May 28, 2025

Cloud-based MTD

Hugo Forraz
Gilles Grimaud, David Nowak
2XS/SISE - CRIStAL/IRCICA - CNRS/Univ. Lille
Phd founded by the IPCEI-CIS

Université de Lille

1 Model the use case

2 Formal model

3 Problem encountered

Hugo Forraz
Gilles Grimaud, David Nowak
2XS/SISE - CRIStAL/IRCICA - CNRS/Univ. Lille
Phd founded by the IPCEI-CIS

Université de Lille

1 Model the use case

2 Formal model

3 Problem encountered

Hugo Forraz
Gilles Grimaud, David Nowak
2XS/SISE - CRIStAL/IRCICA - CNRS/Univ. Lille
Phd founded by the IPCEI-CIS

Université de Lille

Use Case : Provide a high level of guarantees on untrusted architectures

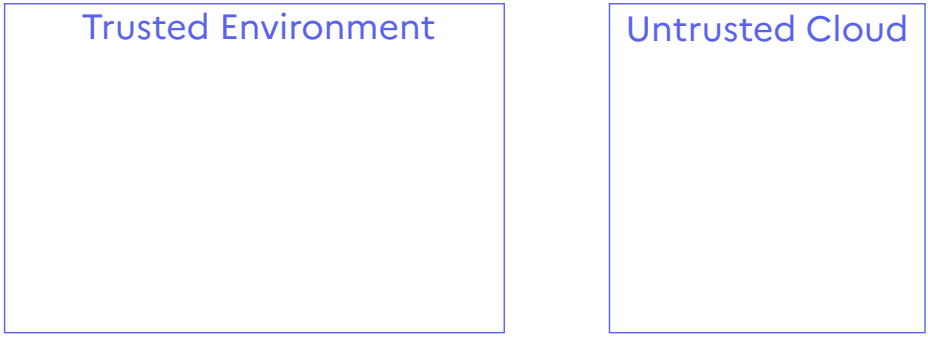
- Model a distributed architecture
- Idea : Machine Learning on lung radiographies to show if a tumour is present or not
- Cloud is a promising solution to mutualize resources
- Solutions like AWS or GCP are easy to use but can not respect the europeans laws to guarantee the privacy of the data
- How can we model a solution that will allow us to take the best in all of those worlds ?

Hugo Forraz
Gilles Grimaud, David Nowak
2XS/SISE - CRIStAL/IRCICA - CNRS/Univ. Lille
Phd founded by the IPCEI-CIS

Université de Lille

Schematic of the use-case

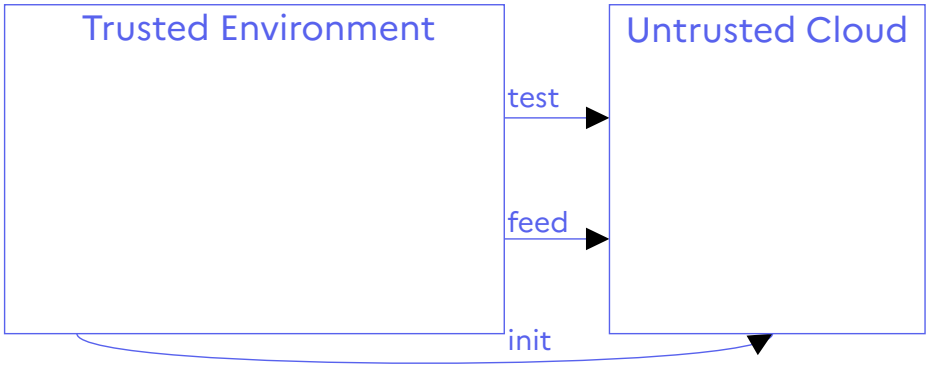
Top Level



Hugo Forraz
Gilles Grimaud, David Nowak
2XS/SISE - CRIStAL/IRCICA - CNRS/Univ. Lille
Phd founded by the IPCEI-CIS

Schematic of the use-case

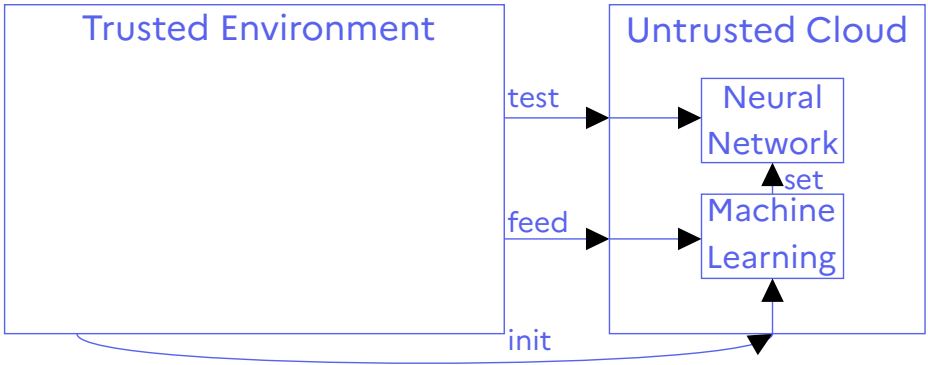
Interactions



Hugo Forraz
Gilles Grimaud, David Nowak
2XS/SISE - CRIStAL/IRCICA - CNRS/Univ. Lille
Phd founded by the IPCEI-CIS

Schematic of the use-case

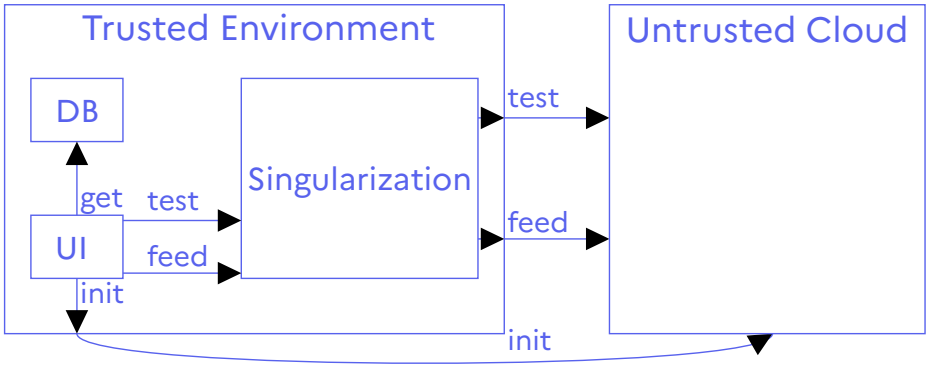
Test interaction



Hugo Forraz
Gilles Grimaud, David Nowak
2XS/SISE - CRIStAL/IRCICA - CNRS/Univ. Lille
Phd founded by the IPCEI-CIS

Schematic of the use-case

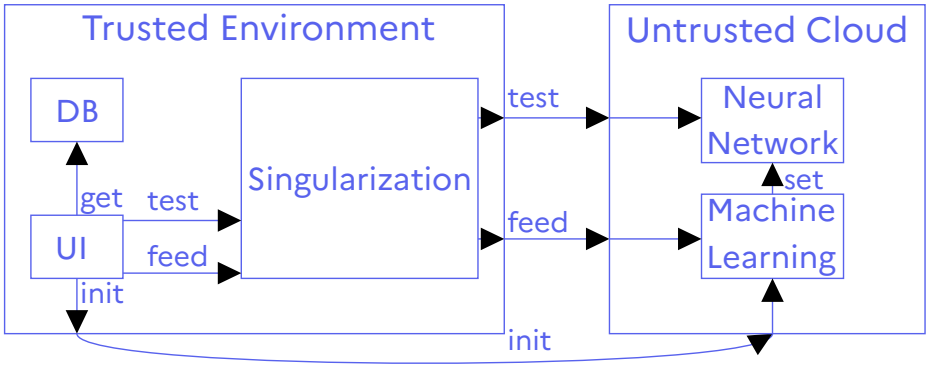
Looking inside the trusted environment



Hugo Forraz
Gilles Grimaud, David Nowak
2XS/SISE - CRIStAL/IRCICA - CNRS/Univ. Lille
Phd founded by the IPCEI-CIS

Schematic of the use-case

Bringing everything together



Hugo Forraz
Gilles Grimaud, David Nowak
2XS/SISE - CRIStAL/IRCICA - CNRS/Univ. Lille
Phd founded by the IPCEI-CIS

Université de Lille

Desired level of guarantees

Common Criteria

The Common Criteria (CC) is an international standard for evaluating information technology security products, providing guidelines and specifications to ensure these products meet recognized security standards, especially for government and other high-security environments.

© https://www.commoncriteriaportal.org/index.cfm

Hugo Forraz
Gilles Grimaud, David Nowak
2XS/SISE - CRIStAL/IRCICA - CNRS/Univ. Lille
Phd founded by the IPCEI-CIS

Université de Lille

Desired level of guarantees

Formal specifications : Evaluation Assurance Levels (EAL)

Level 5

You write a specification for the desired security properties

Level 6

You write a specification for the whole system

Level 7

Once you start to prove things on the spec

Hugo Forraz
Gilles Grimaud, David Nowak
2XS/SISE - CRIStAL/IRCICA - CNRS/Univ. Lille
Phd founded by the IPCEI-CIS

How can we model specifications ?

Why the Rocq Proover?

- Can extract proven code to OCaml
- Interactive
- Metaprogramming allows to write really complex tactics
- Keeps program verification close to the mathematical world
- Famous in the academic world

Hugo Forraz
Gilles Grimaud, David Nowak
2XS/SISE - CRIStAL/IRCICA - CNRS/Univ. Lille
Phd founded by the IPCEI-CIS

Université de Lille

How can we model specifications ?

Why FreeSpec ?

- Written with Rocq
- Automatized the tedious work
- Allows to write complex architectures through composition of components

Hugo Forraz
Gilles Grimaud, David Nowak
2XS/SISE - CRIStAL/IRCICA - CNRS/Univ. Lille
Phd founded by the IPCEI-CIS

Université de Lille

1 Model the use case

2 Formal model

3 Problem encountered

Hugo Forraz
Gilles Grimaud, David Nowak
2XS/SISE - CRIStAL/IRCICA - CNRS/Univ. Lille
Phd founded by the IPCEI-CIS

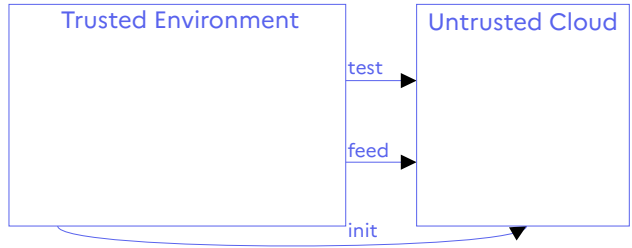
Université de Lille

How can we write the specification using Rocq ?

Top Level

```
Parameter Content : Type.

Inductive Data : Type :=
| private : Content → Data
| public : Content → Data.
Inductive CLOUD : interface :=
| feed : Data → CLOUD unit
| init : CLOUD unit
| test : Data → CLOUD bool.
Inductive TOP_LEVEL : interface :=
| ux_init : TOP_LEVEL unit
| ux_feed : TOP_LEVEL unit
| ux_test : TOP_LEVEL bool.
```



Hugo Forraz
Gilles Grimaud, David Nowak
2XS/SISE - CRIStAL/IRCICA - CNRS/Univ. Lille
Phd founded by the IPCEI-CIS

Université de Lille

How can we write the specification using Rocq ?

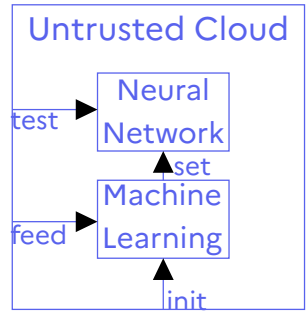
Cloud : Component

```
Parameter NeuralNetwork : Type.

Inductive ML : interface :=
| feedWith : Data → ML unit
| initNN : ML NeuralNetwork.

Inductive NN : interface :=
| set : NeuralNetwork → NN unit
| testWith : Data → NN bool.

Definition ml_feed `{Provide ix ML} (d : Data)
    : impure ix unit := request (feedWith d).
```



Hugo Forraz
Gilles Grimaud, David Nowak
2XS/SISE - CRIStAL/IRCICA - CNRS/Univ. Lille
Phd founded by the IPCEI-CIS

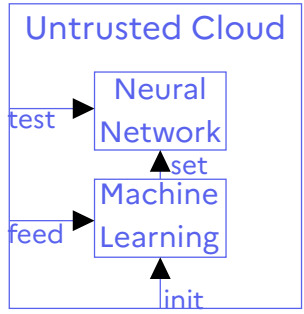
Université de Lille

How can we write the specification using Rocq ?

Cloud : Component

```
Definition ml_init `{Provide ix ML}
    : impure ix NeuralNetwork := request initNN.

Definition cloud `{Provide2 ix NN ML}
    : component CLOUD ix :=
fun _ op ⇒
    match op with
    | init ⇒ ml_init >>= (request set)
    | feed data ⇒ ml_feed
    | test data ⇒ request (testWith data)
    end.
```



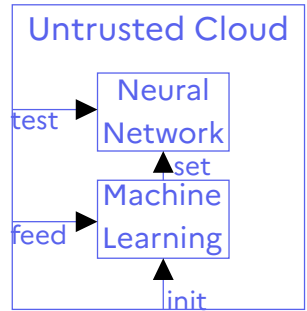
Hugo Forraz
Gilles Grimaud, David Nowak
2XS/SISE - CRIStAL/IRCICA - CNRS/Univ. Lille
Phd founded by the IPCEI-CIS

How can we write the specification using Rocq ?

Cloud : Contract : Witness State

```
Definition c_state := bool * bool.
Definition cs_is_fed : c_state → bool := fst.
Definition cs_is_ini : c_state → bool := snd.

Definition step
    (s: c_state) (a: Type) (e: CLOUD a) (x: a)
: c_state :=
match e with
| feed _ ⇒ (true, cs_is_ini s)
| init ⇒ (cs_is_fed s, true)
| test _ ⇒ s
end.
```



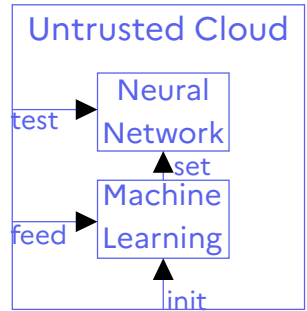
Hugo Forraz
Gilles Grimaud, David Nowak
2XS/SISE - CRIStAL/IRCICA - CNRS/Univ. Lille
Phd founded by the IPCEI-CIS

Université de Lille

How can we write the specification using Rocq ?

Cloud : Contract : Requirements

```
Inductive o_caller
    : c_state → ∀ (a: Type), CLOUD a → Prop :=
| req_feed (data: Data) (s: c_state)
    (H: ∀ c: Content, data = public c)
    : o_caller s unit (feed data)
| req_init (s: c_state) (H: cs_is_fed s = true)
    : o_caller s unit init
| req_test (data: Data) (s: c_state)
    (H: cs_is_ini s = true ∧ (∀ c: Content), data = public c)
    : o_caller s bool (test data).
```




```
Definition cloud_contract : contract CLOUD c_state :=
    make_contract step o_caller no_o_callee.
```


Hugo Forraz
Gilles Grimaud, David Nowak
2XS/SISE - CRIStAL/IRCICA - CNRS/Univ. Lille
Phd founded by the IPCEI-CIS

Université de Lille

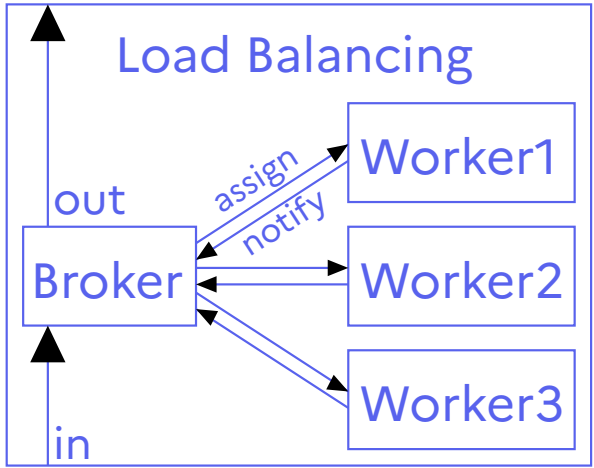
1 Model the use case

2 Formal model

3 Problem encountered

Hugo Forraz
Gilles Grimaud, David Nowak
2XS/SISE - CRIStAL/IRCICA - CNRS/Univ. Lille
Phd founded by the IPCEI-CIS

Schematic that we can’t model using FreeSpec



Hugo Forraz
Gilles Grimaud, David Nowak
2XS/SISE - CRIStAL/IRCICA - CNRS/Univ. Lille
Phd founded by the IPCEI-CIS

Interaction Trees

Hugo Forraz
Gilles Grimaud, David Nowak
2XS/SISE - CRIStAL/IRCICA - CNRS/Univ. Lille
Phd founded by the IPCEI-CIS

Interaction Trees

Bisimulation

Relation of equivalence between two processes. The processess P and Q are bisimilar if we cannot distinguish them while they are running.

Hugo Forraz
Gilles Grimaud, David Nowak
2XS/SISE - CRIStAL/IRCICA - CNRS/Univ. Lille
Phd founded by the IPCEI-CIS

Interaction Trees

Simple infinite loop

```
CoInductive itree (E: Type → Type) (R: Type): Type :=
| Ret (r: R)
| Vis {A: Type} (e : E A) (k : A → itree E R).

Inductive IO : Type → Type :=
| Input : IO nat
| Output : nat → IO unit.

CoFixpoint echo : itree IO void :=
    Vis Input (fun x ⇒ Vis (Output x) (fun _ ⇒ echo)).
```

Li-yao Xia, Yannick Zakowski, Paul He, Chung-Kil Hur, Gregory Malecha, Benjamin C. Pierce and Steve Zdancewic. 2020. Interaction Trees: Representing Recursive and Impure Programs in Coq. Proc. ACM Program. Lang. 4, POPL, Article 51 (January 2020), 32 pages

Hugo Forraz
Gilles Grimaud, David Nowak
2XS/SISE - CRIStAL/IRCICA - CNRS/Univ. Lille
Phd founded by the IPCEI-CIS

Conclusions & Perspectives

Realisation

We managed to make the specification for the whole system using the Rocq Proover.

Reproducibilty

We reproduced the issues pointed out by Letan *et al.* (authors of FreeSpec).

Future works

We want to rework Interaction Trees in a way that would be closer to FreeSpec.

orraz
owak
2XS/SISE - CRIStAL/IRCICA - CNRS/Univ. Lille
Phd founded by the IPCEI-CIS

Université de Lille

Thanks for your attention

Hugo Forraz
Gilles Grimaud, David Nowak
2XS/SISE - CRIStAL/IRCICA - CNRS/Univ. Lille
Phd founded by the IPCEI-CIS