TELECOM ParisTech

Une école de l'IMT

Model-Driven Engineering for Safety, Security and Performance: SysML-Sec

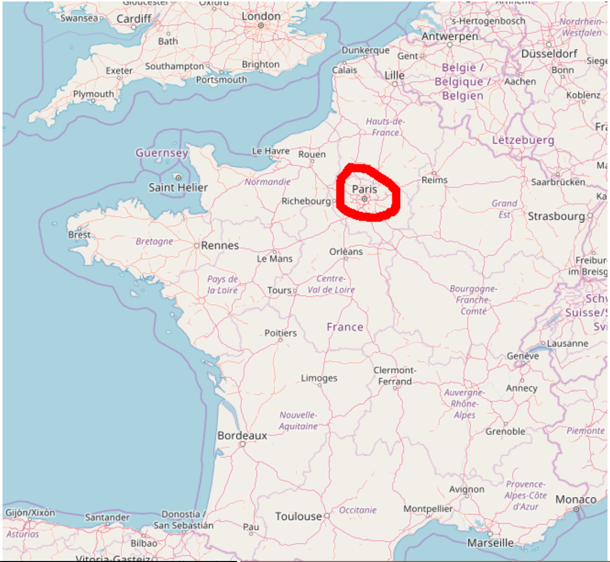
Ludovic APVRILLE
ludovic.apvrille@telecom-paristech.fr

université PARIS-SACLAY

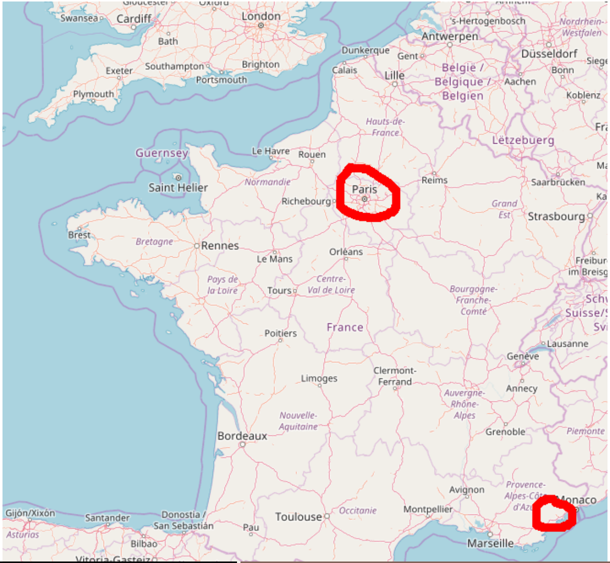
Seminar - City University

Telecom ParisTech

Telecom ParisTech



Telecom ParisTech



Outline

Context: Security for Embedded Systems

- Embedded systems

SysML-Sec

- Method
- SysML-Sec

Case study

- Case Study

Demo

- Demo

Conclusion

- Conclusion, future work and references

Examples of Threats

Transport systems

- Use of exploits in Flight Management System (FMS) to control ADS-B/ACARS [Teso 2013]
- Remote control of a car through Wifi [Miller 2015] [Tecent 2017]

Wired
TAKING ACTION FOR YOU
TAKING ACTION FOR YOU
HACKERS USE WIFI TO TAKE CONTROL OF VEHICLES
INTRUDERS CAN DISABLE BRAKES, STEER & KILL ENGINE
abc ACTION NEWS
11:04 84°

(C) Wired - ABC News

Medical appliances

- Infusion pump vulnerability, April 2015. http://www.scip.ch/en/?vuldb.75158

(C) Hospira

Examples of Threats (Cont.)

Internet of Things

- Proof of concept of attack on IZON camera [Stanislav 2013]

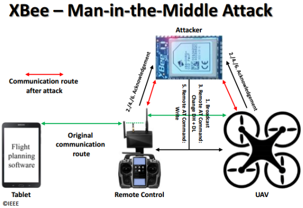
- Vulnerability on fitbit [Apvrille 2015]

Geek usages for your Fitbit Flex Tracker
Et si je cours à 4 pattes, je fais plus de pas ?
Fitness tracker
A. Apvrille
FortiGuard Labs
Fortinet
Hack.Lu
October 2015
FORTINET

A. Apvrille, Hack.lu'2015

(C) beforeitnews

- Hacking a professional drone [Rodday 2016]

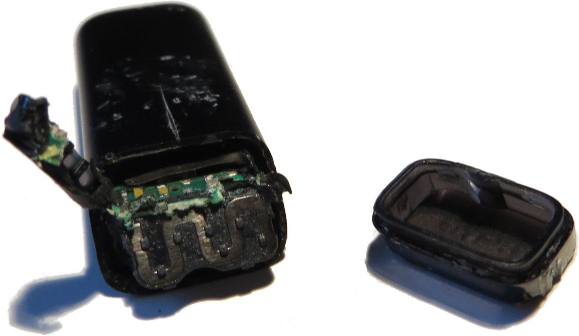

XBee – Man-in-the-Middle Attack
Attacker
Communication route after attack
2./4./6. Acknowledgement
2./4./6. Acknowledgement
5. Remote AT Command: Write
3. Remote AT Command: Change DH + DL
1. Broadcast
Original communication route
Flight planning software
Tablet
Remote Control
UAV
©IEEE

N. Rodday, BlackHat Asia'2016

Finding Vulnerabilities on IoTs

What's inside? Let's look together!

Inside a Fitbit

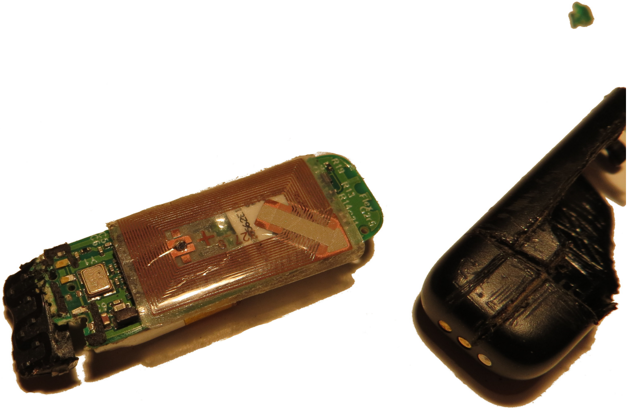


Don’t try this at home!

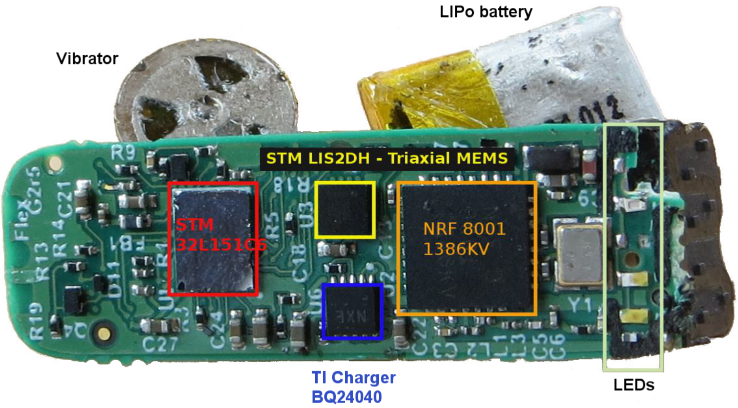
Inside a Fitbit (Cont.)

Again: don’t try this at home!

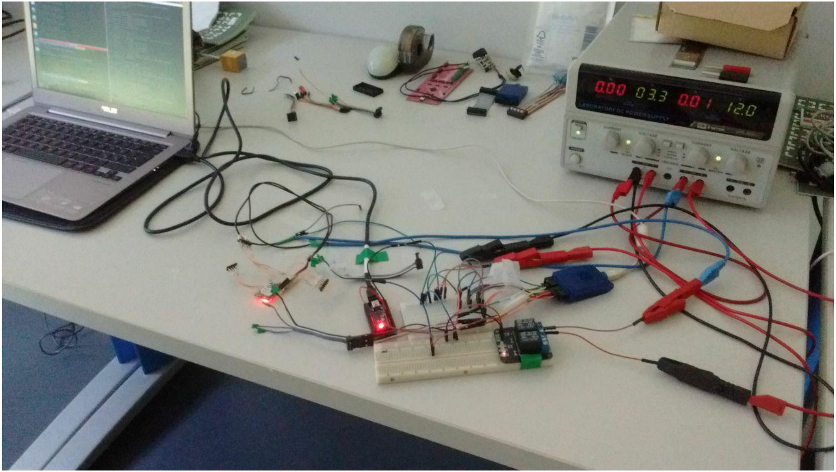
Inside a Fitbit (Cont.)



Fitbit: Hardware Components



Firmware Dumping



Then, How to Identify Vulnerabilities?

Investigations

- Testing ports (JTAG interface, UART, . . .)
- Firmware analysis
- Memory dump
- Side-channel analysis (e.g. power consumption, electromagnetic waves)
- Fault injection
- . . .

Secure your systems!

Develop your system with security in mind from the very beginning

Our solution: SysML-Sec, supported by TTool

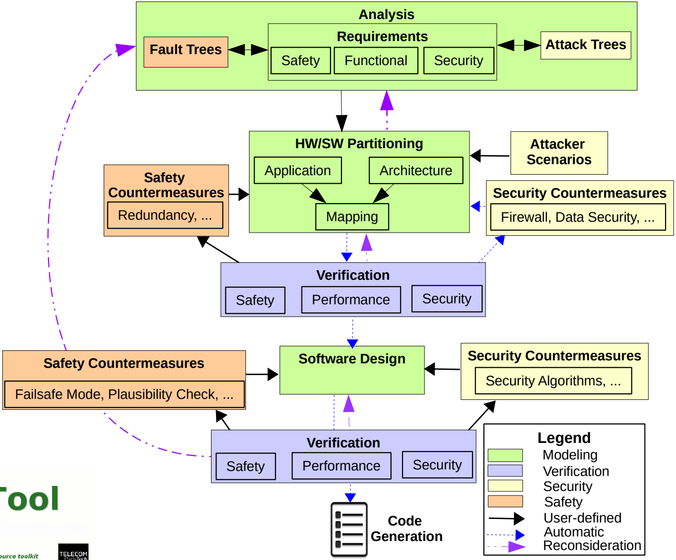
Designing Safe and Secure Embedded Systems: SysML-Sec

Main idea

- **Holistic approach**: bring together experts in embedded systems, system architects, system designers and security experts (with SysML)

Common issues (addressed by SysML-Sec):

- Adverse effects of security over safety/real-time/performance properties
  - Commonly: only the design of security mechanisms
- Hardware/Software partitioning
  - Commonly: no support for this in tools/approaches in MDE and security approaches



Analysis

Fault Trees

Requirements: Safety, Functional, Security

Attack Trees

HW/SW Partitioning: Application, Architecture, Mapping

Attacker Scenarios

Safety Countermeasures: Redundancy, ...

Security Countermeasures: Firewall, Data Security, ...

Verification: Safety, Performance, Security

Software Design

Safety Countermeasures: Failsafe Mode, Plausibility Check, ...

Security Countermeasures: Security Algorithms, ...

Verification: Safety, Performance, Security

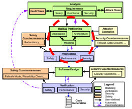
Code Generation

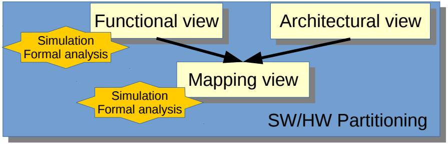
Legend

🟩	Modeling
🟦	Verification
🟨	Security
🟧	Safety
→	User-defined
⇢	Automatic
⇢	Reconsideration

Fully supported by TTool

Partitioning





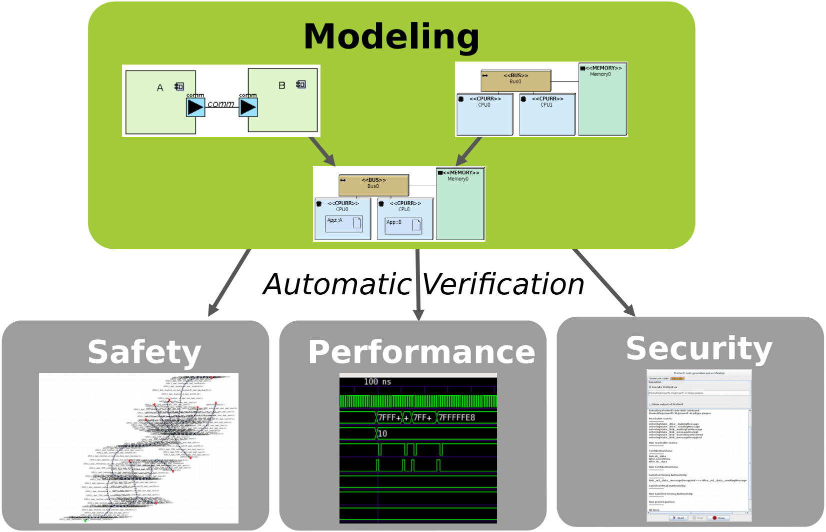
Before mapping

- Security mechanisms can be captured but not verified

After mapping

- Verify security (confidentiality, authenticity) according to attacker capabilities
  - Whether different HW elements are or not on the same die
  - Where are stored the cryptographic materials (keys)
  - Where are performed encrypt/decrypt operations
- Impact of security mechanisms on performance and safety
  - e.g. increased latency when inserting security mechanisms

Partitioning Verification



● Execute ProVerif as

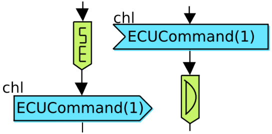
/home/lle/proverif1.91/proverif -in pitype pvspec

□ Show output of ProVerif

```
Executing ProVerif code with command:
/home/lle/proverif1.91/proverif -in pitype pvspec

Reachable states:
--------------
enteringState__Alice__makingMessage
enteringState__Alice__sendingMessage
enteringState__Bob__waitingForMessage
enteringState__Bob__messageDecrypt
enteringState__Bob__SecretDataReceived
enteringState__Bob__messageDecrypted

Non reachable states:
--------------

Confidential Data:
--------------
Bob.sk__data
Alice.secretData
Alice.sk__data

Non Confidential Data:
--------------

Satisfied Strong Authenticity:
--------------
Bob__m2__data__messageDecrypted ---> Alice__m1__data__sendingMessage

Satisfied Weak Authenticity:
--------------

Non Satisfied Strong Authenticity:
--------------

Non proved queries:
--------------

All done
```

Safety and Security Mechanisms

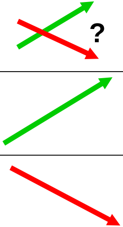
Data Encryption/ Authentication



Safety

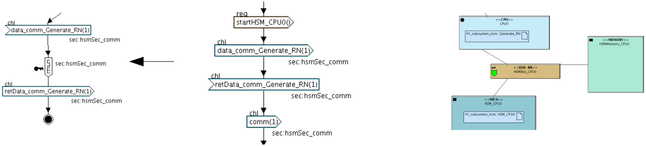
Security

Performance



Safety and Security Mechanisms (Cont.)

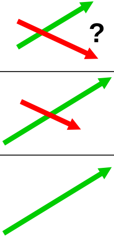
Data Security with Hardware Security Module



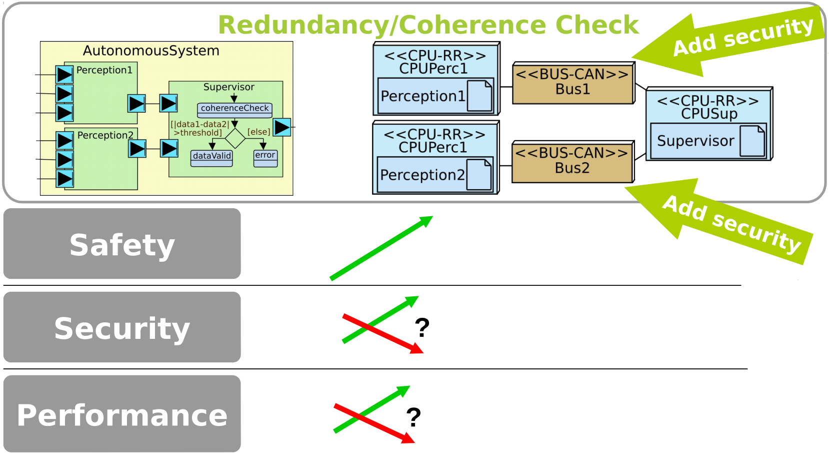
Safety

Security

Performance



Safety and Security Mechanisms (Cont.)

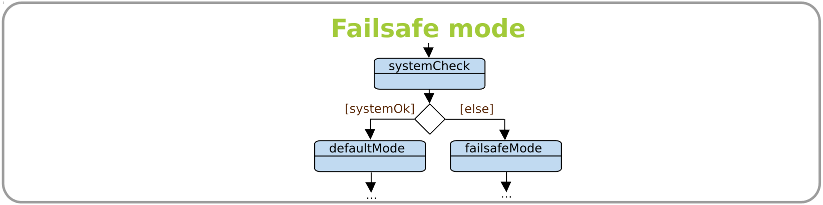

Redundancy/Coherence Check

Safety

Security ?

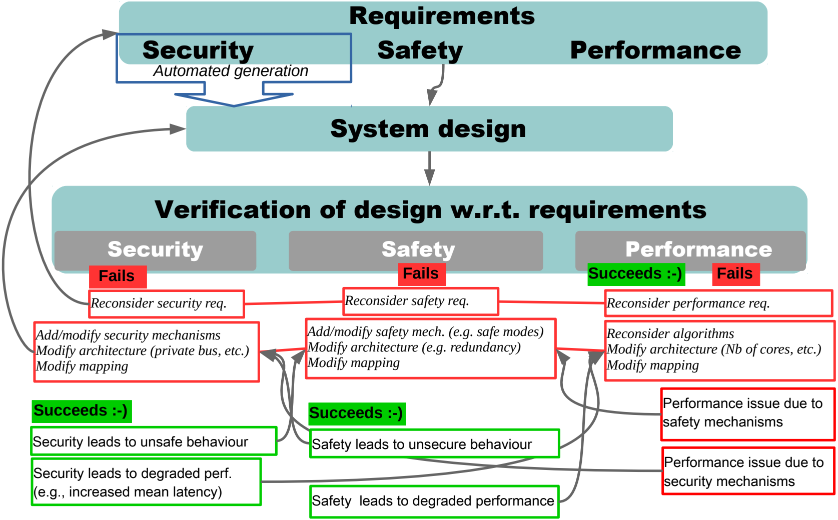
Performance ?

Safety and Security Mechanisms



Safety	↗
Security	↗ ↘ ?
Performance	↗ ↘ ?

Safety/Security/Performance



Requirements

Security

Automated generation

Safety

Performance

System design

Verification of design w.r.t. requirements

Security

Fails

Reconsider security req.

Add/modify security mechanisms
Modify architecture (private bus, etc.)
Modify mapping

Succeeds :-)

Security leads to unsafe behaviour

Security leads to degraded perf. (e.g., increased mean latency)

Safety

Fails

Reconsider safety req.

Add/modify safety mech. (e.g. safe modes)
Modify architecture (e.g. redundancy)
Modify mapping

Succeeds :-)

Safety leads to unsecure behaviour

Safety leads to degraded performance

Performance

Succeeds :-)

Fails

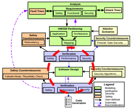
Reconsider performance req.

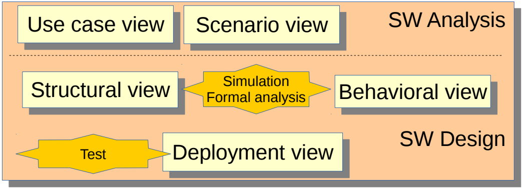
Reconsider algorithms
Modify architecture (Nb of cores, etc.)
Modify mapping

Performance issue due to safety mechanisms

Performance issue due to security mechanisms

SysML-Sec: SW Design





- Precise model of security mechanisms (security protocols)
- Proof of security properties : confidentiality, authenticity
- Channels between software blocks can be defined as private or public
  - This should be defined according to the hardware support defined during the partitioning phase

Case Studies

Cyber security of connected vehicles

- Safety/Security/Performance
- EVITA FP7 Partners: Continental, BMW, Bosch, . . .
- VEDECOM

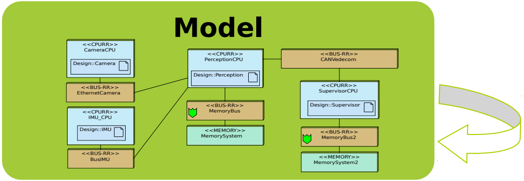
H2020 AQUAS

- Automated train sub-systems (ClearSy): Safety/Security/Performance
- Industrial Drives (Siemens): Safety/Security/Performance

Nokia

- Digital architectures for 5G networks (Safety/Performance)

Case Study: VEDECOM Autonomous Vehicle



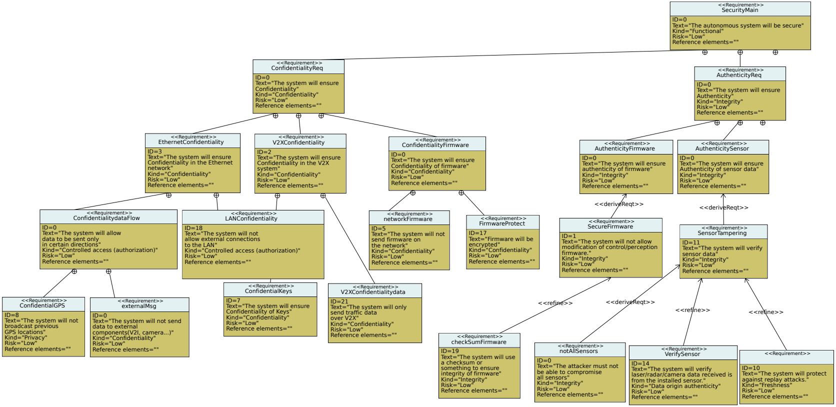
Verification

Tests

Constraints

- Standard: ISO26262
  - SOTIF: Safety Of The Intended Function
- Security: impact of potential attacks on safety

Requirements



<<Requirement>>
SecurityMain
ID=0
Text="The autonomous system will be secure"
Kind="Functional"
Risk="Low"
Reference elements=""

<<Requirement>>
ConfidentialityReq
ID=0
Text="The system will ensure Confidentiality"
Kind="Confidentiality"
Risk="Low"
Reference elements=""

<<Requirement>>
AuthenticityReq
ID=0
Text="The system will ensure Authenticity"
Kind="Integrity"
Risk="Low"
Reference elements=""

<<Requirement>>
EthernetConfidentiality
ID=3
Text="The system will ensure Confidentiality in the Ethernet network"
Kind="Confidentiality"
Risk="Low"
Reference elements=""

<<Requirement>>
V2XConfidentiality
ID=2
Text="The system will ensure Confidentiality in the V2X system"
Kind="Confidentiality"
Risk="Low"
Reference elements=""

<<Requirement>>
ConfidentialityFirmware
ID=0
Text="The system will ensure Confidentiality of firmware"
Kind="Confidentiality"
Risk="Low"
Reference elements=""

<<Requirement>>
AuthenticityFirmware
ID=0
Text="The system will ensure authenticity of firmware"
Kind="Integrity"
Risk="Low"
Reference elements=""

<<Requirement>>
AuthenticitySensor
ID=0
Text="The system will ensure Authenticity of sensor data"
Kind="Integrity"
Risk="Low"
Reference elements=""

<<Requirement>>
ConfidentialitydataFlow
ID=0
Text="The system will allow data to be sent only in certain directions"
Kind="Controlled access (authorization)"
Risk="Low"
Reference elements=""

<<Requirement>>
LANConfidentiality
ID=18
Text="The system will not allow external connections to the LAN"
Kind="Controlled access (authorization)"
Risk="Low"
Reference elements=""

<<Requirement>>
networkFirmware
ID=5
Text="The system will not send firmware on the network"
Kind="Confidentiality"
Risk="Low"
Reference elements=""

<<Requirement>>
FirmwareProtect
ID=17
Text="Firmware will be encrypted"
Kind="Confidentiality"
Risk="Low"
Reference elements=""

<<Requirement>>
SecureFirmware
ID=1
Text="The system will not allow modification of control/perception firmware."
Kind="Integrity"
Risk="Low"
Reference elements=""

<<Requirement>>
SensorTampering
ID=11
Text="The system will verify sensor data"
Kind="Integrity"
Risk="Low"
Reference elements=""

<<Requirement>>
ConfidentialGPS
ID=8
Text="The system will not broadcast previous GPS locations"
Kind="Privacy"
Risk="Low"
Reference elements=""

<<Requirement>>
externalMsg
ID=0
Text="The system will not send data to external components(V2I, camera...)"
Kind="Confidentiality"
Risk="Low"
Reference elements=""

<<Requirement>>
ConfidentialKeys
ID=7
Text="The system will ensure Confidentiality of Keys"
Kind="Confidentiality"
Risk="Low"
Reference elements=""

<<Requirement>>
V2XConfidentialitydata
ID=21
Text="The system will only send traffic data over V2X"
Kind="Confidentiality"
Risk="Low"
Reference elements=""

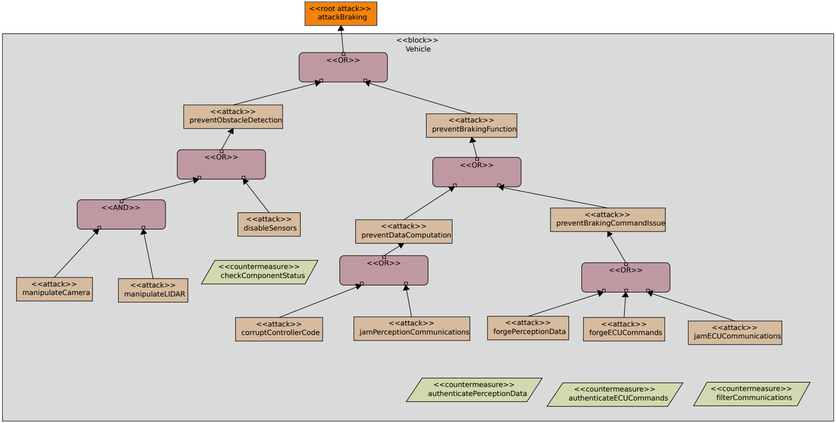
<<Requirement>>
checkSumFirmware
ID=19
Text="The system will use a checksum or something to ensure integrity of firmware"
Kind="Integrity"
Risk="Low"
Reference elements=""

<<Requirement>>
notAllSensors
ID=0
Text="The attacker must not be able to compromise all sensors"
Kind="Integrity"
Risk="Low"
Reference elements=""

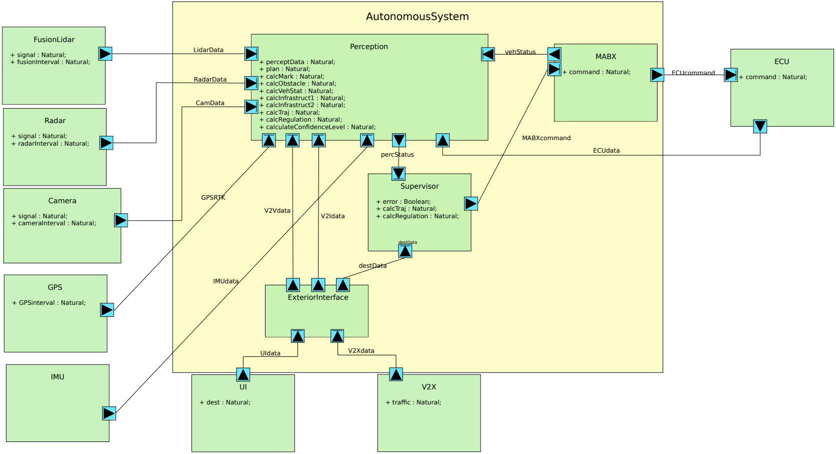
<<Requirement>>
VerifySensor
ID=14
Text="The system will verify laser/radar/camera data received is from the installed sensor."
Kind="Data origin authenticity"
Risk="Low"
Reference elements=""

<<Requirement>>
ID=10
Text="The system will protect against replay attacks."
Kind="Freshness"
Risk="Low"
Reference elements=""

Attacks


<<root attack>> attackBraking
<<block>> Vehicle
<<OR>>
<<attack>> preventObstacleDetection
<<attack>> preventBrakingFunction
<<OR>>
<<OR>>
<<AND>>
<<attack>> disableSensors
<<attack>> preventDataComputation
<<attack>> preventBrakingCommandIssue
<<attack>> manipulateCamera
<<attack>> manipulateLIDAR
<<countermeasure>> checkComponentStatus
<<OR>>
<<OR>>
<<attack>> corruptControllerCode
<<attack>> jamPerceptionCommunications
<<attack>> forgePerceptionData
<<attack>> forgeECUCommands
<<attack>> jamECUCommunications
<<countermeasure>> authenticatePerceptionData
<<countermeasure>> authenticateECUCommands
<<countermeasure>> filterCommunications

Functional View



AutonomousSystem

FusionLidar
+ signal : Natural;
+ fusionInterval : Natural;

Radar
+ signal : Natural;
+ radarInterval : Natural;

Camera
+ signal : Natural;
+ cameraInterval : Natural;

GPS
+ GPSinterval : Natural;

IMU

Perception
+ perceptData : Natural;
+ plan : Natural;
+ calcMark : Natural;
+ calcObstacle : Natural;
+ calcVehStat : Natural;
+ calcInfrastruct1 : Natural;
+ calcInfrastruct2 : Natural;
+ calcTraj : Natural;
+ calcRegulation : Natural;
+ calculateConfidenceLevel : Natural;

MABX
+ command : Natural;

ECU
+ command : Natural;

Supervisor
+ error : Boolean;
+ calcTraj : Natural;
+ calcRegulation : Natural;

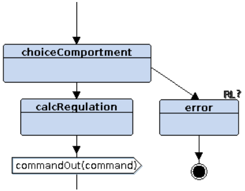
ExteriorInterface

UI
+ dest : Natural;

V2X
+ traffic : Natural;

Safety Verification (Before Mapping)

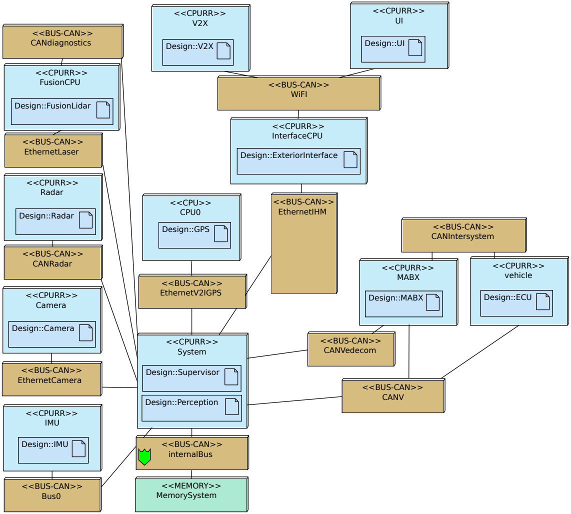
Reachability/Liveness



Queries

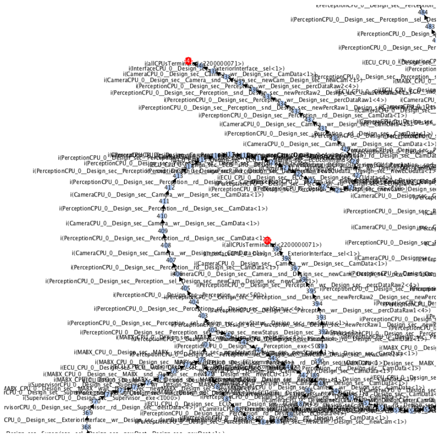
Safety Pragma
A[] Supervisor.running
Perception.distance<threshold -->
Supervisor.brakingOrder

Architecture and Mapping Views

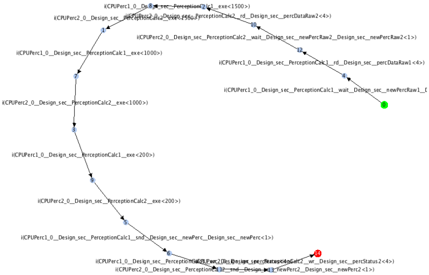


Safety Verification (After Mapping)

Reachability Graph

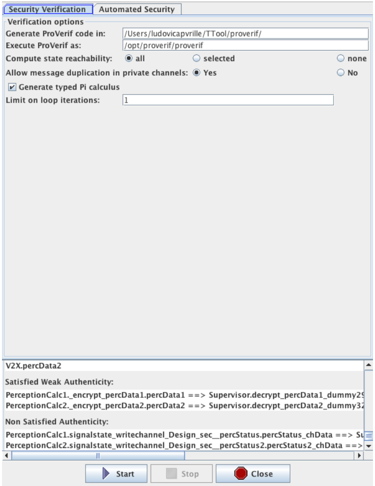

i(allCPUsTerminated<2200000071>)
i(InterfaceCPU_0__Design_sec__ExteriorInterface__sel<1>)
i(CameraCPU_0__Design_sec__Camera__wr__Design_sec__CamData<1>)
i(PerceptionCPU_0__Design_sec__Perception__rd__Design_sec__CamData<1>)
i(CameraCPU_0__Design_sec__Camera__snd__Design_sec__newCam__Design_sec__newCam<1>)
i(PerceptionCPU_0__Design_sec__Perception__wr__Design_sec__percDataRaw2<4>)
i(PerceptionCPU_0__Design_sec__Perception__wr__Design_sec__percDataRaw1<4>)
i(SupervisorCPU_0__Design_sec__Supervisor__exe<1000>)

Minimized RG


0
4
12
10
2
8
1
7
3
9
5
6
11
13
14
i(CPUPerc1_0__Design_sec__PerceptionCalc1__wait__Design_sec__newPercRaw1_[
i(CPUPerc1_0__Design_sec__PerceptionCalc1__rd__Design_sec__percDataRaw1<4>)
i(CPUPerc2_0__Design_sec__PerceptionCalc2__wait__Design_sec__newPercRaw2__Design_sec__newPercRaw2<1>)
i(CPUPerc2_0__Design_sec__PerceptionCalc2__rd__Design_sec__percDataRaw2<4>)
i(CPUPerc1_0__Design_sec__PerceptionCalc1__exe<1500>)
i(CPUPerc2_0__Design_sec__PerceptionCalc2__exe<1500>)
i(CPUPerc1_0__Design_sec__PerceptionCalc1__exe<1000>)
i(CPUPerc2_0__Design_sec__PerceptionCalc2__exe<1000>)
i(CPUPerc1_0__Design_sec__PerceptionCalc1__exe<200>)
i(CPUPerc2_0__Design_sec__PerceptionCalc2__exe<200>)
i(CPUPerc1_0__Design_sec__PerceptionCalc1__snd__Design_sec__newPerc__Design_sec__newPerc<1>)
i(CPUPerc2_0__Design_sec__PerceptionCalc2__snd__Design_sec__newPerc2__Design_sec__newPerc2<1>)
i(CPUPerc2_0__Design_sec__PerceptionCalc2__wr__Design_sec__percStatus2<4>)

Security Verification

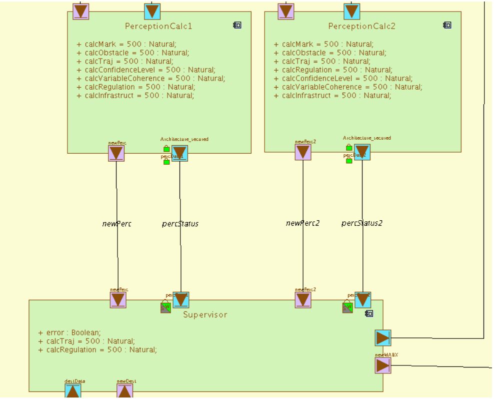
Dialog window



Verification options
Generate ProVerif code in: /Users/ludovicapvrille/TTool/proverif/
Execute ProVerif as: /opt/proverif/proverif
Compute state reachability: ● all ○ selected ○ none
Allow message duplication in private channels: ● Yes ○ No
☑ Generate typed Pi calculus
Limit on loop iterations: 1

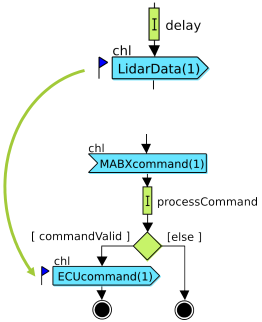
```
V2X.percData2

Satisfied Weak Authenticity:
PerceptionCalc1._encrypt_percData1.percData1 ==> Supervisor.decrypt_percData1_dummy29
PerceptionCalc2._encrypt_percData2.percData2 ==> Supervisor.decrypt_percData2_dummy32

Non Satisfied Authenticity:
PerceptionCalc1.signalstate_writechannel_Design_sec__percStatus.percStatus_chData ==> Su
PerceptionCalc2.signalstate_writechannel_Design_sec__percStatus2.percStatus2_chData ==>
```

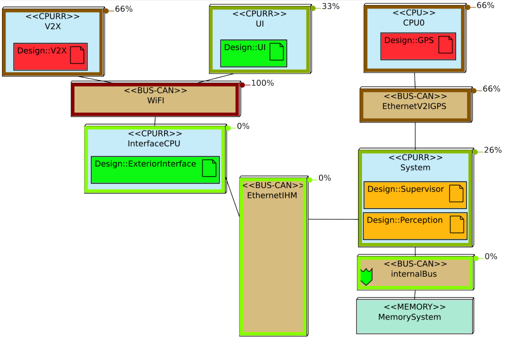
Backtracing


PerceptionCalc1
+ calcMark = 500 : Natural;
+ calcObstacle = 500 : Natural;
+ calcTraj = 500 : Natural;
+ calcConfidenceLevel = 500 : Natural;
+ calcVariableCoherence = 500 : Natural;
+ calcRegulation = 500 : Natural;
+ calcInfrastruct = 500 : Natural;
PerceptionCalc2
+ calcMark = 500 : Natural;
+ calcObstacle = 500 : Natural;
+ calcTraj = 500 : Natural;
+ calcRegulation = 500 : Natural;
+ calcConfidenceLevel = 500 : Natural;
+ calcVariableCoherence = 500 : Natural;
+ calcInfrastruct = 500 : Natural;
newPerc
Architecture_secured
percData1
newPerc2
percData2
newPerc
percStatus
newPerc2
percStatus2
Supervisor
+ error : Boolean;
+ calcTraj = 500 : Natural;
+ calcRegulation = 500 : Natural;
newMABX
destData
newDest

Performance Verification

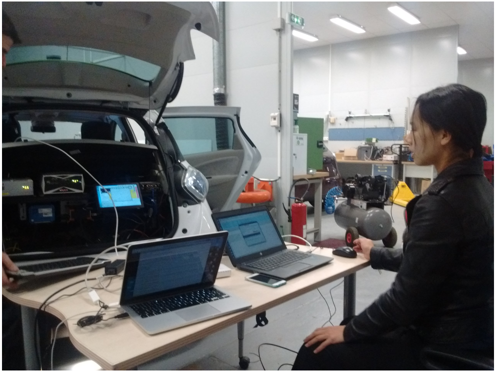
Latency	Bus/CPU Load





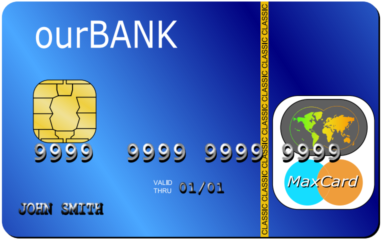
SW Design, Code generation, Test

- First SW model from mapping models
- SW model refinement
- SW model verification (safety, security)
- Code generation
  - (Virtual) Prototyping, test



Demo: SmartCard

- Main functions of the system
- Safety of the system (before mapping, after mapping)
- Performance
- Model enhanced with Security
- Impact on performance



Conclusion and Future Work

Achievements: SysML-Sec

- Methodology for designing safe and secure embedded systems
- Fully supported by TTool
- Applied to different domains, e.g., automotive systems, IoTs, malware

Future work

- Security risk assistance and backtracing
- Assistance to handle conflicts between security/safety/performance
  - Design space exploration

To Go Further ...

Web sites

- https://sysml-sec.telecom-paristech.fr
- https://ttool.telecom-paristech.fr

References

- Ludovic Apvrille, Yves Roudier, "SysML-Sec: A SysML Environment for the Design and Development of Secure Embedded Systems", Proceedings of the INCOSE/APCOSEC 2013 Conference on system engineering, Yokohama, Japan, September 8-11, 2013.
- Ludovic Apvrille, Yves Roudier, "Designing Safe and Secure Embedded and Cyber-Physical Systems with SysML-Sec", Chapter in Model-Driven Engineering and Software Development, p293–308, Springer International Publishing, 2015