MATLAB EXPO 2019 Bern

KYBURZ

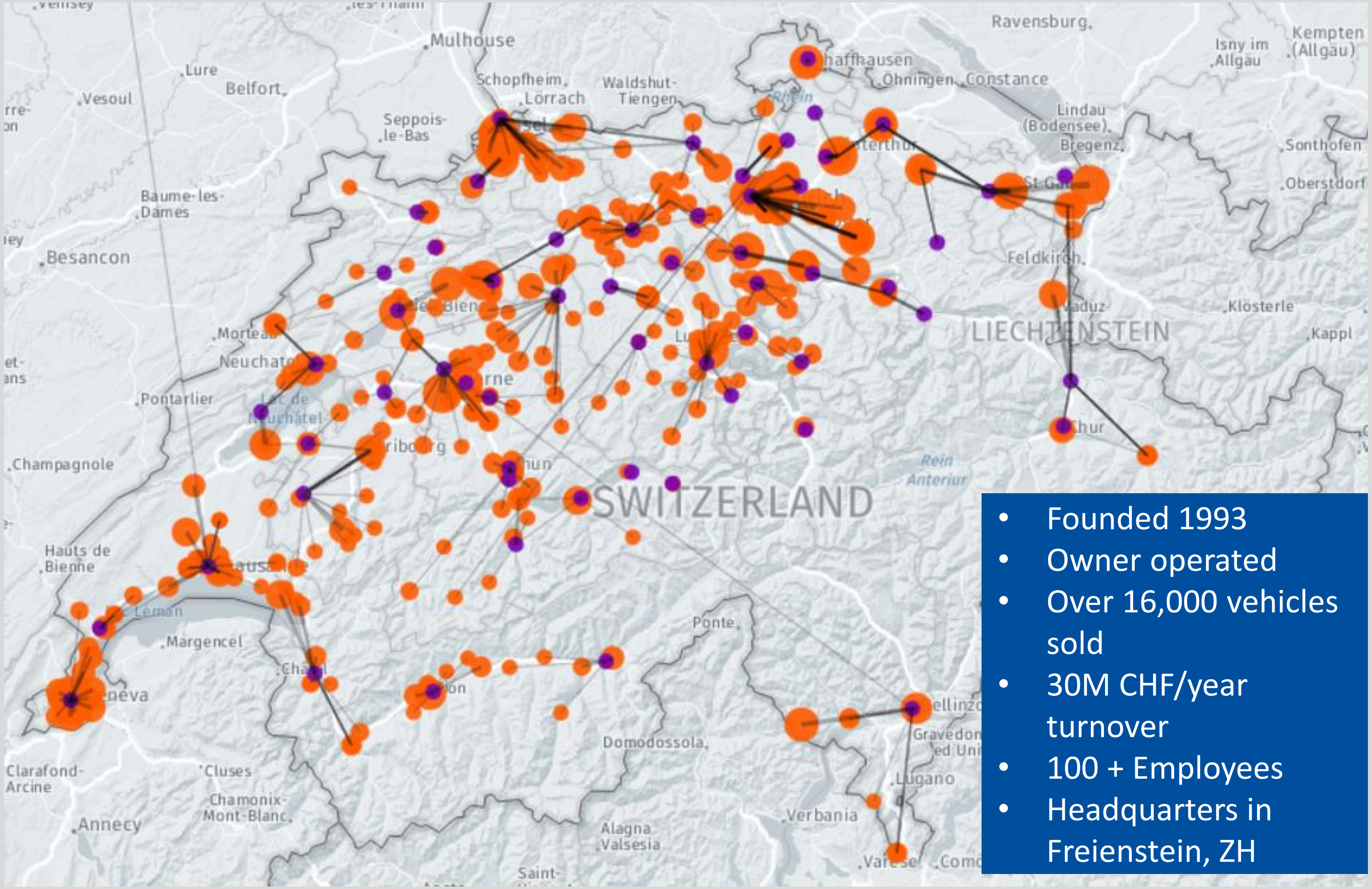
Courtesy of ETH-Zurich, ARC

Designing and controlling safe self-driving systems

Dr. Erik Wilhelm
Head of Research
KYBURZ Switzerland

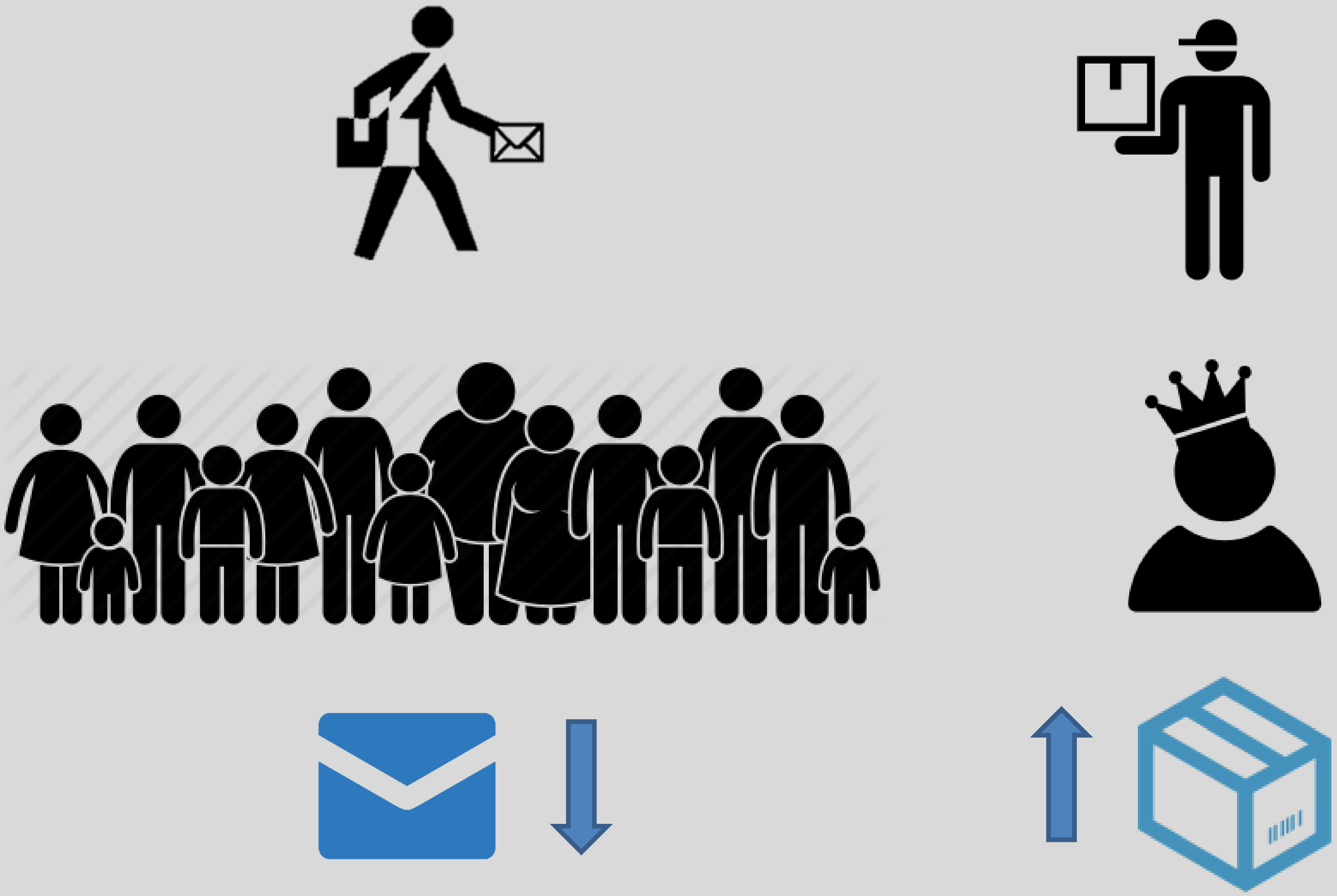
23rd May, 2019

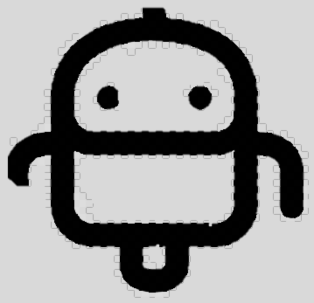
A well-established brand



- Founded 1993
- Owner operated
- Over 16,000 vehicles sold
- 30M CHF/year turnover
- 100 + Employees
- Headquarters in Freienstein, ZH

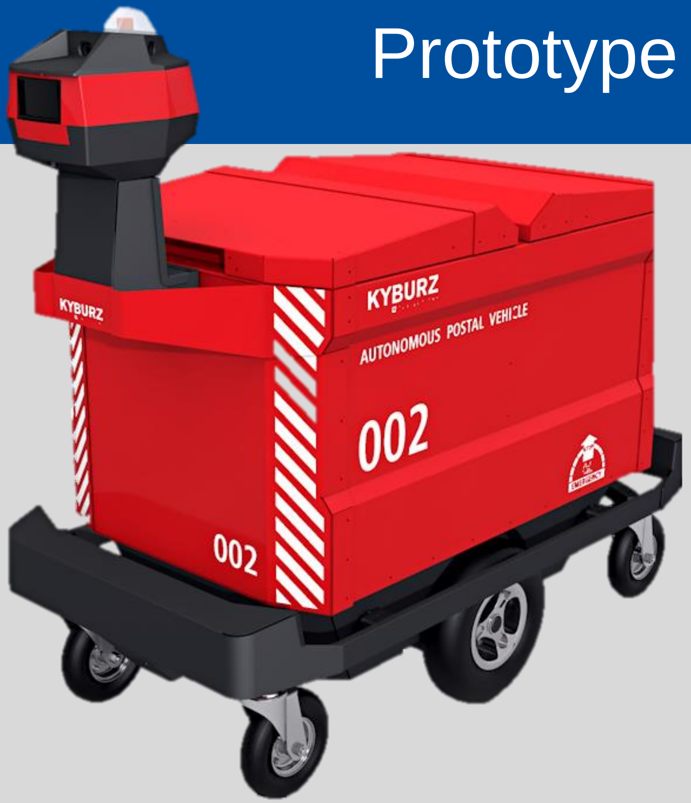
Changing postal delivery landscape



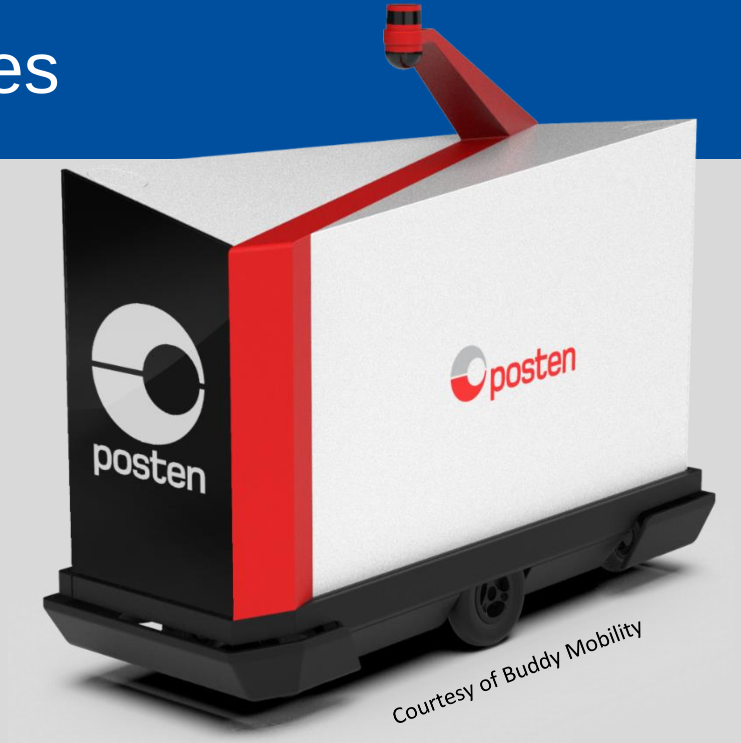


- Must be:
  - Cheaper
  - Faster
  - More reliable
  - ... More personal?

Prototype series

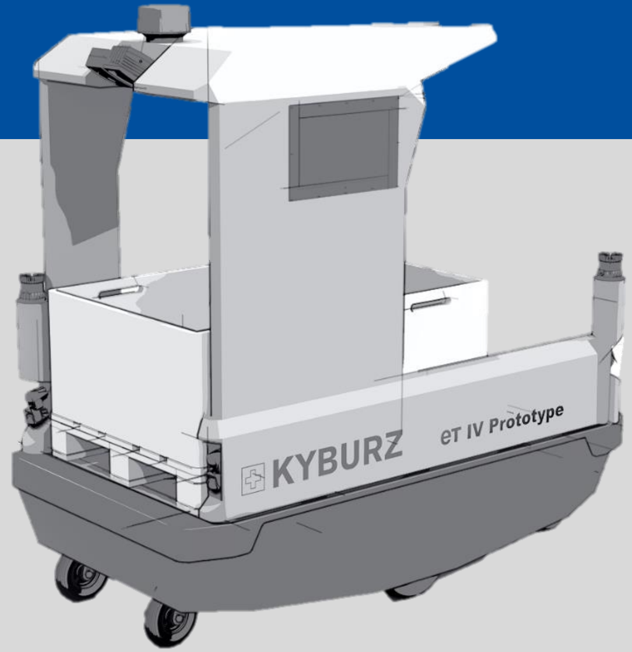


- Mobile depot box (eT2)
- Sensors
  - 2D Lidar
  - Ultrasonic
  - 360 camera
  - GPS
  - Bump-stop



Courtesy of Buddy Mobility

- Autonomous delivery agent (eT3)
- Sensors
  - 3D Lidar
  - Ultrasonic
  - Infraded
  - INS
  - Bump-stop



- Flexible delivery system (eT4)
- Sensors
  - 3D Lidar (2x)
  - Ultrasonic (8x)
  - Infrared (8x)
  - Radar (4x)
  - GPS (INS)
  - 360 Cameras (localization)
  - 360 Cameras (comprehension)
  - Time-of-flight camera
  - Bump-stop

Autonomous System Design Challenges

High availability

Image: ABC news

Ap(proved) safety

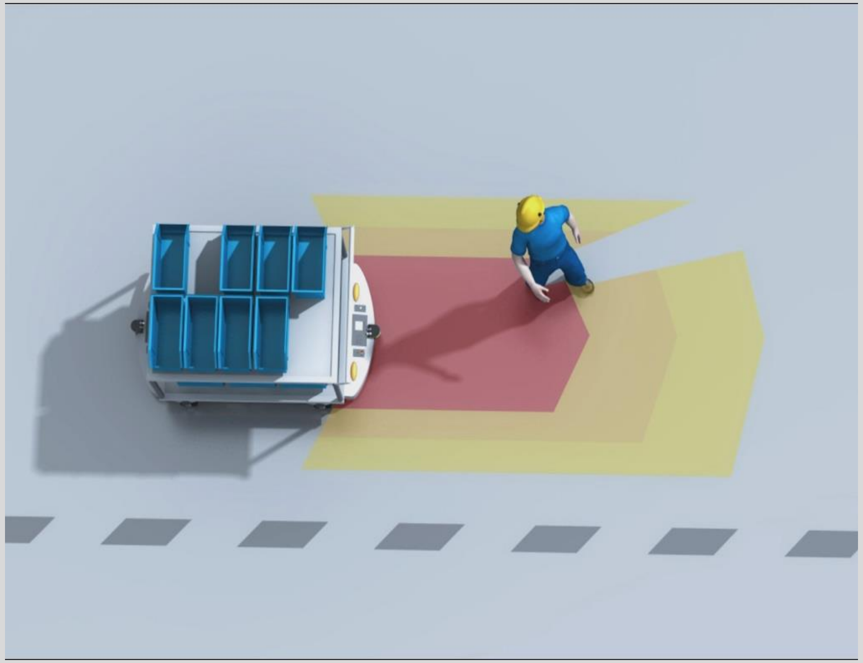


Image: sick.com

Test coverage

Image: youtube.com

Availability Requirement

Image: Frugal Entrepreneur

- 300 parcels/day
- 8.25 hr/day
- 56 kCHF/year

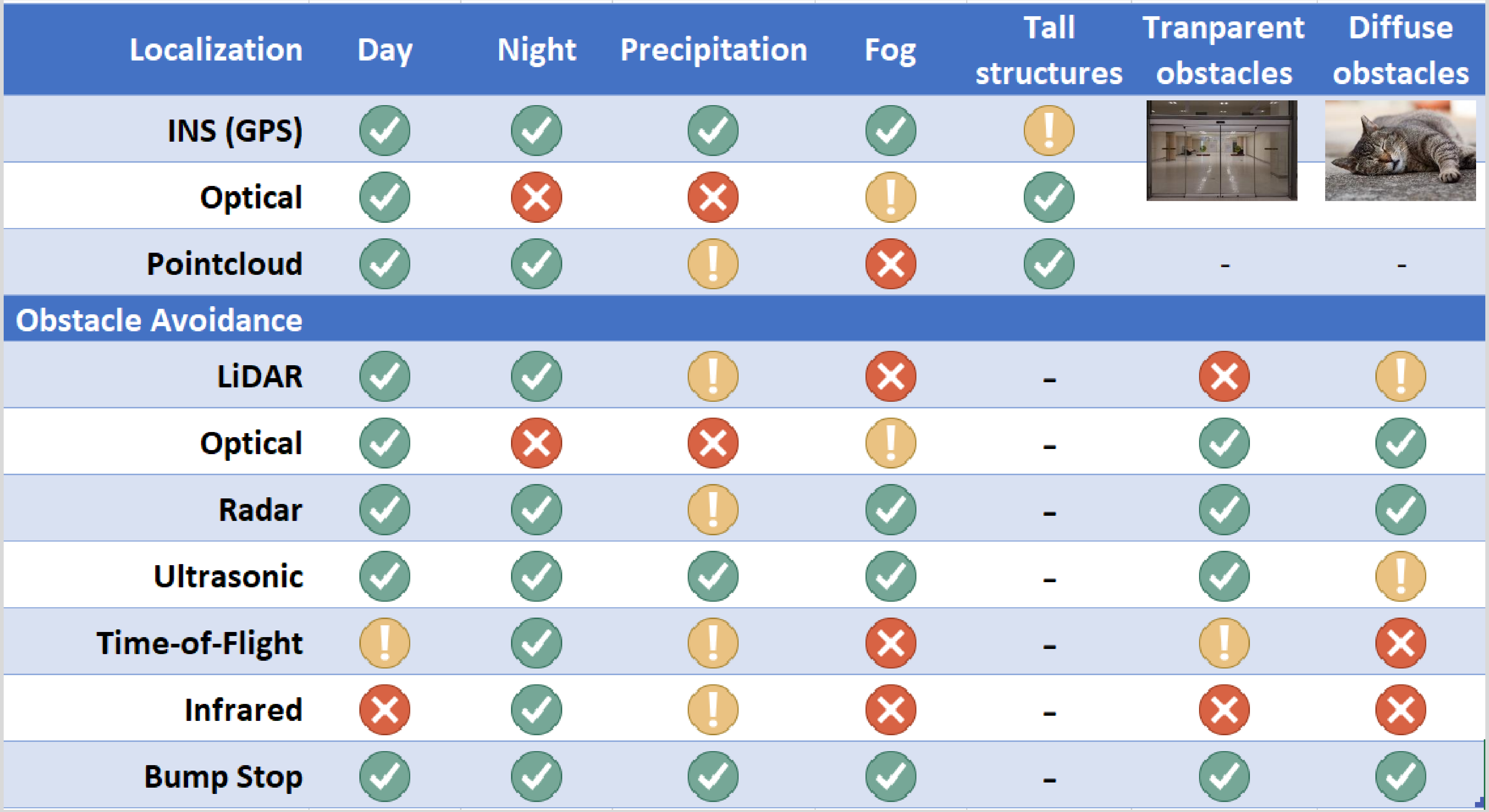
- 40 parcels/day
- 24 hr/day
- 50 kCHF purchase

Image: cnbc.com

- 1 disengagement/day
- 56 hours per year
- 3 kCHF per year

- Robotic delivery amortized with 1 disengagement/day, never with 3 disengagements/day

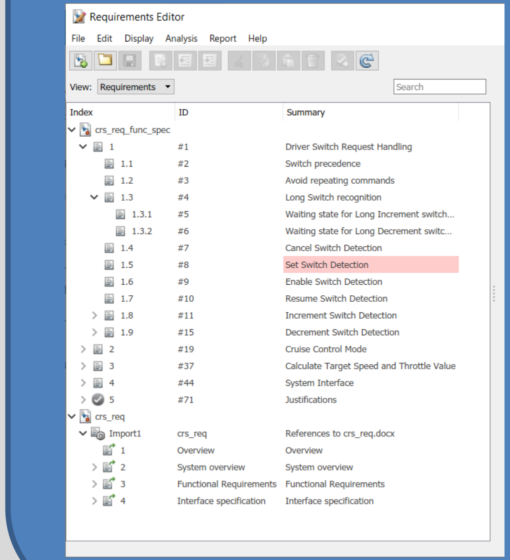
Sensor and controller redundancy



Localization	Day	Night	Precipitation	Fog	Tall structures	Tranparent obstacles	Diffuse obstacles
INS (GPS)	✓	✓	✓	✓	!		
Optical	✓	✗	✗	!	✓		
Pointcloud	✓	✓	!	✗	✓	-	-
Obstacle Avoidance							
LiDAR	✓	✓	!	✗	-	✗	!
Optical	✓	✗	✗	!	-	✓	✓
Radar	✓	✓	!	✓	-	✓	✓
Ultrasonic	✓	✓	✓	✓	-	✓	!
Time-of-Flight	!	✓	!	✗	-	!	✗
Infrared	✗	✓	!	✗	-	✗	✗
Bump Stop	✓	✓	✓	✓	-	✓	✓

Workflow

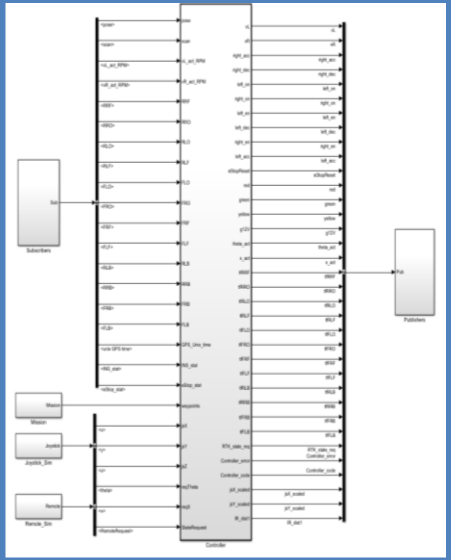
Specifications



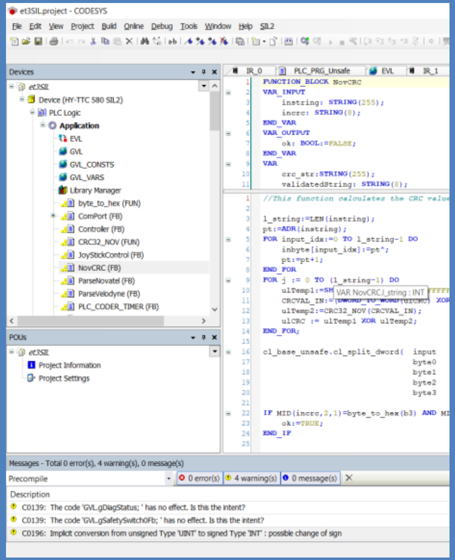
Requirements Editor

Index	ID	Summary
crs_req_func_spec		
1	#1	Driver Switch Request Handling
1.1	#2	Switch precedence
1.2	#3	Avoid repeating commands
1.3	#4	Long Switch recognition
1.3.1	#5	Waiting state for Long Increment switch...
1.3.2	#6	Waiting state for Long Decrement switc...
1.4	#7	Cancel Switch Detection
1.5	#8	Set Switch Detection
1.6	#9	Enable Switch Detection
1.7	#10	Resume Switch Detection
1.8	#11	Increment Switch Detection
1.9	#15	Decrement Switch Detection
2	#19	Cruise Control Mode
3	#37	Calculate Target Speed and Throttle Value
4	#44	System Interface
5	#71	Justifications
crs_req		
Import1	crs_req	References to crs_req.docx
1	Overview	Overview
2	System overview	System overview
3	Functional Requirements	Functional Requirements
4	Interface specification	Interface specification

Model Based Design



Code Generation



et3SIL.project - CODESYS

```
FUNCTION_BLOCK NovCRC
VAR_INPUT
    instring: STRING(255);
    incrc: STRING(8);
END_VAR
VAR_OUTPUT
    ok: BOOL:=FALSE;
END_VAR
VAR
    crc_str:STRING(255);
    validatedString: STRING(8);
```

```
//This function calculates the CRC value

l_string:=LEN(instring);
pt:=ADR(instring);
FOR input_idx:=0 TO l_string-1 DO
    inbyte[input_idx]:=pt^;
    pt:=pt+1;
END_FOR
FOR j := 0 TO (l_string-1) DO
    ulTemp1:=SH VAR NovCRC.l_string : INT FFFFF
    CRCVAL_IN:=(DWORD_TO_WORD(ulCRC) XOR
    ulTemp2:=CRC32_NOV(CRCVAL_IN);
    ulCRC := ulTemp1 XOR ulTemp2;
END_FOR;

cl_base_unsafe.cl_split_dword(  input
                                byte0
                                byte1
                                byte2
                                byte3

IF MID(incrc,2,1)=byte_to_hex(b3) AND MI
    ok:=TRUE;
END_IF
```

Messages - Total 0 error(s), 4 warning(s), 0 message(s)

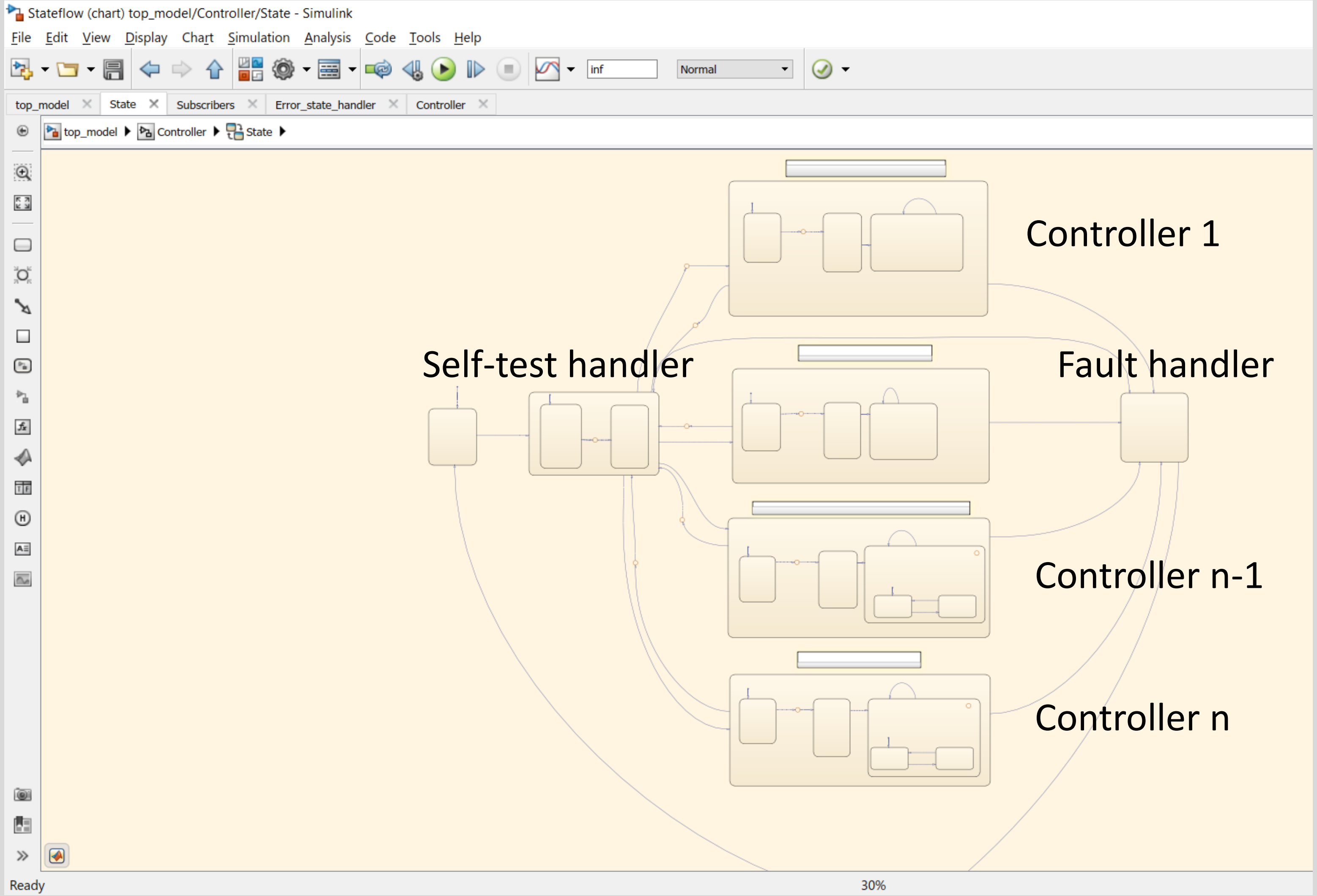
Precompile | 0 error(s) | 4 warning(s) | 0 message(s)

Description
- C0139: The code 'GVL.gDiagStatus; ' has no effect. Is this the intent?
- C0139: The code 'GVL.gSafetySwitch0Fb; ' has no effect. Is this the intent?
- C0196: Implicit conversion from unsigned Type 'UINT' to signed Type 'INT' : possible change of sign

Compilation

- This workflow allows SIL2 certifiable code to be generated using model-based design
- Review and testing occurs within each phase and before each release

Availability Solution


Stateflow (chart) top_model/Controller/State - Simulink
File Edit View Display Chart Simulation Analysis Code Tools Help
inf
Normal
top_model
State
Subscribers
Error_state_handler
Controller
top_model ▸ Controller ▸ State ▸
Controller 1
Self-test handler
Fault handler
Controller n-1
Controller n
Ready
30%

- Supervisory controller invokes multiple independent and redundant motion control paradigms
  - Local
  - Remote
  - Mission training
  - Mission running
- Graphical state modeling of control logic allows streamlined, debuggable, testable strategies

Functional Safety and Approvals

- Kyburz is designing autonomous machines not vehicles
  - IEC 61508
- Voluntarily following automotive functional safety norms
  - ISO 13849:2015
  - ISO 26262:2018
- Primary implications
  - Development process
  - Documentation system
  - Component selection
  - Software development toolchains

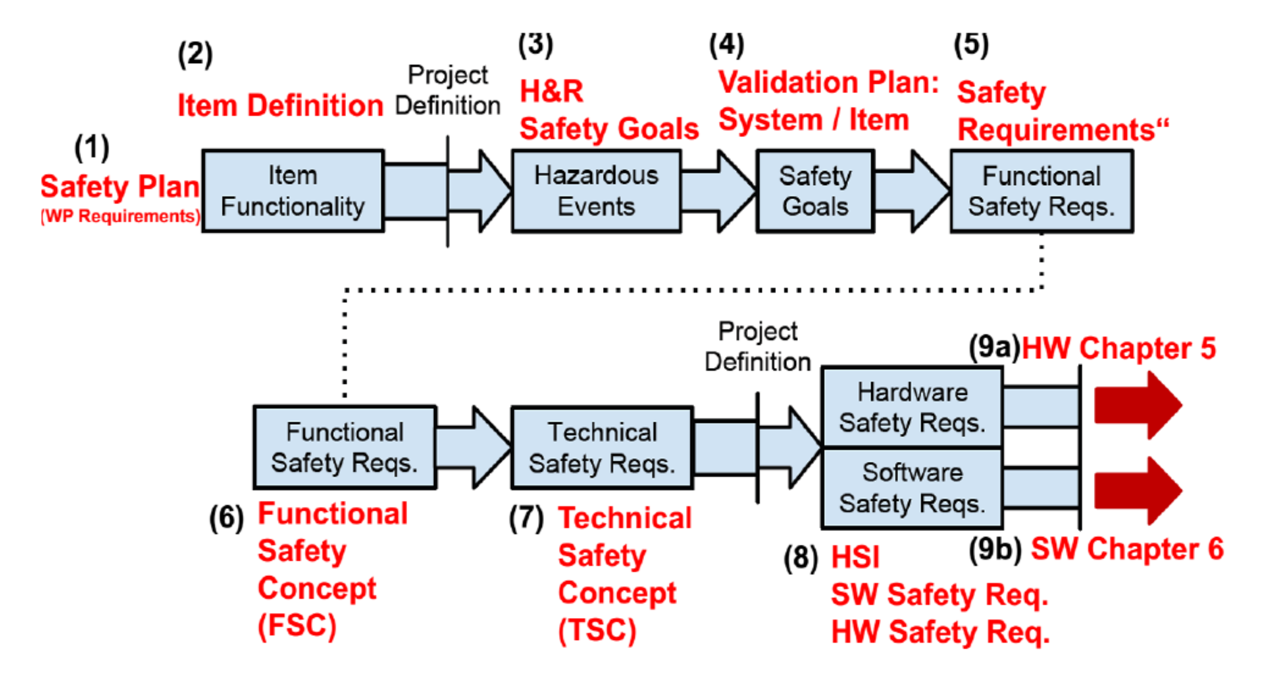
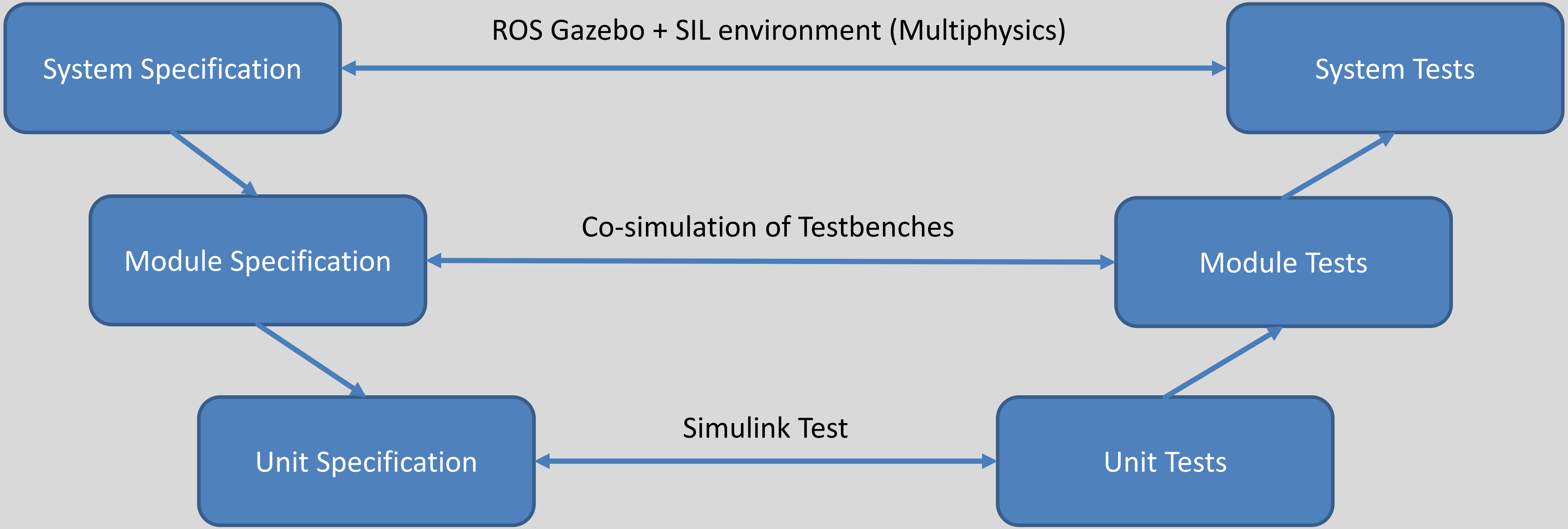


Image: ROSAS Freiburg, Paria Amini

Safety Solution



System Specification

ROS Gazebo + SIL environment (Multiphysics)

System Tests

Module Specification

Co-simulation of Testbenches

Module Tests

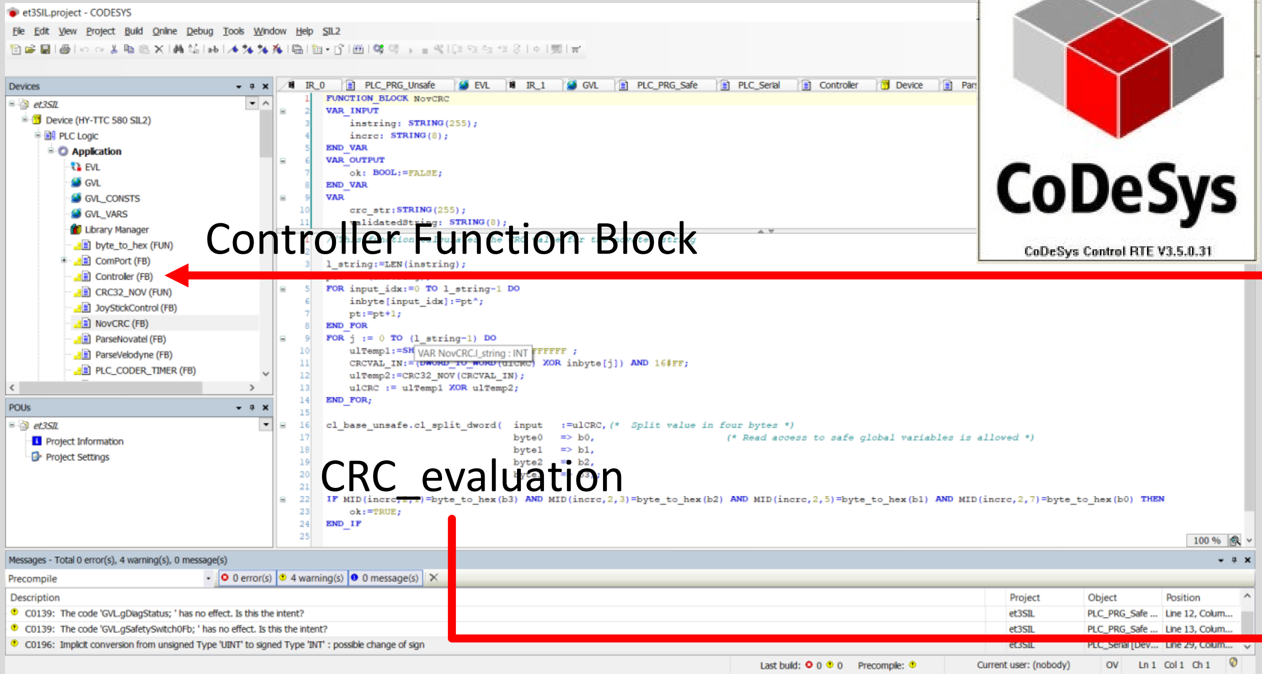
Unit Specification

Simulink Test

Unit Tests

- Kyburz toolchain uses layered verification techniques and model-based design
- All requirements are easily documented for traceability

Safety Example



```
FUNCTION_BLOCK NovCRC
VAR_INPUT
    instring: STRING(255);
    incrc: STRING(8);
END_VAR
VAR_OUTPUT
    ok: BOOL:=FALSE;
END_VAR
VAR
    crc_str:STRING(255);
    validatedString: STRING(8);
```

```
//This function calculates the CRC value for the novatel string

l_string:=LEN(instring);

FOR input_idx:=0 TO l_string-1 DO
    inbyte[input_idx]:=pt^;
    pt:=pt+1;
END_FOR
FOR j := 0 TO (l_string-1) DO
    ulTemp1:=SH                FFFFFF ;
    CRCVAL_IN:=(DWORD_TO_WORD(ulCRC) XOR inbyte[j]) AND 16#FF;
    ulTemp2:=CRC32_NOV(CRCVAL_IN);
    ulCRC  := ulTemp1 XOR ulTemp2;
END_FOR;

cl_base_unsafe.cl_split_dword(  input    :=ulCRC, (*  Split value in four bytes *)
                                byte0    => b0,                     (* Read access to safe global variables is allowed *)
                                byte1    => b1,
                                byte2    => b2,

IF MID(incrc,2,1)=byte_to_hex(b3) AND MID(incrc,2,3)=byte_to_hex(b2) AND MID(incrc,2,5)=byte_to_hex(b1) AND MID(incrc,2,7)=byte_to_hex(b0) THEN
    ok:=TRUE;
END_IF
```

VAR NovCRC.l_string : INT

100 %

Messages - Total 0 error(s), 4 warning(s), 0 message(s)

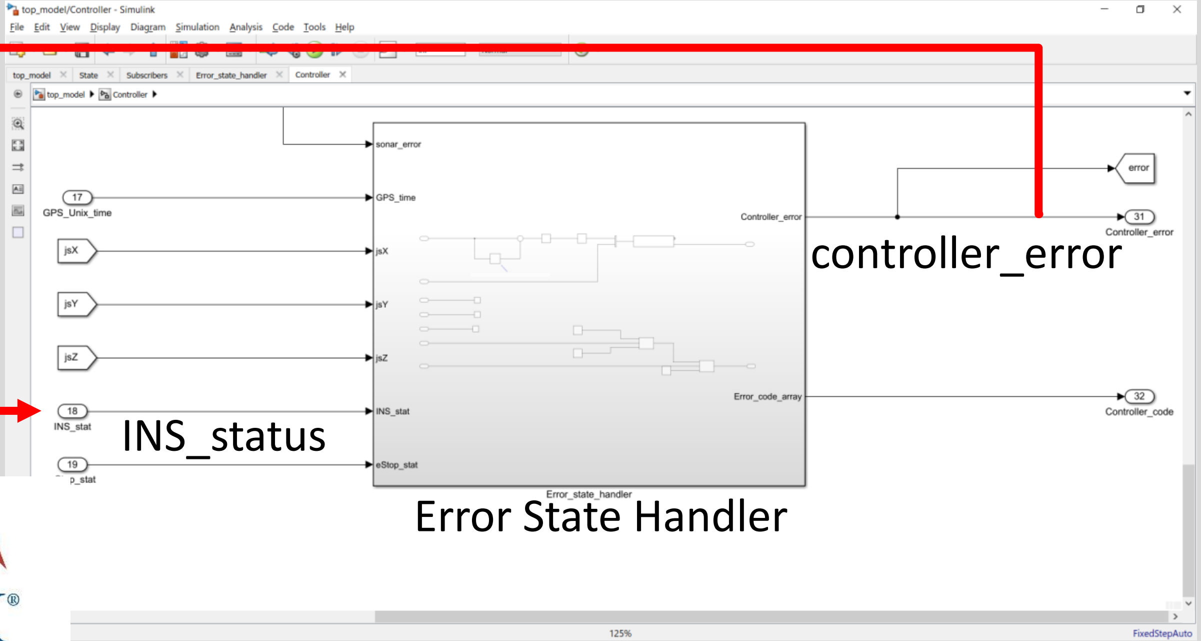
Precompile | 0 error(s) | 4 warning(s) | 0 message(s)

Description	Project	Object	Position
C0139: The code 'GVL.gDiagStatus; ' has no effect. Is this the intent?	et3SIL	PLC_PRG_Safe ...	Line 12, Colum...
C0139: The code 'GVL.gSafetySwitch0Fb; ' has no effect. Is this the intent?	et3SIL	PLC_PRG_Safe ...	Line 13, Colum...
C0196: Implicit conversion from unsigned Type 'UINT' to signed Type 'INT' : possible change of sign	et3SIL	PLC_Serial [Dev...	Line 29, Colum...

Last build: 0 0 Precompile: Current user: (nobody) OV Ln 1 Col 1 Ch 1

Controller Function Block

CRC_evaluation



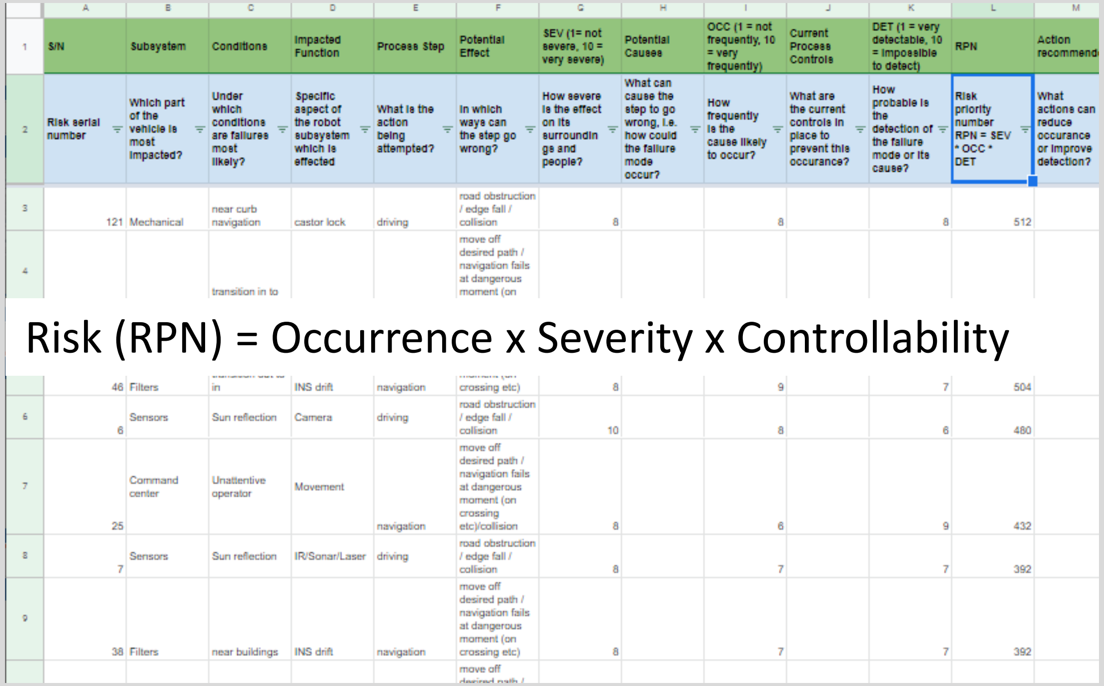
controller_error

INS_status

Error State Handler

- Serial communication errors are detected and handled gracefully in control logic

Corner Cases



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	S/N	Subsystem	Conditions	Impacted Function	Process Step	Potential Effect	SEV (1= not severe, 10 = very severe)	Potential Causes	OCC (1 = not frequently, 10 = very frequently)	Current Process Controls	DET (1 = very detectable, 10 = impossible to detect)	RPN	Action recommend
2	Risk serial number	Which part of the vehicle is most impacted?	Under which conditions are failures most likely?	Specific aspect of the robot subsystem which is effected	What is the action being attempted?	In which ways can the step go wrong?	How severe is the effect on its surroundings and people?	What can cause the step to go wrong, i.e. how could the failure mode occur?	How frequently is the cause likely to occur?	What are the current controls in place to prevent this occurance?	How probable is the detection of the failure mode or its cause?	Risk priority number RPN = SEV * OCC * DET	What actions can reduce occurance or improve detection?
3	121	Mechanical	near curb navigation	castor lock	driving	road obstruction / edge fall / collision	8		8		8	512	
4	46	Filters	transition in to in	INS drift	navigation	move off desired path / navigation fails at dangerous moment (on crossing etc)	8		9		7	504	
6	6	Sensors	Sun reflection	Camera	driving	road obstruction / edge fall / collision	10		8		6	480	
7	25	Command center	Unattentive operator	Movement	navigation	move off desired path / navigation fails at dangerous moment (on crossing etc)/collision	8		6		9	432	
8	7	Sensors	Sun reflection	IR/Sonar/Laser	driving	road obstruction / edge fall / collision	8		7		7	392	
9	38	Filters	near buildings	INS drift	navigation	move off desired path / navigation fails at dangerous moment (on crossing etc)	8		7		7	392	
						move off desired path /							

Risk (RPN) = Occurrence x Severity x Controllability

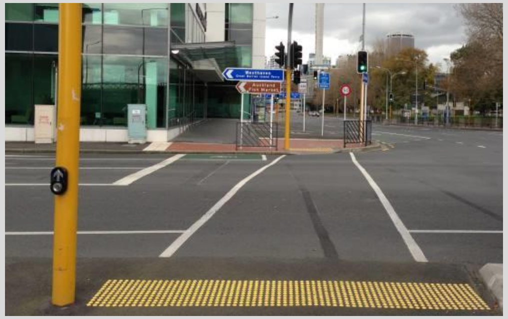


Image: drivingtests.co.nz

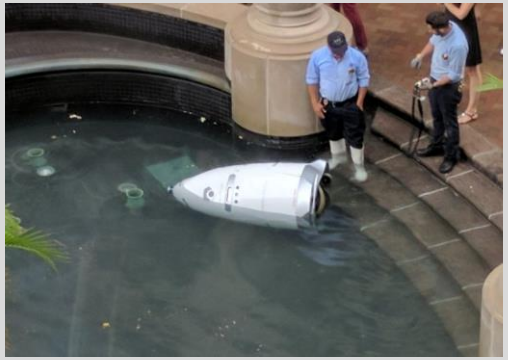
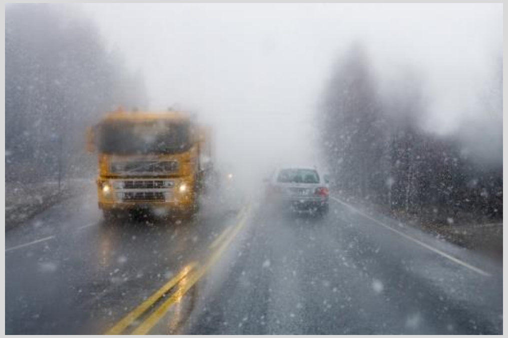
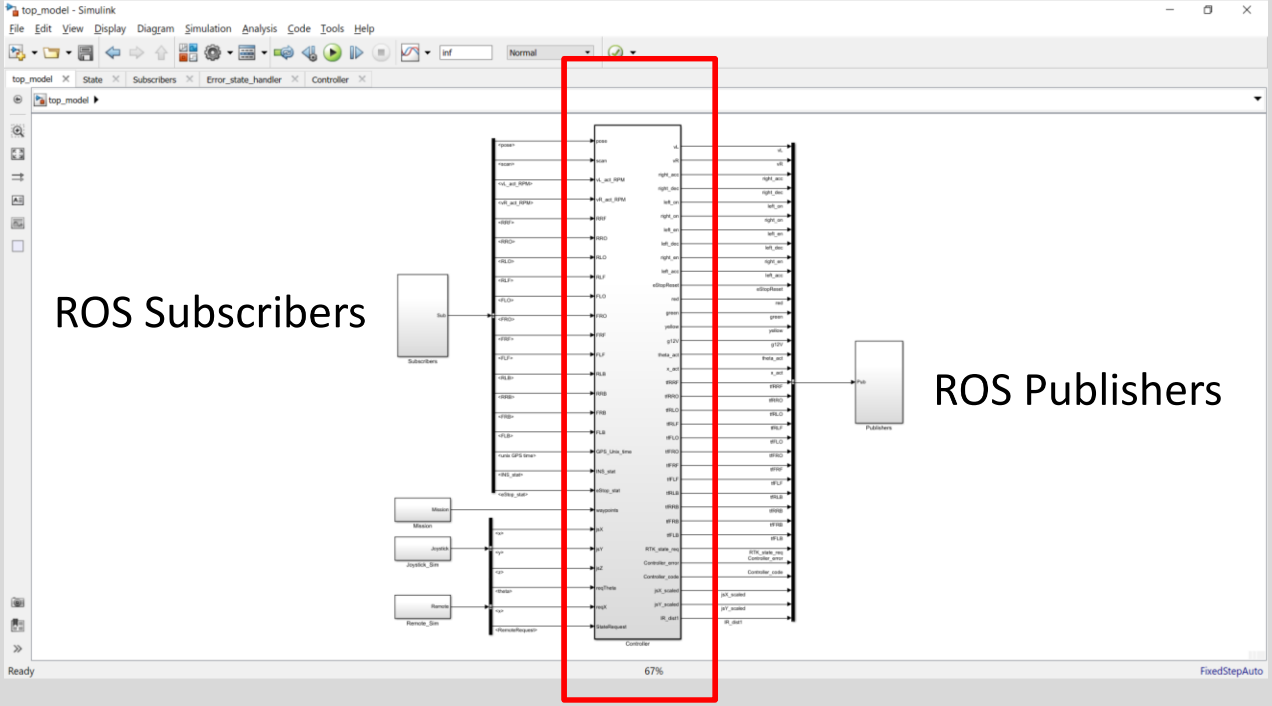


Image: arstechnica.com



- Hazard and Risk Assessment (HARA) identified 30 failure modes with Risk Priority Number (RPN) > 200, some which are challenging to simulate

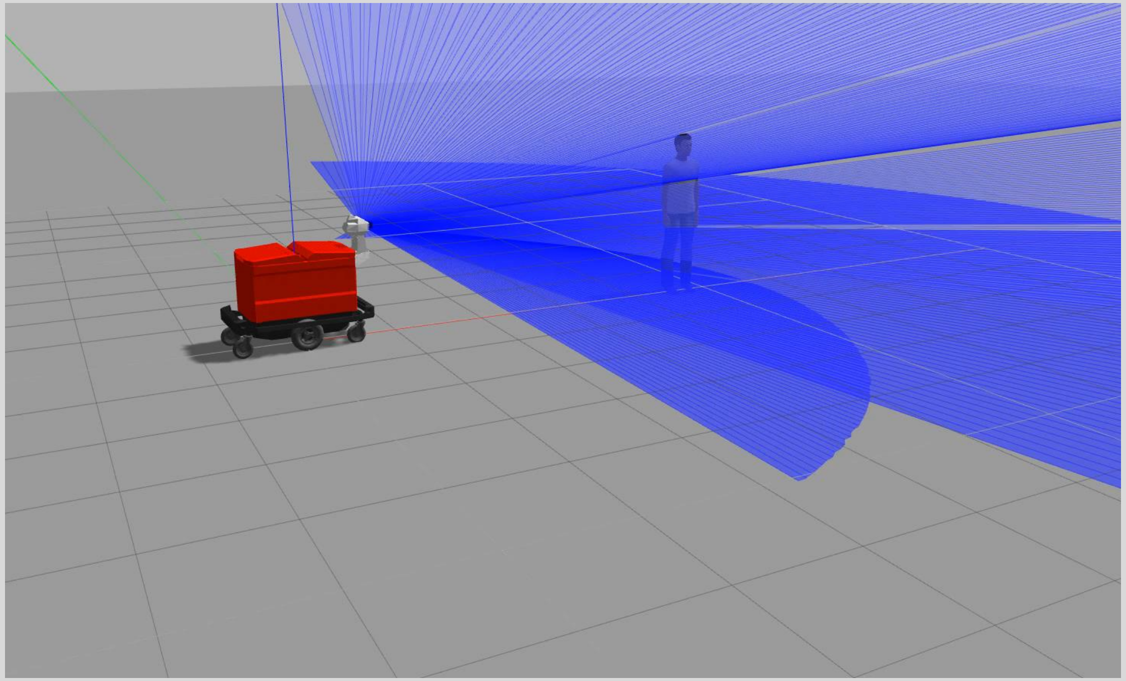
Corner Cases Solution



ROS Subscribers

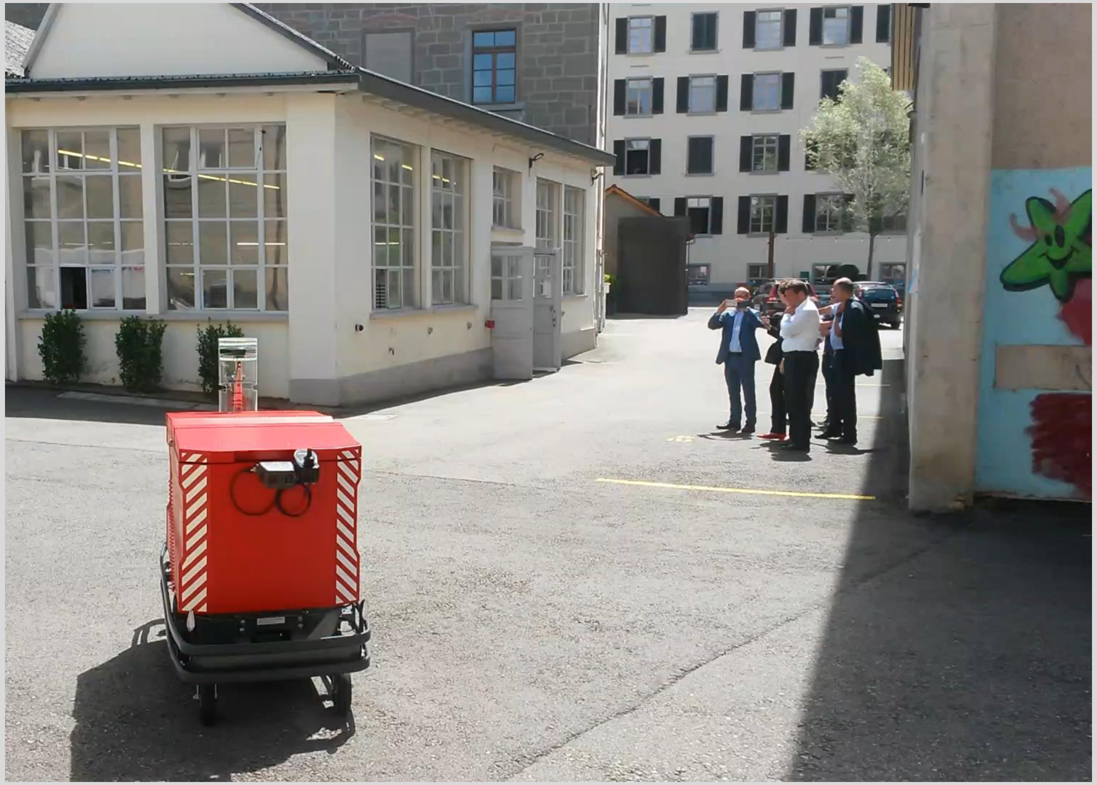
ROS Publishers

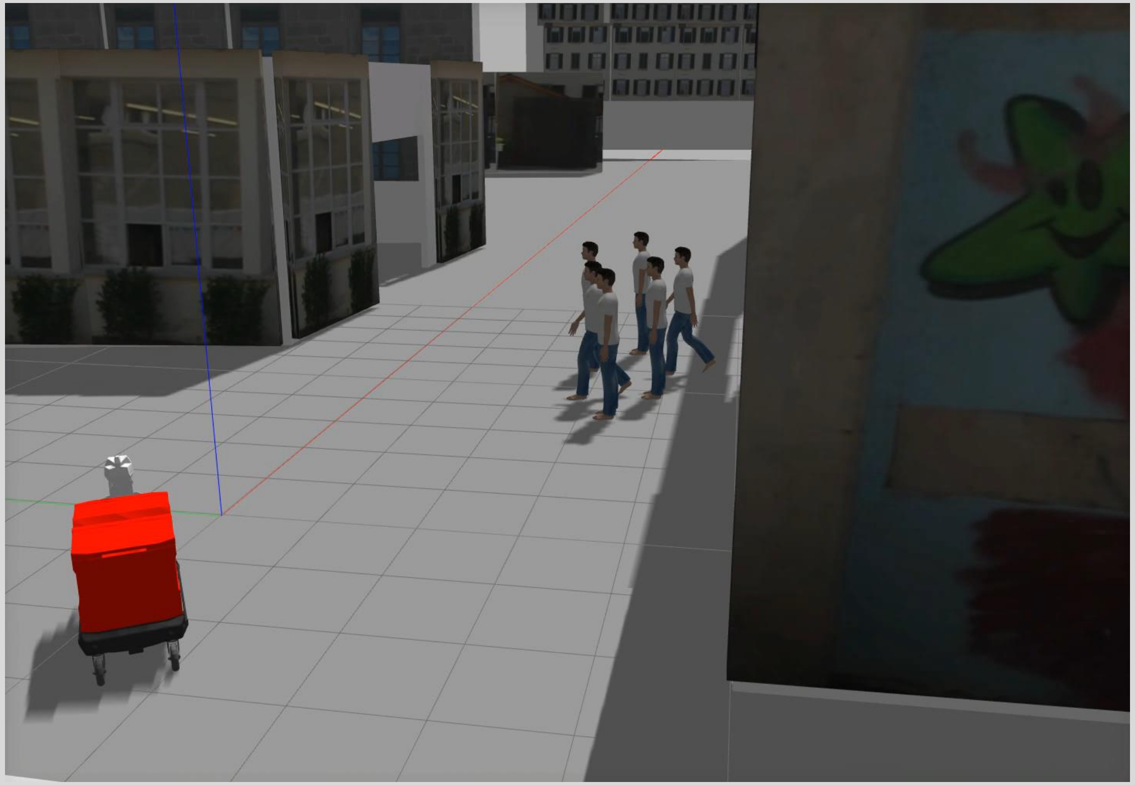
Autogenerated



- ROS Gazebo enables detailed sensor measurement-level simulation
- With co-simulation testing is drastically streamlined

Corner Cases Example





Summary

- Kyburz Switzerland’s autonomous system developments have saved substantial development time from
  - Enabling seamless and testable control redundancy with finite state machines
  - Integrated toolboxes for streamlining development following functional safety norms
  - Simulation of difficult to test corner-cases with controller to environment interfaces

Thank you for your attention