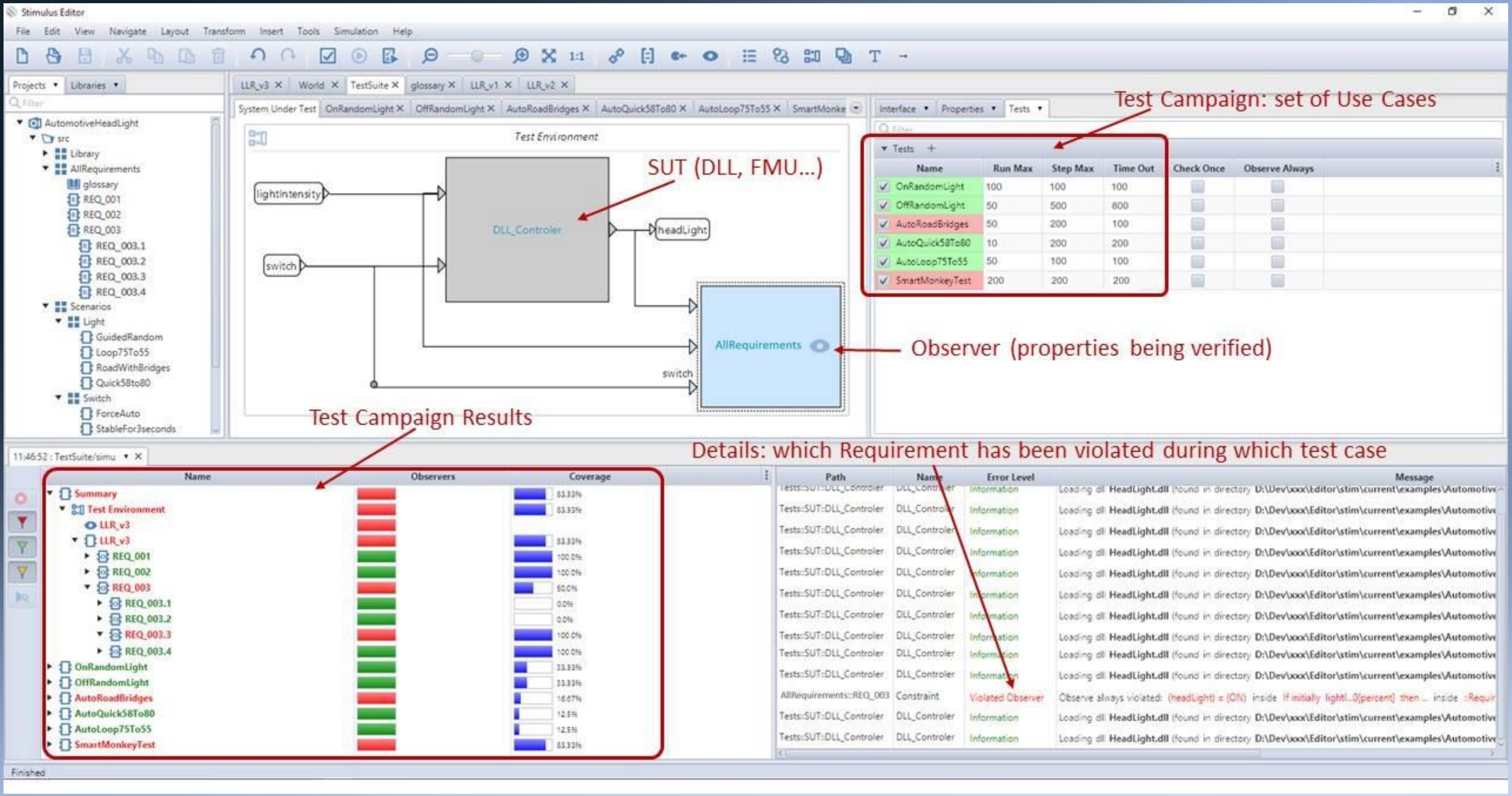
argosim
SYSTEM VALIDATION

STIMULUS

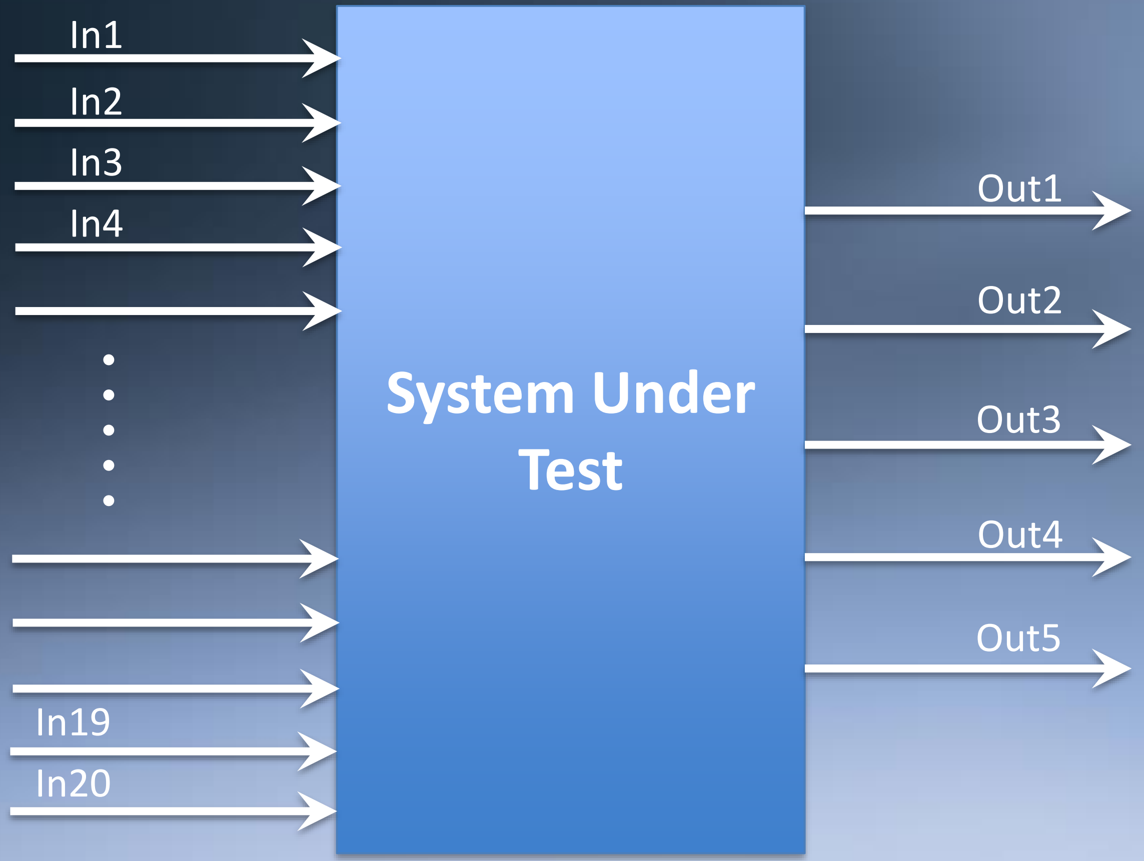
Debug Your Requirements

Additional Slides

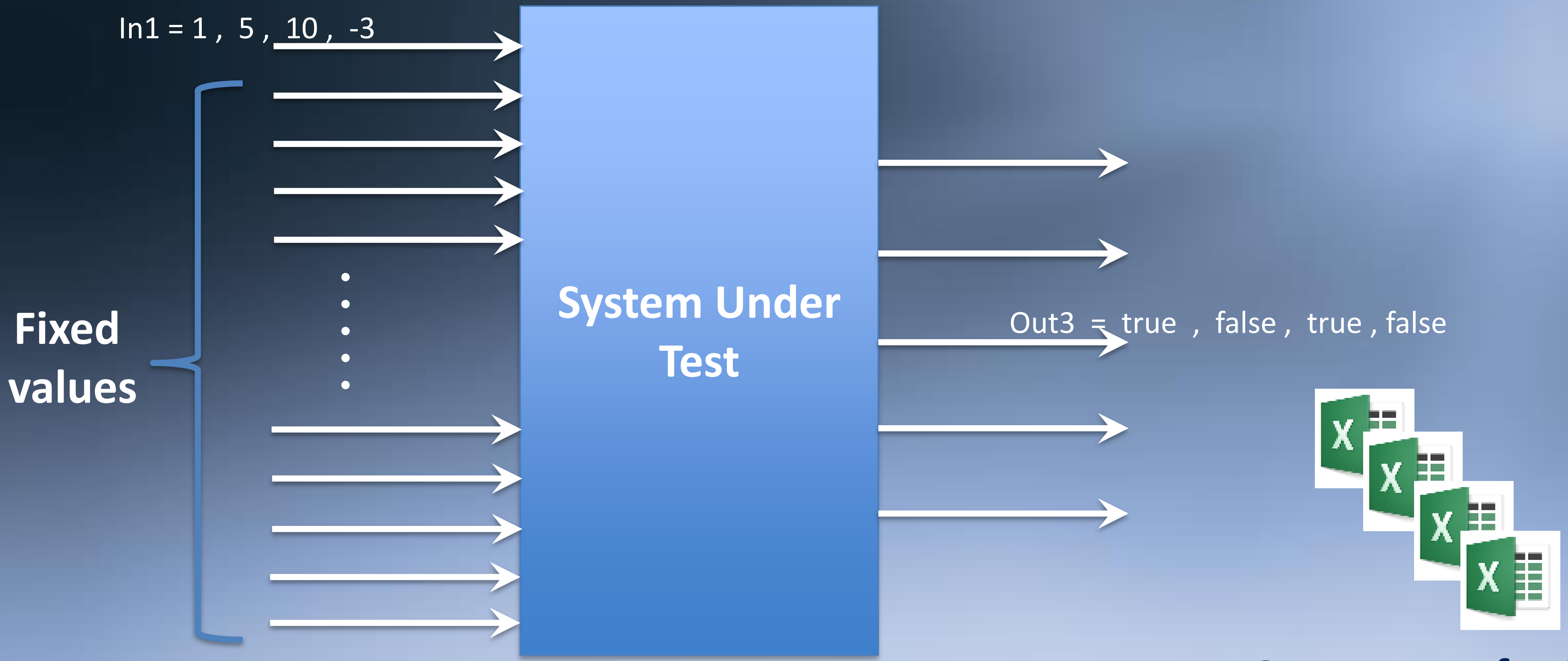
STIMULUS Test Campaign



Today’s Industry Practice (2/4)

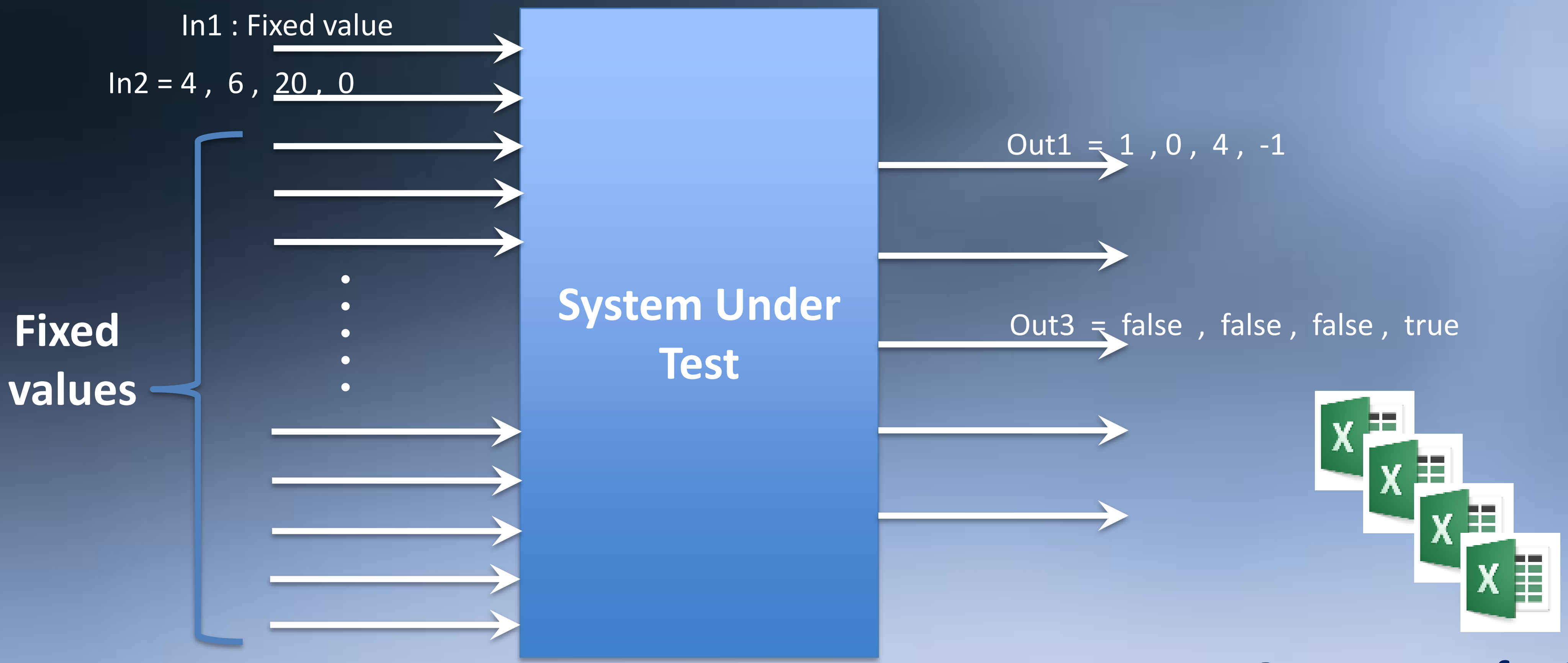


Today’s Industry Practice (3/4)



4 test cases for In1

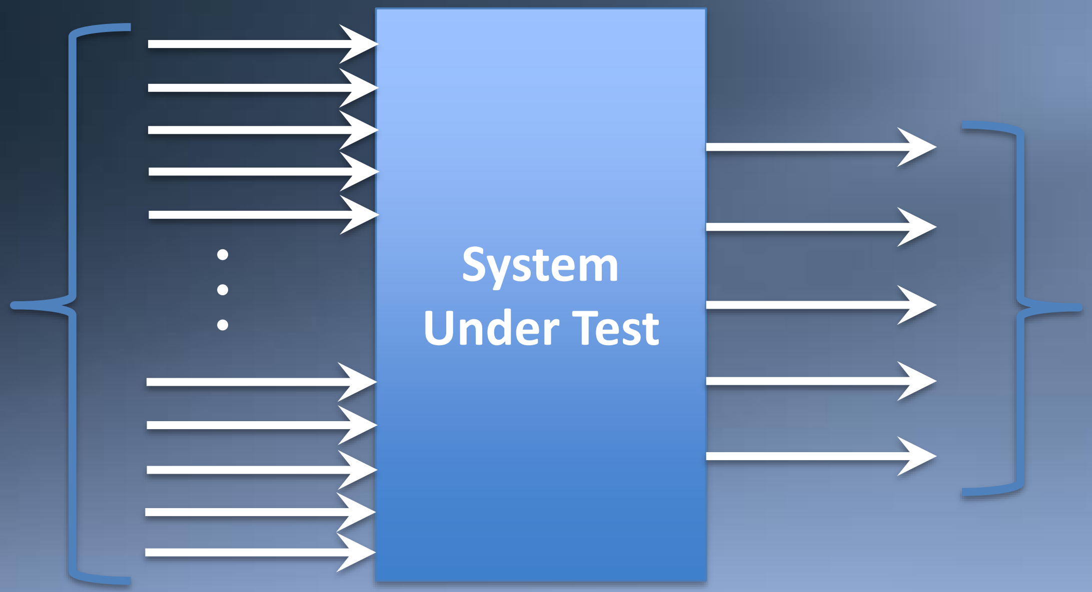
Today's Industry Practice (4/4)



4 test cases for In2

What you **really** want to do

1. Define **constraints** on the inputs and between the inputs
2. **Generate** numerous test vectors within the constraints



Define the test acceptance criteria **independently** of each individual test vector: use **requirement**-based oracles

argosim

SYSTEM VALIDATION

STIMULUS

Debug Your Requirements

Refine & Validate Requirements over a System Architecture

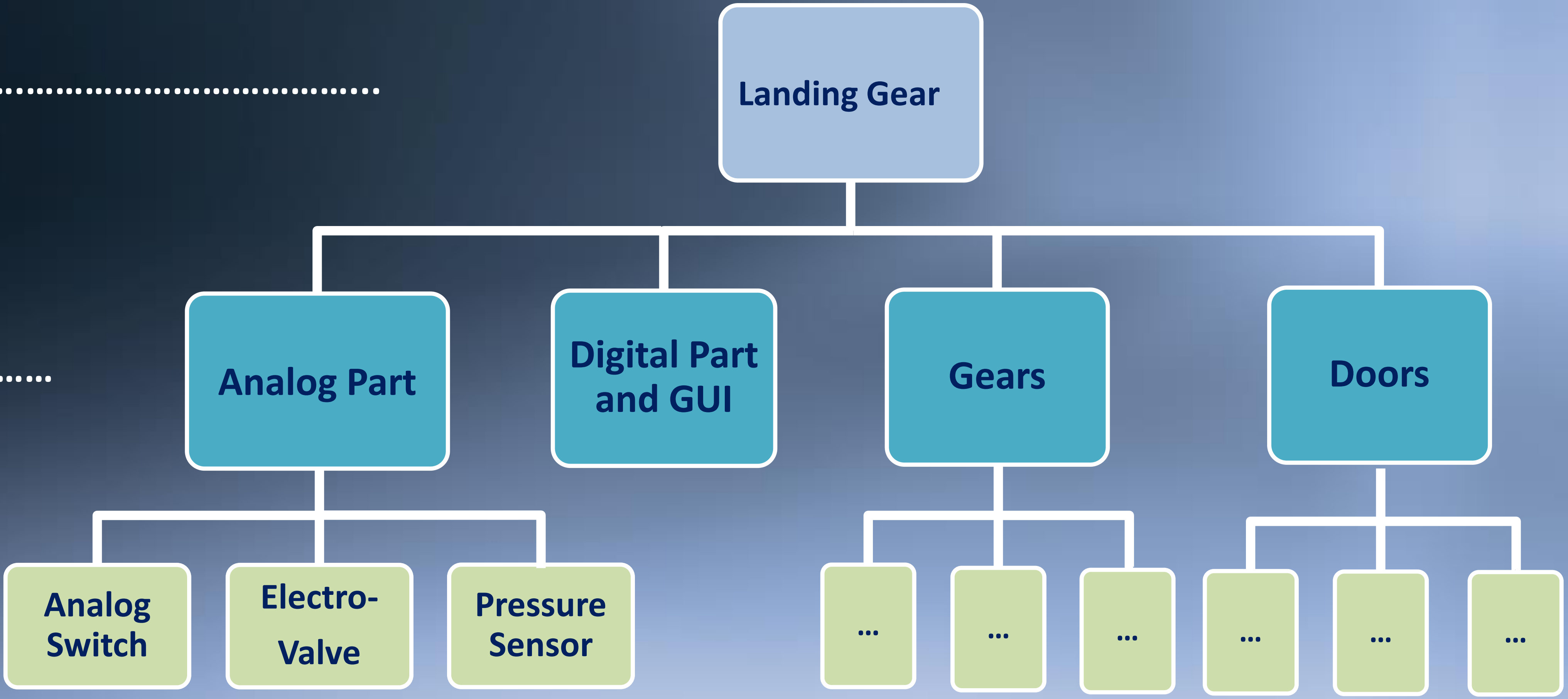
argosim
SYSTEM VALIDATION

STIMULUS
Debug Your Requirements

Level 1 ……………………………………………..

Level 2 ………………

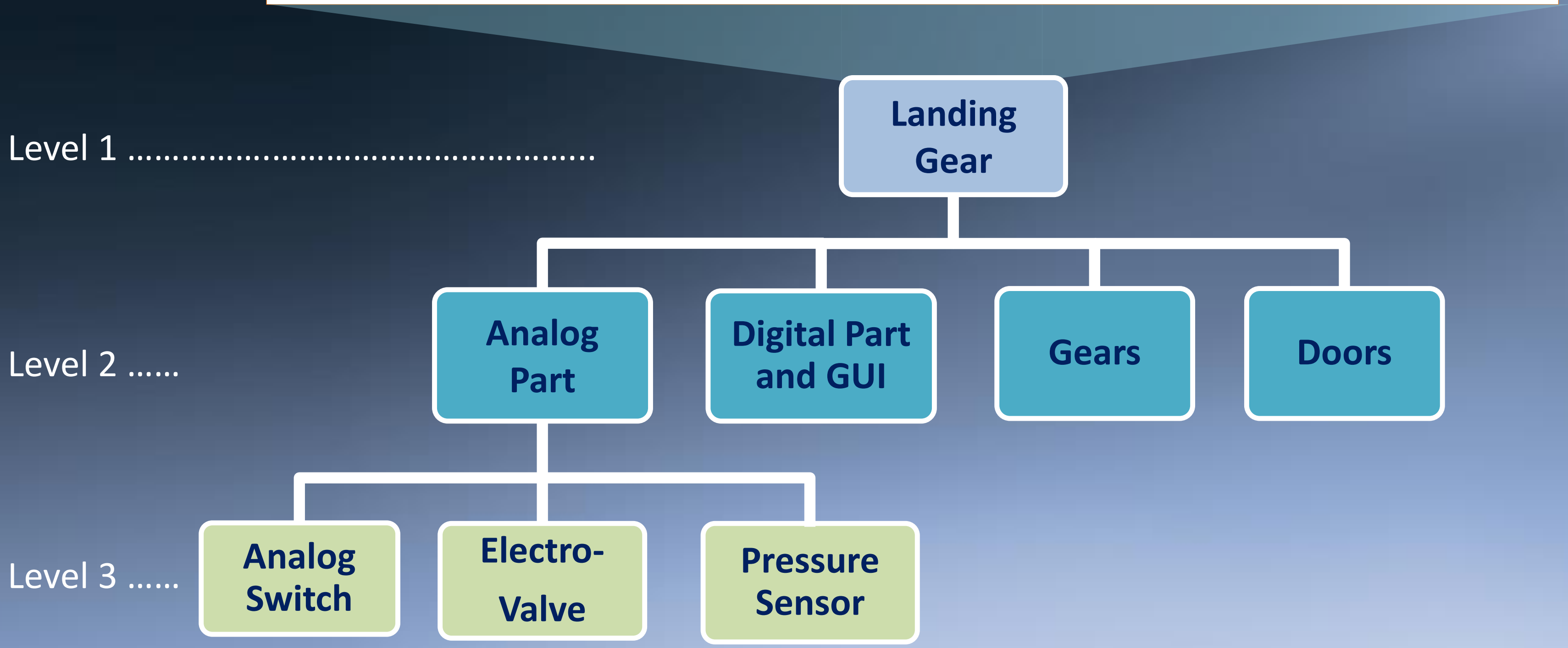
Level 3 …



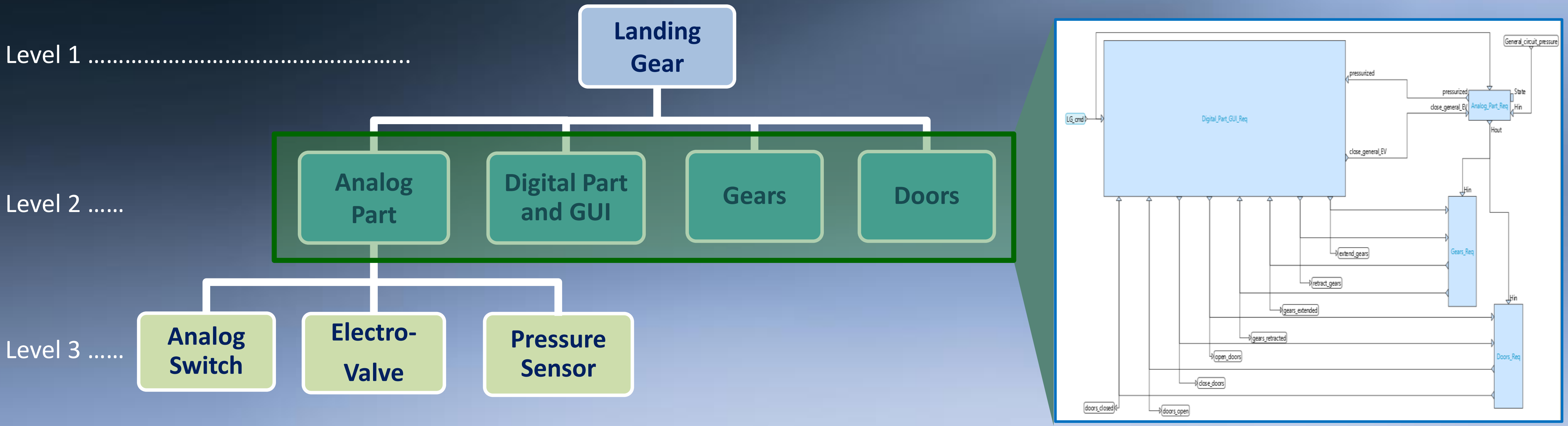
Level 1 Requirements

[LS_RQ_001] *When* LG_cmd *is* 'DOWN' , *Do*
gears_extended *shall be* true *and* doors_closed *shall be* true *once within* 15 [second]
afterwards
gears_extended *shall be* true
doors_closed *shall be* true

[LS_RQ_002] *When* LG_cmd *is* 'UP' , *Do*
gears_retracted *shall be* true *and* doors_closed *shall be* true *once within* 15 [second]
afterwards
gears_retracted *shall be* true
doors_closed *shall be* true

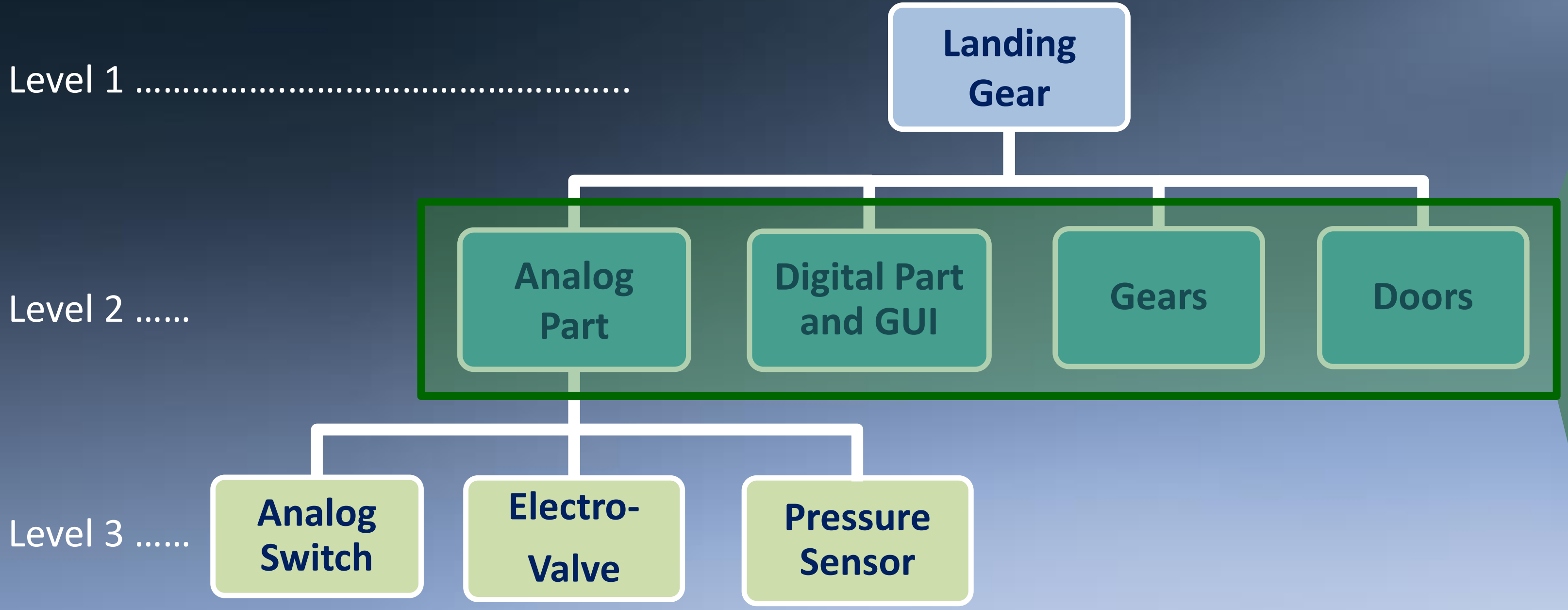


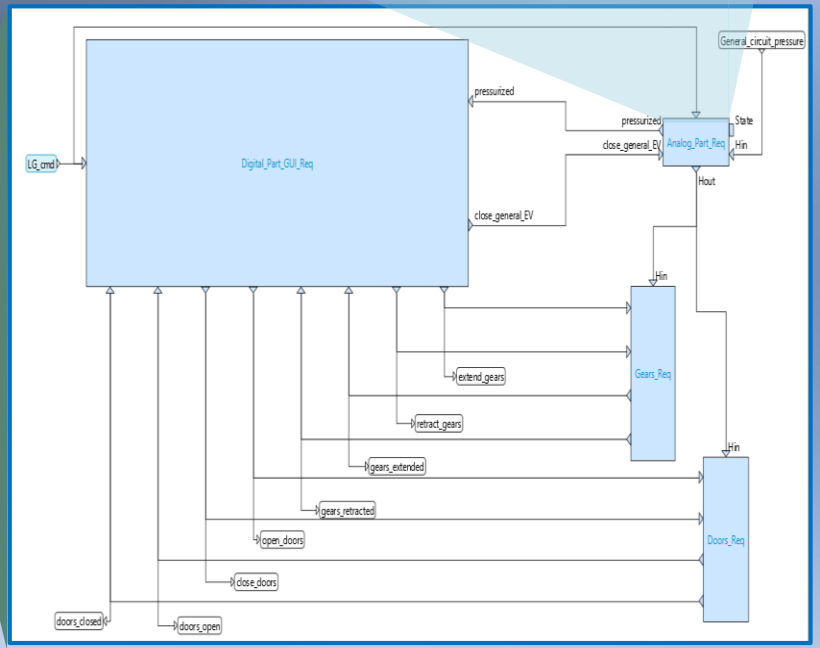
Level 2 Architecture



Level 2 Component Requirements

[AP_REQ_002] *From each* LG_cmd *is unstable* ,
Do
State *shall be* 'closed *once within* 2 [second]
afterwards
During 20[second] - period , State *shall be* 'closed
[AP_REQ_003] *When* LG_cmd *is stable and* last State *is* 'closed *has been true during more than* 20 [second] , State *shall be* 'open
[AP_REQ_004] *When* State *is* 'closed ,
When close_general_EV *has been true during more than* 1 [second] , pressurized *shall be* true



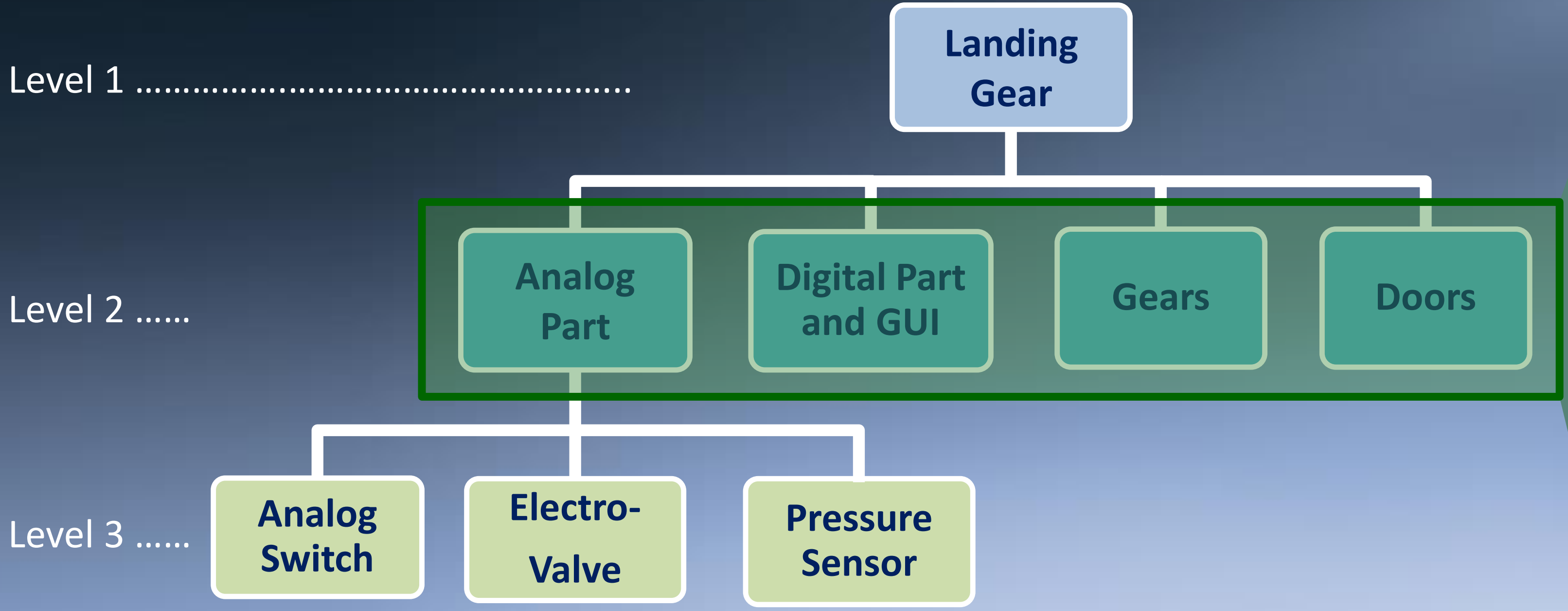


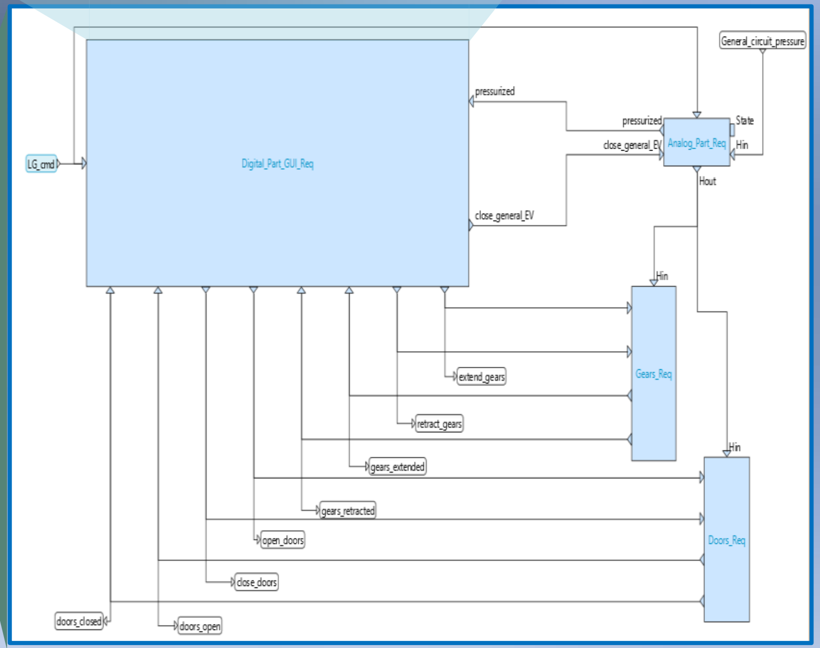
Level 2 Component Requirements

[DPI_REQ_007]*When* ((retractation_sequence *is* true) *and* (outgoing_sequence *is* false)),
Before ((doors_closed *is* true) *and* (gears_retracted *is* true)),
close_general_EV *shall be* true
afterwards
open_doors *shall be* false
close_doors *shall be* false
close_general_EV *shall be* false

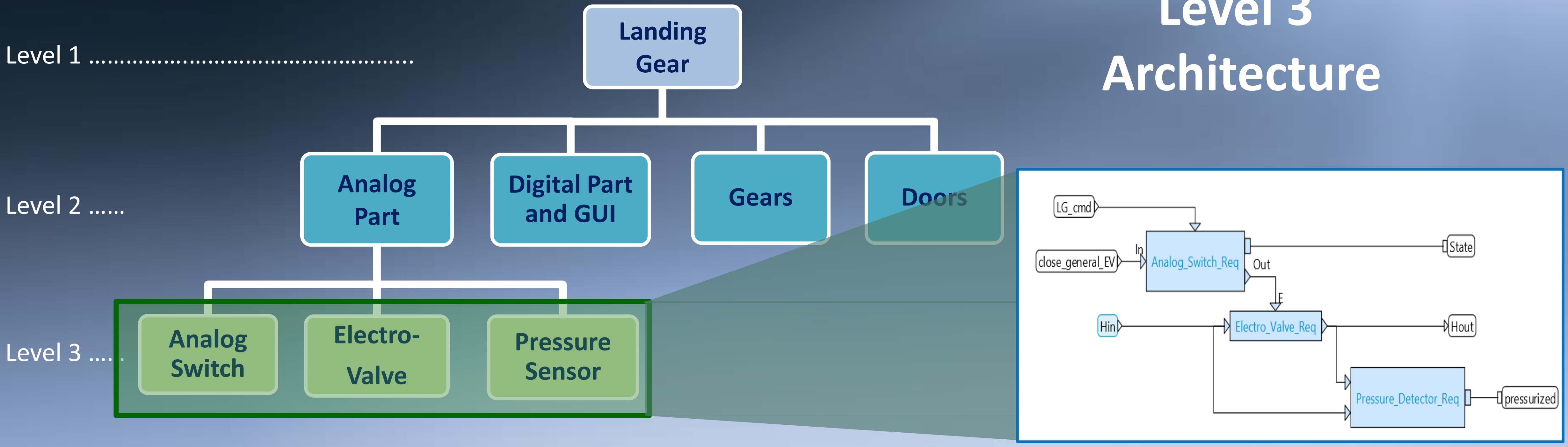
[DPI_REQ_008]*When* ((retractation_sequence *is* true) *and* (outgoing_sequence *is* false)),
When (pressurized *is* true), *As long as* (gears_retracted *is* false), open_doors *shall be* true

[DPI_REQ_009]*When* ((retractation_sequence *is* true) *and* (outgoing_sequence *is* false)),
When ((pressurized *is* true) *and* (doors_open *is* true)), *As long as* (gears_retracted *is* false), retract_gears *shall be* true





Level 3 Architecture



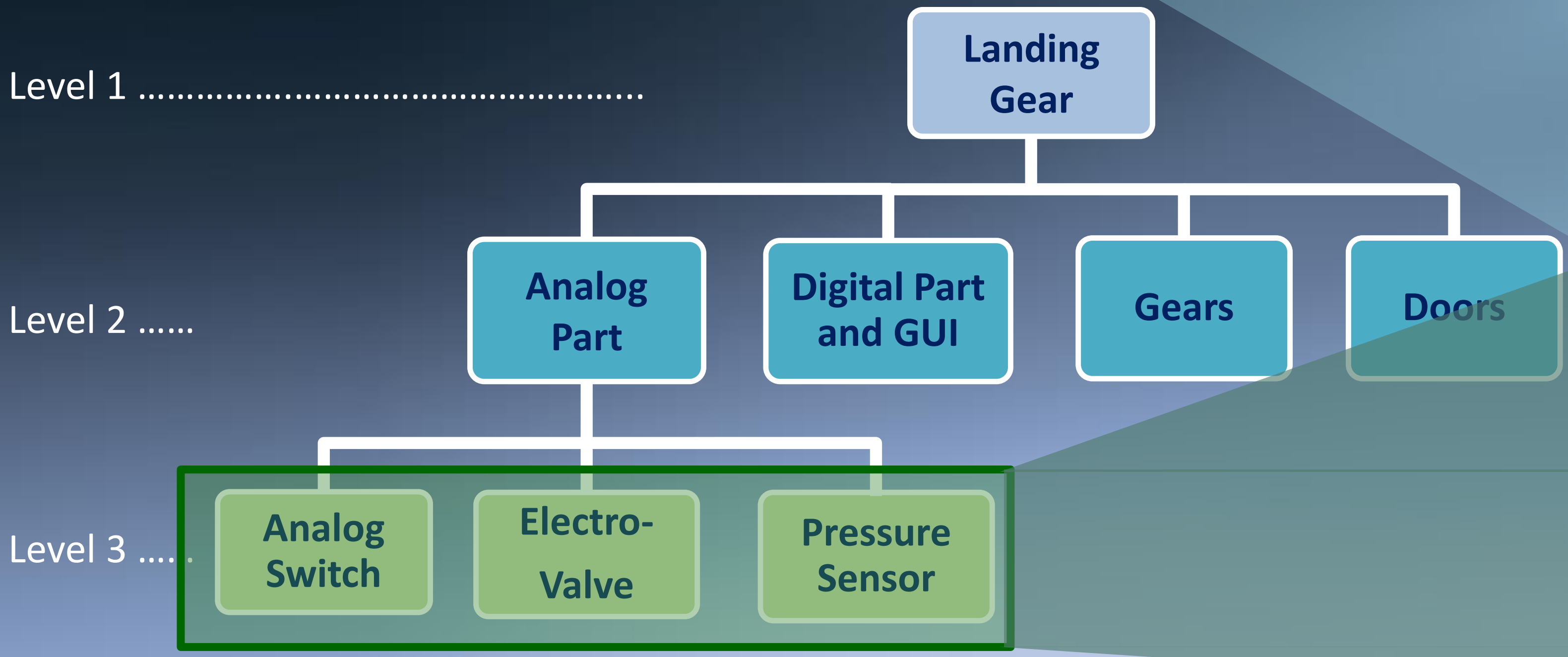
Level 3 Component Requirements

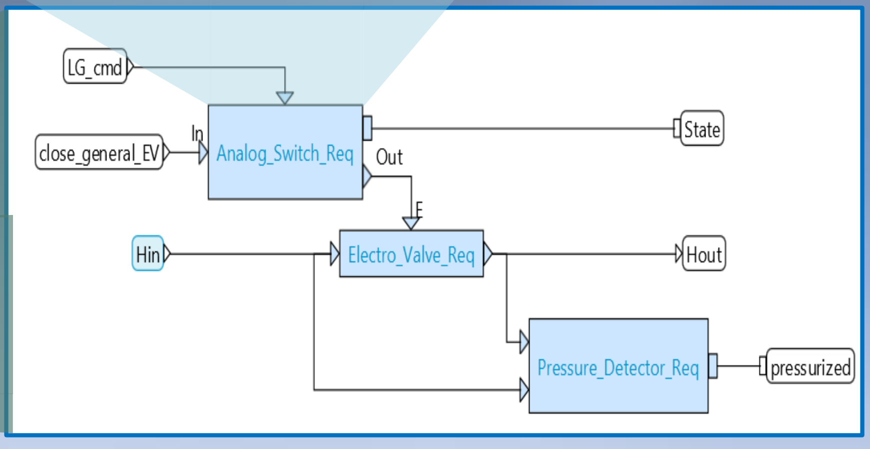
[AS_REQ_001] *When* (State *is* 'closed), Out *shall be* In *otherwise* Out *shall be* false

[AS_REQ_002] *Before* Handle , State *shall be* 'open

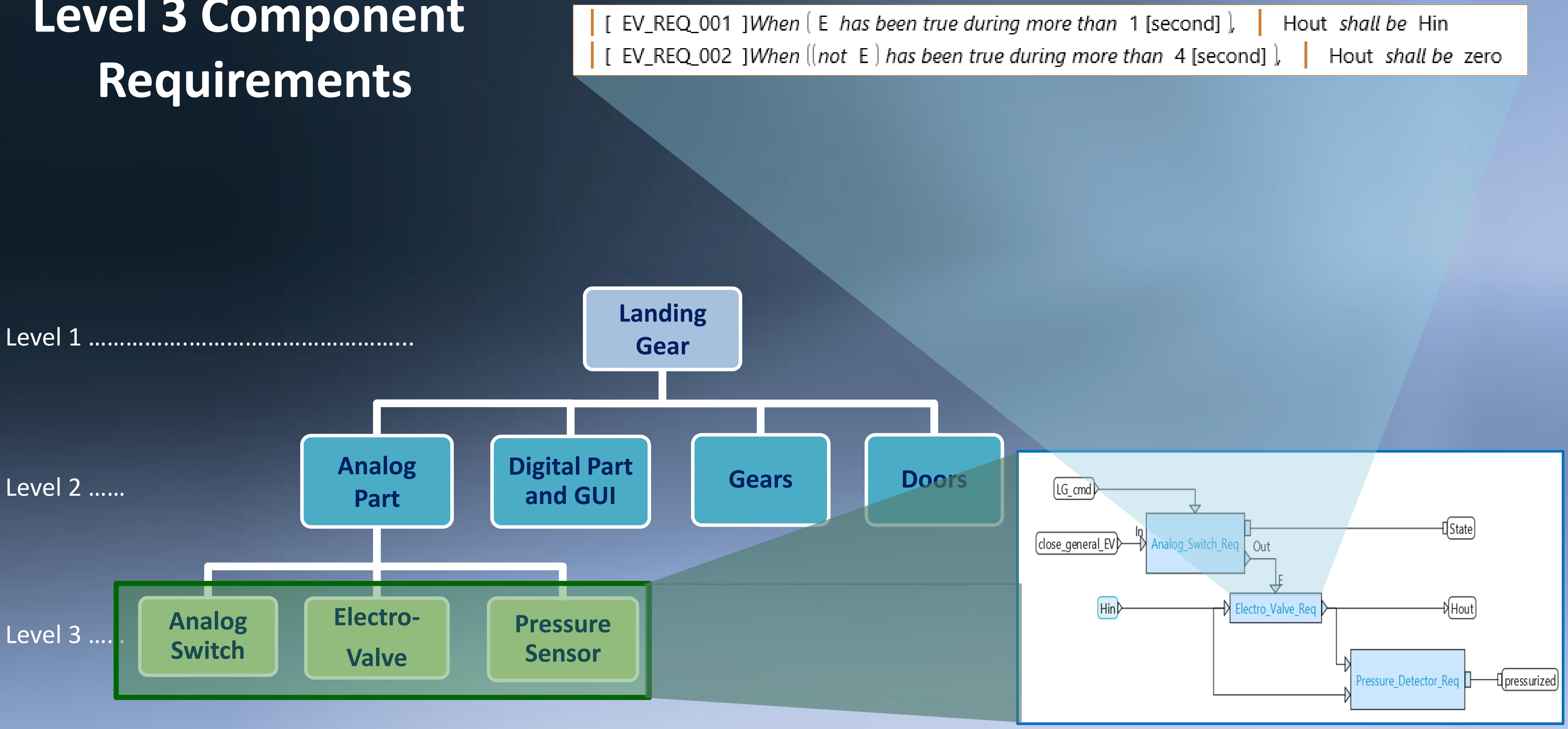
[AS_REQ_003] *From each* Handle , *Do*
(State *shall be* 'closed) *once within* 2 [second]
afterwards
During (20[second] - period), State *shall be* 'closed

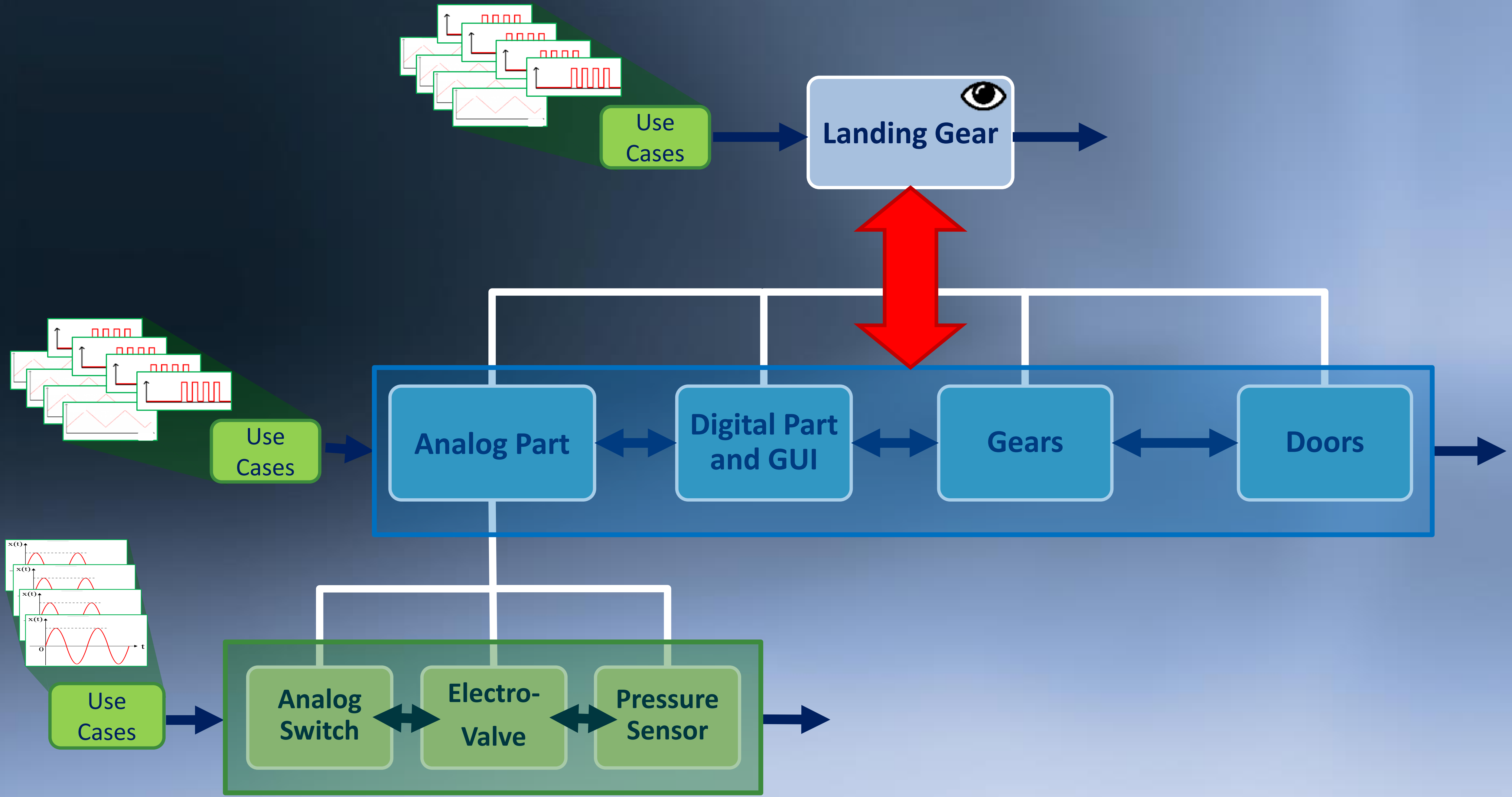
[AS_REQ_004] *When* (((*not* Handle) *and* (last State *is* 'closed)) *has been true during more than* 20 [second]), State *shall be* 'open

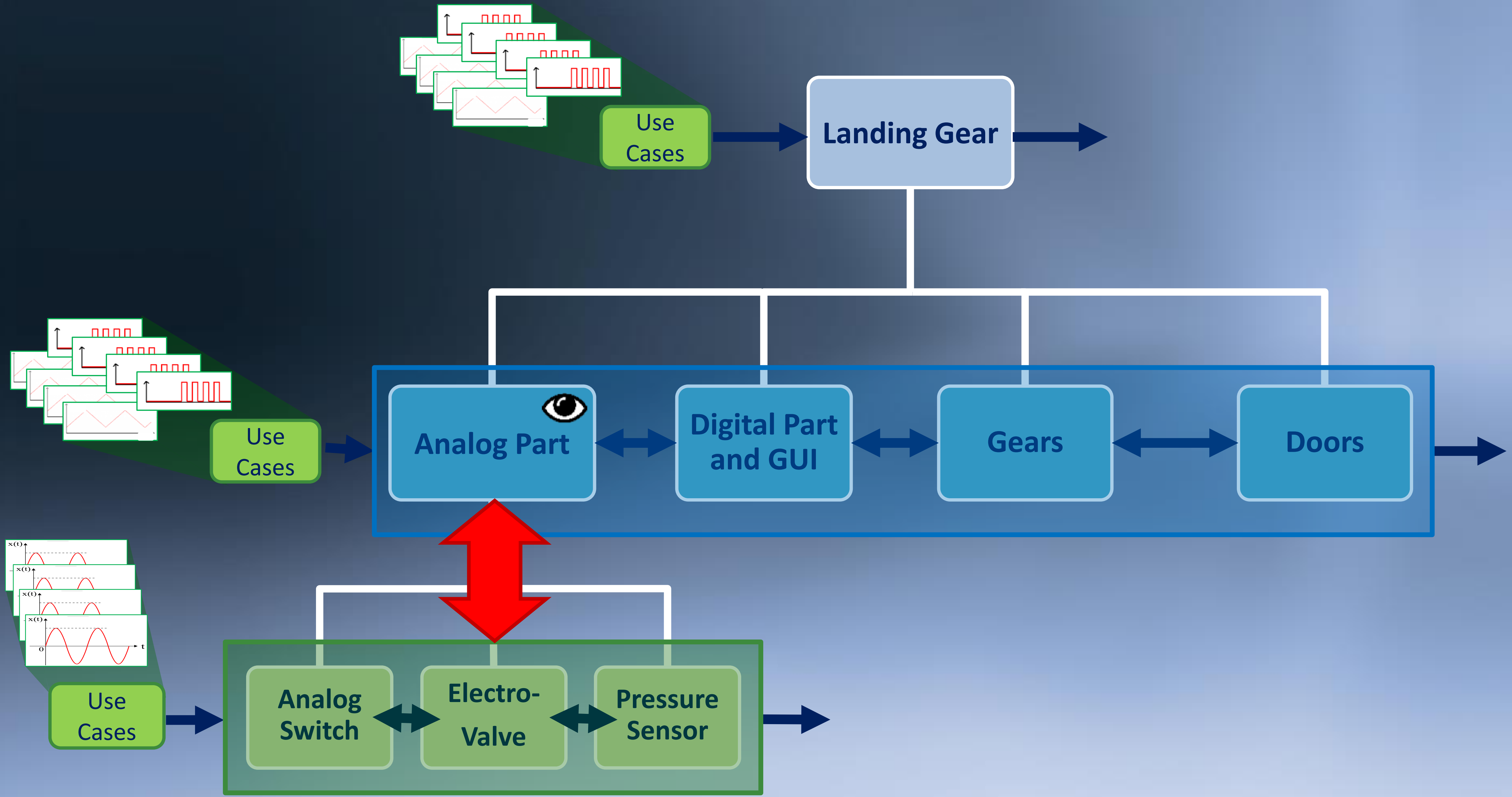




Level 3 Component Requirements




Use Cases
Landing Gear
Use Cases
Analog Part
Digital Part and GUI
Gears
Doors
Use Cases
Analog Switch
Electro-Valve
Pressure Sensor

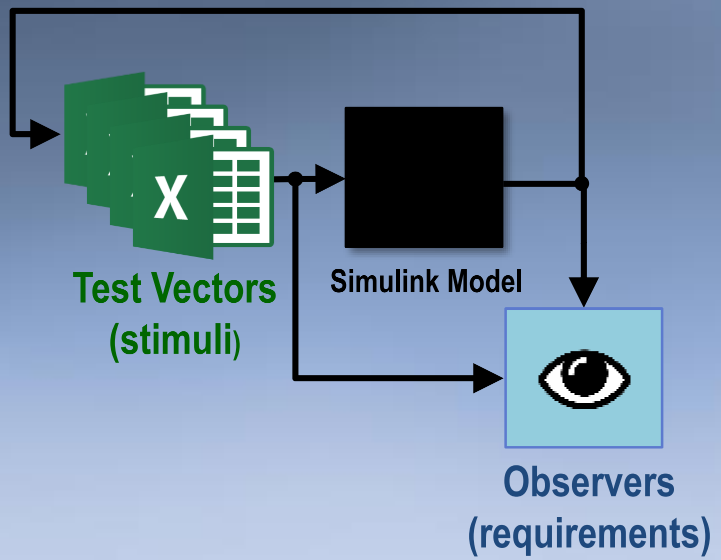


Use Cases
Landing Gear
Use Cases
Analog Part
Digital Part and GUI
Gears
Doors
Use Cases
Analog Switch
Electro-Valve
Pressure Sensor

STIMULUS – Simulink (or SCADE)

Step 1:

- Test automatically an FMU containing the Simulink model in STIMULUS
- STIMULUS identifies the violations of the specification
- Export the test cases leading to the errors



STIMULUS – Simulink (or SCADE)

Step 2:

- Use the generated test cases (csv) to debug the Simulink model in Simulink

SIMULINK