

Méthodes formelles : De l'ENS Paris-Saclay à Thales Research & Technology

10 ans de l'Institut Farman



ENS Cachan

- 6 mois de stage, 3 ans de thèse
 - Laboratoire Spécification et Vérification
 - 5 Projets Farman

école _____
normale _____
supérieure _____
paris-saclay _____

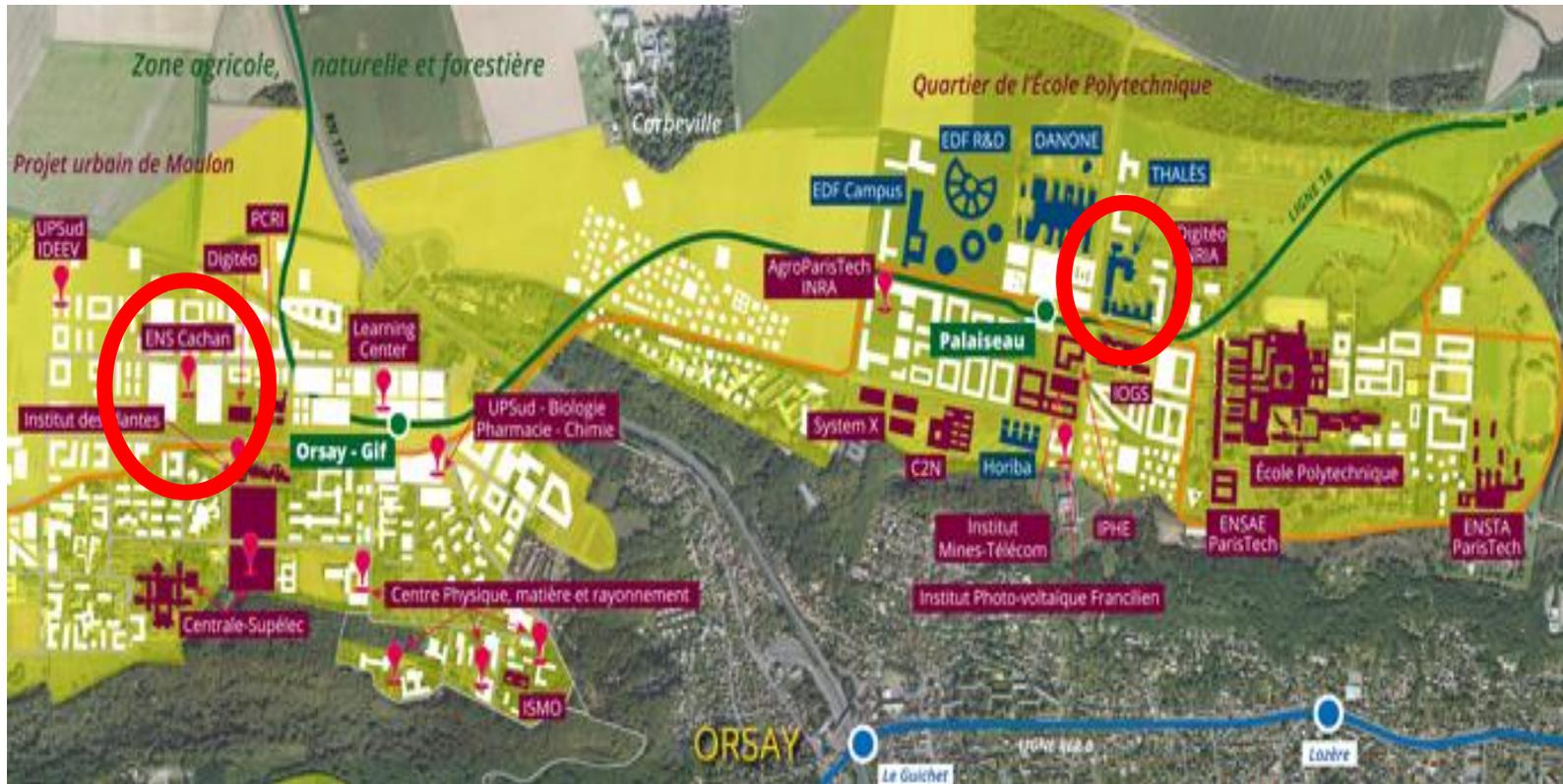
Thales Research & Technology

- 3 ans et 6 mois
 - Equipe Méthodes Formelles
 - Laboratoire Système Embarqués Critiques
 - n projets de recherche interne ($n > 0$)

THALES

TRT

- Centre de recherche « Corporate » du groupe
- Sur le plateau de Saclay



L'intelligence collective pour un monde plus sûr

Partout où des décisions critiques doivent être prises, Thales est présent. Sur les marchés que le Groupe sert (aéronautique, espace, transport terrestre, sécurité, défense), **ses équipements et systèmes aident ses clients à choisir la meilleure option et à agir en conséquence.**

L'expertise de ses **64 000 collaborateurs** et sa présence opérationnelle dans **56 pays** en font ainsi **un acteur clé de la sécurité des citoyens, des infrastructures et des États.**



Salariés

64 000



Une présence mondiale

56 pays



R & D Autofinancée* 2016

731 millions d'euros

* N'inclut donc pas la R & D réalisée sur financements externes.

Une structure équilibrée du chiffre d'affaires

Défense
50 %

Civil
50 %

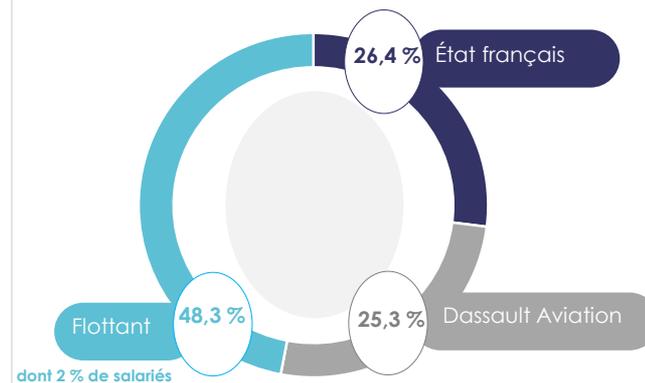


Chiffre d'affaires 2016

15 milliards d'euros

Répartition du capital

(au 31 décembre 2014)



MARCHÉS DUAUX Militaires & Civils



AÉRONAUTIQUE



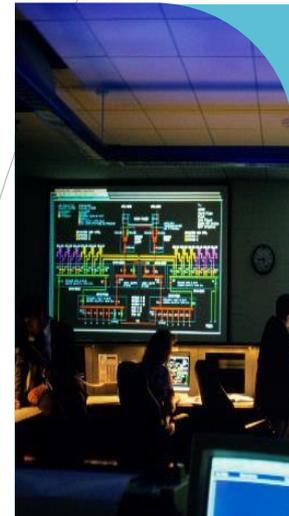
ESPACE



**TRANSPORT
TERRESTRE**



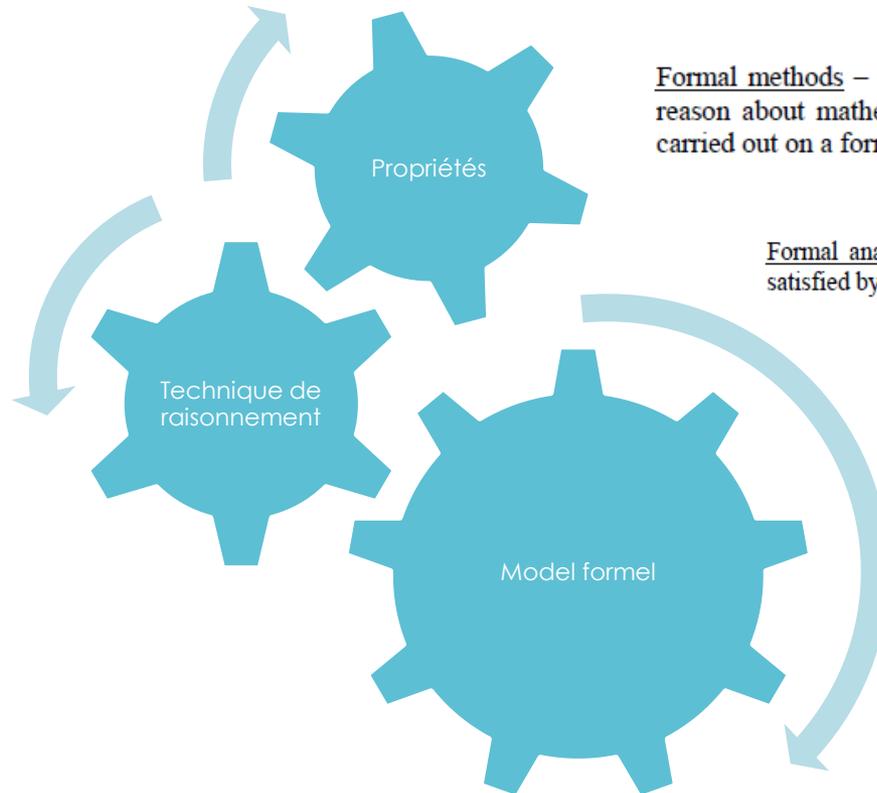
DÉFENSE



SÉCURITÉ

■ Définition de la DO-333

➤ Norme avionique pour le logiciel



Formal methods – Descriptive notations and analytical methods used to construct, develop and reason about mathematical models of system behavior. A formal method is a formal analysis carried out on a formal model.

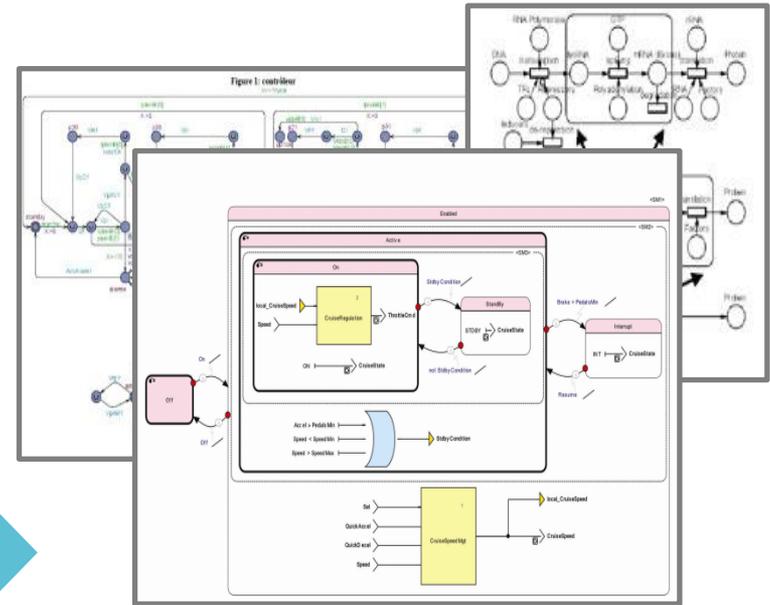
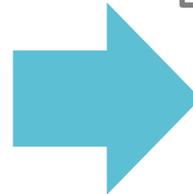
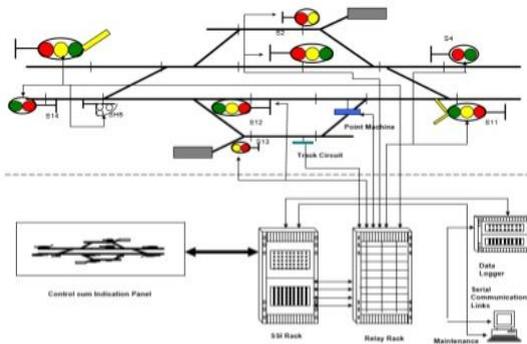
Formal analysis – The use of mathematical reasoning to guarantee that properties are always satisfied by a formal model.

Méthodes formelles

Ce document ne peut être reproduit, modifié, adapté, publié, en tout ou partie, ni divulgué à un tiers sans l'accord préalable et écrit de Thales - ©Thales 2017 Tous Droits réservés.



RAILWAY INTERLOCKING



$$\forall i, j \in \text{Trains} \ \&\& \ \text{NumTrain}(i).\text{Cross} \ \&\& \ \text{Train}(j).\text{Cross} \Rightarrow i == j$$



OPEN

Recherche « appliquée »

- Entre TRT et une ou plusieurs entités Thales
- Projet d'un an pour montrer l'applicabilité technique et/ou industrielle
- Dossier à soumettre en fin d'année

- A peu près un projet Farman

Recherche « amont »

- Projets de recherche : ANR, H2020, FUI, etc.
- Vision long terme

Thales & l'écosystème « formel »



OPEN

Ce document ne peut être reproduit, modifié, adapté, publié, traduit, d'une quelconque façon, en tout ou partie, ni divulgué à un tiers sans l'accord préalable et écrit de Thales. - @Thales 2017 Tous Droits réservés.

Est-ce que l'on s'en sert vraiment ?

Utilisation dans Thales

- Analyse de code C/Ada
 - Communications
 - Sécurité
 - Entre autres
- Model checking
 - Ferroviaire
 - Entre autres
- Theorem proving
 - Ferroviaire
 - Entre autres
- Entre autres utilisations également
 - CESTI Thales

Success stories hors Thales

- Intel : Theorem Proving
- Microsoft : Model checking
- Siemens/Alstom : B (Theorem proving)
- Airbus : Analyse de code, analyse de WCET

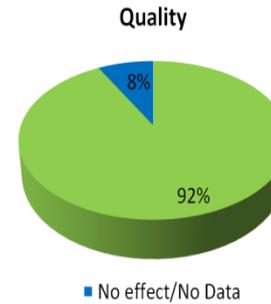
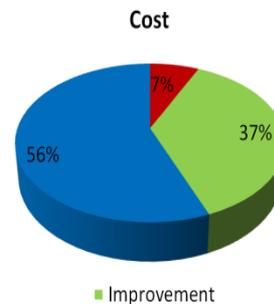
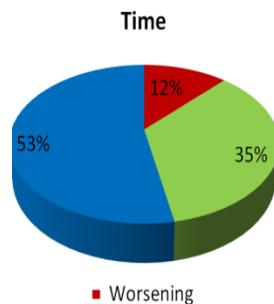
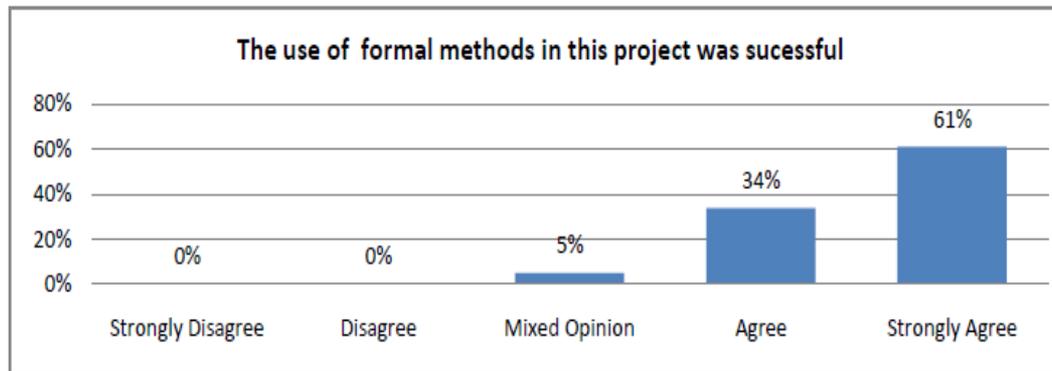
Autres utilisations

- Amazon, Apple, Dassault, Facebook, Gemalto, RockwellCollins, SNCF, RATP, etc.

➤ Retours de 62 projets industriels

- Du monde entier
- Dans toutes les branches de l'industrie des systèmes critiques

➤ Résultats



Quels sont les principaux obstacles ?

Multitude d'outils/de techniques

➤ Quoi essayer pour un novice ?

Name	Model-Creation			Fidelity-checking		SIL		Availability			
	main resources / lessons	learning ongoing	reporting ongoing	support / open source	cost / energy / granular	sil / support / granular	cost / energy / granular	learning curve	reporting ongoing / on a	runtime / sil	
ARUP	Aggressive Passivists	Passive models	PCT, PLS		Yes	Yes	No	No	PLSC	C	Linux & others
ARUP	Plain	CoFlow	Yastula, CT*		Yes	Yes	No	No	PLSC	ANSI C	Linux & others
BEAD&UP	Code analysis	Java	CT, LT		Yes	Yes	Yes	Yes	PLSC	Java	Windows and Linux related
BLIST	Code analysis	C	Veritas automate		Yes	No	No	No	PLSC	CC++	Windows and Linux related
CADENCE EDA/UP	Plain	Codeless SIM, VHDL, Verilog	CT, LT		Yes	Yes	No	No	PLSC	-	Windows and Linux related
CDSP	Passivists	LOTUS, REP, LOTUS NT	ARIC	22, 118, 22, 22, 278, 1178, 22, 22, 22	Yes	Yes	Yes	Yes	PLSC	-	Mac OS, Linux, Solaris, Windows
CDSP	Code analysis	C, C++ Java	Coastline		Yes	Yes	No	No	PLSC	C++	Windows and Linux related
CDLanet/UP	Code analysis	C	Veritas automate		Yes	Yes	No	Yes	PLSC	Java	Any
CDLanet/UP	Plain and Test	COE, CEP, LOTUS, TOCB	ARIC, CT, COE	22, 118, 22, 118	Yes	Yes	No	No	PLSC	SIL	Windows and Linux related, Mac OS
CDKase	Test	Lab, C, C++ Java, VHDL, Verilog	LT, LT*		No	Yes	Yes	Yes	Veritas Commercial Use only	-	Windows and Linux related
CDLanet/UP	Plain	CD++ (via LLVM), COE, COE language, Test automate	LT, Coastline, Veritas verify		Yes	Yes	No	Yes	PLSC	C++	Linux and related
CDLanet/UP	Realtime	C++, Test automate	Veritas automate		Yes	No	No	No	PLSC	C++	Windows and Linux related
CDLanet/UP	Digital system analysis	C, C++	Coastline		Yes	Yes	No	No	PLSC	C++	Windows and Linux related
CDLanet/UP	Model checking	SIM, Verilog	SIL		Yes	Yes	No	No	PLSC	C++	Windows and Linux related
CDLanet/UP	Plain	COE, TOCB, COE	Yastula	22, 118, 22, 118, 22, 118	Yes	No	No	No	PLSC	SIL	Windows and Linux related
Emulated/Veritas/UP	Hybrid	Emulated/Veritas/UP/COE	Veritas automate		Yes	Yes	Yes	Yes	Veritas Commercial Use only	-	Windows
Emulated/Veritas/UP	Code analysis	C, C++	Coastline		Yes	Yes	No	No	PLSC	C++	Windows and Linux related
Espresso/UP	Hybrid		ARIC, CT	22, 22	No	Yes	No	No	PLSC	C++/Verilog	Linux related
FAST/UP	Plain	PCT	-	22, 118, 22	Yes	No	Yes	Yes	PLSC	-	Linux related
CDLanet/UP	Plain	-	ARIC, CT, Yastula		Yes	Yes	Yes	Yes	PLSC	Java	Windows and Linux related
Infra/UP	Plain	Model	Coastline		Yes	No	No	No	PLSC	Model	Linux, Windows, Mac OS
Java Path/UP	Plain and Test	Java	Veritas		No	Yes	No	No	NOEL	Java	Mac OS, Windows, Linux
LLIC/UP	Code analysis	C, C++ all languages supported by LLVM	Coastline		Yes	No	No	No	PLSC	C++	Windows and Linux related
LT&UP	Plain	REP	LT		Yes	Yes	No	Yes	PLSC	Java	Windows and Linux related
LT&UP	Plain, Realtime	Formals, JCR, hCRU, COE Input Language	Yastula, LT, CT*	22, 22	Yes	No	No	No	PLSC	C, C++	Linux, Mac OS X, Windows
MDL&UP	Plain, Realtime	EP, L	CT, CT*		Yes	Yes	No	Yes	PLSC	C++	Linux, Windows, Mac OS
hCRU	Plain, Realtime	hCRU	Yastula	22, 22, 118, 278, 1178	Yes	Yes	Yes	Yes	PLSC	C++	Mac OS, Linux, Solaris, Windows
ARIC	Realtime, Passivists	Plain MC	CEI, CEAL, PCT, PRCT	22	No	No	No	No	PLSC	C	Windows, Linux, Mac OS
NA&UP	Plain	SIM	CT, LT, PRL		Yes	No	No	No	PLSC	C	Linux, Windows, Mac OS X
simple OpenSP (C analysis)	software symbolic simulation with DPI kernel	CD++ programs with OpenSP simulator	high precision or flexible precision Inspec (PI)		Yes	Yes	No	Yes	PLSC	C, C++	Linux, Linux, Windows (not on available user)
PCT	Plain, Realtime, Passivists	CEP, Test CEP, Passivists CEP	LT, Coastline		Yes	Yes	Yes	Yes	PLSC	C++	Windows, other OS, with Java
PR&UP	Passivists	PR&UP, PR&UP language, Plain MC	CEI, PLS, PCT		No	Yes	No	No	PLSC	C++ Java	Windows, Linux, Mac OS
PR&UP	Plain	EP, Veritas, B+REP, Z, T&N, CEP	Coastline, LT, CT		Yes	Yes	No	No	PLSC	Verilog, C, Java, Verilog	Linux, Mac OS, Windows
PR&UP	Passivists	SystemC	PLS		No	Yes	No	No	PLSC	C++ Java	Windows, Linux, Mac OS
Realtime Test/UP	Hybrid	Emulated/Veritas/UP	-		No	Yes	Yes	Yes	Veritas Commercial Use only	SIL	Windows, Linux
R&UP	abstract, linear hybrid, fully symbolic	abstracting formal automate (TA), linear hybrid automate (LH)	TCT, with formal assumptions, CT, with formal assumptions	formal simulation, bit simulation	Yes	Yes	Yes	Yes	PLSC	C++	Linux, Linux
S&UP	symbolic bounded and infinite	S&UP	LT		Yes	No	No	No	PLSC	Verilog	Linux, Mac OS X, Windows (Eggh)
S&UP/UP	Code analysis	C, C++	Coastline		Yes	Yes	No	No	PLSC	C++	Windows and Linux related
S&UP/UP	Plain, bounded	CEI, CEI, L&M	LT, Coastline		Yes	No	No	No	PLSC	Verilog	Windows and Linux related
S&UP/UP	Plain	Yastula	COE		Yes	No	No	No	PLSC	CC++	Windows and Linux related
SPN	Plain	Formals	LT		Yes	Yes	No	Yes	PLSC	C, C++	Windows and Linux related
SPN	Plain	Plain test, COE Input Language	LT, PRL, subal		Yes	No	No	No	PLSC	C, C++	Linux & related
TOP&UP	Realtime	Test/UP Plain test, open Inspec, initializers, language	TCT, subal		No	Yes	Yes	Yes	PLSC	C++ Java	Mac OS, Windows, Linux
TOP&UP	Plain	CDSP	CT, Yastula	22, 118, 22, 278, 1178, 22, 118, 22	Yes	Yes	Yes	Yes	PLSC	Java	Windows, Mac OS and Linux related
UP&UP	Realtime	Test automate, C subal	TCT, subal		Yes	Yes	Yes	Yes	PLSC	C++ Java	Mac OS, Windows, Linux
RD&UP	Realtime	Test Plain test, abstract generate Plain test	TCT, subal		Yes	Yes	Yes	Yes	PLSC	C++ Verilog	Mac OS, Windows, Linux
T&UP	Plain	T&UP, P&UP	T&UP		Yes	Yes	Yes	No	PLSC	Java	Mac OS, Windows, Linux

Ce document ne peut être reproduit, modifié, adapté, publié, traduit, d'une quelconque façon, en tout ou partie, ni divulgué à un tiers sans l'accord préalable et écrit de Thales - ©Thales 2017 Tous Droits réservés.

Quels sont les principaux obstacles ?

■ Multitude d'outils/de techniques

- Quoi essayer pour un novice ?

■ Manque de méthodologie

- Comment se servir des outils ?
- Comment déboguer ?
 - Son modèle
 - Son environnement
 - Ses propriétés

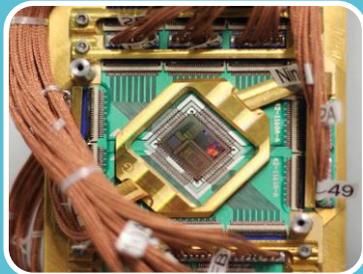
■ Difficulté d'évaluation du ROI

Poussée des normes

- Avionique: DO 333 (Suggestion)
- Ferroviaire: 50128 (Recommandation)
- Security: CC (Obligatoire pour EAL7)
- Automobile: 26262 (Recommandation)
- Défense: 00-55/00-56 (Recommandation)

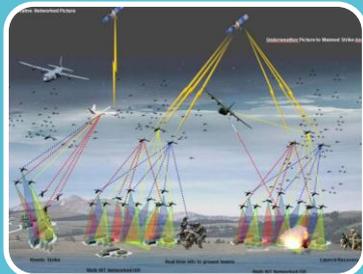
Poussée des clients

- Des produits plus sûrs
- Moins de bug/Moins de maintenance



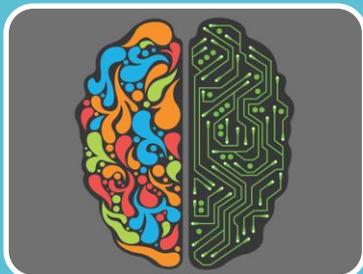
Calcul quantique

- Quels gains pour la vérification ?
- Comment certifier leur comportement ?



Essaim de drones

- Vérification des propriétés souhaitées ?
- Absence de propriétés non désirables ?



Intelligence artificielle

- Absence de bug ?
- Comment certifier son comportement ?