

3/11

## Some companies using formal methods

<ul> <li>Transportation</li> </ul>	
<ul> <li>line 14 of Paris subway</li> </ul>	(RATP)
<ul> <li>primary flight control software of A340 and A3</li> </ul>	380 (Airbus)
AILS (Airbone Information for Lateral Spacing	) (NASA)
• Army : military secured communication softwar	re (DGA)
<ul> <li>Security for Communications/Trading/Banks</li> </ul>	
<ul> <li>online payment protocols</li> </ul>	(Orange)
<ul> <li>cloud computing</li> </ul>	(Amazon Web Services)
Consumer software	
<ul> <li>static analysis of front-end and back-end code</li> </ul>	(Meta)
• processors	(Intel)
<ul> <li>Windows drivers, secured web protocols</li> </ul>	(Microsoft)
<ul> <li>Smartcards, Javacards</li> </ul>	(Gemalto, Fime)
• home networks, secured movie editing devices	(Technicolor)
T. Genet (ISTIC/IRISA) ACF	5 / 11

## How does ACF relates to other M1/L3 courses?

- ACO (M1)
  - ACF is only about programming in the small
  - ACF focuses on the validity of a solution/program
- LOG (L3)
  - Same core logical language as LOG, extended in ACF
  - ACF does not focus on proofs
  - Automation of many aspects of LOG
- ProgC (L3)
  - Functional programming instead of imperative (Why)
  - More complex programs in ACF
  - ... and more complex properties that you can prove!
  - Integration of verified code in Java project

## Objectives of ACF

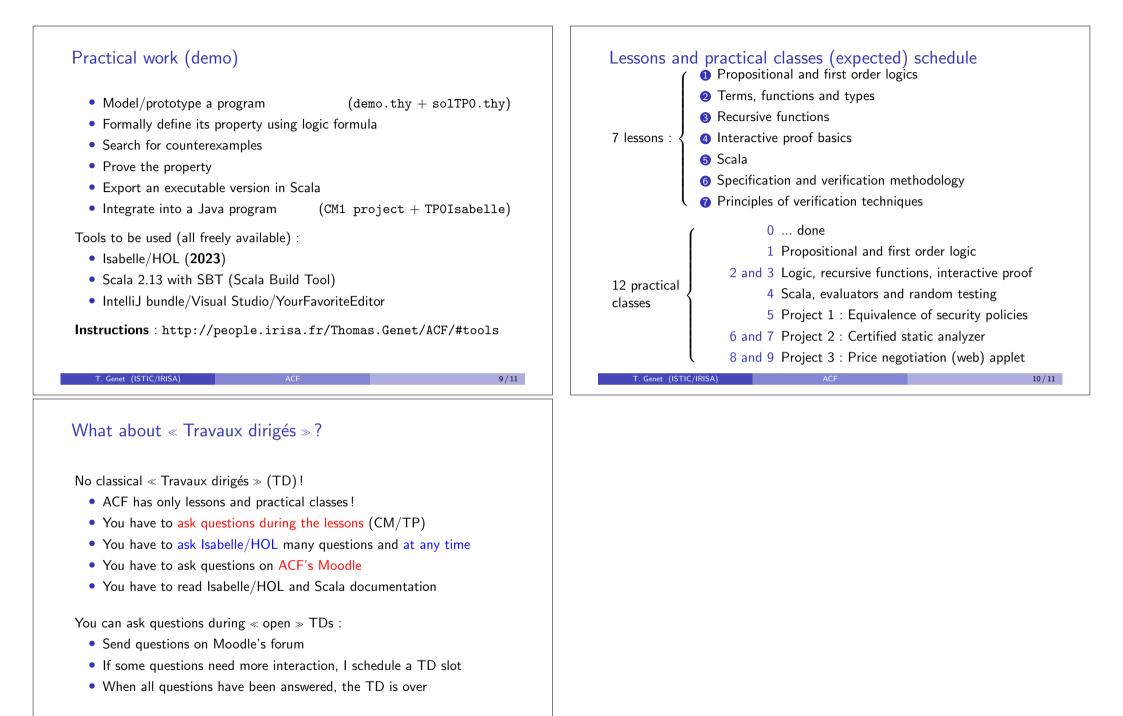
- (Re)-introduce functional programming
  - Good and fast prototyping/modeling language
  - Renewed programming paradigm (Ocaml, Haskell, F#, Scala)
  - Proofs are far easier on a functional program than on an imperative one (*e.g.* Why3 in ProgC)
- Use logic to formally define the properties of a software
  - « The most precise, concise and expressive programming language »
  - Proving one formula can replace infinitely many tests !
  - Testing one formula can replace thousands of tests !

## Evaluation

T. Genet (ISTIC/IRISA)

- Terminal exam (1/2 of the final mark)
- 3 projects (1/2 of the final mark)
  - Model/prototype a software using functional programming
  - Define the expected properties of the software using logic
  - Check that the software satisfies the properties
  - Export a Scala program corresponding to the model
  - Integrate it into a Java program

6/11



T. Genet (ISTIC/IRISA)

11 / 11