# Measurable Safety -
# Standards and Regulatory implications – A perspective
# SIP-adus 2020

**Gil Amid**
**Foretellix Ltd**

# The regulatory and liability prospective:

## Safe?

No Standards In Place

No Rating System In Place


**HOW ???**

**What to demand for certification?**

**What can be tested ?**

**What  data can be used ?**

**What is “safe enough”  ?**

**What is the required minimum ?**

**Simulation**

**X-in-the-Loop**

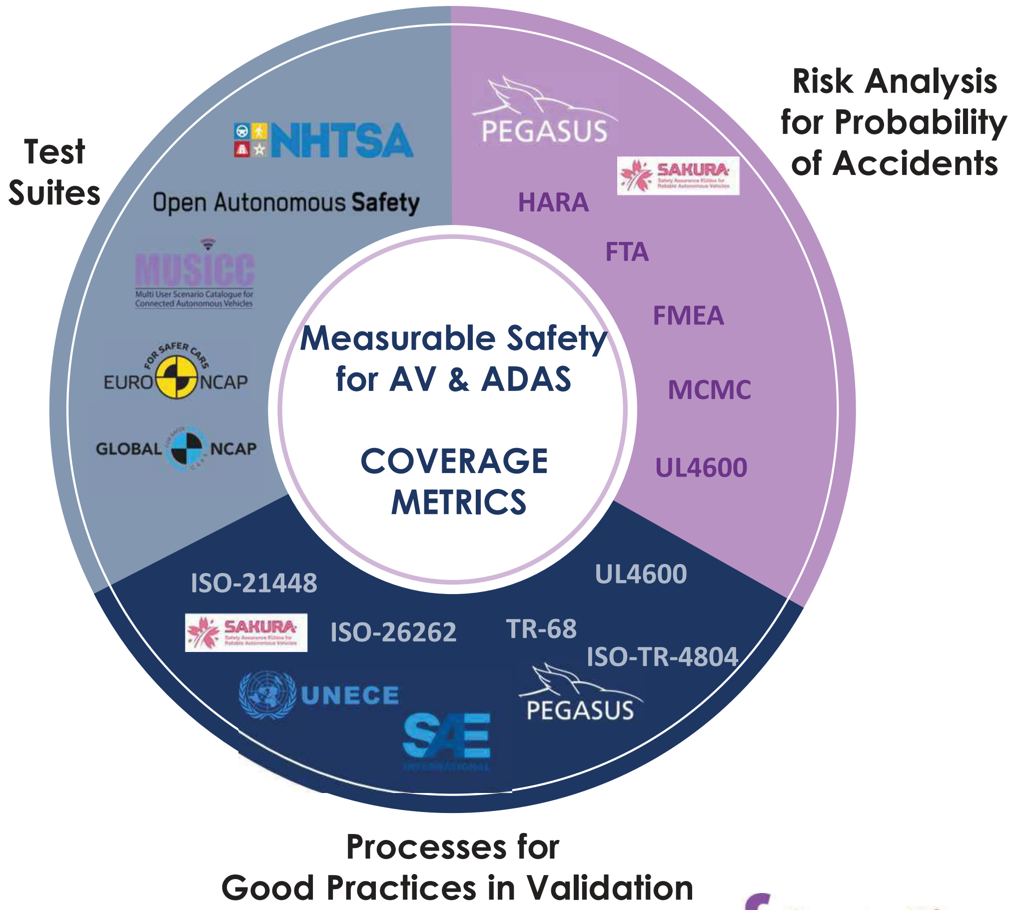
**Test Tracks**

**Test Driving**

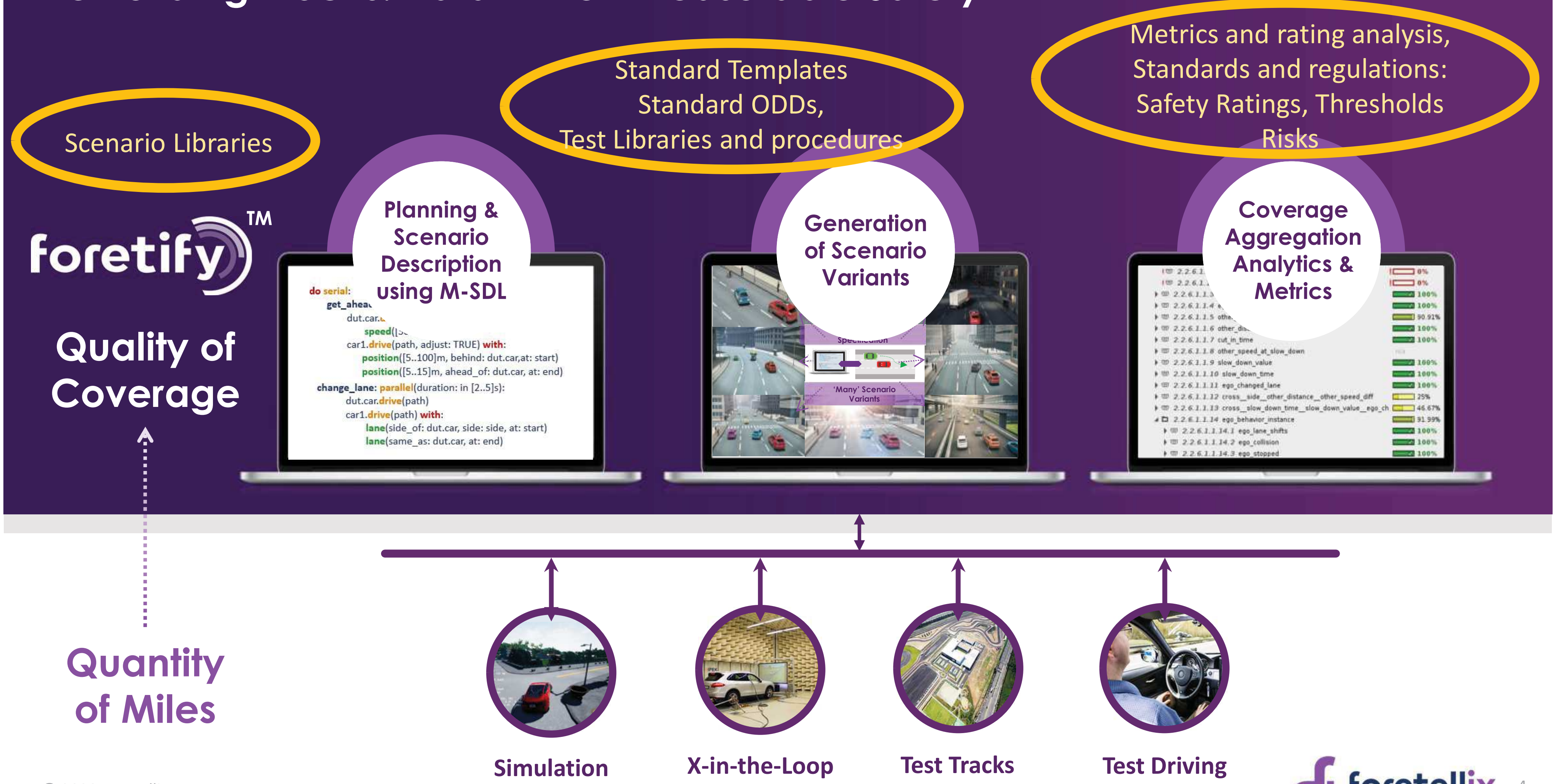
foretellix
verifying autonomous systems

# Building the AV Safety Argument

- Verification & validation metrics are needed for enabling the body of evidence required for building the AV's safety case
- Coverage Metrics measure what actually happens and provides scenario coverage aggregation analytics & metrics
- Coverage metrics supports all existing and emerging safety standards & processes



# The Building Blocks: Data Driven Measurable Safety



THANK YOU JAPAN FOR THE WIDE CONTRIBUTION IN ALL THESE DOMAINS

# The Building blocks are forming....

ISO Standards : 3450* , ASAM

UNECE/GRVA – New Assessment and Test Methods:

Scenario Catalogue → Scenario Libraries

Testing Methods → Standard Templates, Standard ODDs, Test Libraries and procedures

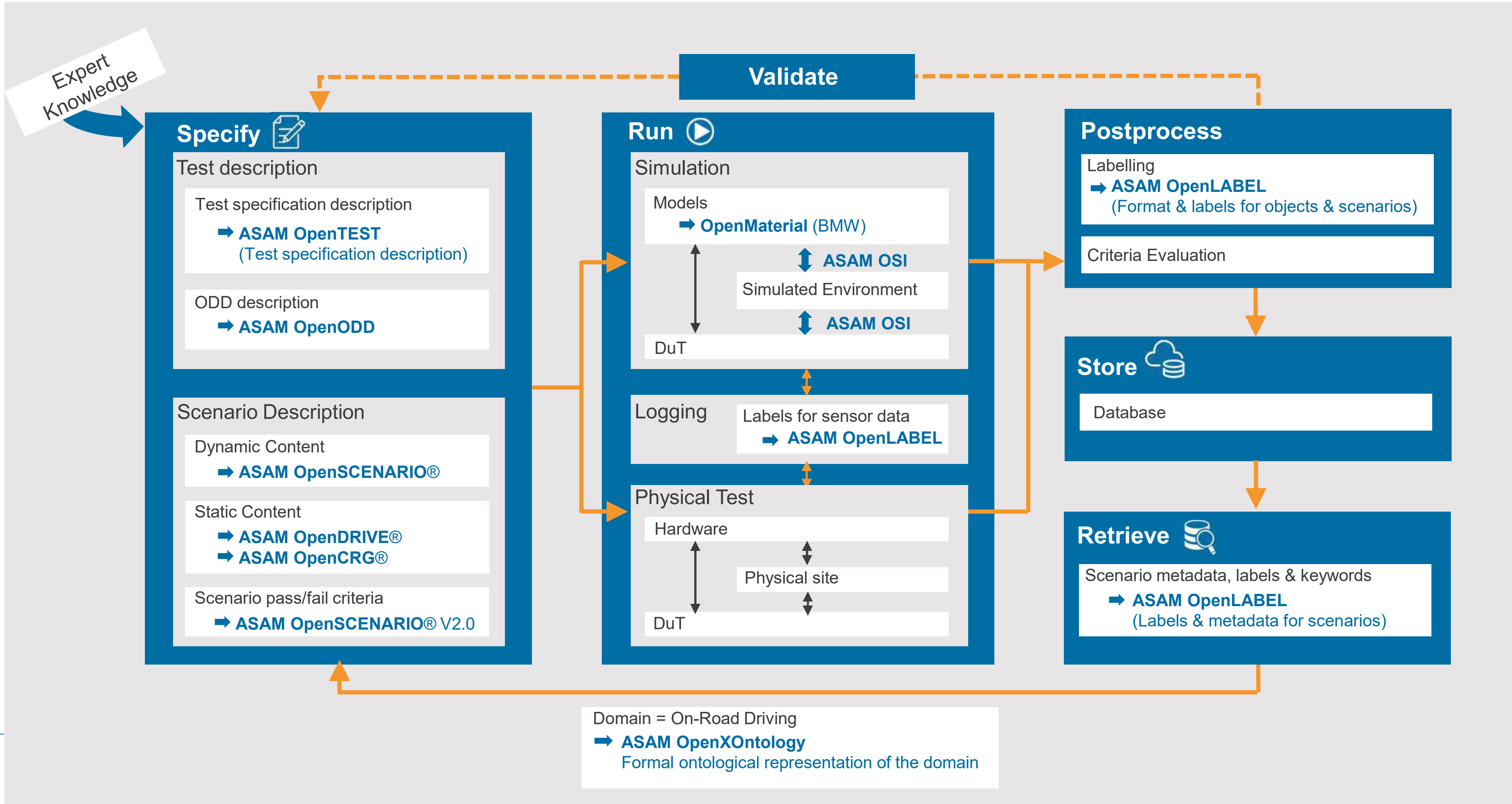
Regulatory Thresholds UNECE, ISO, SAE → Metrics and rating analysis, Standards and regulations: Safety Ratings, Thresholds Risks

ASAM OpenSCENARIO 2.0

ASAM OpenODD, ISO 34502

SAE – Safety metrics AVSC Foretellix's coverage analysis.

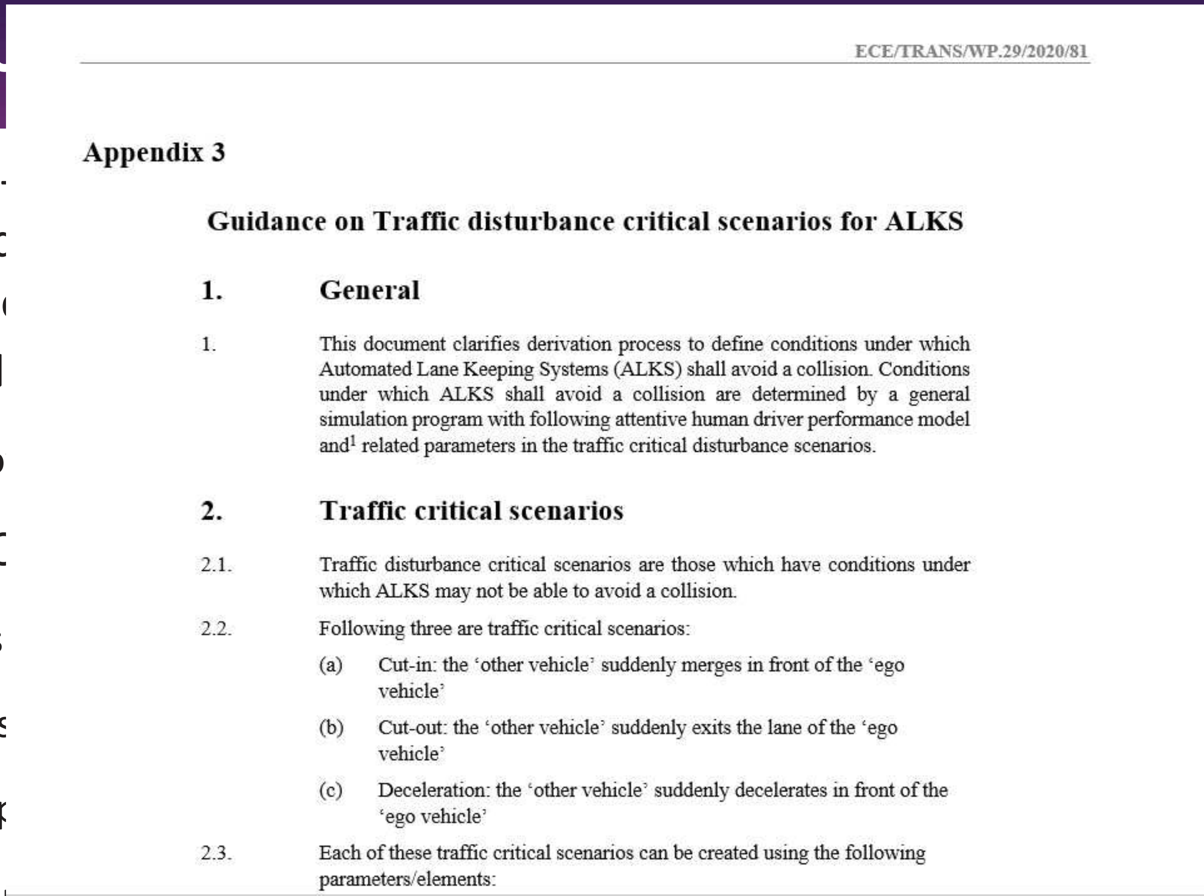
# ASAM Example Workflow for Scenario-Based Testing (SBT



# A Pragmatic Example:

# Applying CDV  to Verify Regulatory Compliance – ALKS regulation.

# ALKS U 157)


ECE/TRANS/WP.29/2020/81

Appendix 3

**Guidance on Traffic disturbance critical scenarios for ALKS**

**1. General**

1. This document clarifies derivation process to define conditions under which Automated Lane Keeping Systems (ALKS) shall avoid a collision. Conditions under which ALKS shall avoid a collision are determined by a general simulation program with following attentive human driver performance model and[1] related parameters in the traffic critical disturbance scenarios.

**2. Traffic critical scenarios**

2.1. Traffic disturbance critical scenarios are those which have conditions under which ALKS may not be able to avoid a collision.

2.2. Following three are traffic critical scenarios:

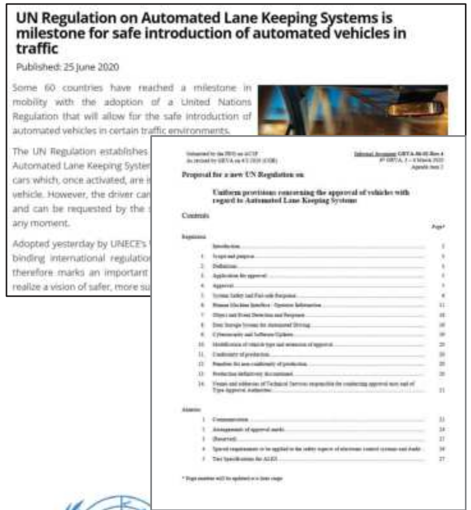
(a) Cut-in: the 'other vehicle' suddenly merges in front of the 'ego vehicle'

(b) Cut-out: the 'other vehicle' suddenly exits the lane of the 'ego vehicle'

(c) Deceleration: the 'other vehicle' suddenly decelerates in front of the 'ego vehicle'

2.3. Each of these traffic critical scenarios can be created using the following parameters/elements:

- ALKS e lateral and longituc ut further driver c
- This UN
  - Appro
- ALKS's C
  - Roads
  - A phys tions
  - The op




UNECE

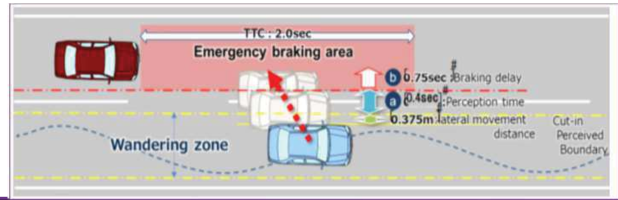
- The regulation specifies guidance for 3 critical scenarios for testing and simulation ( in addition to other testing requirements) – Specific contribution from Japan



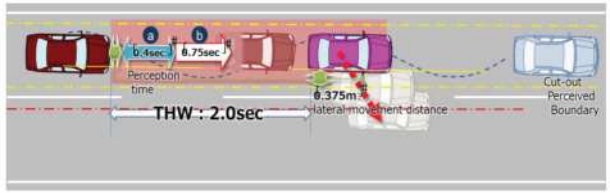
foretellix

# ALKS Scenarios

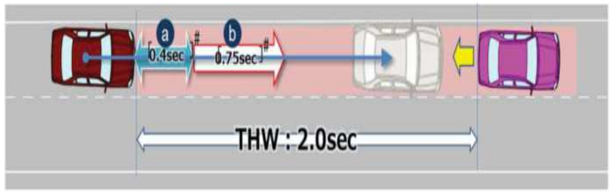
**Cut-in** - A car cuts-in to the ego's lane ( in front of  the ego)




**Cut-out** - A leading car cuts out in front of the ego



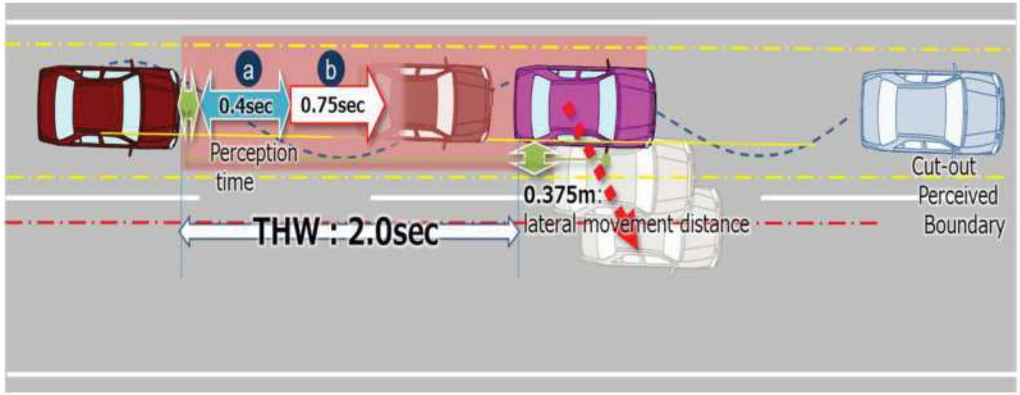

**Deceleration** - A leading car in front of the ego decelerates

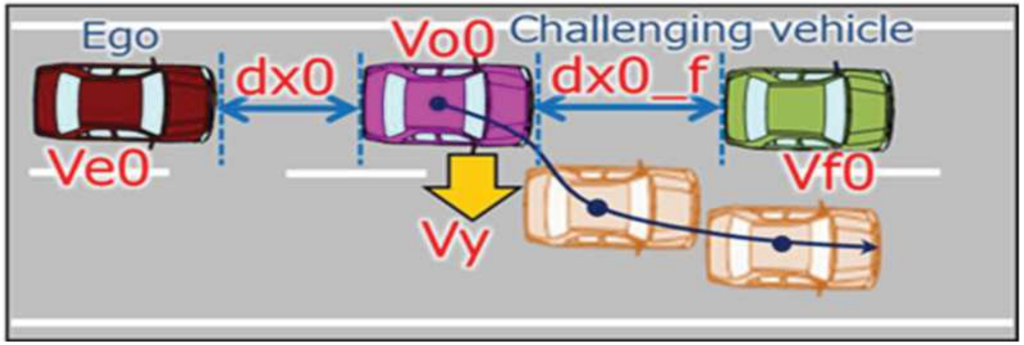


# Cut Out - Terminology and Notations

<u>Initial Velocity</u>
**Ve0** = Ego vehicle
**Vo0** = Leading vehicle in lane or in adjacent lane
**Vf0** = Vehicle in front of leading vehicle in lane

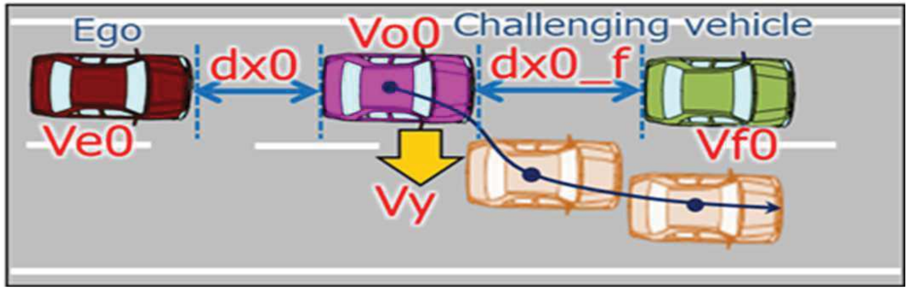
<u>Initial Distance</u>
**dx0** = Distance in Longitudinal direction between the front end of the ego vehicle and the rear end of the leading vehicle
**dx0_f** = Distance in longitudinal direction between front end of leading vehicle and rear end of vehicle in front of leading vehicle



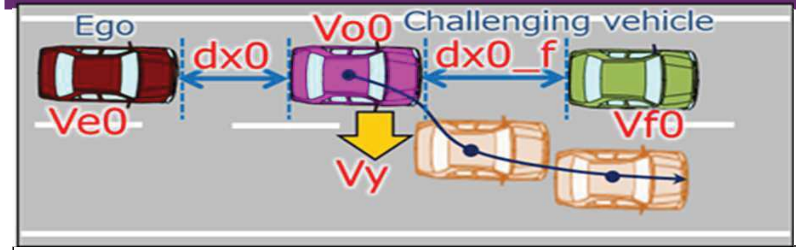

**Vy** =Leading vehicle lateral velocity

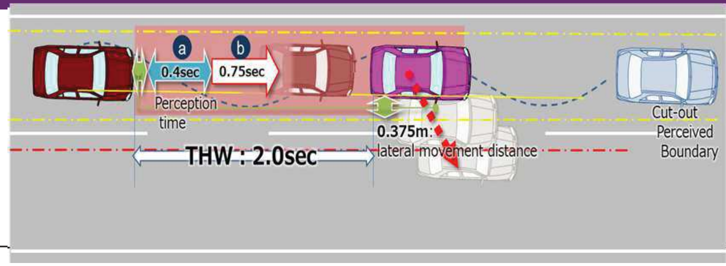
# M-SDL Cut Out Scenario Implementation




```
do serial():

    dut_speed_up: parallel( duration: [6..10]second):
        dut.car.drive(path: path) with:
            ego_mode(alk)
        other_car.drive(path: path, adjust: false)
        in_front_car.drive(path:path)

    lead: parallel(duration: [1..3]second):
        dut.car.drive(path: path) with:
            ego_mode(alk)
        other_car.drive(path: path, adjust: false) with:
            lane(same_as: dut.car)
            position(time: [THW..THW], ahead_of: dut.car, at:end)
            speed([0..0]kph, faster_than: dut.car, at: end )
        in_front_car.drive(path: path, adjust: false) with:
            lane(same_as: other_car)
            speed([0..0]kph)
            position([dxo_f+in_front_car.length ,ahead_of:other_car, at:end )

    cut_out: parallel(duration: [1..4]second):
        dut.car.drive(path: path)
        other_car.drive(path: path, adjust: false) with:
            change_lane()
        in_front_car.drive(path: path, adjust: false) with:
            keep_lane()
            speed(speed: [0..0]kph)
```

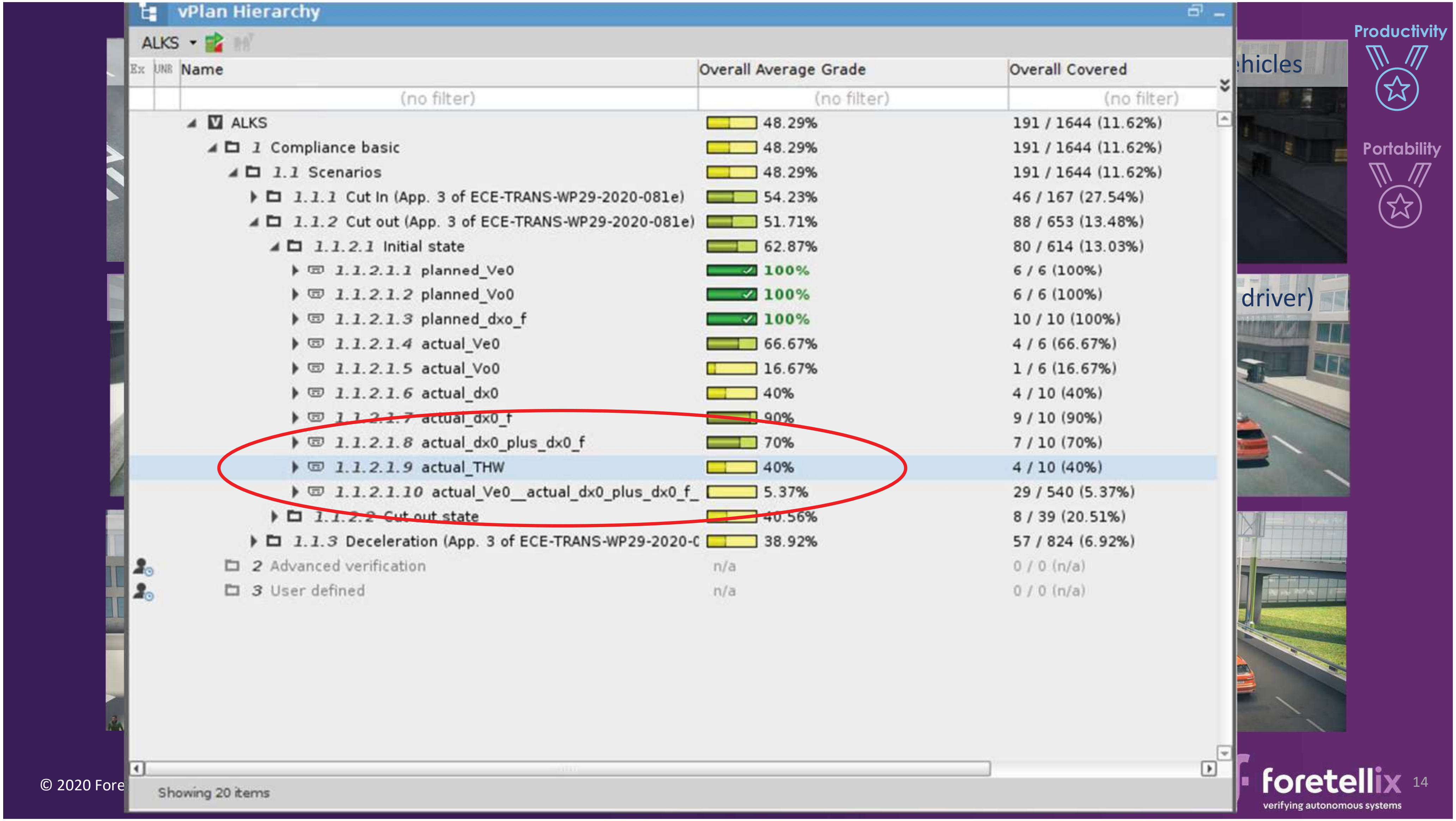
# Cut Out- Coverage and Measurements Definitions




```
!actual_ttc := sample(get_min_ttc(),@cut_out.end) with:
  cover(it,unit:ms,every: 100,range:[0..3000],text:"Minimal time to collision for ego car")

!actual_Ve0 := sample(dut.car.state.speed,@lead.end) with:
  cover(it,unit:kph,range:[0..60],every:10,text:"Actual velocity of ego at cut out start (can go up to 60kph by spec)")


!actual_Vy := sample(other_car.state.avg_lateral_speed,@cut_out.end) with:
      cover(it,unit:kph,range:[1..10],every:1,text:"Actual lateral speed of the cutting out car")

!actual_THW := sample(actual_dx0/actual_Ve0,@lead.end) with:
    cover(it, unit:millisecond, range:[0..5000], every:500, text : "Actual THW when cut-out car starts cutting-out")
```


Pedestrians | Bicyclists
Rain
Low light | Different vehicles
'Many' Scenario Coverage Monitors Across Many Tests & Platforms
Urban roads (Curved road)
driver behaviors (Drunk driver)
'One' Scenario Coverage Metric Dashboard
Urban roads (junction)
Sun glare
Highways



vPlan Hierarchy
ALKS
Ex UNR Name | Overall Average Grade | Overall Covered
(no filter) (no filter) (no filter)
ALKS 48.29% 191 / 1644 (11.62%)
1 Compliance basic 48.29% 191 / 1644 (11.62%)
1.1 Scenarios 48.29% 191 / 1644 (11.62%)
1.1.1 Cut In (App. 3 of ECE-TRANS-WP29-2020-081e) 54.23% 46 / 167 (27.54%)
1.1.2 Cut out (App. 3 of ECE-TRANS-WP29-2020-081e) 51.71% 88 / 653 (13.48%)
1.1.2.1 Initial state 62.87% 80 / 614 (13.03%)
1.1.2.1.1 planned_Ve0 100% 6 / 6 (100%)
1.1.2.1.2 planned_Vo0 100% 6 / 6 (100%)
1.1.2.1.3 planned_dxo_f 100% 10 / 10 (100%)
1.1.2.1.4 actual_Ve0 66.67% 4 / 6 (66.67%)
1.1.2.1.5 actual_Vo0 16.67% 1 / 6 (16.67%)
1.1.2.1.6 actual_dx0 40% 4 / 10 (40%)
1.1.2.1.7 actual_dx0_f 90% 9 / 10 (90%)
1.1.2.1.8 actual_dx0_plus_dx0_f 70% 7 / 10 (70%)
1.1.2.1.9 actual_THW 40% 4 / 10 (40%)
1.1.2.1.10 actual_Ve0__actual_dx0_plus_dx0_f_ 5.37% 29 / 540 (5.37%)
1.1.2.2 Cut out state 40.56% 8 / 39 (20.51%)
1.1.3 Deceleration (App. 3 of ECE-TRANS-WP29-2020-0 38.92% 57 / 824 (6.92%)
2 Advanced verification n/a 0 / 0 (n/a)
3 User defined n/a 0 / 0 (n/a)
Showing 20 items
hicles
driver)
Productivity
Portability
foretellix
verifying autonomous systems

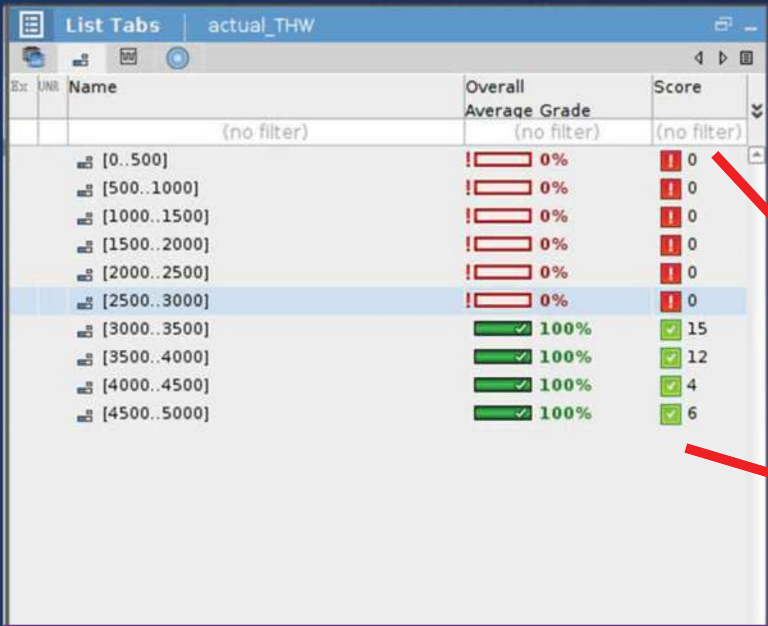
# Non-RSS vs. RSS Controlled Ego

- In testing different EGOs, we have few examples where RSS controlled behavior is preventing a collision ( keeping the ego out of “unpreventable” space)

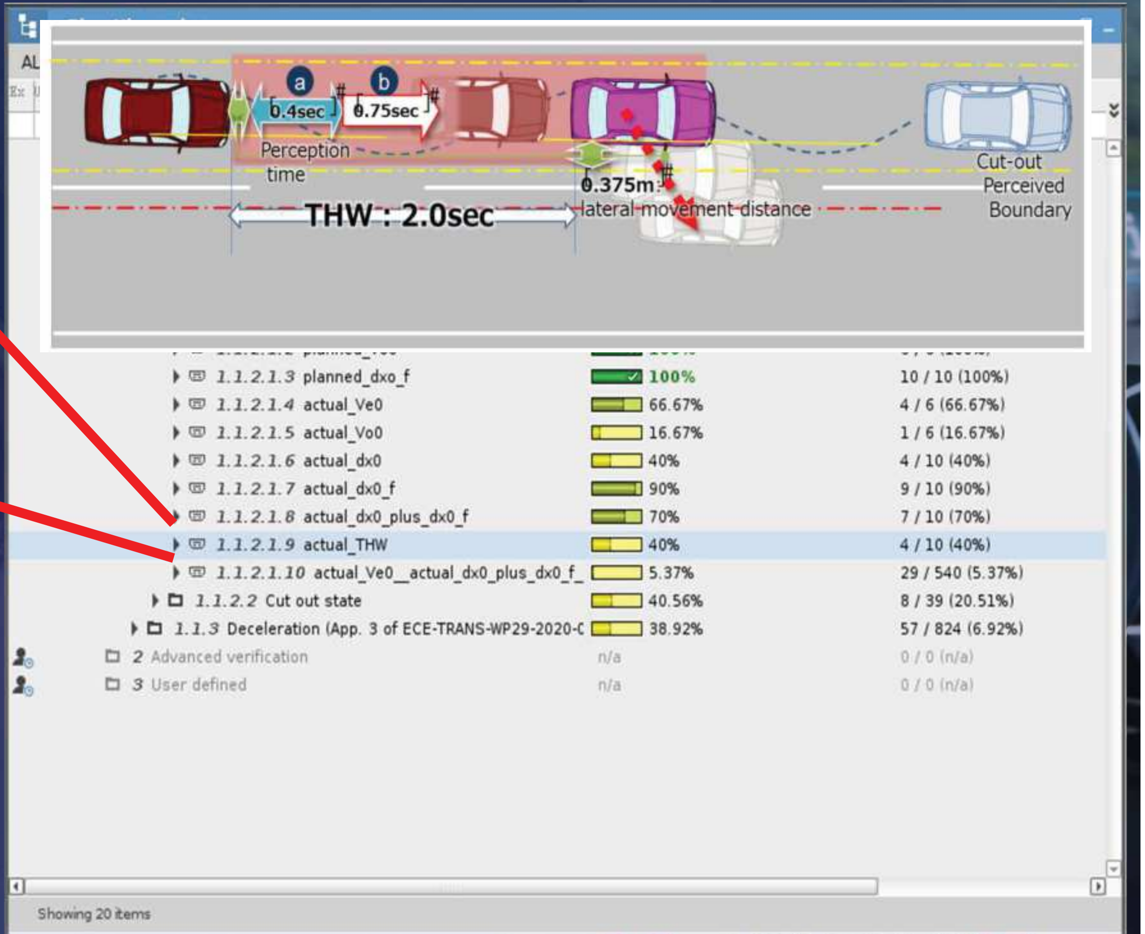
No RSS


RSS Controlled Ego

# THW COVERAGE/TESTING HOLE




**In All Tests, THW > 3s**
**Testing does not meet regulatory spec !**






verifying autonomous systems

# Re-tuning EGO Parameters:   THW issue solved

- Re-tuning solved the issue

Before

List Tabs | actual_THW

| Name | Overall Average Grade | Score |
|---|---|---|
| (no filter) | (no filter) | (no filter) |
| [0..500] | 0% | 0 |
| [500..1000] | 0% | 0 |
| [1000..1500] | 0% | 0 |
| [1500..2000] | 0% | 0 |
| [2000..2500] | 0% | 0 |
| [2500..3000] | 0% | 0 |
| [3000..3500] | 100% | 15 |
| [3500..4000] | 100% | 12 |
| [4000..4500] | 100% | 4 |
| [4500..5000] | 100% | 6 |

Showing 10 items

After

List Tabs | actual_THW

| Name | Overall Average Grade | Score |
|---|---|---|
| (no filter) | (no filter) | (no filter) |
| [0..500] | 100% | 2 |
| [500..1000] | 100% | 35 |
| [1000..1500] | 100% | 171 |
| [1500..2000] | 100% | 103 |
| [2000..2500] | 100% | 44 |
| [2500..3000] | 100% | 31 |
| [3000..3500] | 100% | 22 |
| [3500..4000] | 100% | 12 |
| [4000..4500] | 100% | 7 |
| [4500..5000] | 100% | 7 |
| [5000..5500] | 100% | 2 |
| [5500..6000] | 100% | 1 |
| [6000..6500] | 100% | 1 |
| [6500..7000] | 0% | 0 |
| [7000..7500] | 100% | 1 |
| [7500..8000] | 100% | 1 |

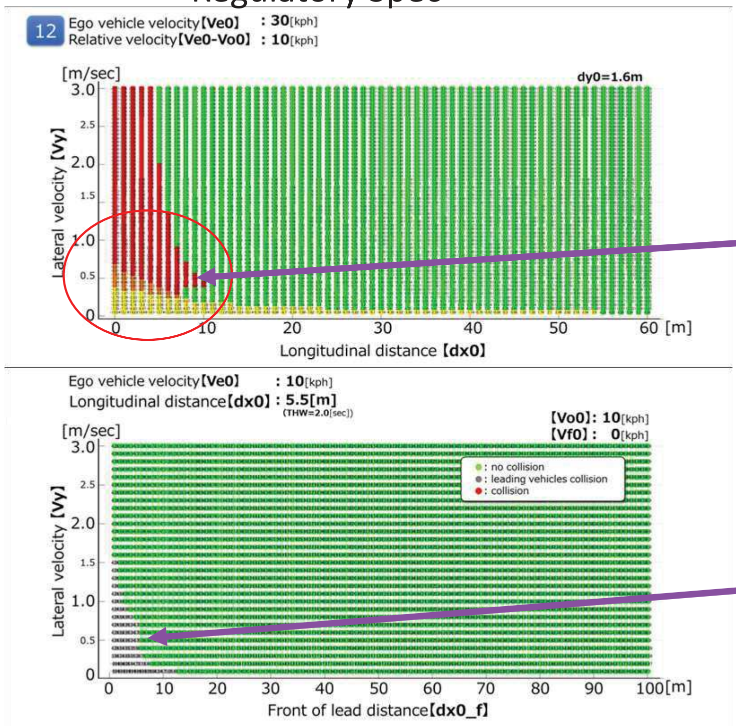
Showing 16 items

**Very good coverage of all risky areas , and regulatory spec.**

# Regulatory Specs vs. Coverage Slice

The regulations includes “expected behaviors” for different ranges – coverage data shows that RSS controlled ego is within these “expected results”

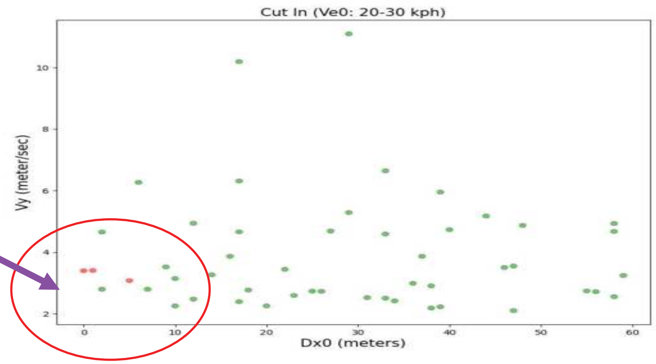
Regulatory Spec

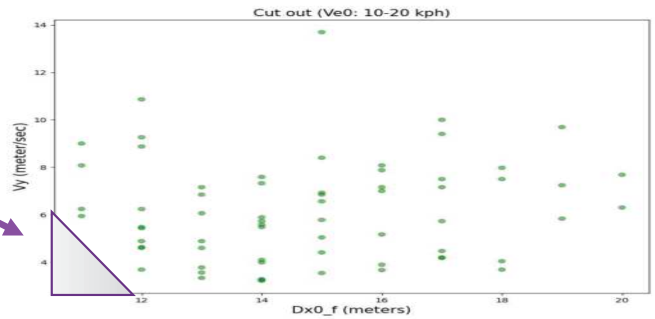



Cut-in:
Unpreventable
Collisions from
the side.

Cut-out:
RSS is avoiding
The collision
area

Coverage Samples






foretellix

# The Building Blocks: Data Driven Measurable Safety

Scenario Libraries

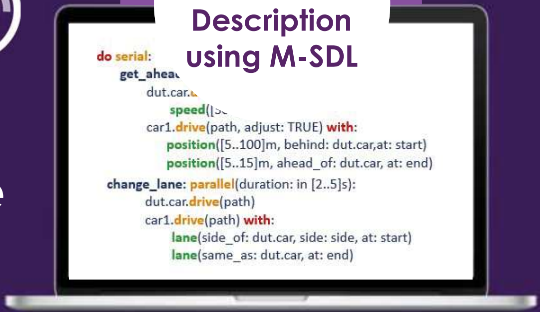
Standard Templates
Standard ODDs,
Test Libraries and procedures

Metrics and rating analysis,
Standards and regulations:
Safety Ratings, Thresholds
Risks

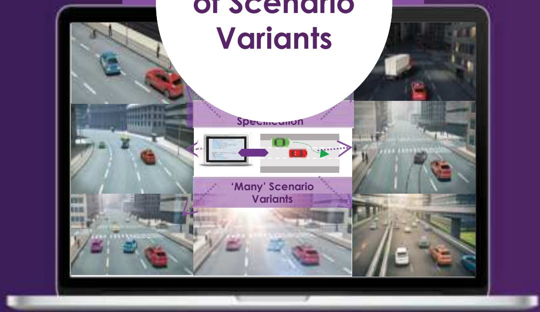
foretify™

Quality of Coverage

Planning & Scenario Description using M-SDL




Generation of Scenario Variants




Coverage Aggregation Analytics & Metrics

Quantity of Miles

Simulation

X-in-the-Loop

Test Tracks

Test Driving

# Summary: Measurable Safety – Coverage Metrics

- **Usage of [Coverage] Metrics Supplies:**
  - **Goals for testing and certification**
  - **Threshold of quality and safe behaviors**
  - **Relative comparison between AVs**

- Regulators seek Using standard templates, standard testing libraries and ODDs – in order to ensure you have a complete, **measurable**, certification system

foretellix
verifying autonomous systems

# For More Information

www.Foretellix.com
info@foretellix.com
blog.foretellix.com

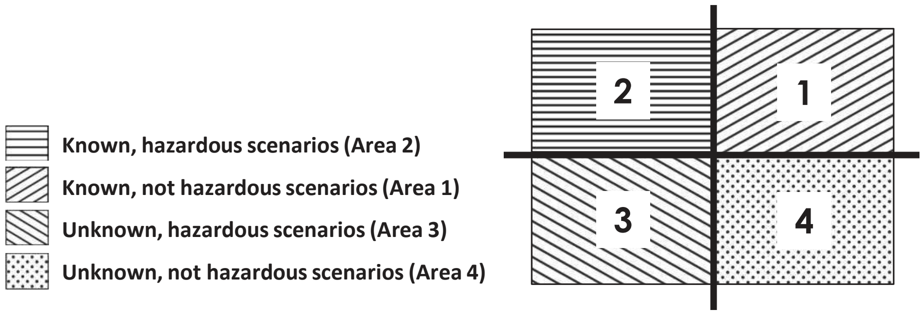
foretellix
verifying autonomous systems

# Safety Of The Intended Functionality (SOTIF)

"Absence of unreasonable risk due to hazards resulting from functional insufficiencies of the intended functionality or from reasonably foreseeable misuse by persons"

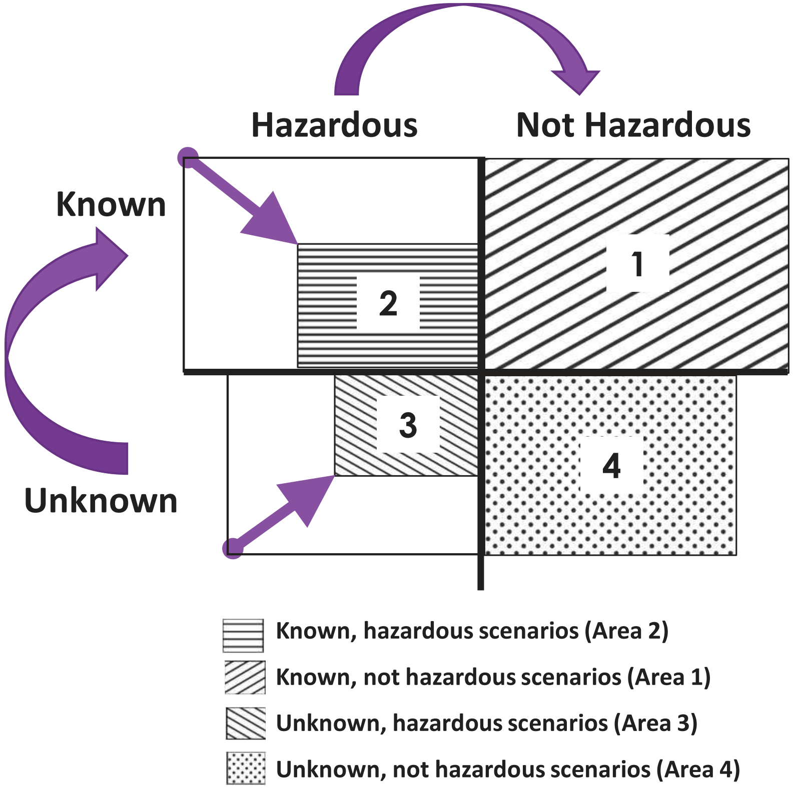


- SOTIF (ISO 21448) is dealing with Safety of Autonomous Systems, and provides guidance on design, verification, and validation measures
- SOTIF breaks down the possible scenario space to 4 categories
- "The ultimate goal is to evaluate the safety in **area 2 and area 3** and to provide an argument that these areas are **sufficiently small and the resulting residual risk is acceptable"**

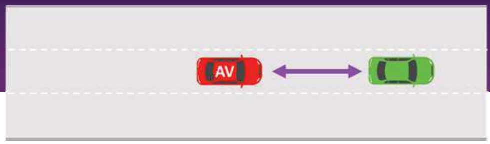


# foretify™ – The Full SOTIF Flow

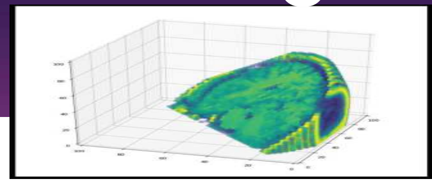
- Foretify™ is an automation and analysis tool, implementing the Coverage Driven Verification methodology
- Foretify™ provides a systematic approach to reduce both area 2 and area 3
- Foretify™ supports the SOTIF process, intended for reaching acceptable levels of risk



# KPI/Measurement vs Coverage







- How did the AV perform within a given ODD?
- KPI/Metrics specify the specific measurements to be analyzed, given specific test conditions /ODD. Usually – "simulation output"
- **<u>Answering:</u>**
  - In ODD X, How did the ego perform for all test variations in the context of "cut in" ? ( aggregate of all specific measurement )
  - What was TTC, when the AV was driving at 55kph, and the other player deceleration was -3 m/s^2 ? Is it above my threshold ?

- What was actually tested, out of the possible space of testing values [per ODD]
- Coverage can be measured both on test input/settings ,as well on output/results of the tests. It can be measure on one ,two, or multiple dimensions
- **<u>Answering:</u>**
  - For "cut in" scenario, on a road with 2 lanes and only green cars, what % of the possible AV speeds between 50KPH and 100KPH did I test ?
  - What % of the TTC space between 0 and 3S was demonstrated during all tests ?