Sur la pratique des méthodes formelles par de non praticiens : Autopsie d'un robot

Eric JENN, IRT Saint Exupéry et Thales Avionics

Journée FMF

LAAS, 10 octobre 2017

FRENCH
INSTITUTES OF
TECHNOLOGY

Context

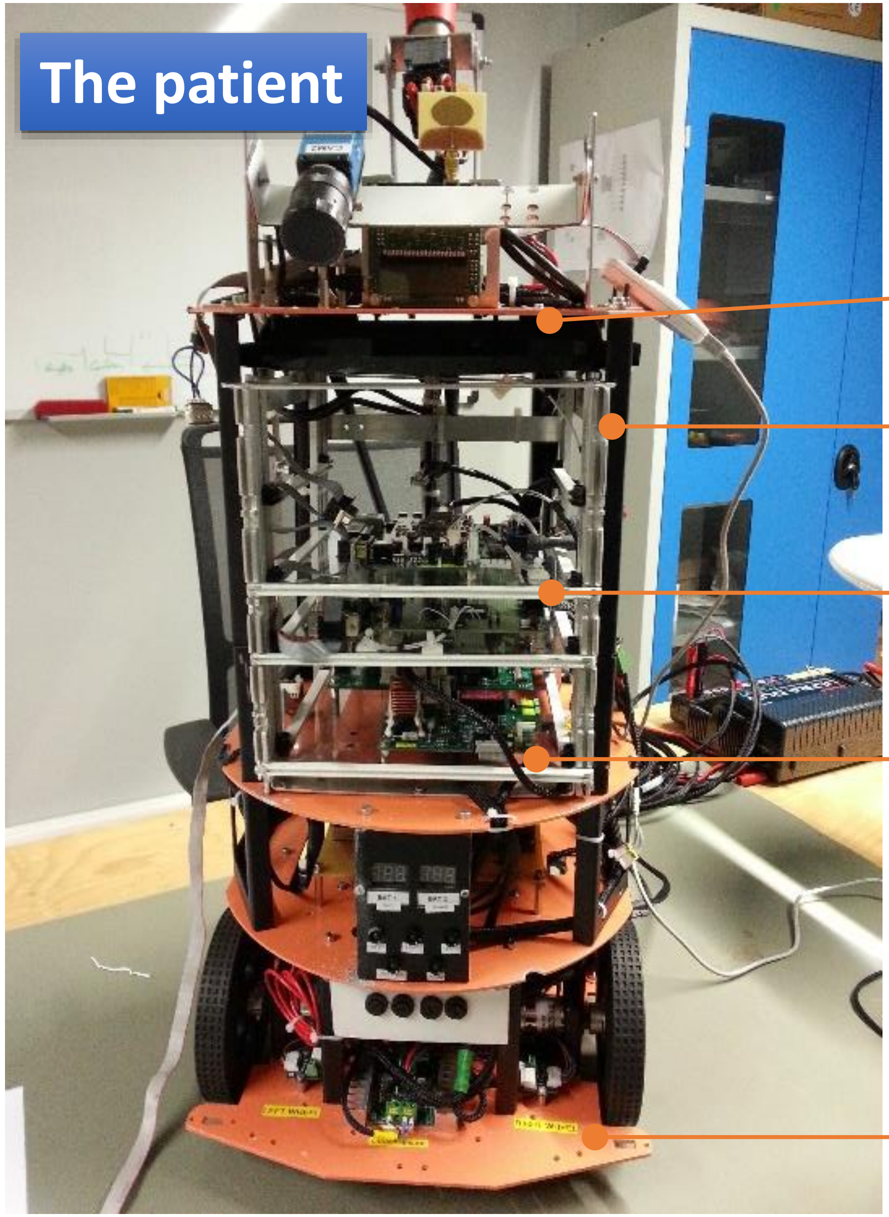
Collaborative work in INGEQUIP project at IRT

Pierre-Alain Bourdil
Arnaud Dieumegard
Ning Ge
Faiez Zalila

with academic support from LAAS, ONERA, IRIT, ISAE

with financial backing from CGI, ANR, and industrial members:

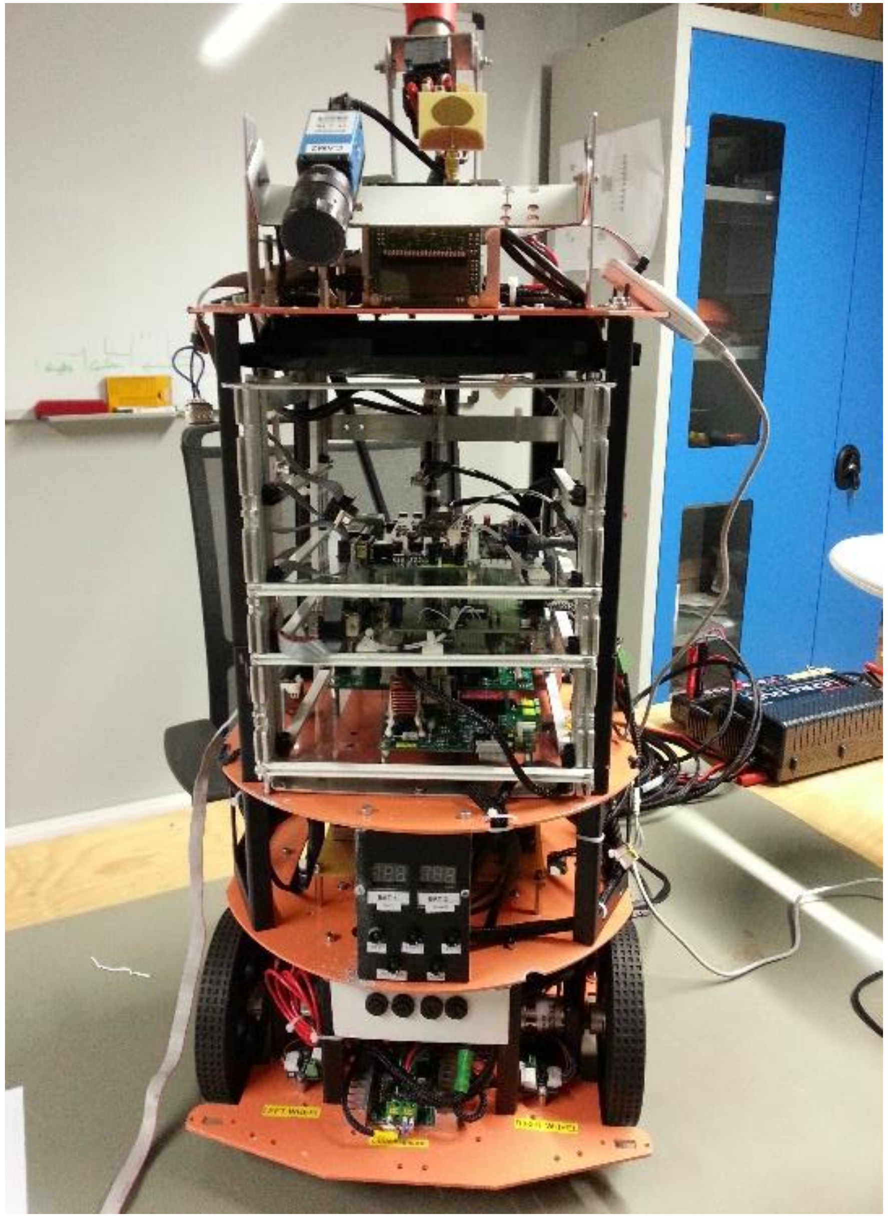
ACTIA, astc, AIRBUS, AIRBUS DEFENCE & SPACE, Continental, Ellidiss Technologies, GreenSocs, Space Codesign, SAFRAN, systerel, THALES AVIONICS

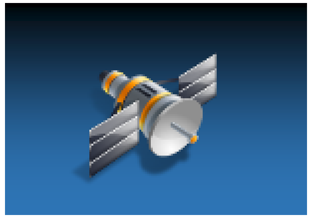


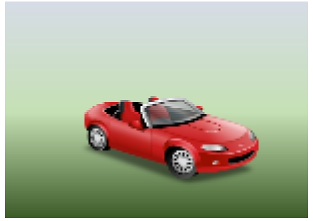
Pathologies

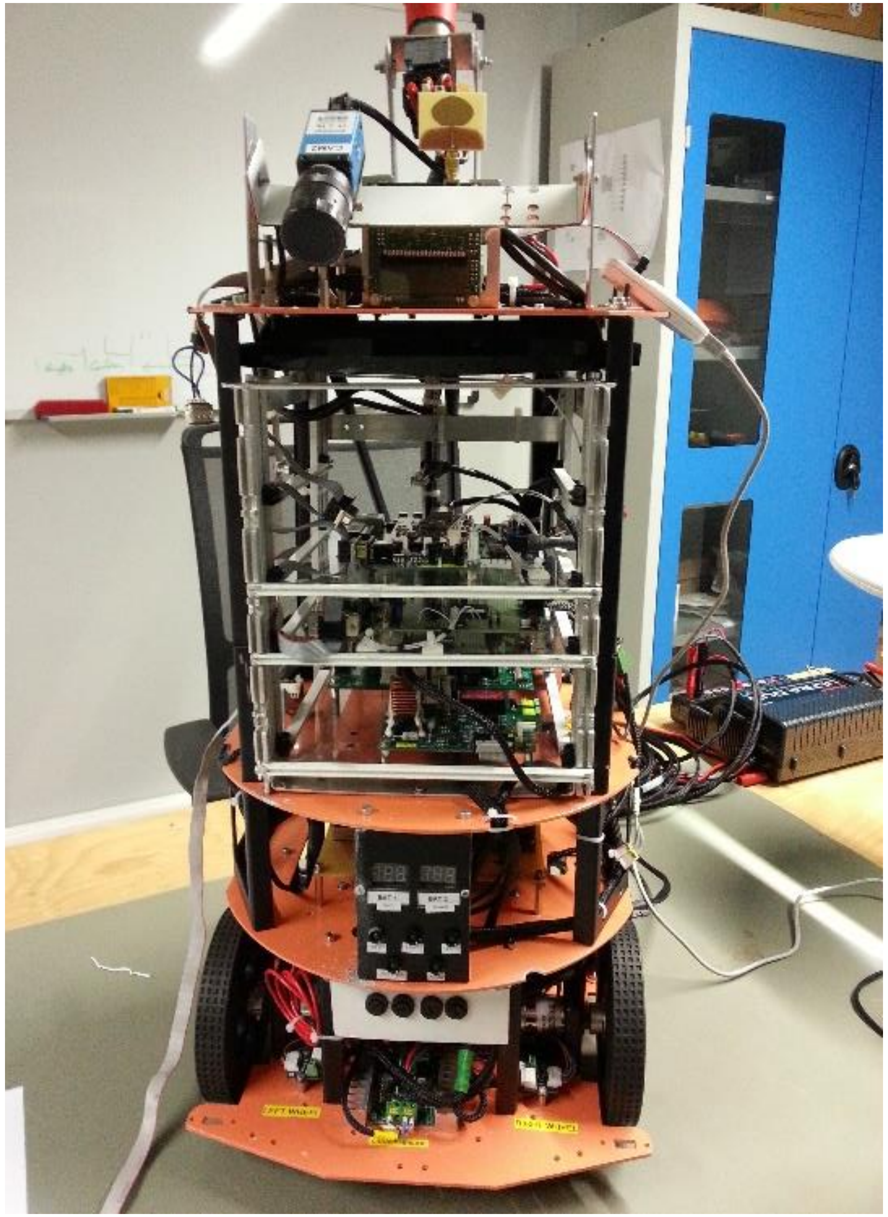
Remedies

Pathologies	Remedies
Data	Formal verif. of conf. data
Time	Formal verif. of timed systems
HMIs	Formal verification of HMIs
SW impl.	Correctness by construction
	Formal code verification
Numerical	



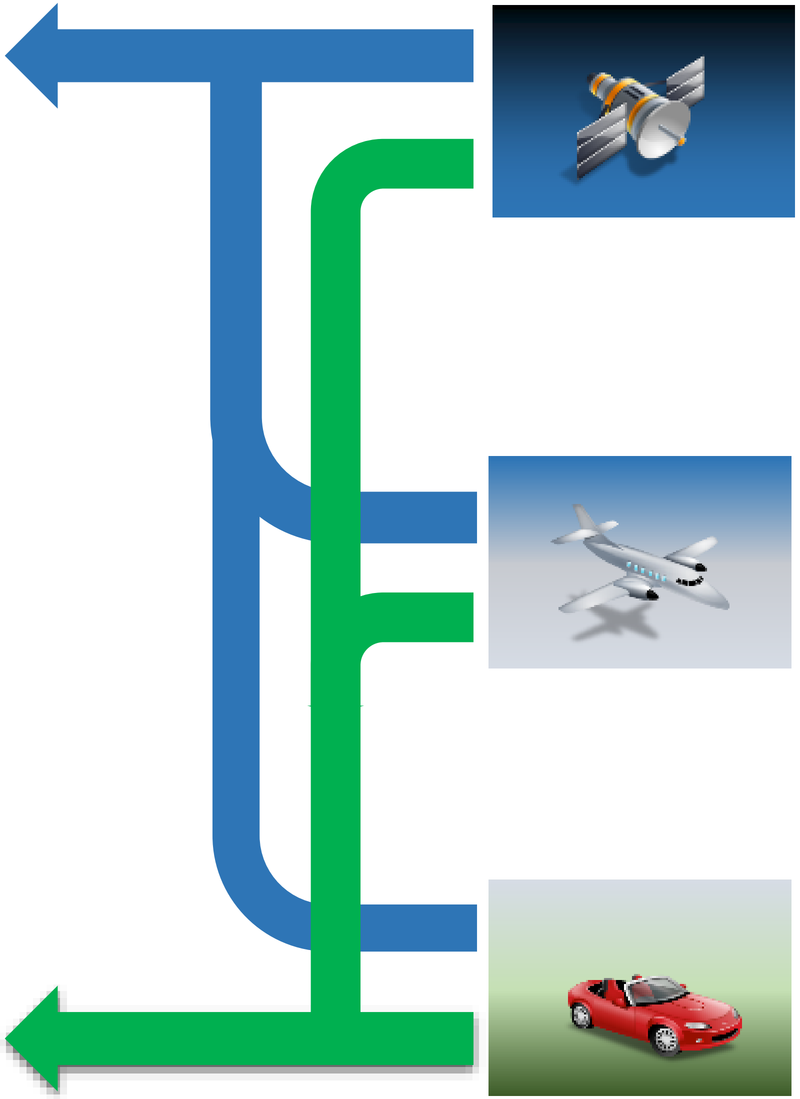






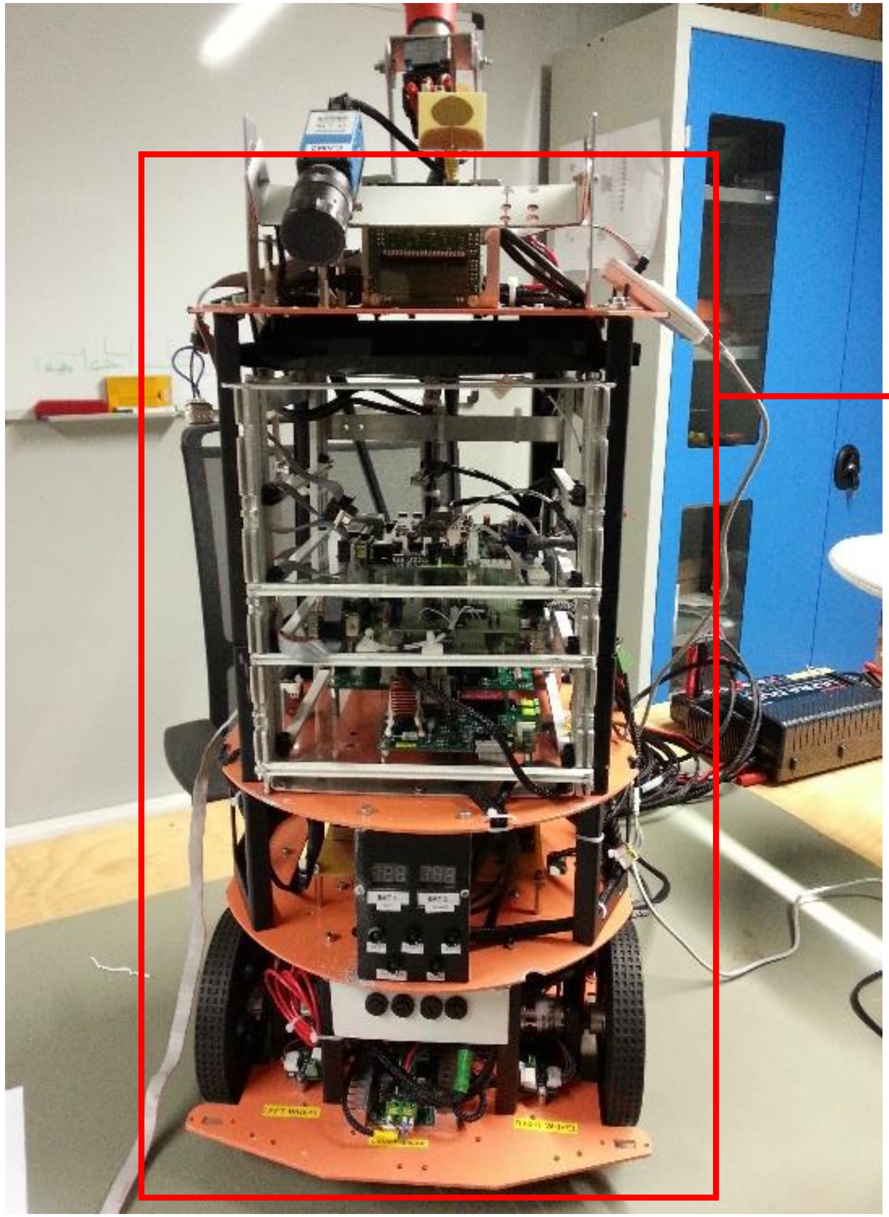
Virtual platforms

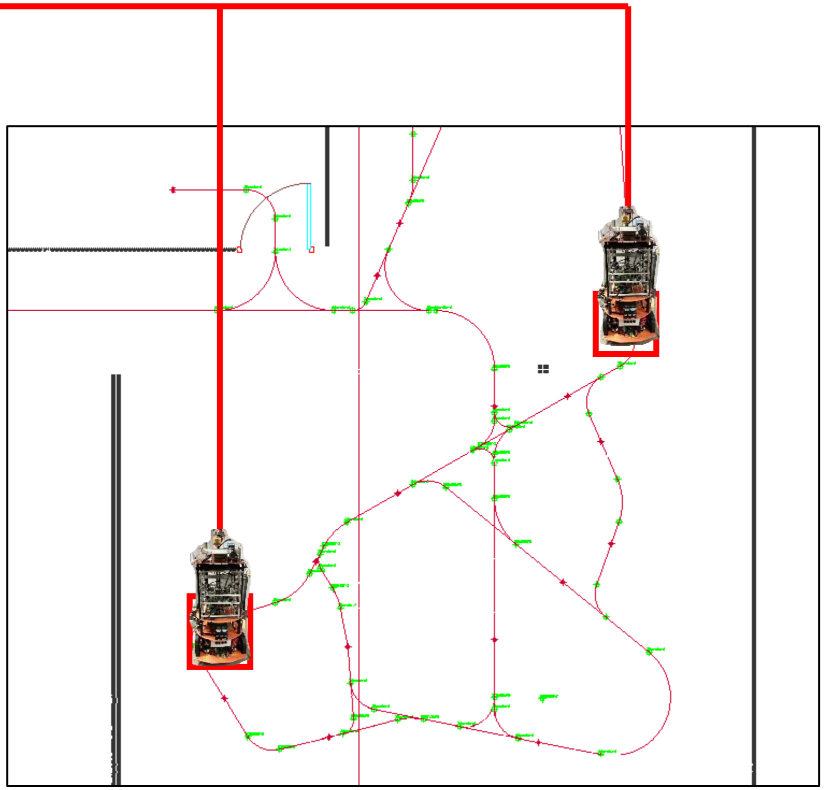
Design space exploration



Formal development methods

A robot

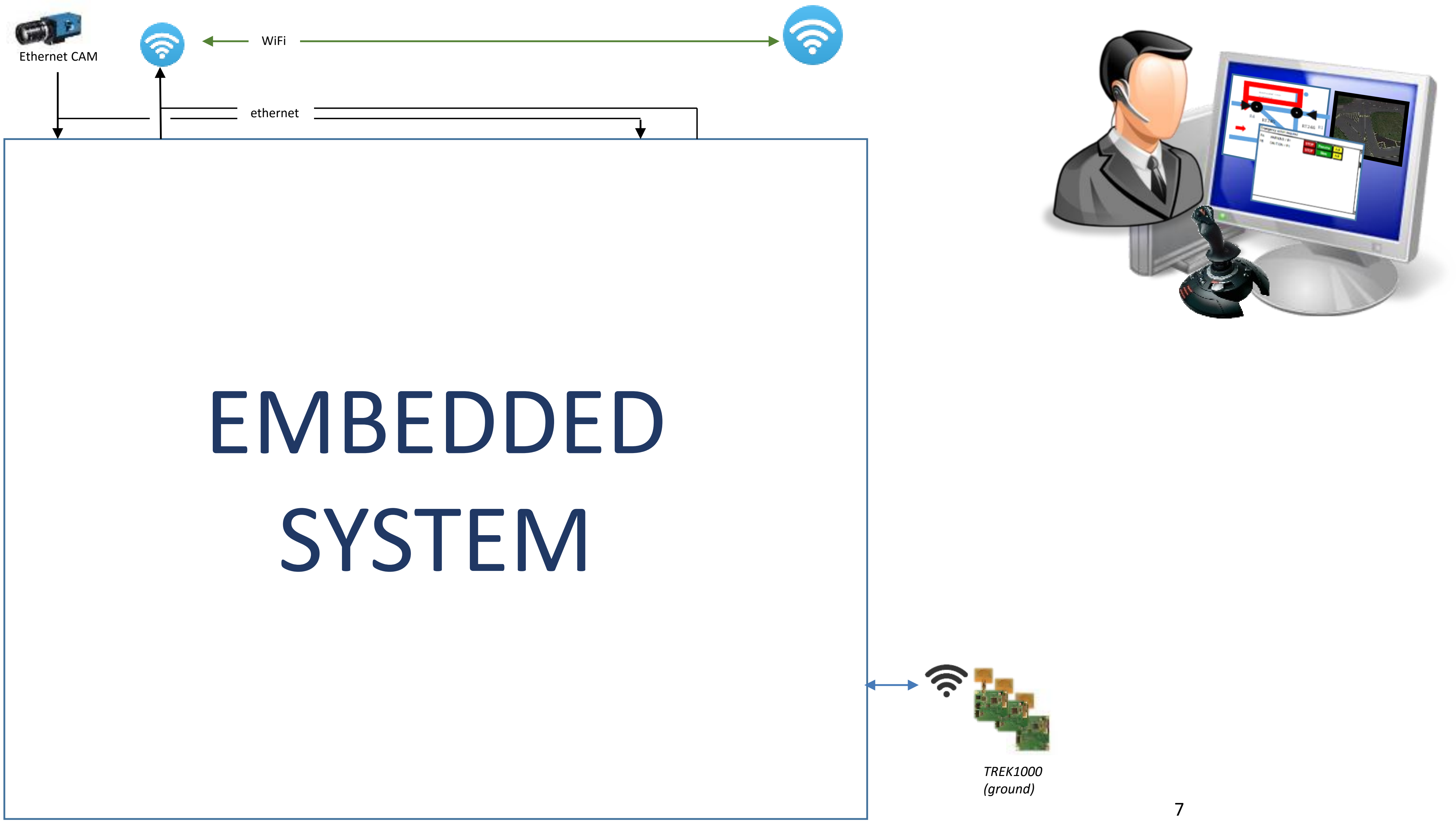




A supervision station

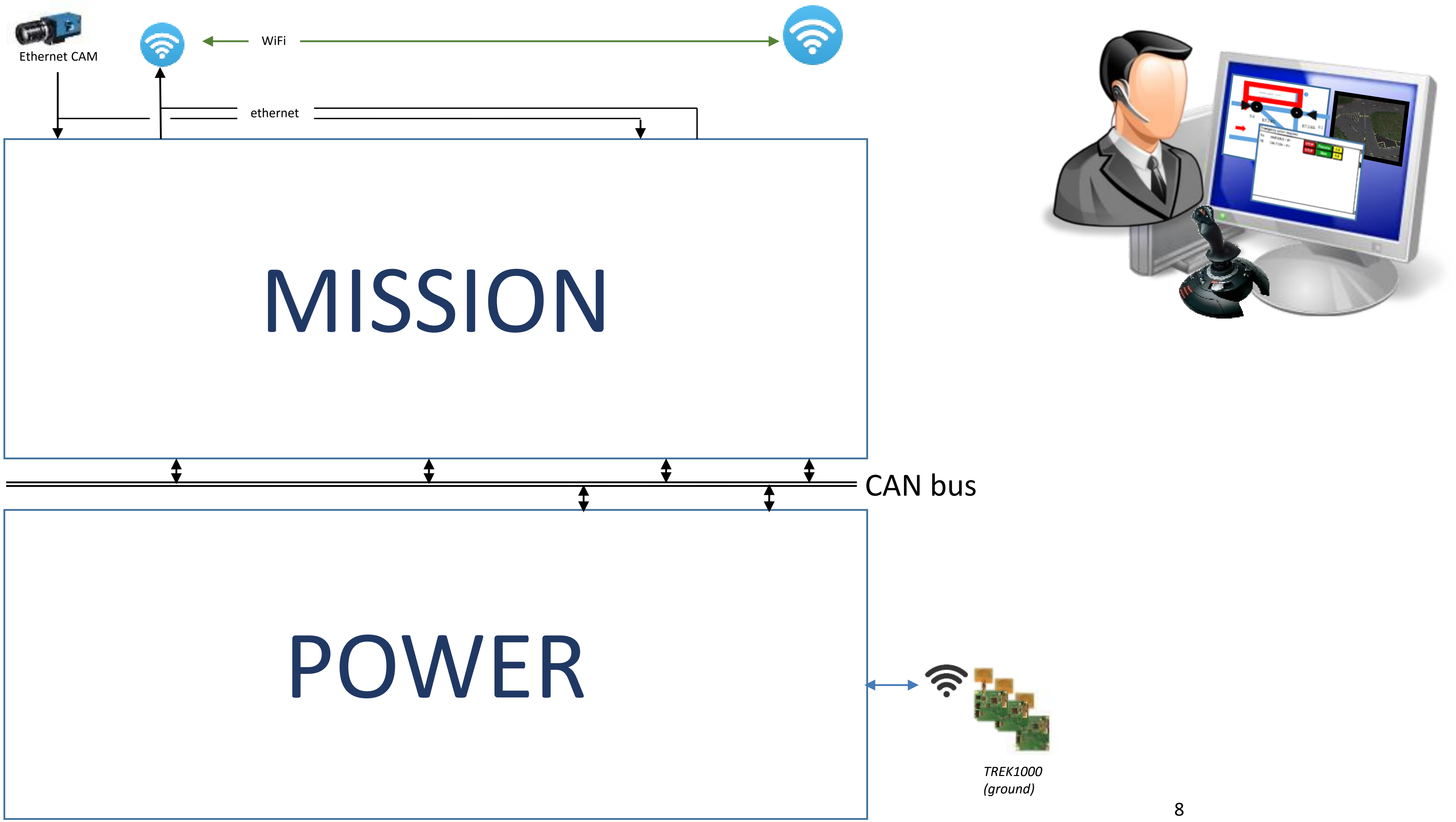
An environment

TwIRTee

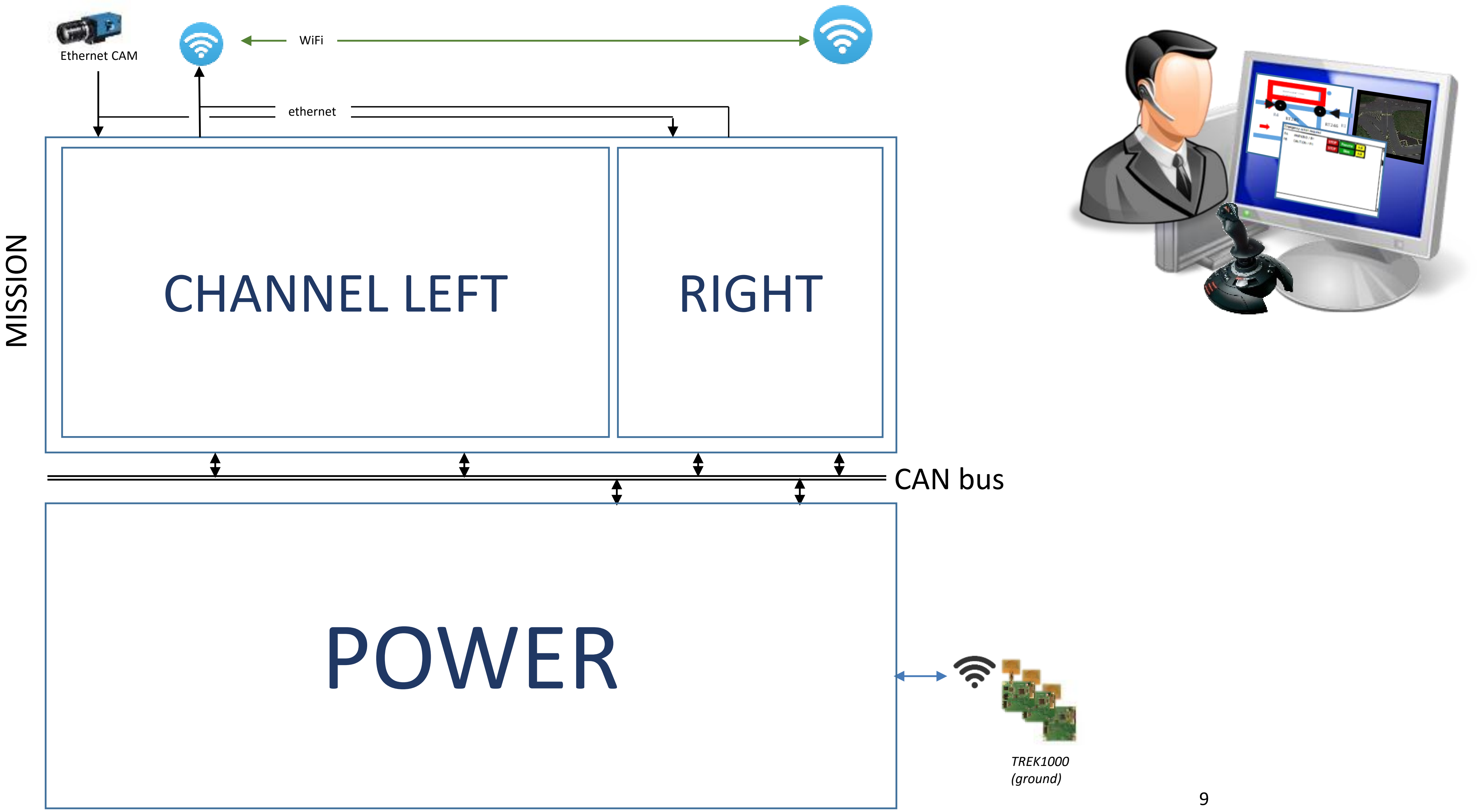


Ethernet CAM
WiFi
ethernet
EMBEDDED SYSTEM
TREK1000 (ground)

TwIRTee

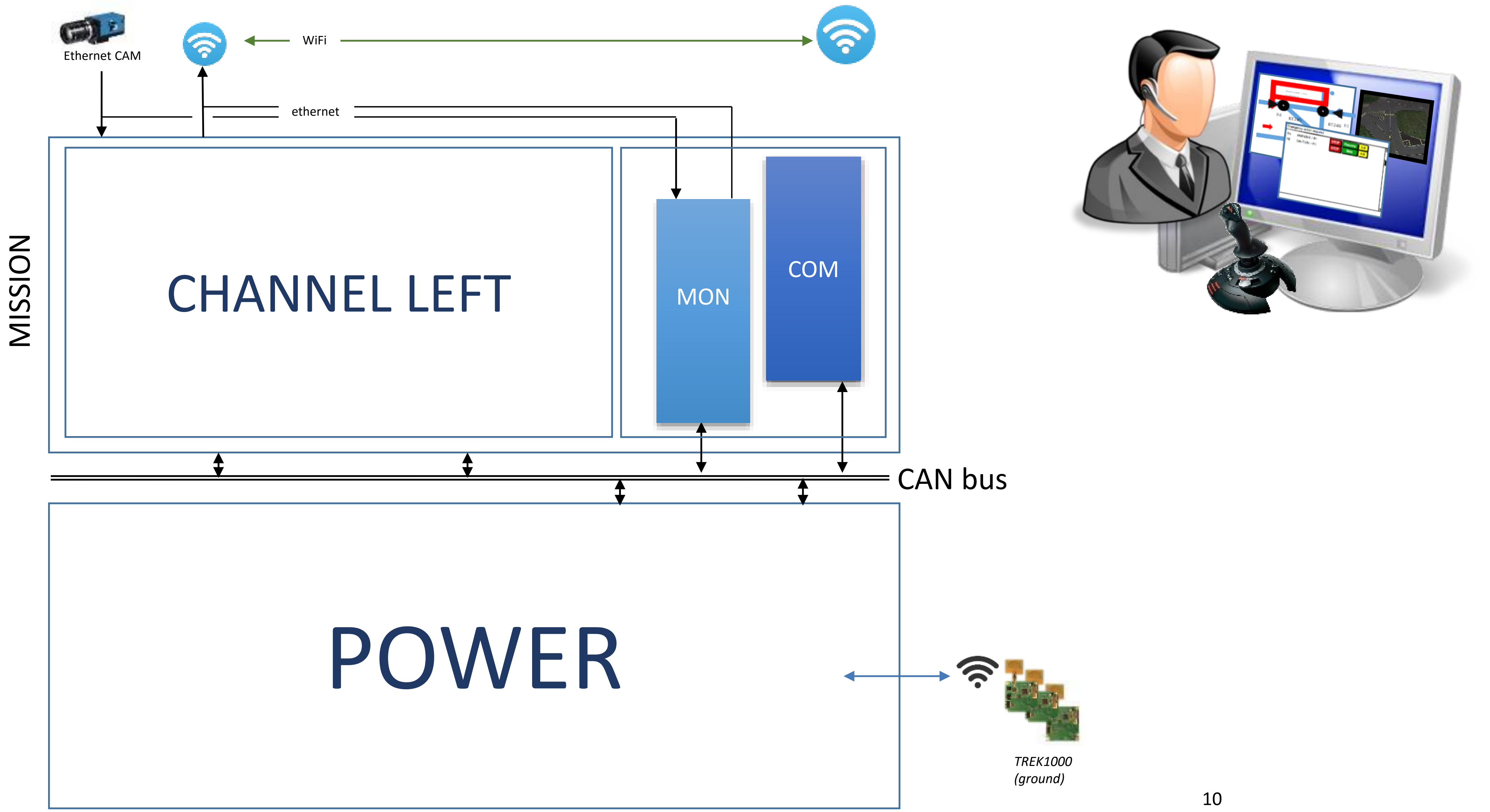

Ethernet CAM
WiFi
ethernet
MISSION
CAN bus
POWER
TREK1000
(ground)

TwIRTee



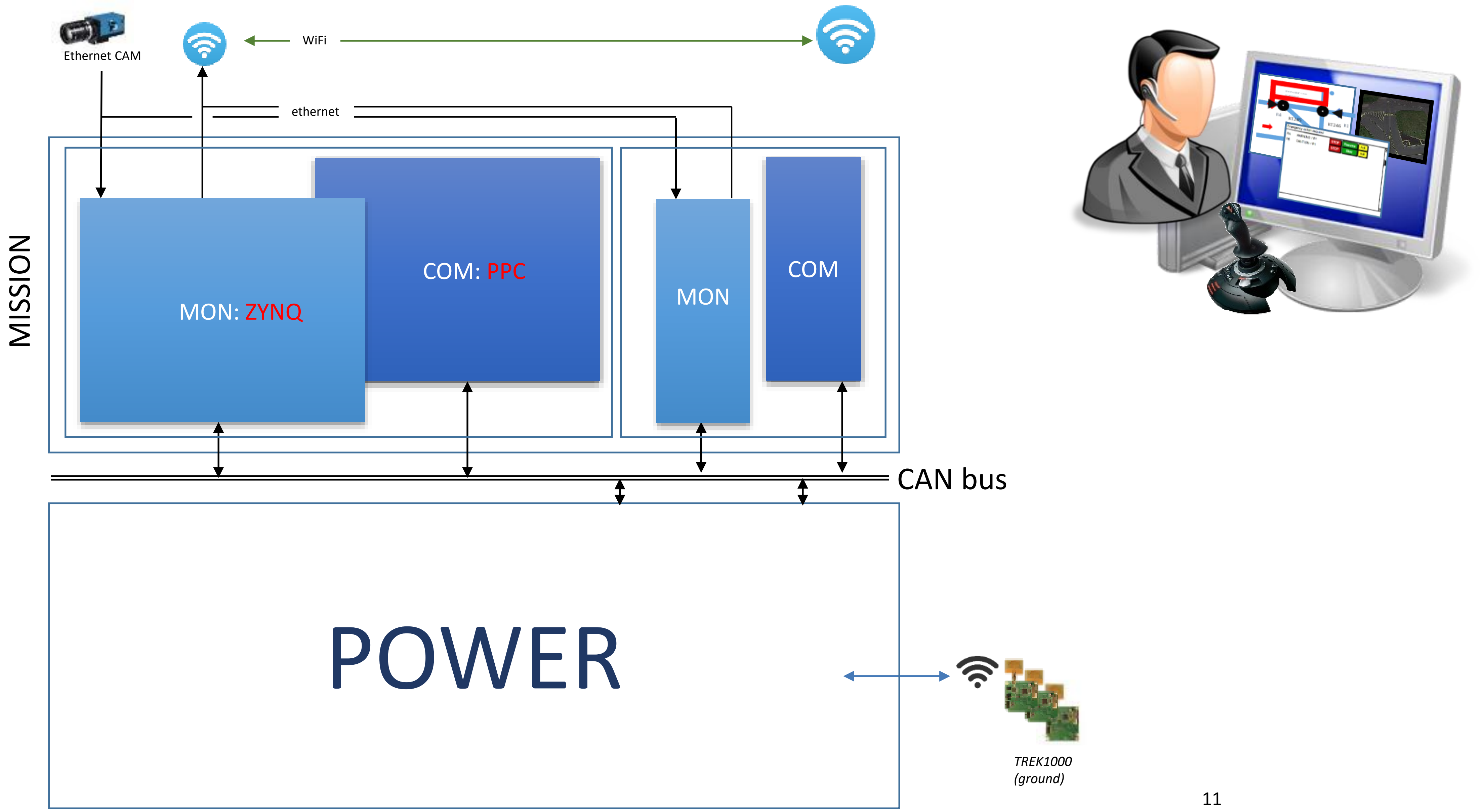
Ethernet CAM
WiFi
ethernet
MISSION
CHANNEL LEFT
RIGHT
CAN bus
POWER
TREK1000
(ground)

TwIRTee

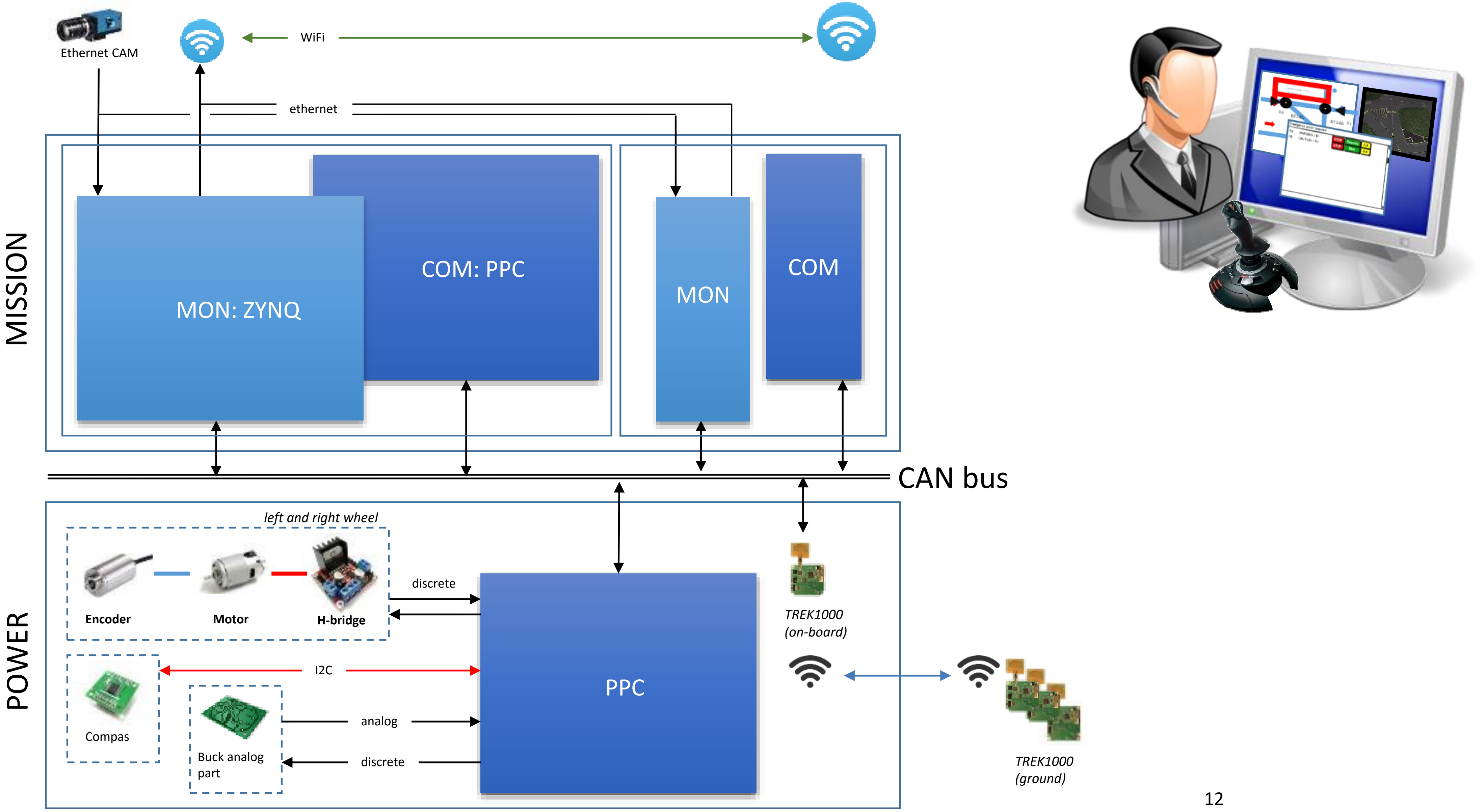


Ethernet CAM
WiFi
ethernet
MISSION
CHANNEL LEFT
MON
COM
CAN bus
POWER
TREK1000
(ground)

TwIRTee

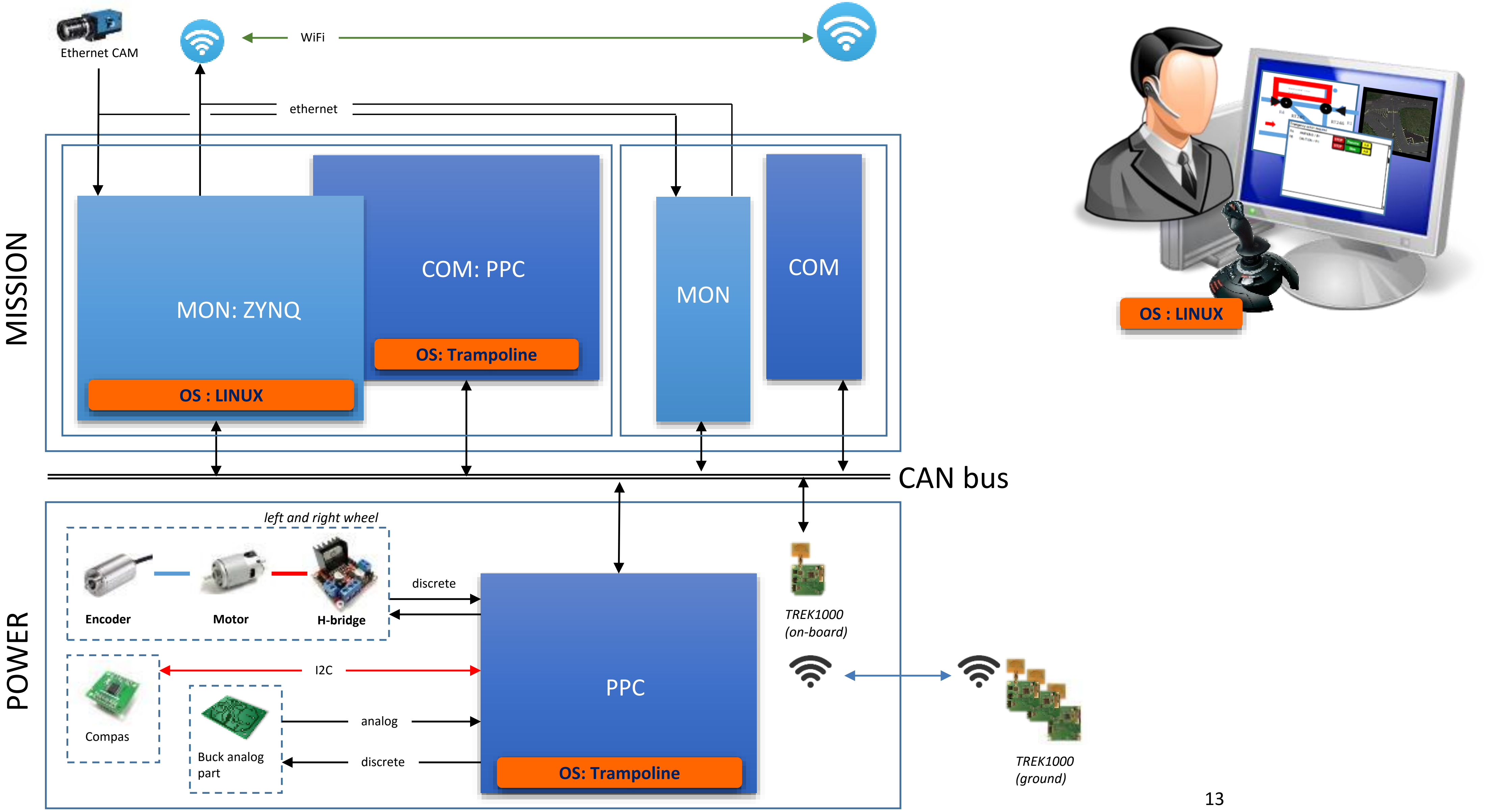

Ethernet CAM
WiFi
ethernet
MISSION
MON: ZYNQ
COM: PPC
MON
COM
CAN bus
POWER
TREK1000
(ground)

TwIRTee



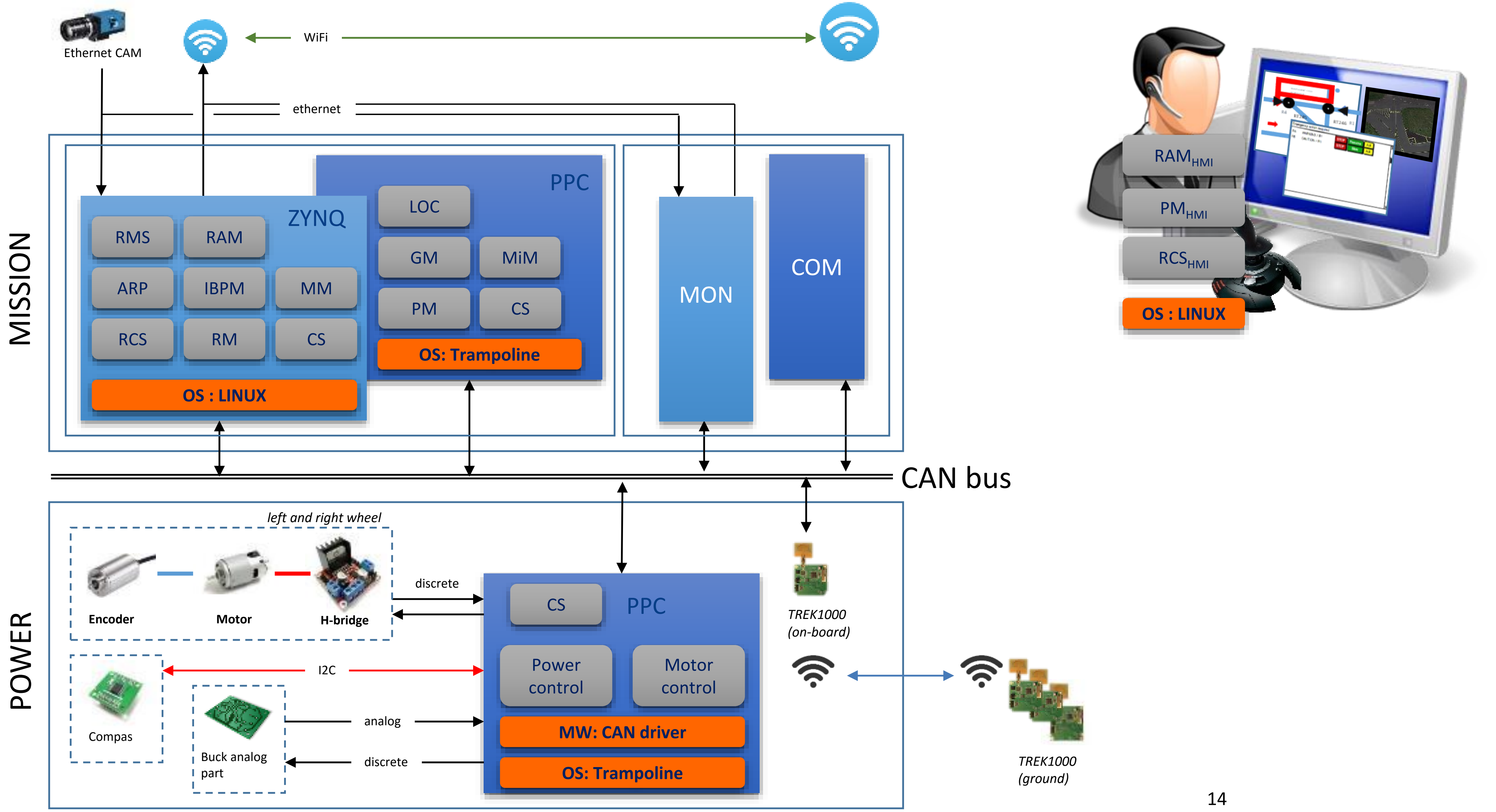
Ethernet CAM
WiFi
ethernet
MISSION
MON: ZYNQ
COM: PPC
MON
COM
CAN bus
POWER
left and right wheel
Encoder
Motor
H-bridge
discrete
I2C
Compas
Buck analog part
analog
discrete
PPC
TREK1000 (on-board)
TREK1000 (ground)

TwIRTee

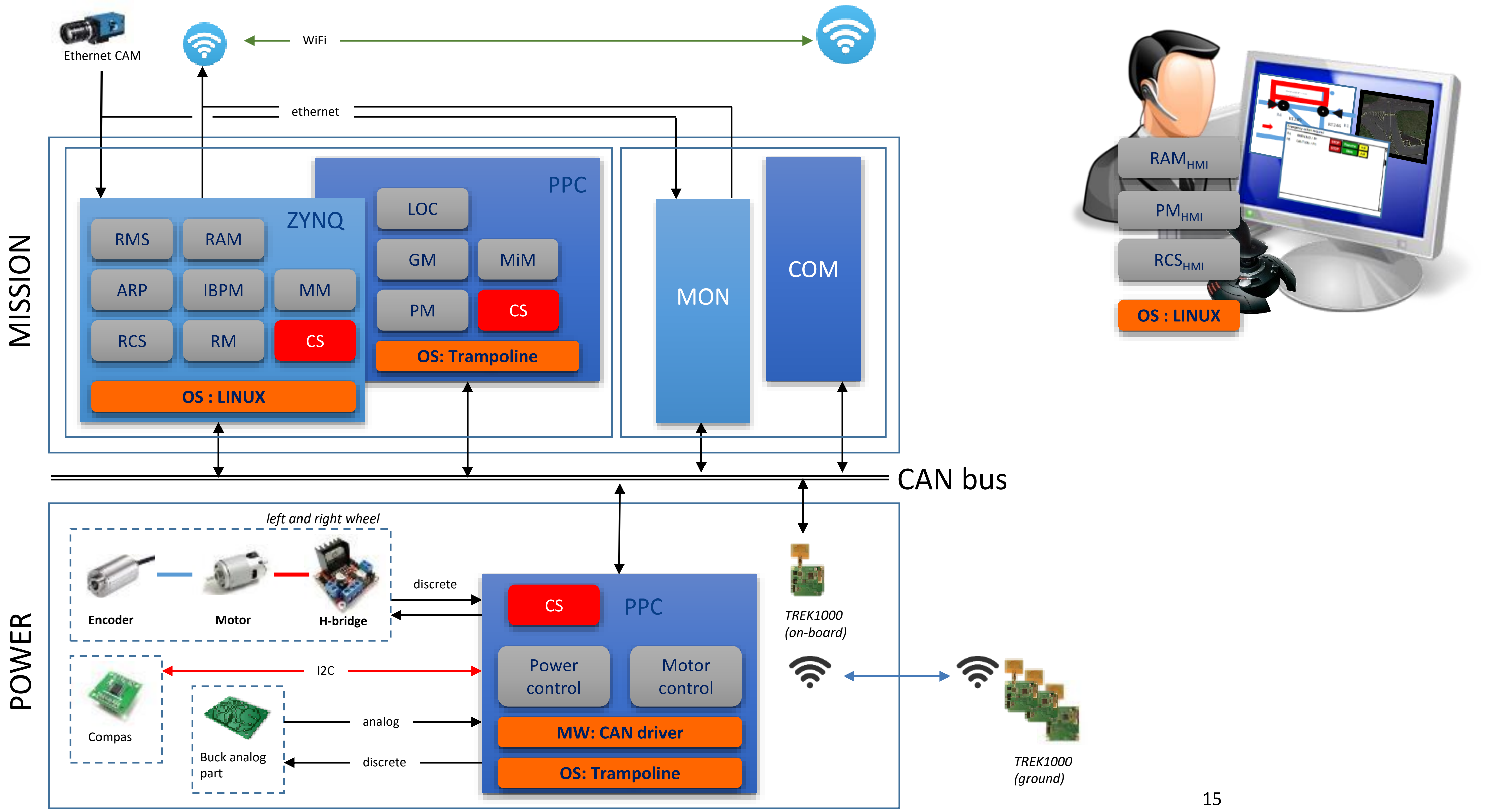


Ethernet CAM
WiFi
ethernet
MISSION
MON: ZYNQ
OS : LINUX
COM: PPC
OS: Trampoline
MON
COM
OS : LINUX
CAN bus
POWER
left and right wheel
Encoder
Motor
H-bridge
discrete
Compas
I2C
Buck analog part
analog
discrete
PPC
OS: Trampoline
TREK1000 (on-board)
TREK1000 (ground)

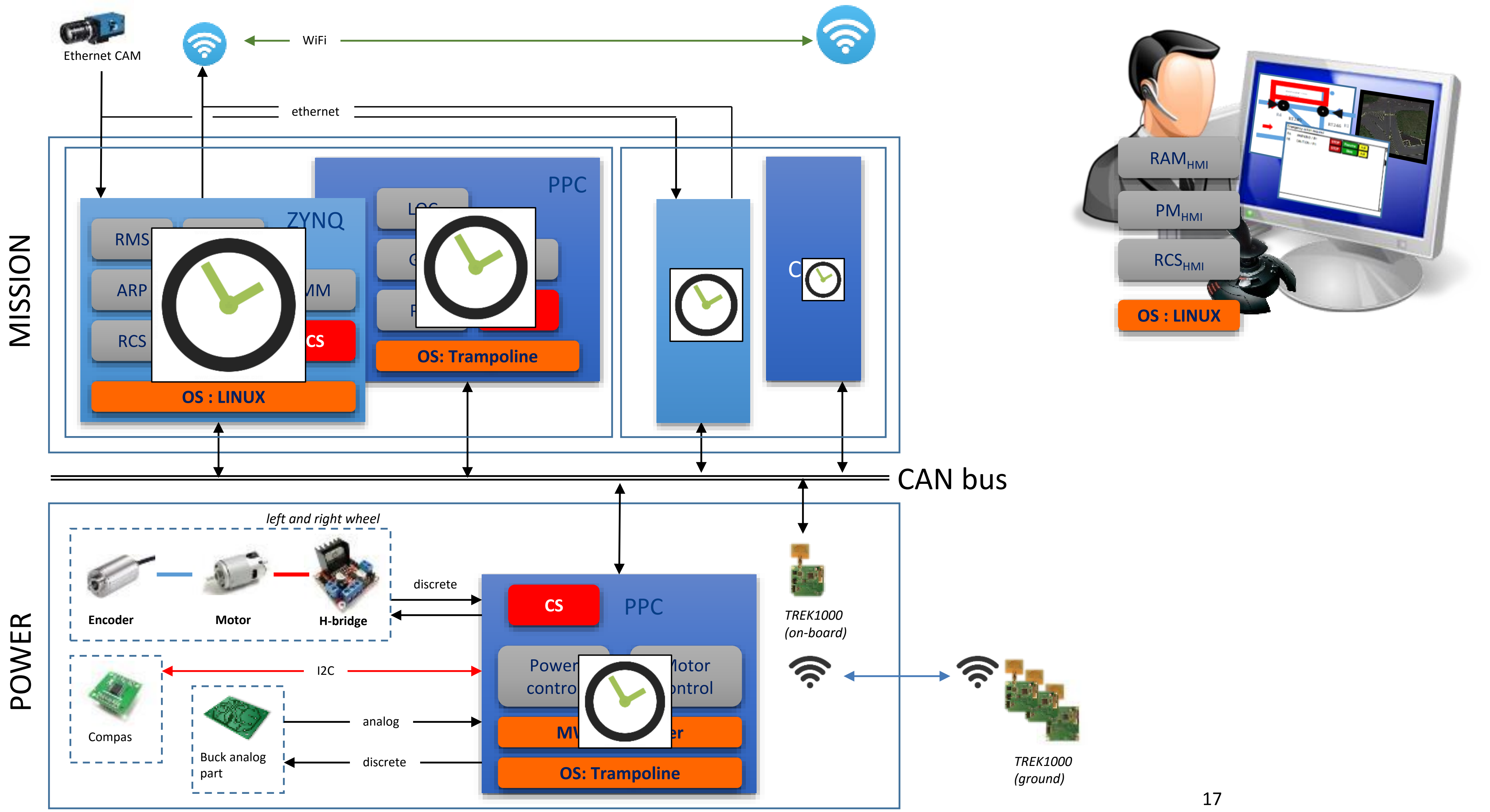
Clock synchronization: Verification of timed systems



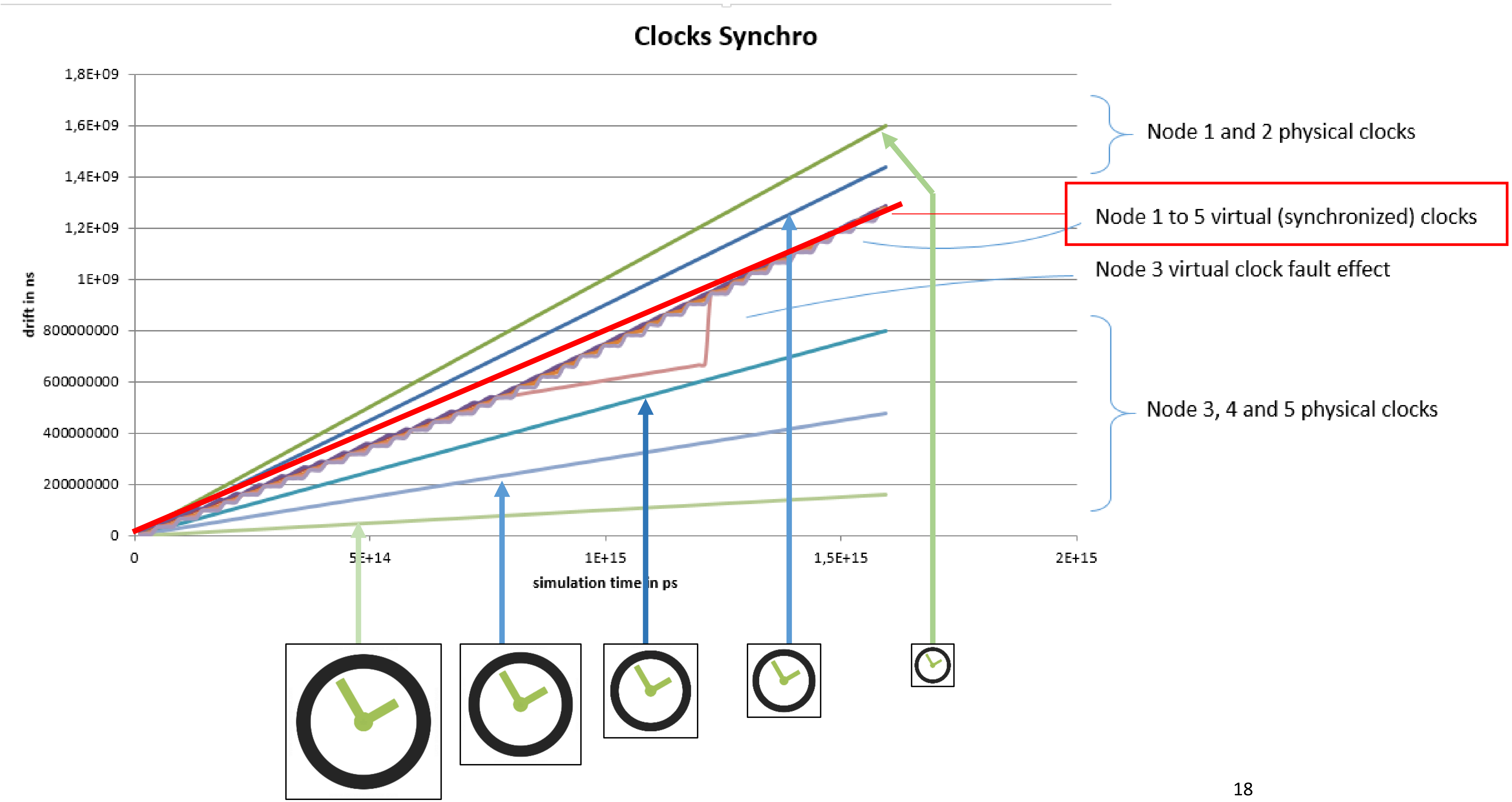
Clock synchronization: Verification of timed systems



Verification of timed systems: Fiacre and Tina

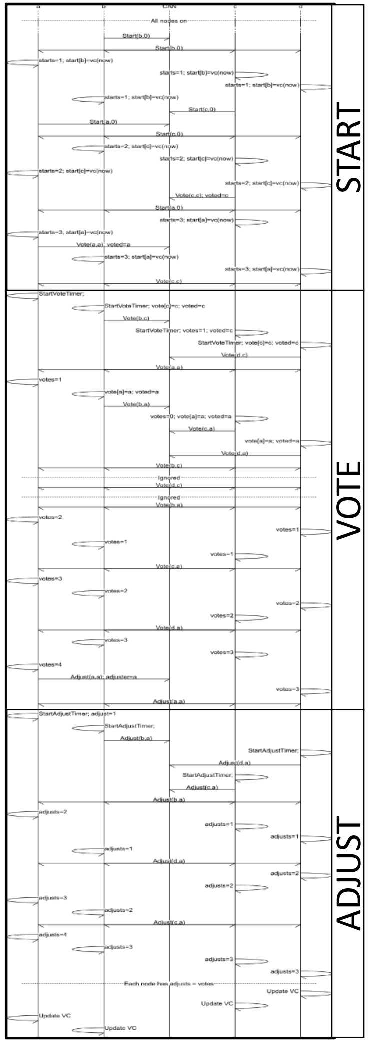


Verification of timed systems: Fiacre and Tina



Verification of timed systems: Fiacre and Tina

Clock synchronization protocol



Formalization (abstractions, hypothesis,…)

2.2.1 Hypothesis

- **CLOCK1**: the drift rate of a correct clock is $\rho = 10^{-6}$

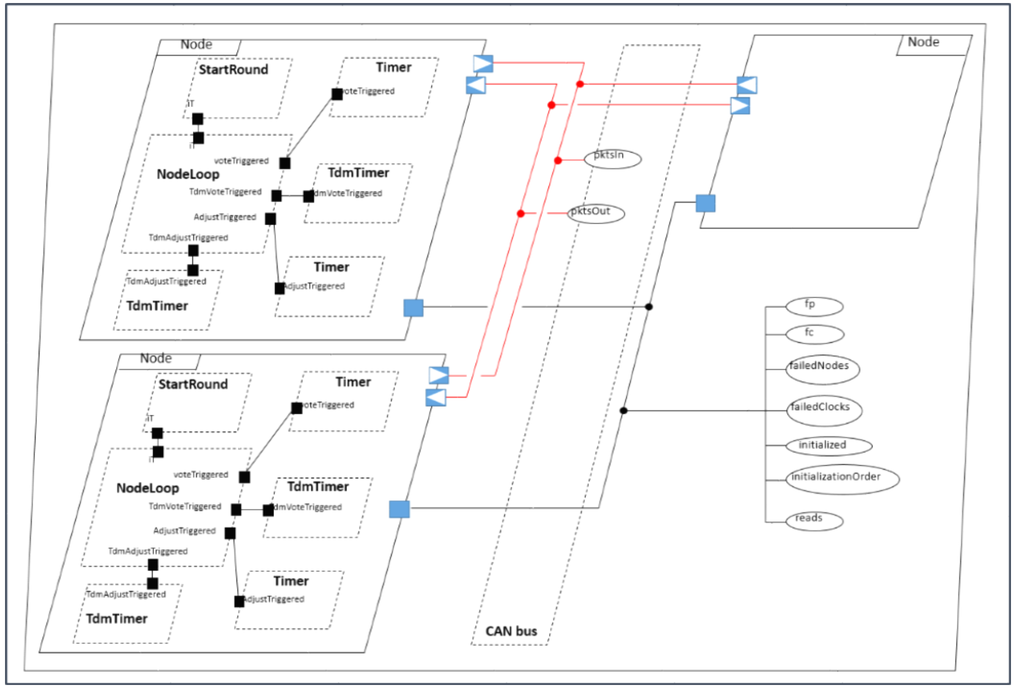
2.2.2 Formalization

Property	Requirements
CLOCK1 : the drift rate of the clock is $\rho = 10^{-6}$	Each temporal constraint of the form $[a, b]$ which relative to a duration measure by the physical local correct clock of a node is replaced with $[a(1-\rho), b(1-\rho)]$

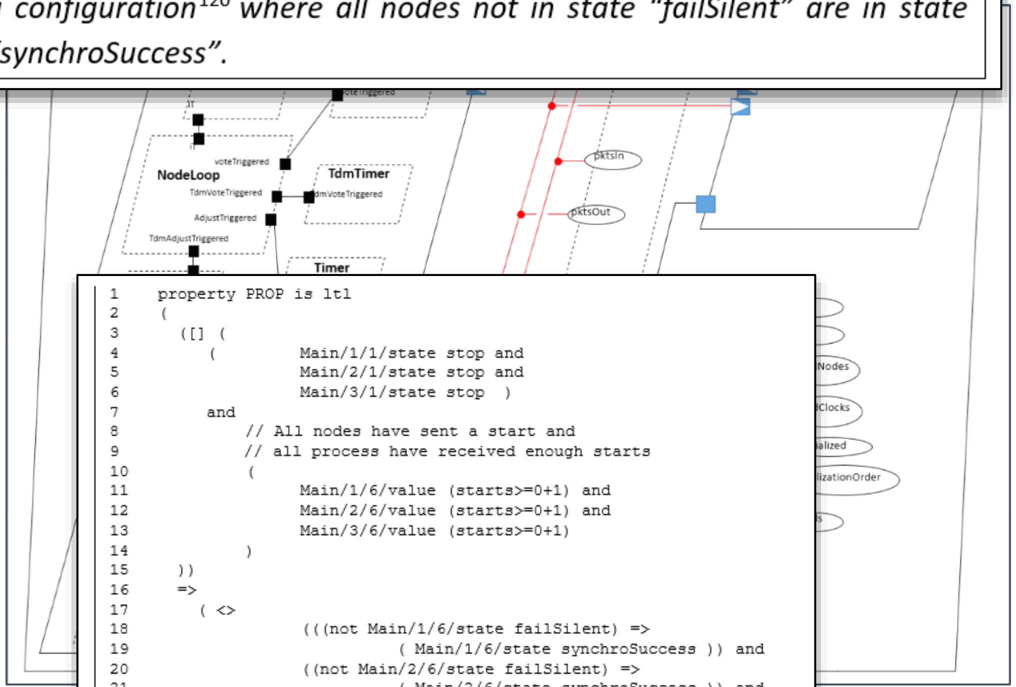
Requirements	Model elements	Rationale
Each temporal constraint of the form $[a, b]$ which relative to a duration measure by the physical local correct clock of a node is replaced with $[a(1-\rho), b(1-\rho)]$	Each temporal constraint of the form $[a, b]$ which model a duration relative to a timer is replaced with $[a(1-\rho), b(1-\rho)]$	Design choice => Duration measure local to a clock are done using timer. This are relative to local clock

Verification of timed systems: Fiacre and Tina

The model (structure)



Verification of timed systems: Fiacre and Tina



It is always true that starting from a configuration where all nodes are in state "stop" with values "starts" greater or equal to 1, the system reaches a configuration[120] where all nodes not in state "failSilent" are in state "synchroSuccess".

```
property PROP is ltl
(
  ([] (
      (        Main/1/1/state stop and
               Main/2/1/state stop and
               Main/3/1/state stop  )
    and
         // All nodes have sent a start and
         // all process have received enough starts
         (
               Main/1/6/value (starts>=0+1) and
               Main/2/6/value (starts>=0+1) and
               Main/3/6/value (starts>=0+1)
         )
  ))
  =>
    ( <>
               (((not Main/1/6/state failSilent) =>
                       ( Main/1/6/state synchroSuccess )) and
               ((not Main/2/6/state failSilent) =>
                       ( Main/2/6/state synchroSuccess )) and
               ((not Main/3/6/state failSilent) =>
                       ( Main/3/6/state synchroSuccess )))
```

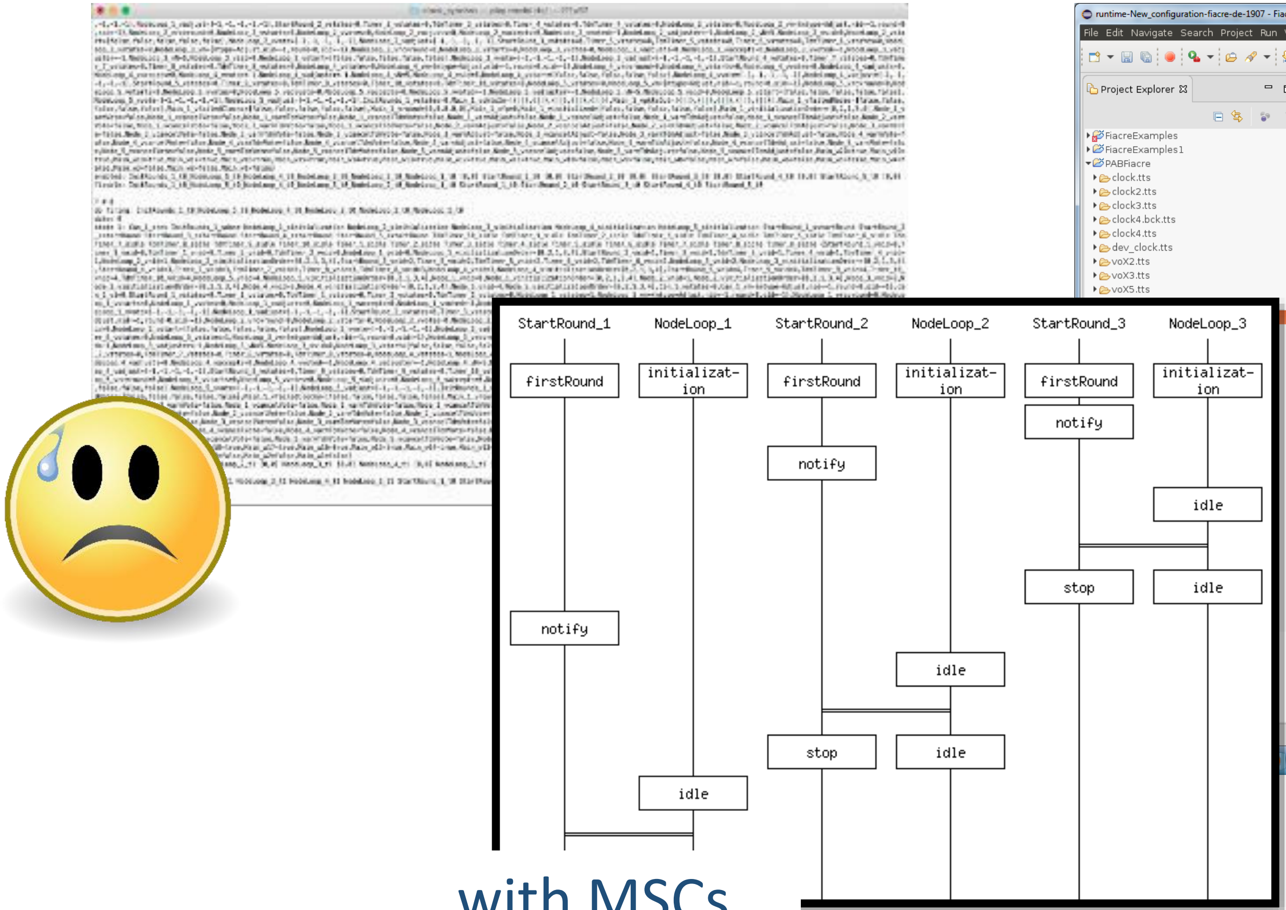
The model (behaviour)

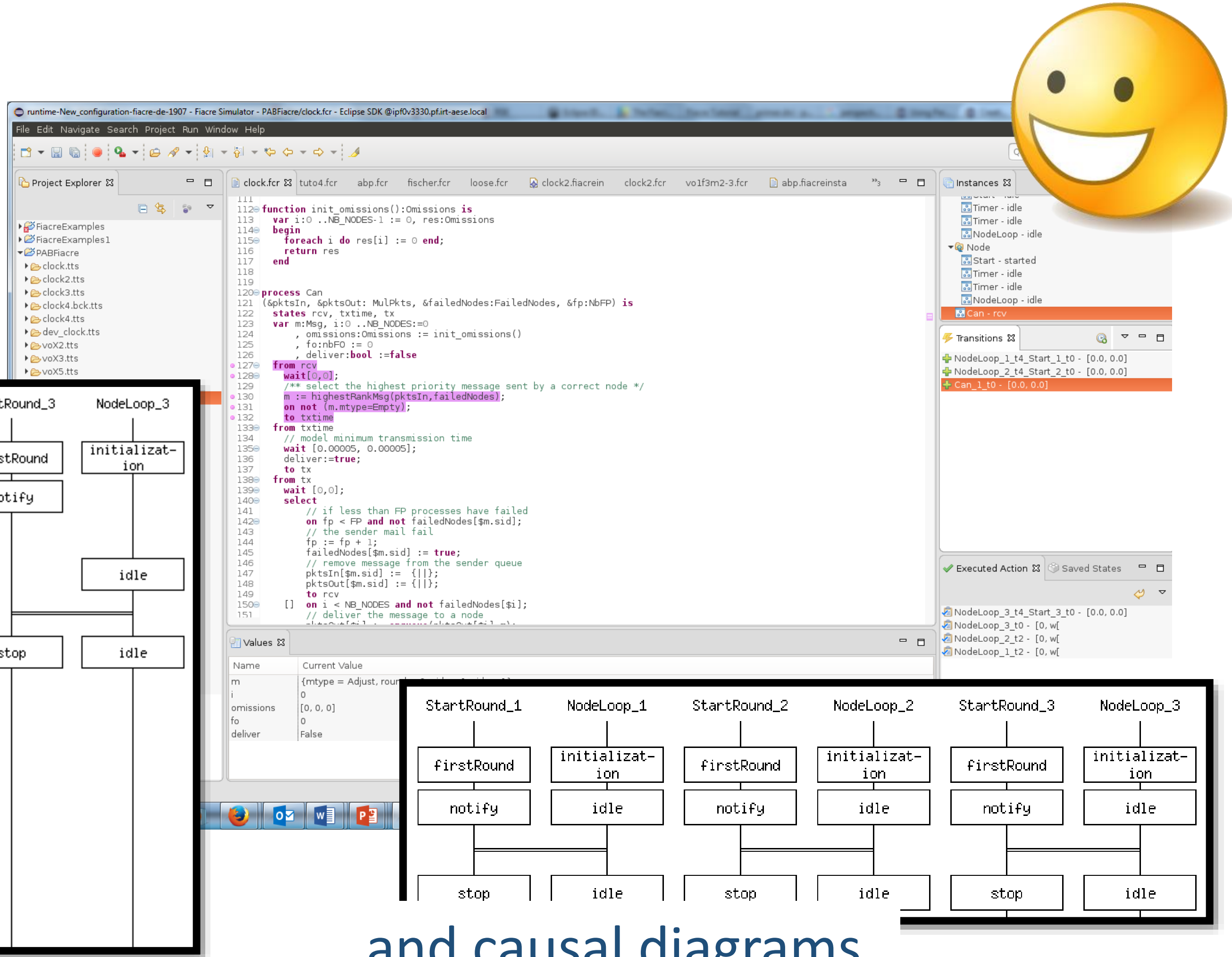
```
// Timer model a timer
process Timer[triggered:none](nid:Nodes, &arm:bool, &cancel:bool,
      &fc:nbFC, &failedClocks:FailedClocks
) is
states idle, armed, failing, triggered
from idle
  on arm;
  wait [0,0];
    cancel := false;
  to armed
from armed
  select
      on arm and not cancel;
      select
         on fc < FC and not failedClocks[$nid];
         // clock failing
         wait [0,0.005]; failedClocks[$nid] := true; fc := fc +1; to failing
      [] on failedClocks[$nid];
         wait [0, ...[
      [] on not failedClocks[$nid];
         wait [0.005,0.005]
      end
    []  on cancel; wait[0,0]; arm:=false; cancel:=false; to idle
  end;
    to triggered
from failing
  select
    on arm and not cancel;  wait [0, ...[
  // nevertheless, cancel is still possible
    []  on cancel; wait[0,0]; arm:=false; cancel:=false;  to idle
  end;
    to triggered
from triggered
  select
    on cancel; wait[0,0]
  [] triggered
  end
    cancel:=false;
    arm:=false;
    idle
```


```
[] on failedClocks[$nid];
   // clock failed
   wait [0, ...[
```

Verification of timed systems: Fiacre and Tina

From a crude HMI… … to a fancy Eclipse simulation environment





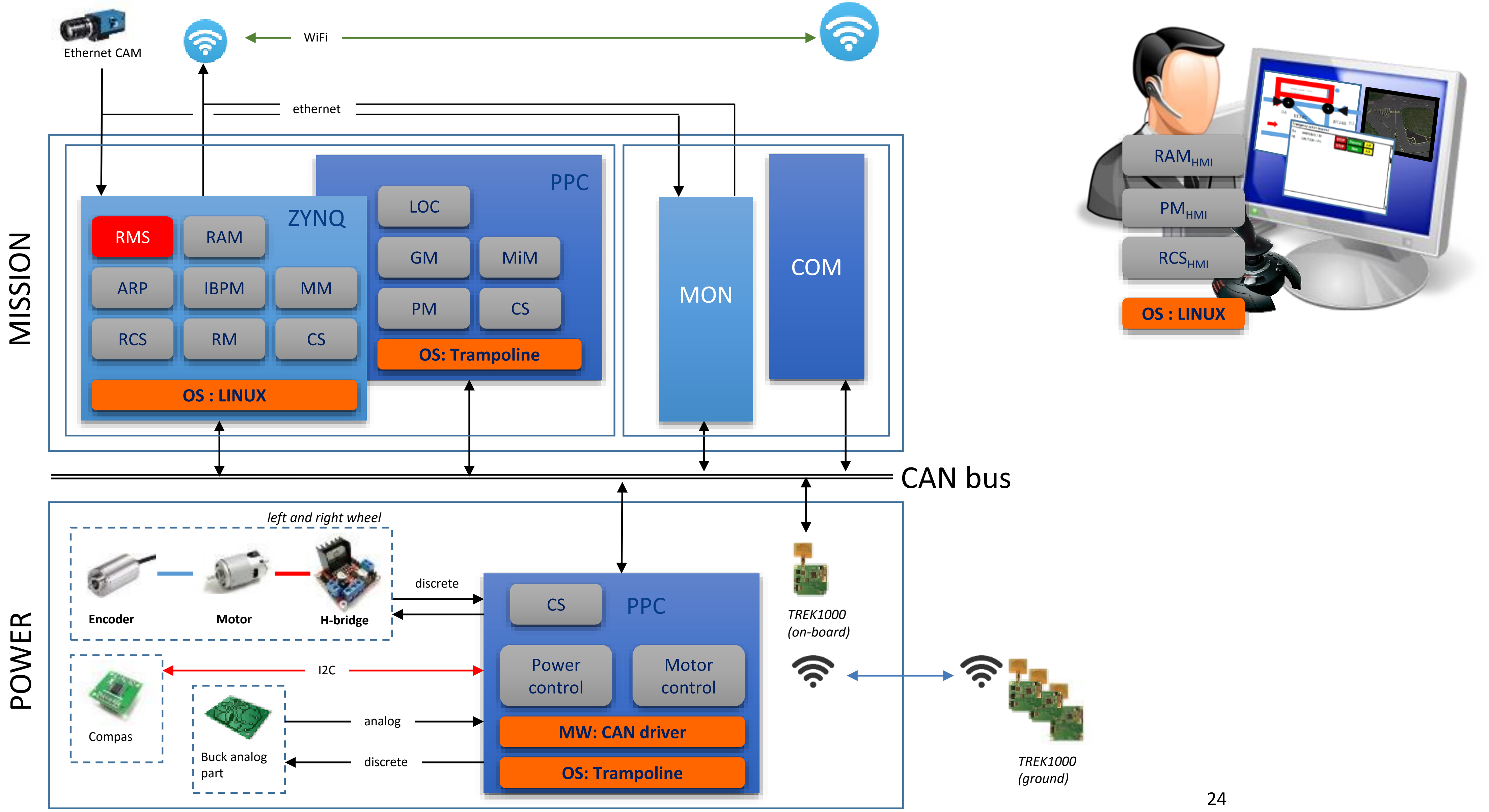
… with MSCs… … and causal diagrams.

Verification of timed systems: Fiacre and Tina

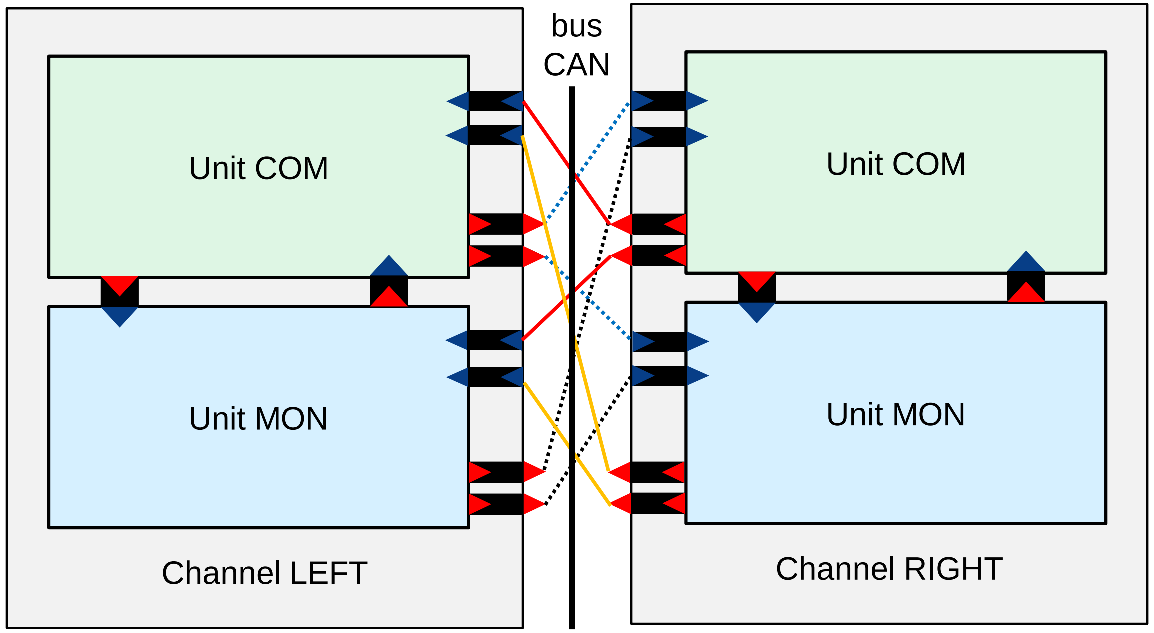
- Some specific properties have been verified: are they worth the effort?
- How to gain confidence on the model?
- How to debug the model ?

- From "simulation" to "advanced simulation"
  - Guided-simulation (property-driven simulation)

Redundancy Mngt: Verification of synchronous SW:



Verification of synchronous SW: Redundancy Mngt



CRM_R3. *There shall never be two confirmed MASTERS in different channels.*

CRM_P3.

$\sim(masterChannel = RIGHT \;\&\; masterChannel = LEFT);$

CRM_R5. *The time window shall end at the same time on all channels.*

CRM_P5.

$ALL\ i: [0, NB_{UNITS} - 1](env_{isOn[i]})$
$\&\ SOME\ i: [0, NB\_UNITS - 1]\ (\sim InSW[i])$
$\rightarrow\ ALL\ j: [0, NB\_UNITS - 1]\ (\sim InSW[j]);$

Verification of synchronous SW: Redundancy Mngt

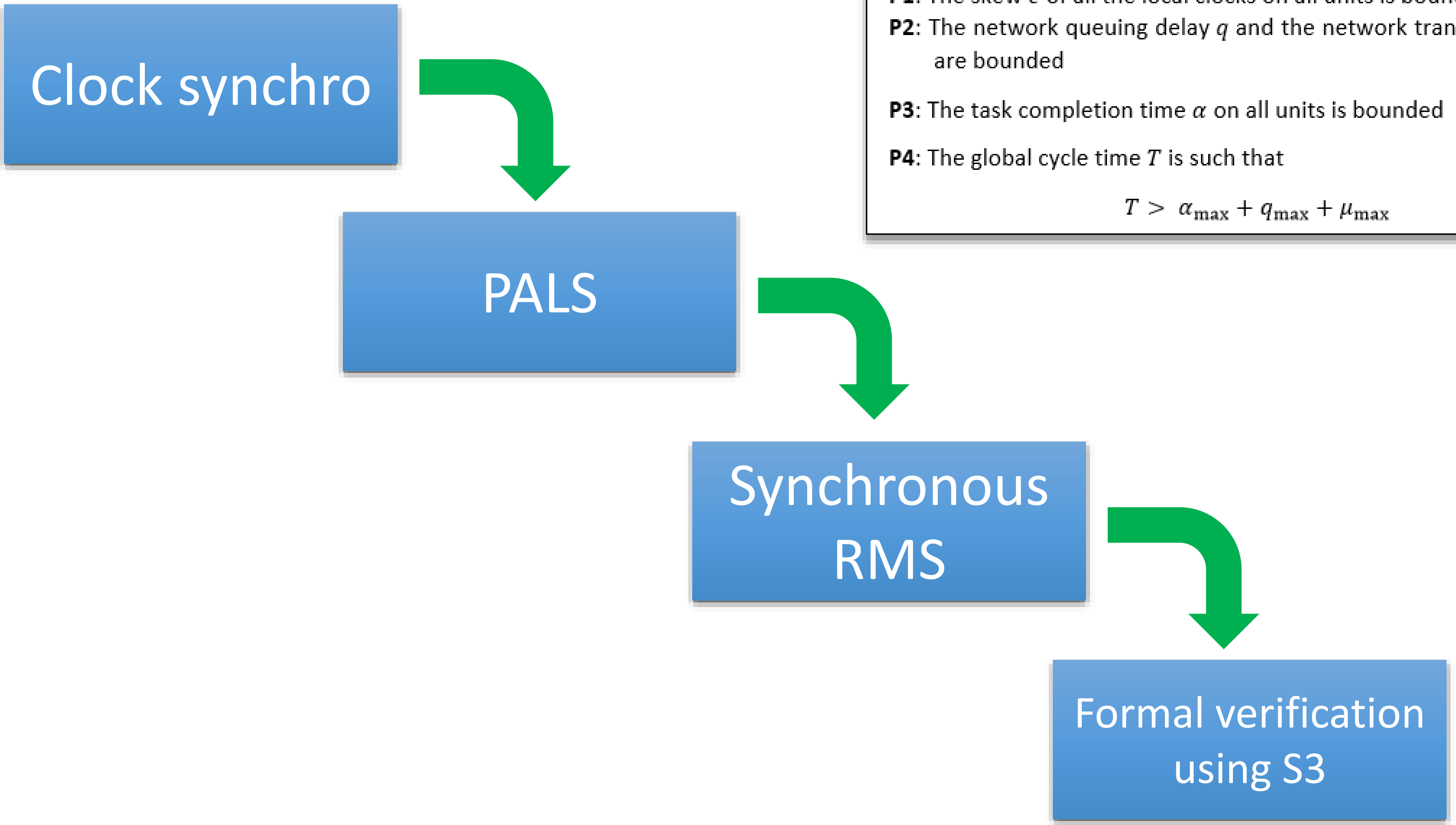
P1: The skew ϵ of all the local clocks on all units is bounded

P2: The network queuing delay q and the network transmission delay μ are bounded

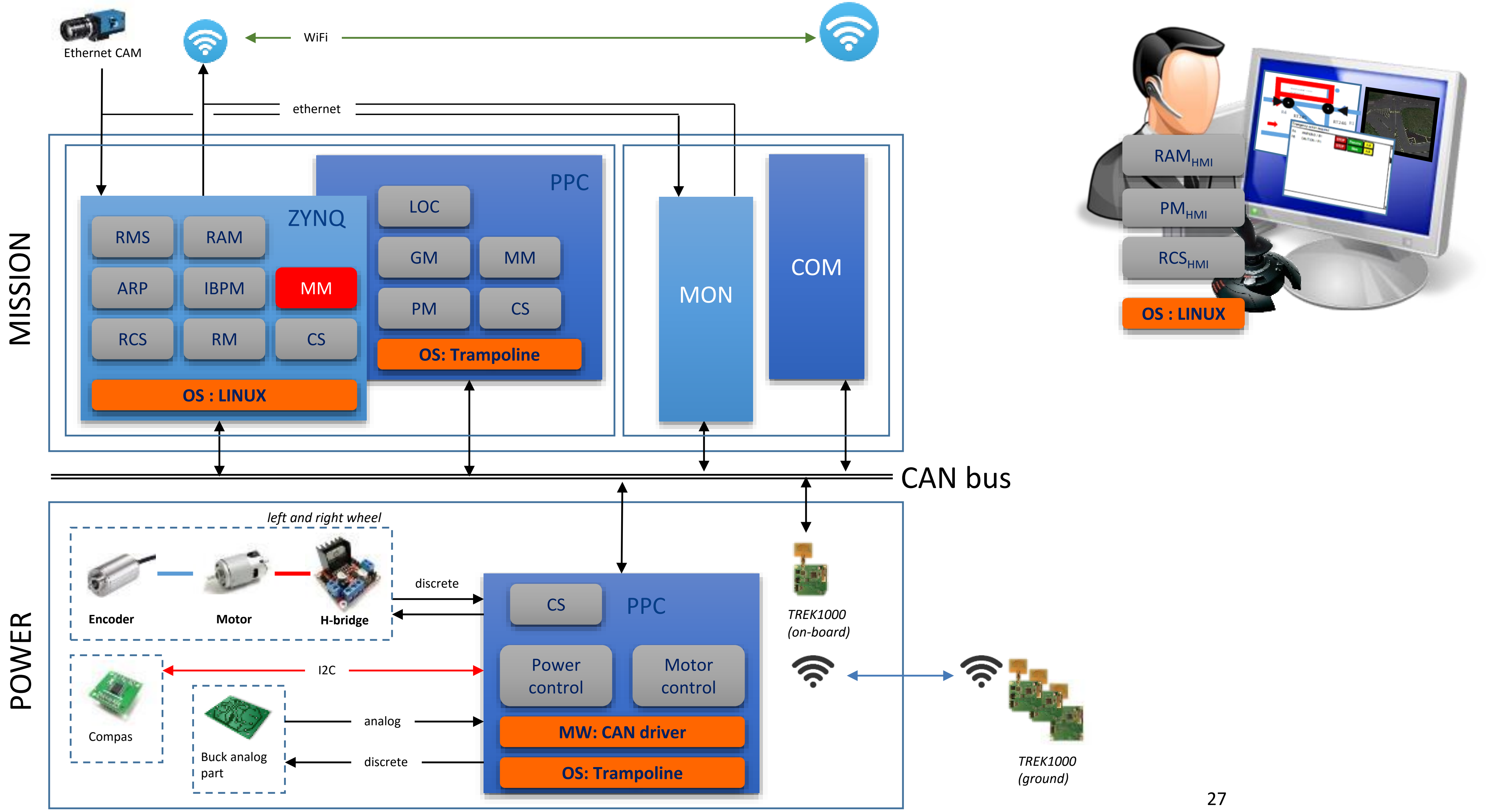
P3: The task completion time α on all units is bounded

P4: The global cycle time T is such that

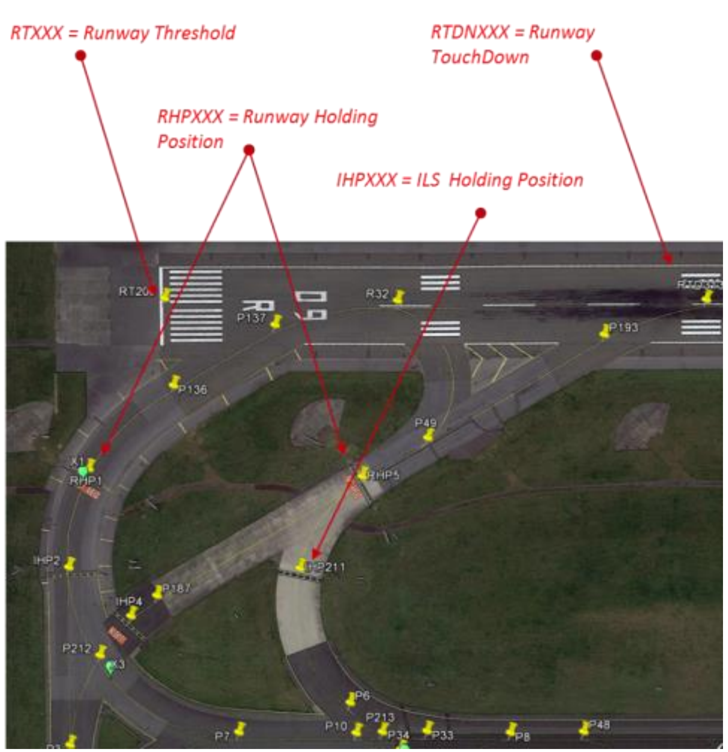
$$T > \alpha_{\max} + q_{\max} + \mu_{\max}$$



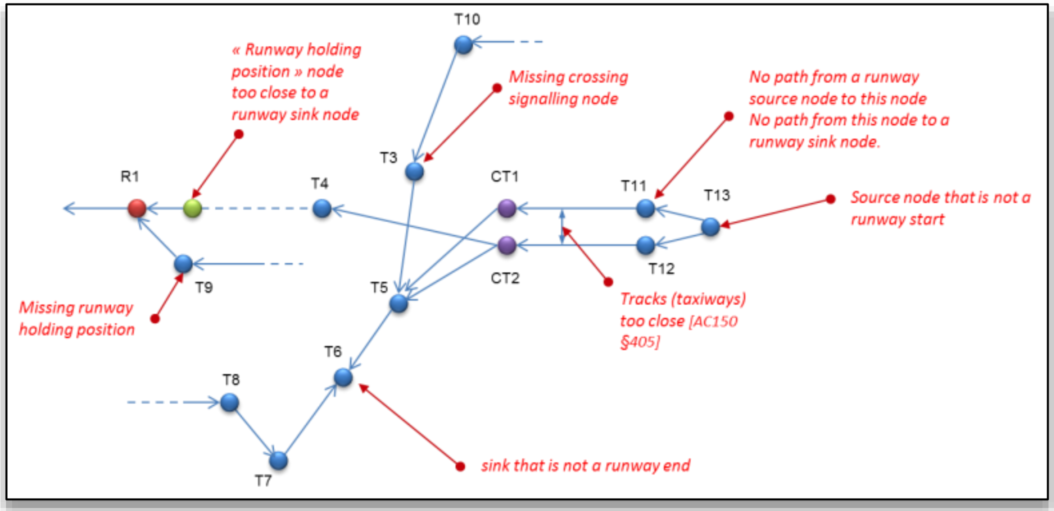
Mission management: Verification of configuration data



Formal verification of configuration data



No path from a runway source node to this node
No path from this node to a runway sink node.



Formal verification of configuration data

[Carto-P001] No intersection of edges

$\forall A, B, C, D \in Wps \wedge e1,\ e2 \in Edges \wedge A, B \in e1 \wedge C, D \in e2\ \wedge (A\ \neq C) \wedge\ (A\ \neq D) \wedge (B\ \neq\ C) \wedge (B\ \neq\ D),$

$\text{intersectant}(e1,\ e2) = true\ \Leftrightarrow \big(ccw(A,\ B,\ D) \neq ccw(B,\ C,\ D)\big) \wedge \big(ccw(A,\ B,\ C) \neq ccw(A,\ B,\ D)\big),$

where

$$\text{ccw}(A, B, C) = (C.y - A.y)(B.x - A.x) > (B.y - A.y)(C.x - A.x)$$

[Carto-P004] Connected runway nodes

$$\forall\, r1,\ r2\ \in RUN$$

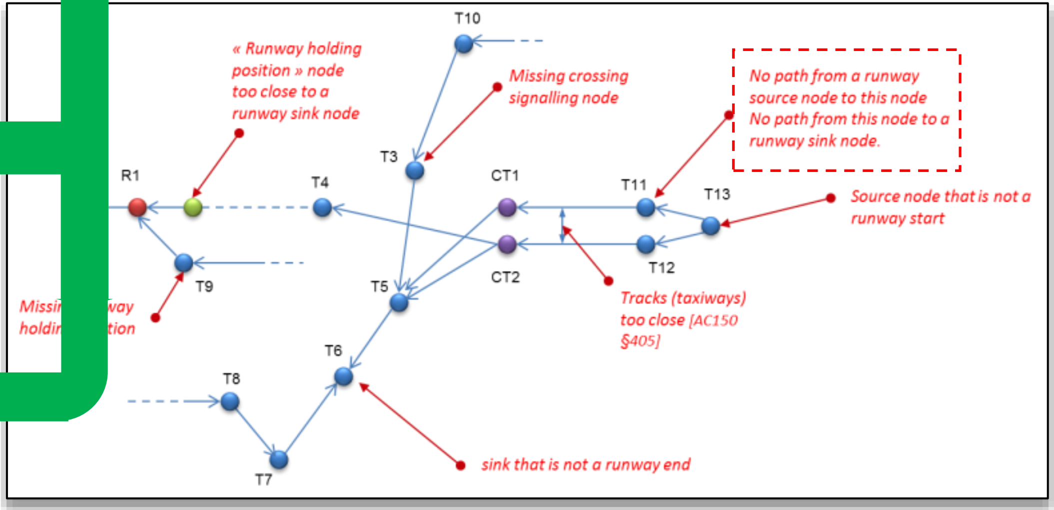
$$\big(r1 \neq r2\ \rightarrow\ reachable(r1,\ r2)\ \wedge\ reachable(r2,\ r1)\big),$$

where reachable (a, b) stands for the node can reach the node b.

[Carto-P005] Aligned runway nodes

$\forall\, r1,\ r2,\ r3 \in RUNWAY,\ r1 \neq r2 \wedge r2 \neq r3\ \rightarrow aligned(r1,\ r2,\ r3, \delta)$, where RUNWAY is the set of runway nodes, and δ is the tolerance alignment value.

No path from a runway source node to this node
No path from this node to a runway sink node.



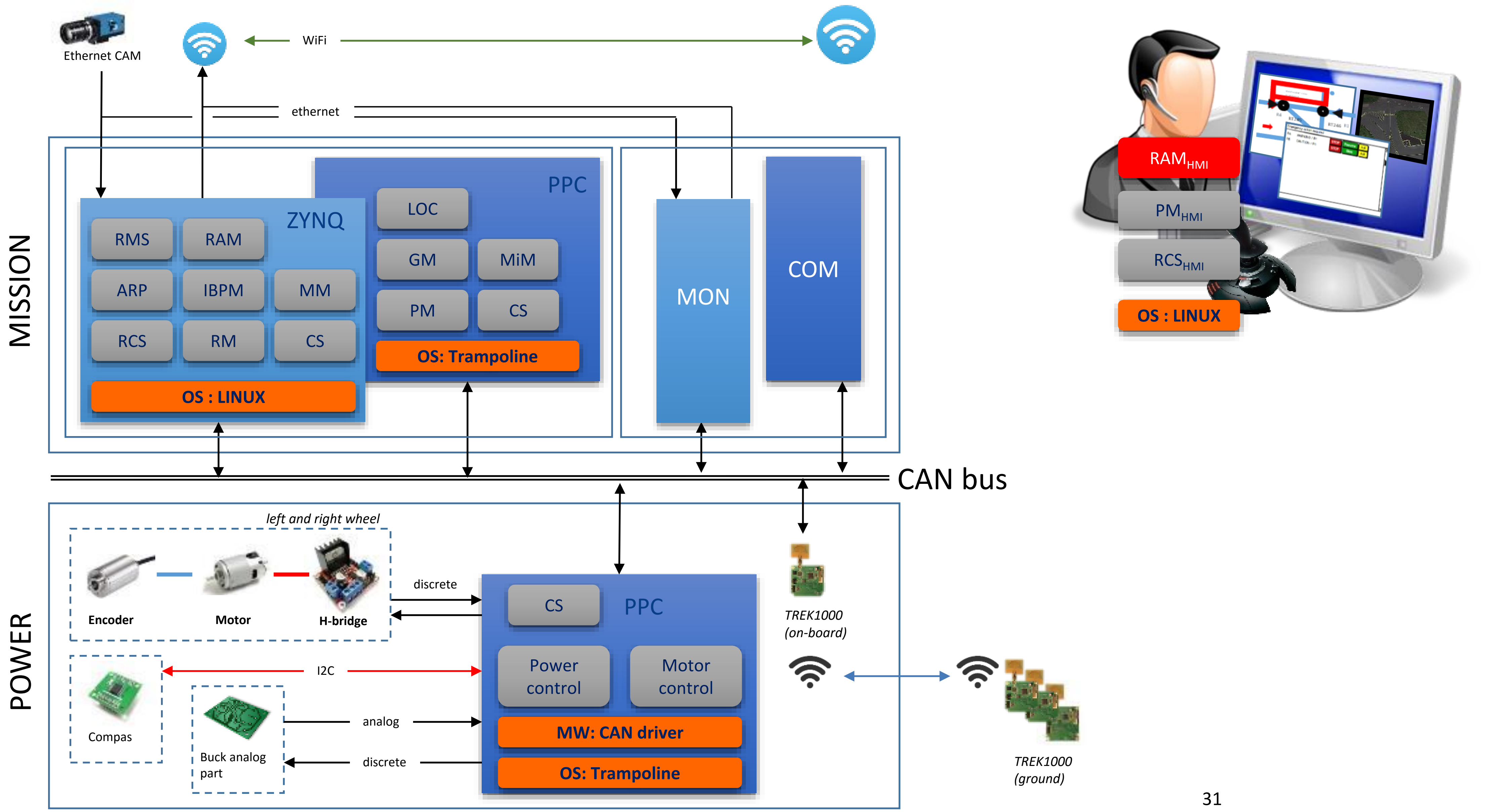
HLL → **S3**

Formal verification of configuration data

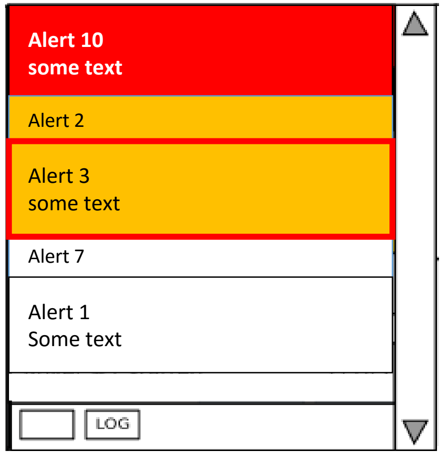
- A way to get acquainted to the formalism and tool…

- Consider dedicated tools!

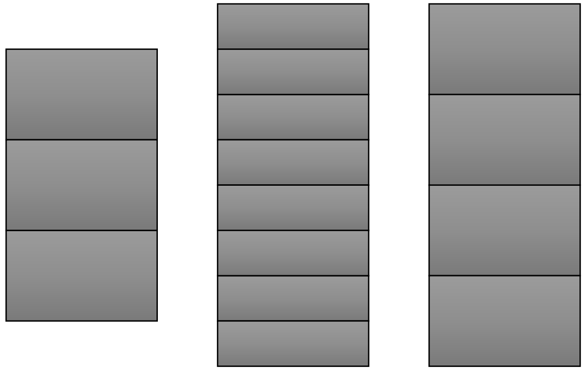
Supervision: Formal verification of HMIs



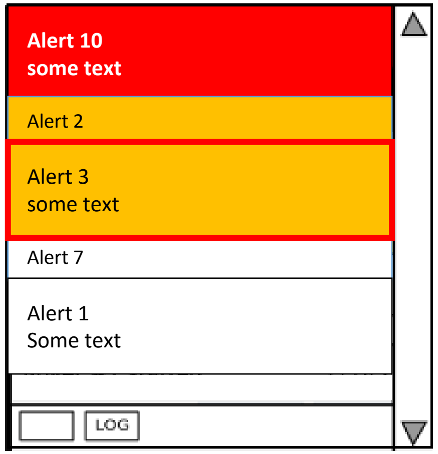
Supervision: Formal verification of HMIs



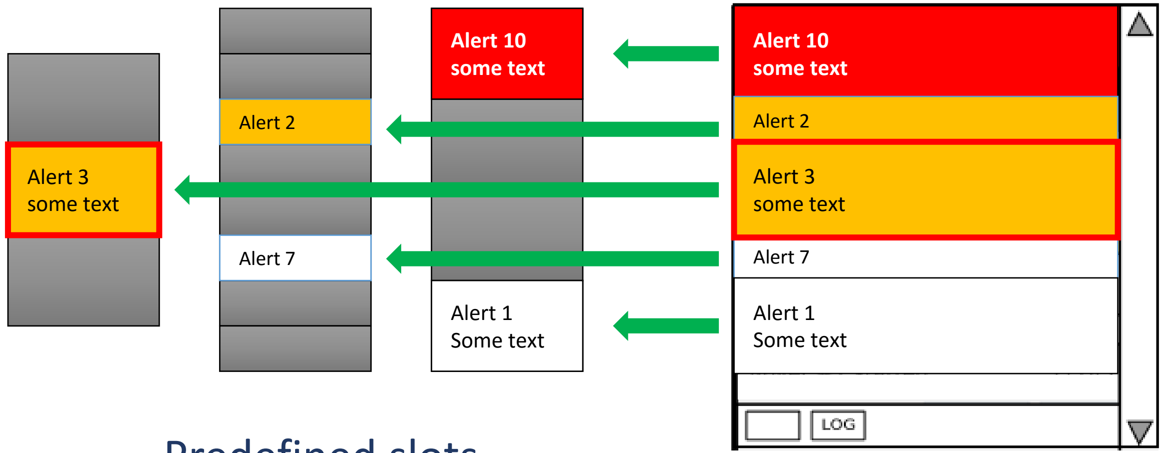
Supervision: Formal verification of HMIs



Predefined slots

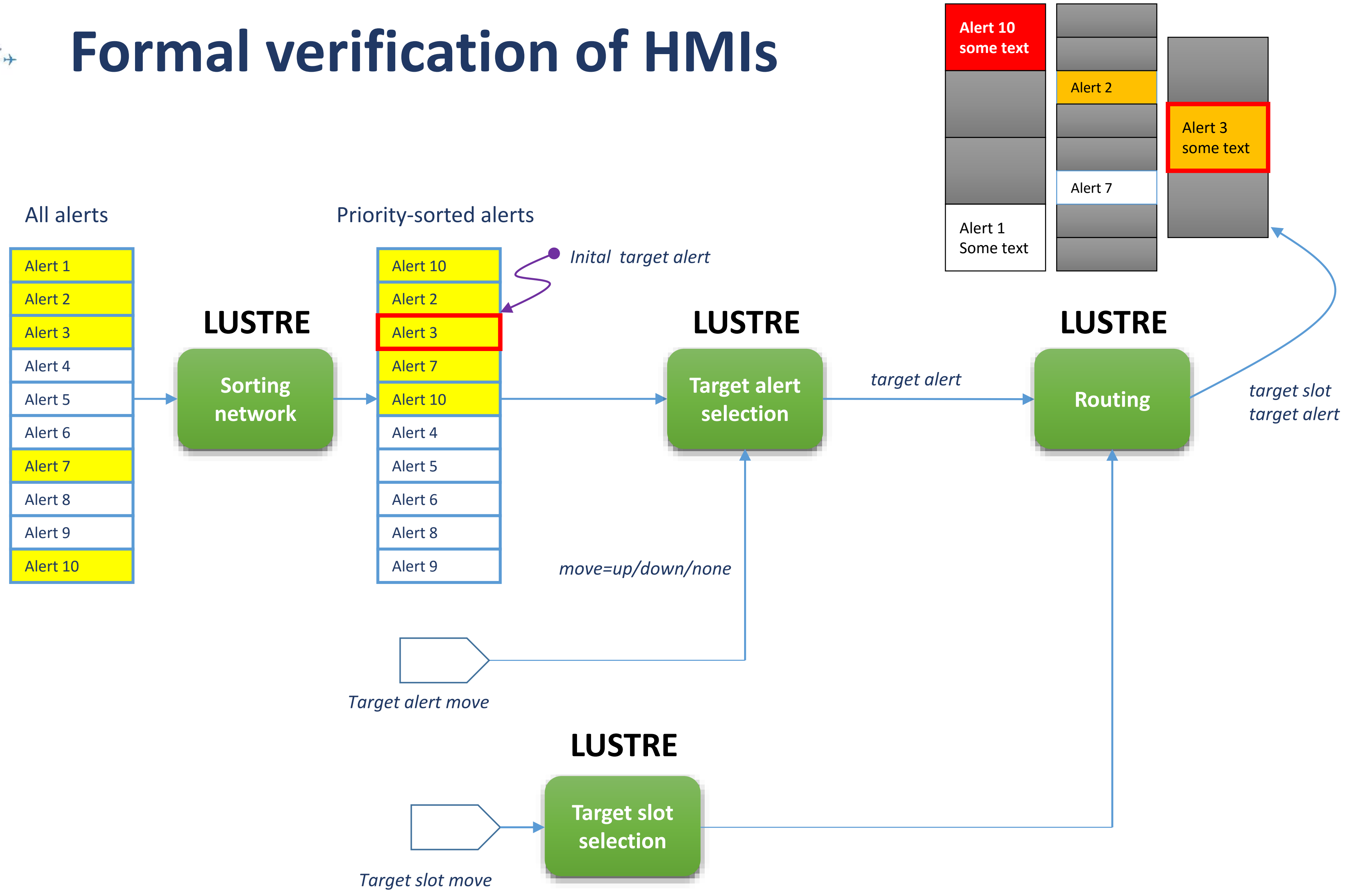


Supervision: Formal verification of HMIs



Predefined slots

Formal verification of HMIs



Formal verification of HMI

VISIBILITY	**REQ-16:** Only active alerts shall be displayed. **PROP-16:** $\mathbf{ALL}\ s: slots\ (s.id\ != -1 \rightarrow\ \mathbf{ALL}\ i: [0,7](a[i].id = s.id \rightarrow a[i].act))$
TASK RELATED	**REQ-8-1:** If the user clicks the [target alert up] button and the priority of previous selected alert is greater than the priorities of any current alerts, then the index of the returned alert shall be 0. **PROP-8-1:** $(aselmov = up\ \&\ \mathbf{ALL}\ i: [0,7](\ p > a[i].p\ )) \rightarrow idx = 0$
RELIABILITY	**REQ-17:** The target alert shall always be displayed in the target slot. **PROP-17:** $ALL\ s: Slots, a: Alerts\ ((s.x = target\_x\ \&s.y = target\_y\ \&a.idx = target\_index) \rightarrow s.id = a.id)$

Formal verification of HMIs

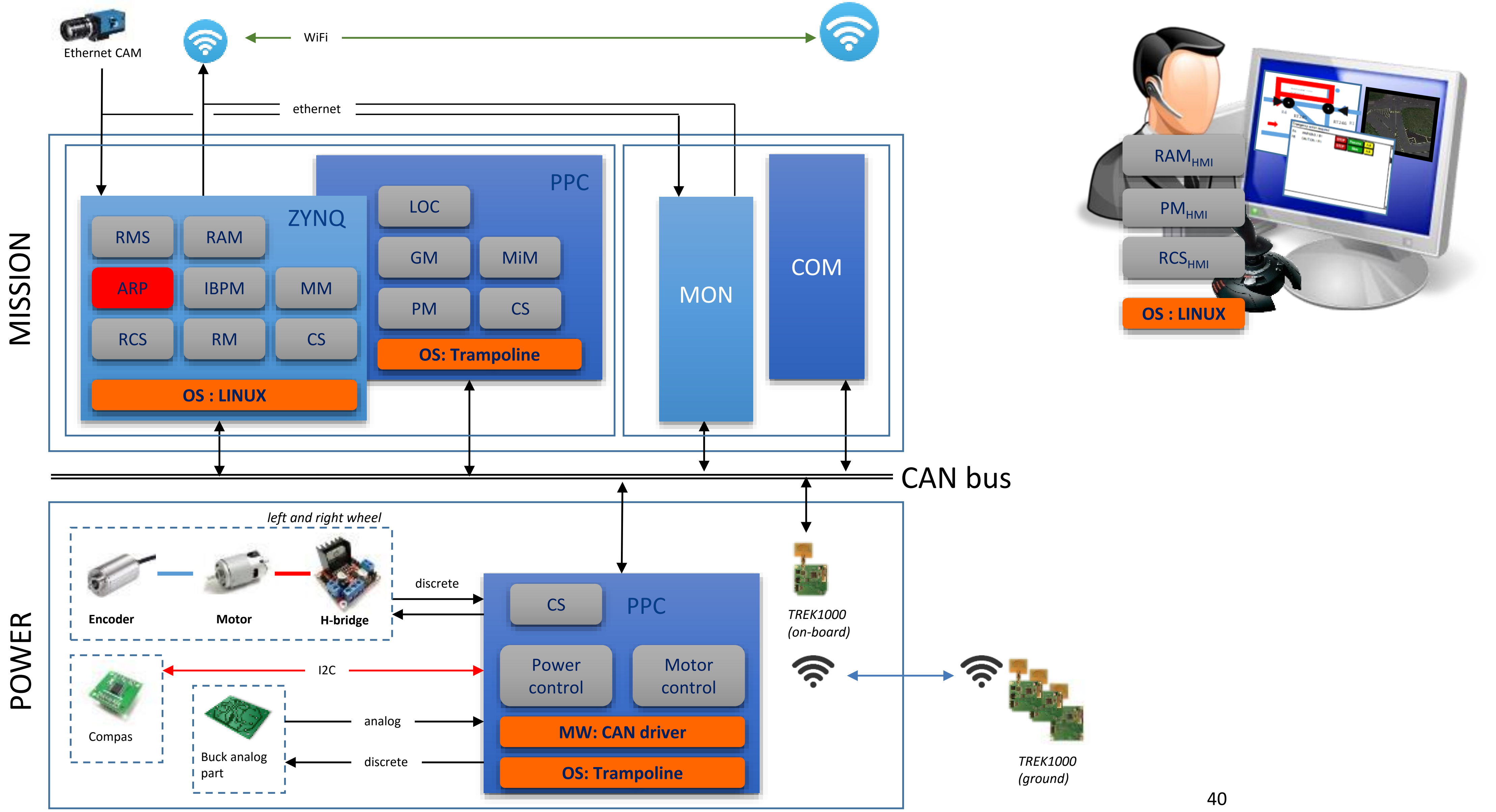
Informal specification
Formal specification
Formal design
Formal verification of implementation

Formal verification of HMIs

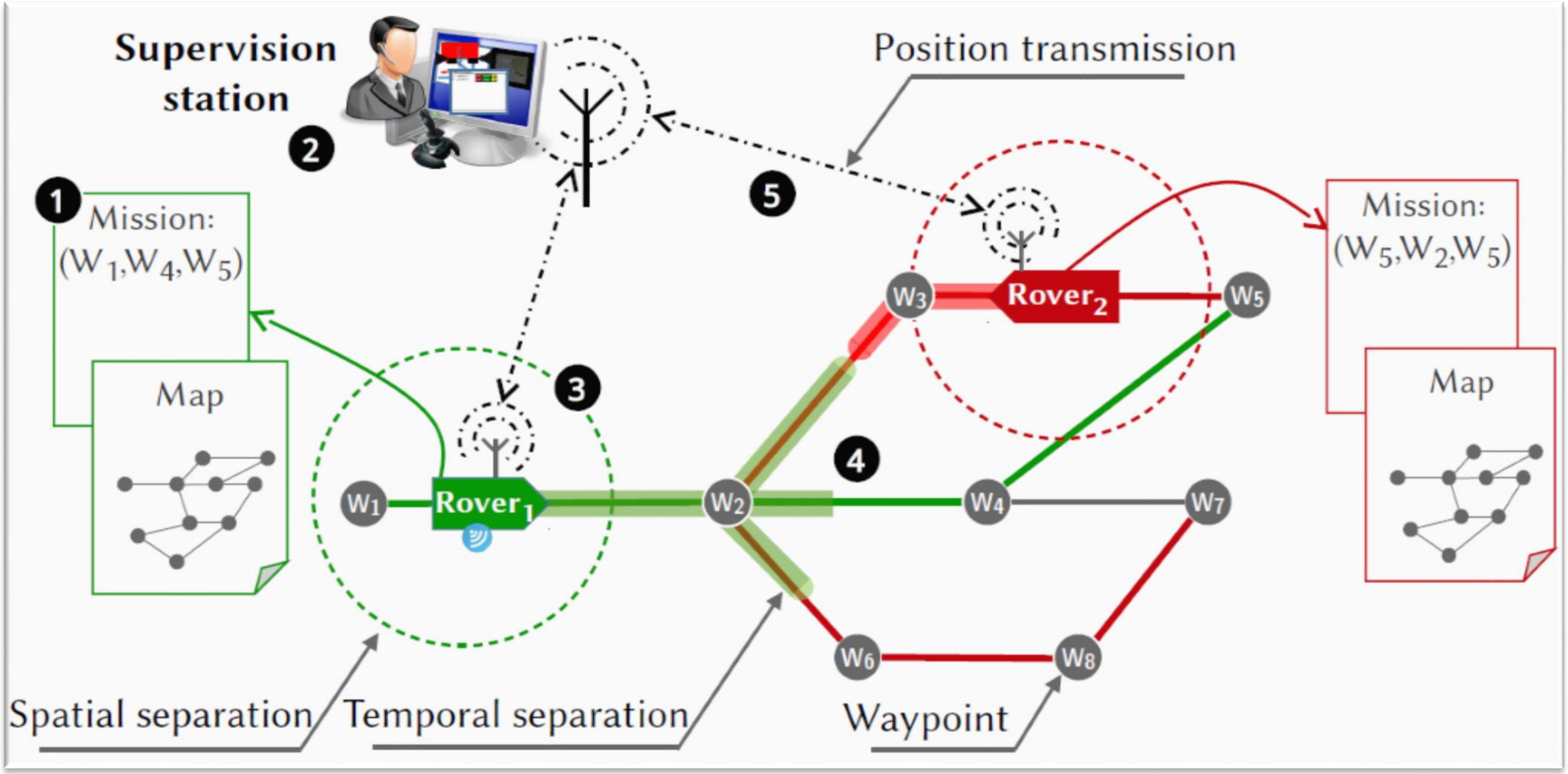
- A *very simple* HMI designed using formal models and methods…

- HMIs deserve formal specification and verification
- HMIs are (probably) in the area where model-checking techniques are very efficient

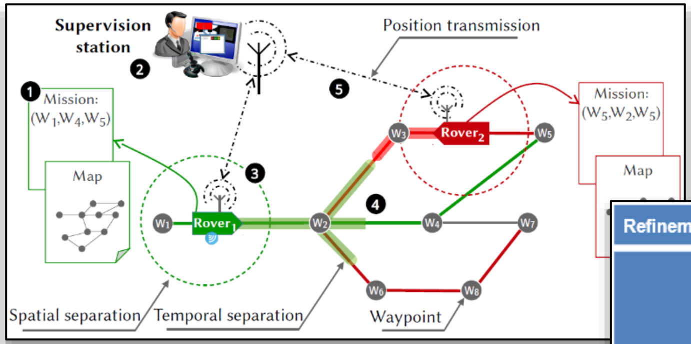
Anticollision function: Correction *by construction*



Anticollision function: Correction *by construction*



Anticollision function: Correction *by construction*



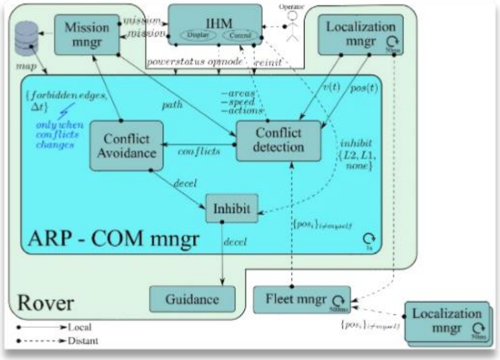
Refinement	Rover (HL-RV)	Operator (HL-OP)
Initial 1 (one rover)	HL-RV-ENV-R001, HL-RV-ENV-R001.01, HL-RV-ENV-R001.02, HL-RV-ENV-R002, HL-RV-ENV-R002.01, HL-RV-ENV-R003, HL-RV-ENV-R003.01, HL-RV-ENV-R003.02, HL-RV-ENV-R004, HL-RV-ENV-R006, HL-RV-ENV-R007.	--
Initial 2.1 (several rovers)	HL-RV-ENV-R005, HL-RV-ENV-R009 (Position emission) HL-RV-ENV-R010 (Correct position) HL-RV-SAF-R001 (Auto Safety prop.) HL-RV-FUN-R001 (Max decel)	HL-OP-FUN-R001

Refinement strategy

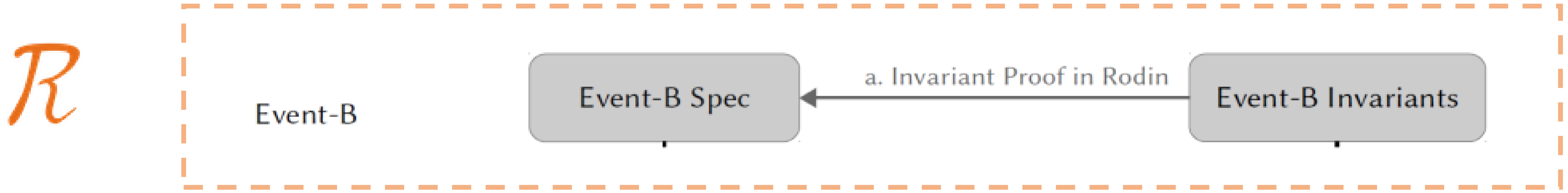
Requirements classification

Requirements layering

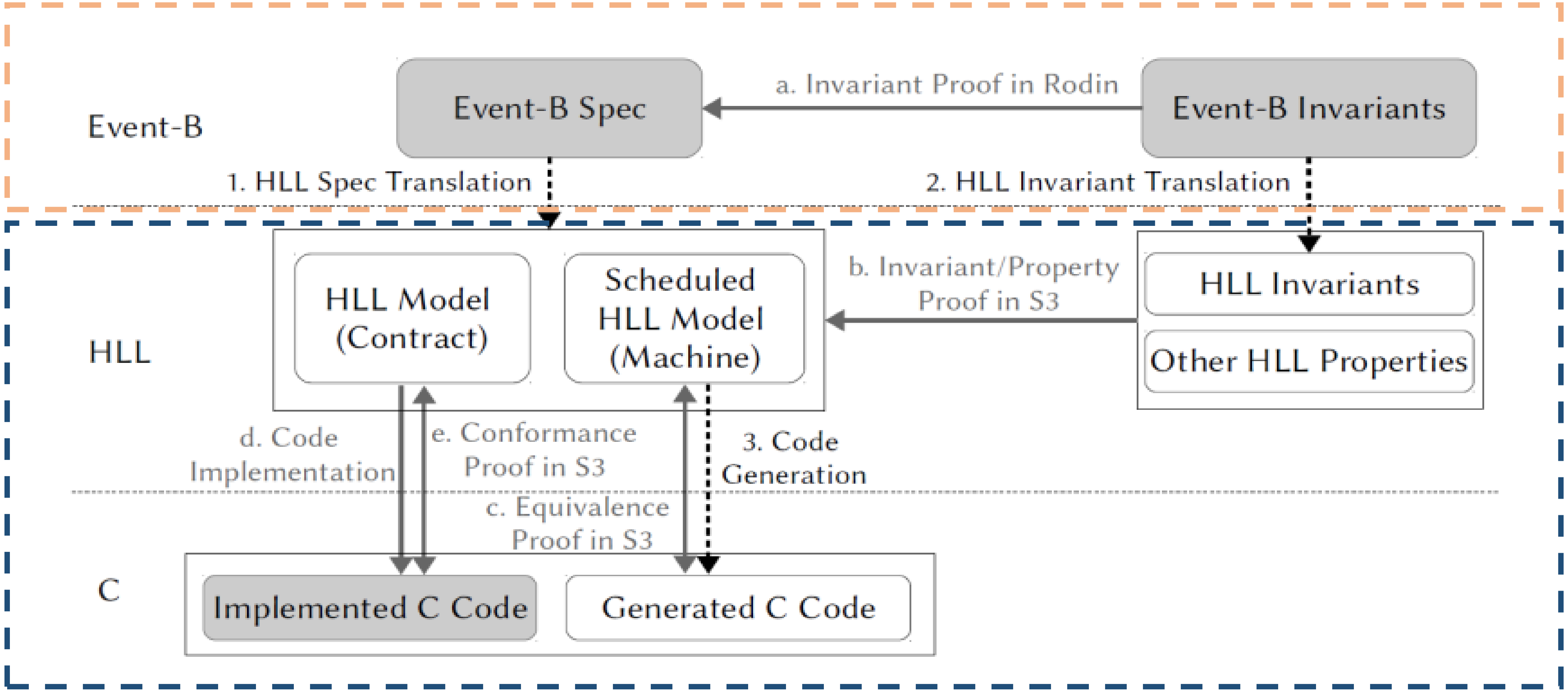
Event-B model



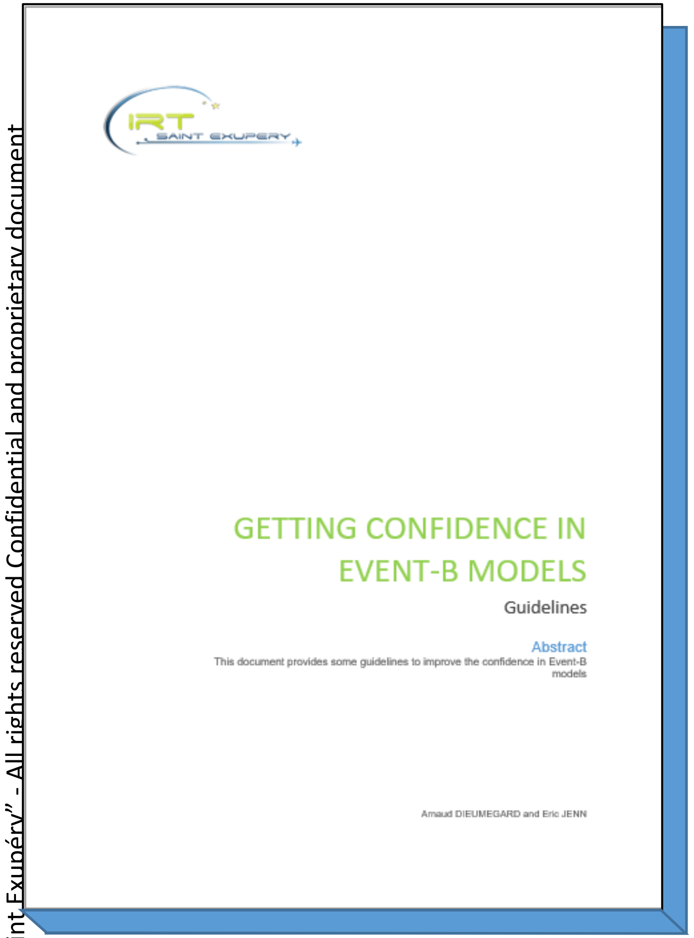
Anticollision function: Correction *by construction*



Anticollision function: Correction *by construction*



Anticollision function: Correction *by construction*



GETTING CONFIDENCE IN
EVENT-B MODELS

Guidelines

Abstract

This document provides some guidelines to improve the confidence in Event-B models

Amaud DIEUMEGARD and Eric JENN

Category 1.a: writing rules aimed at avoiding **modelling errors**

Categor[y 1.b:] writing rules aimed at facilitating the **proof process**

Rule 7: Discriminate necessary and sufficient modelling elements	
Statement (a) Make a clear distinction between the model elements that are actual parts of the specification and/or design from those introduced to support the proof process. A naming convention can be applied to discriminate those modelling elements.	
Rationales: The formal specification shall be kept as simple and short as possible. However, some modelling elements may be useful to simplify the proof. It is up to the designer to identify which one is to be kept according to the verification effort.	
Examples:	

Category 1.c: writing rules aimed at facilitating the **review process**

Category 2.a: review rules aimed at revealing **modelling errors**

Anticollision function: Correction *by construction*

- A complete (simple) function formally developed top to bottom using a combination of Robin and S3
- No floating point operations

- Automation?

Image-based position monitoring: Numerical issues

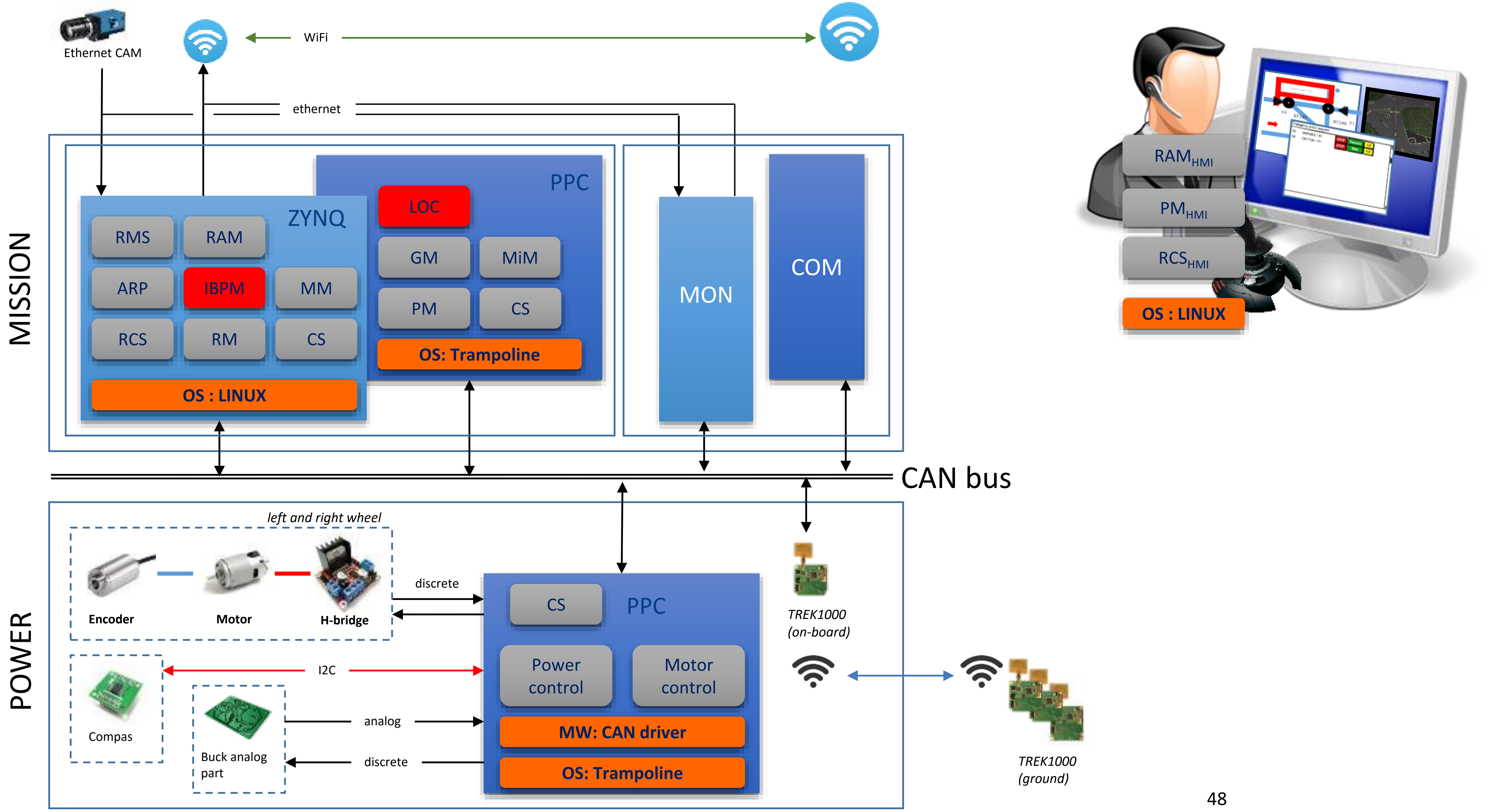


Image-based position monitoring: Numerical issues

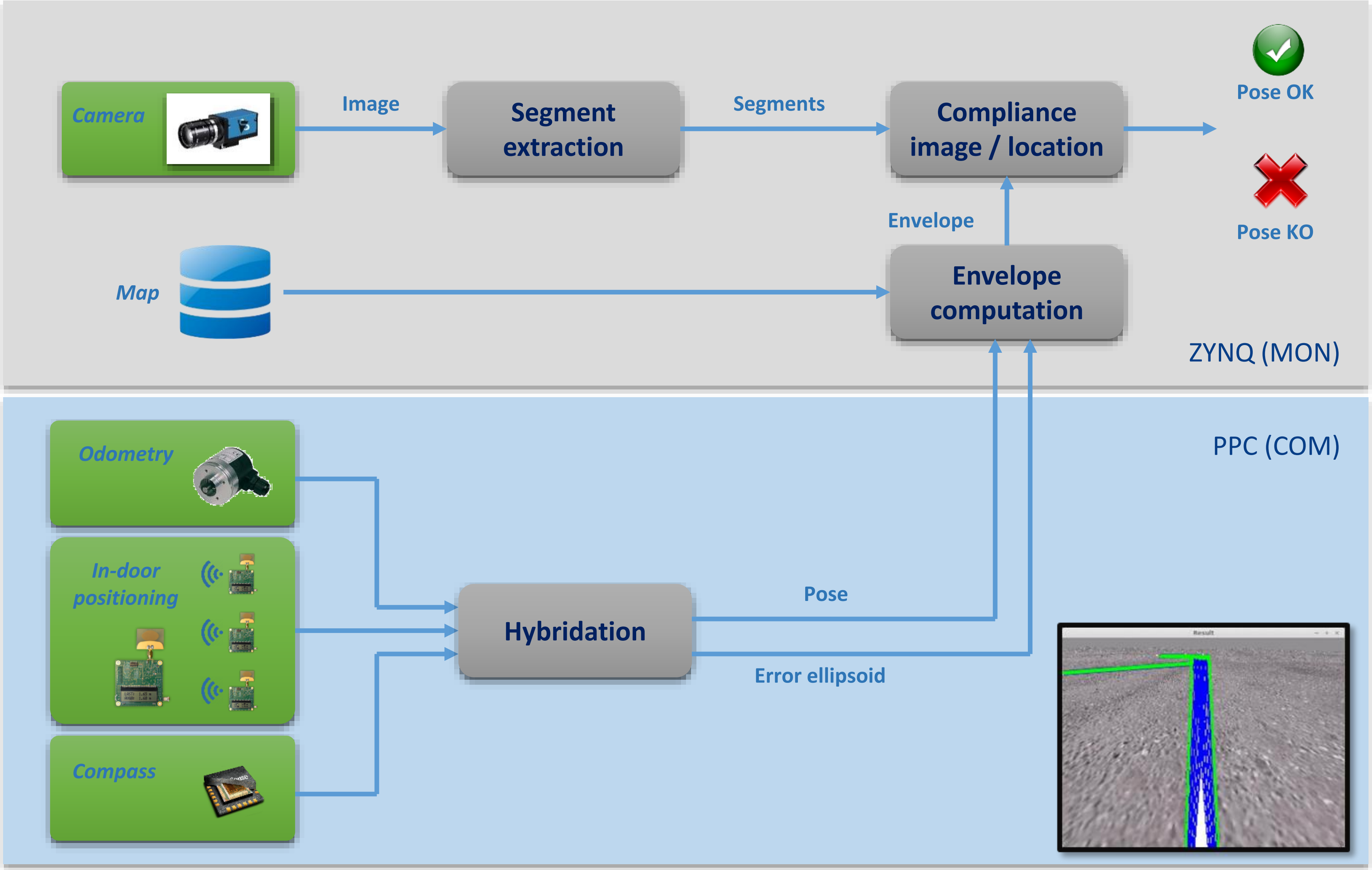


Image-based position monitoring: Numerical issues

- Matrix inversion (Kalman)

```
// Calcul du gain de Kalman
// On rappelle que S est semi-définie positive.
// K = P*H'*pinv(S);
double Kd[3][1];
T_mat K = {3,1, _MAT_ Kd };
double SId[1][1];
T_mat SI={1,1, _MAT_ SId};
double HTd[3][1];
T_mat HT = {3,1, _MAT_ HTd};
double TMP4d[3][1];
T_mat TMP4 = {3,1, _MAT_ TMP4d};

mat_transp(&H,&HT);
mat_inv(&S, &SI);
mat_mul(&HT, &SI, &TMP4);
mat_mul(Pp, &TMP4, &K);
```

Image-based position monitoring: Numerical issues

- Matrix inversion (Kalman)
- Eigenvectors (error ellipsoïd)

```
// ----------------------------------------------------------------------------
// Calculates the eigenvalues of a symmetric 3x3 matrix A using Cardano's
// analytical algorithm.
// Only the diagonal and upper triangular parts of A are accessed. The access
// is read-only.
// ----------------------------------------------------------------------------
// Parameters:
//   A: The symmetric input matrix
//   w: Storage buffer for eigenvalues
// ----------------------------------------------------------------------------

{
  double m, c1, c0;

  // Determine coefficients of characteristic poynomial. We write
  //       | a   d   f  |
  //  A =  | d*  b   e  |
  //       | f*  e*  c  |
  double de = A[0][1] * A[1][2];                                    // d * e
  double dd = SQR(A[0][1]);                                         // d^2
  double ee = SQR(A[1][2]);                                         // e^2
  double ff = SQR(A[0][2]);                                         // f^2
  m  = A[0][0] + A[1][1] + A[2][2];
  c1 = (A[0][0]*A[1][1] + A[0][0]*A[2][2] + A[1][1]*A[2][2])        // a*b + a*c + b*c - d^2 - e^2 - f^2
          - (dd + ee + ff);
  c0 = A[2][2]*dd + A[0][0]*ee + A[1][1]*ff - A[0][0]*A[1][1]*A[2][2]
            - 2.0 * A[0][2]*de;                                     // c*d^2 + a*e^2 + b*f^2 - a*b*c -
2*f*d*e)

  double p, sqrt_p, q, c, s, phi;
  p = SQR(m) - 3.0*c1;
  q = m*(p - (3.0/2.0)*c1) -
  sqrt_p = sqrt(fabs(p));

  phi = 27.0 * ( 0.25*SQR(c1
  phi = (1.0/3.0) * atan2(sq

  c = sqrt_p*cos(phi);
  s = (1.0/M_SQRT3)*sqrt_p*s

  w[1]  = (1.0/3.0)*(m - c);
  w[2]  = w[1] + s;
  w[0]  = w[1] + c;
  w[1] -= s;

  return 0;
}
```

$$\begin{pmatrix} 10^{20} & 10^{9} & 10^{9} \\ 10^{9} & 10^{20} & 10^{9} \\ 10^{9} & 10^{9} & 1 \end{pmatrix}$$

Exact

$$v_1 = (1 + 10^{-11}) \cdot 10^{20}$$
$$v_2 = (1 - 10^{-11}) \cdot 10^{20}$$
$$v_3 = 0.98$$

$$\mathbf{v}_1 = \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \\ 0 \end{pmatrix}, \quad \mathbf{v}_2 = \begin{pmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \\ 0 \end{pmatrix}, \quad \mathbf{v}_3 = \begin{pmatrix} 10^{-11} \\ 10^{-11} \\ 1 \end{pmatrix}$$

CORRECT

Approx

$$v_1 = (1) \cdot 10^{20}$$
$$v_2 = (1) \cdot 10^{20}$$
$$v_3 = 0.98$$

$$\mathbf{v}_1 = \begin{pmatrix} 1/\sqrt{3} \\ 1/\sqrt{3} \\ -1/\sqrt{3} \end{pmatrix}, \quad \mathbf{v}_2 = \begin{pmatrix} 2/\sqrt{6} \\ -1/\sqrt{6} \\ 1/\sqrt{6} \end{pmatrix}, \quad \mathbf{v}_3 = \begin{pmatrix} 0 \\ 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix}$$

ERRONEOUS

Image-based position monitoring: Numerical issues

- Matrix inversion (Kalman)
- Eigenvectors (error ellipsoïd)
- Geometrical computations (enveloppe computation)

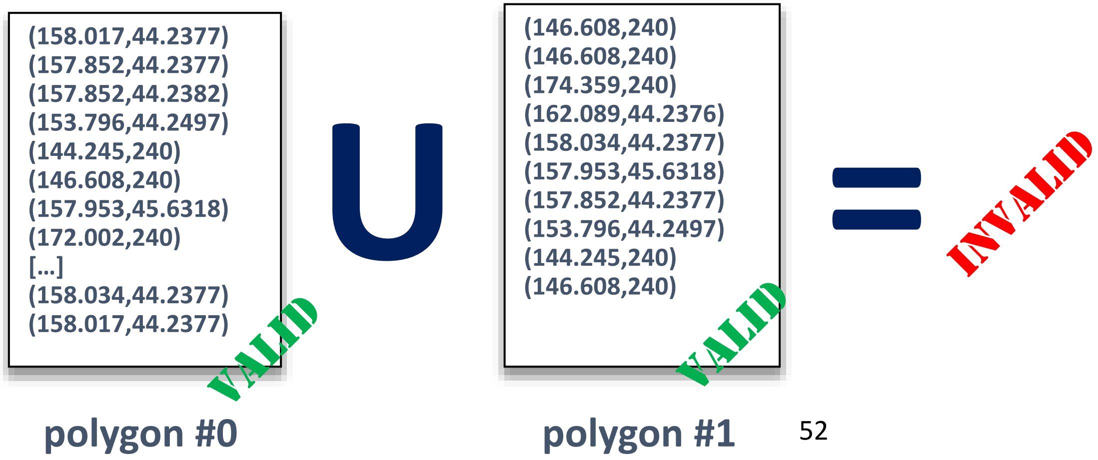


Image-based position monitoring: Numerical issues

- Excerpt from the BOOST library

there are some other rules that valid

Besides the Concepts, which are checks on compile-time, there are some other rules that valid polygons must fulfill. This follows the opengeospatial rules (see link

polygons must fulfill.

- Polygons are simple geometric objects (See also wiki but ho polygons).
- If the polygons underlying `ring_type` is defined as clockwise, the exterior ring must have the clockwise orientation, and any interior ring must be counter clockwise. If the `ring_type` is defined counter clockwise, it is vice versa.
- If the polygons underlying `ring_type` is defined as closed, all rings must be closed: the first point must be spatially equal to the last point.

There should be no self intersections

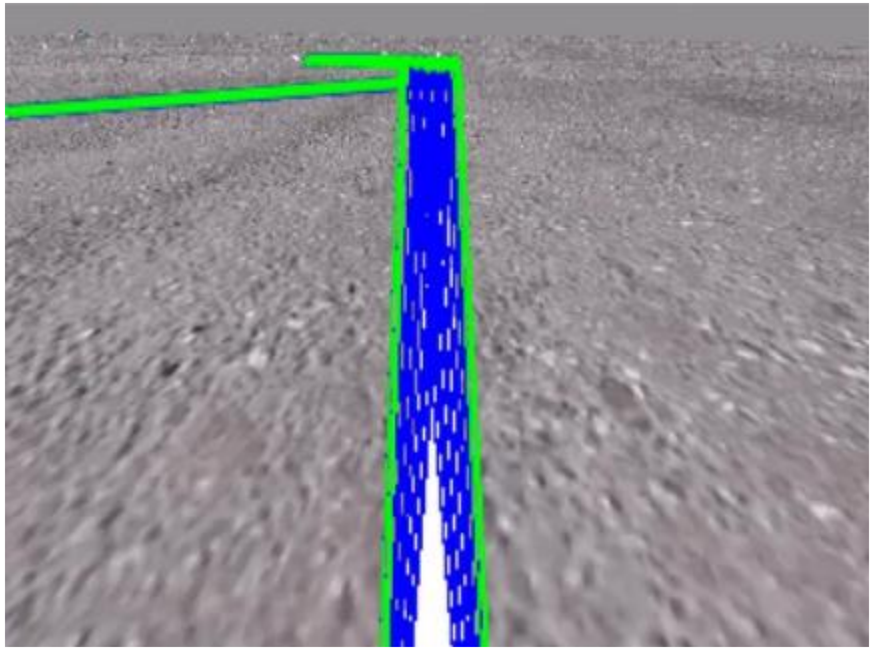
/interior rings) are

- There should be no cut lines, spikes or punctures.
- The interior rings should be located within the exterior ring. Interior rings may not be located within each other.

There should be no cut lines, spikes or punctures.

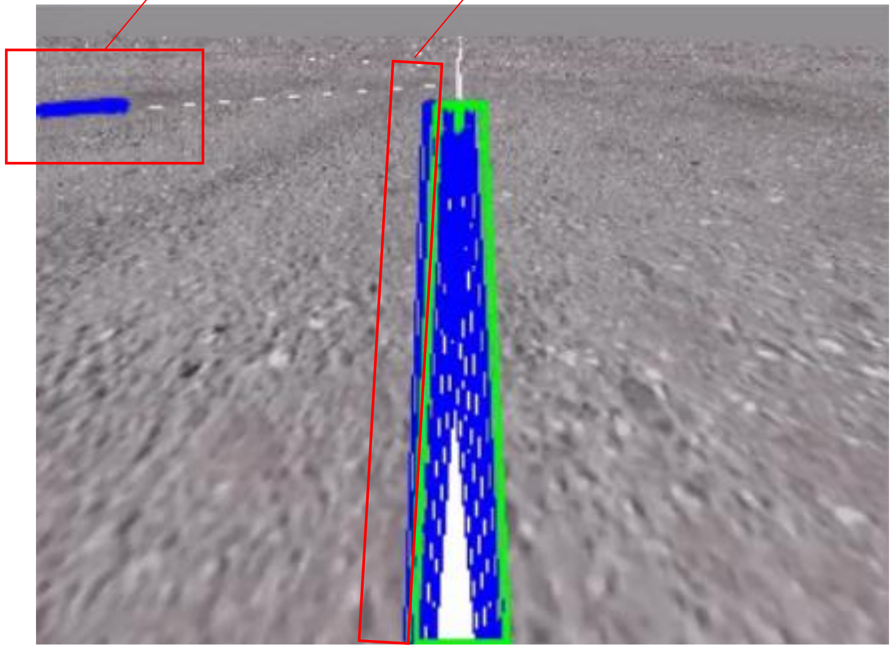
Image-based position monitoring: Numerical issues

Strange, strange…



CGAL

Speed = 1

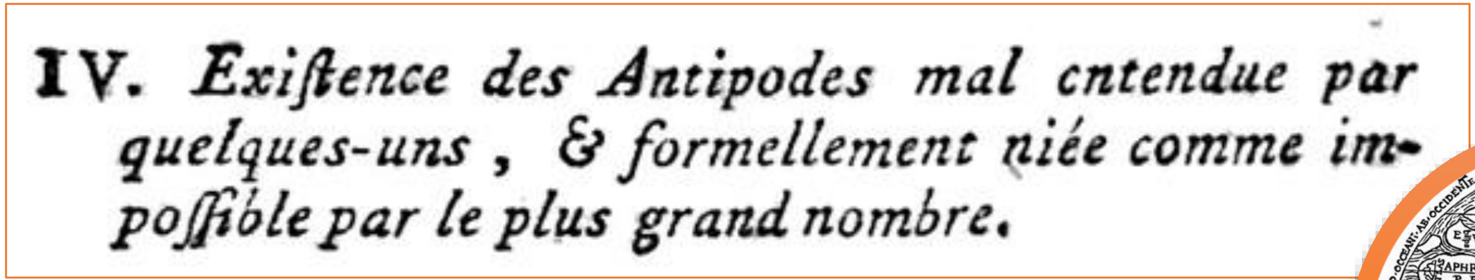


BOOST

Speed = 50

Formal verification of configuration data

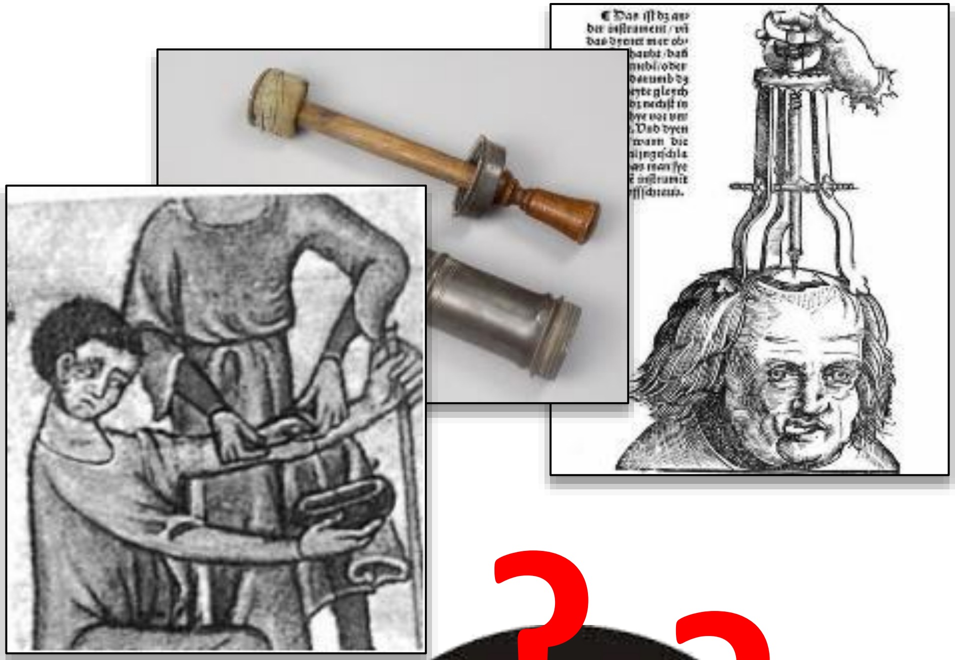
- Found a place where $0.1 + 0.1 \neq 0.2$… and where it <u>actually</u> matters…



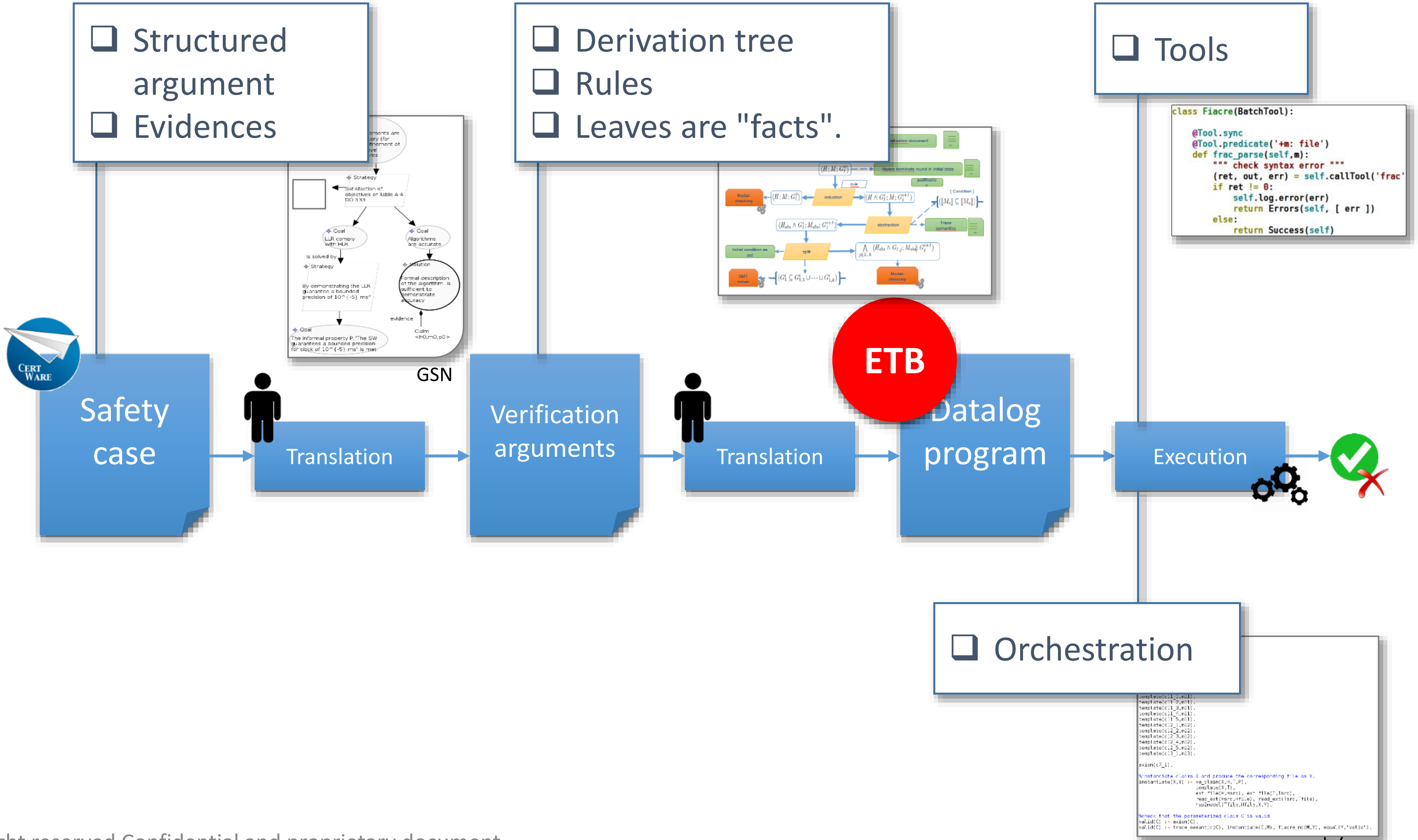
IV. *Exiſtence des Antipodes mal cntendue par quelques-uns, & formellement niée comme impoſſible par le plus grand nombre.*

- Find the best representation for hardware implementation

To conclude: dealing with (so) many methods and tools…



To conclude: dealing with (so) many methods and tools…



Merci de votre attention