FROM RESEARCH TO INDUSTRY

cea tech

list

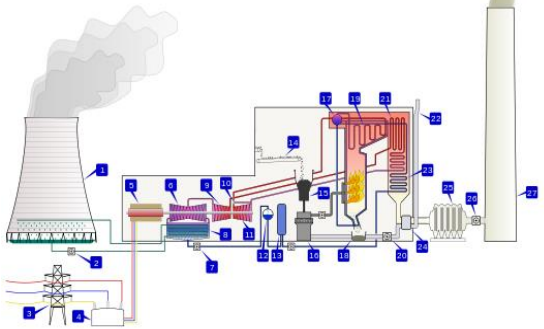
COMBINING FORMAL TOOLS

SOURCE CODE VERIFICATION OF FUNCTIONAL TEMPORAL PROPERTIES

L. CORRENSON, P. BAUDIN

F. KIRCHNER ★

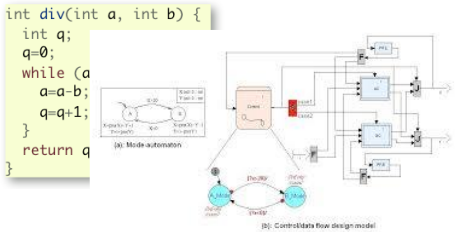
CONTEXT | CRITICAL COMMAND AND CONTROL SOFTWARE



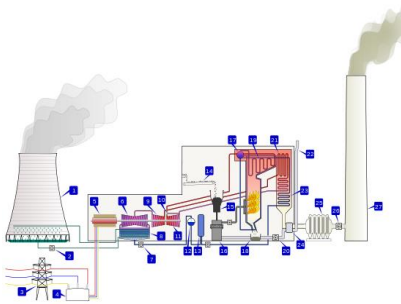
CONTEXT | DEVELOPMENT CYCLE

Design

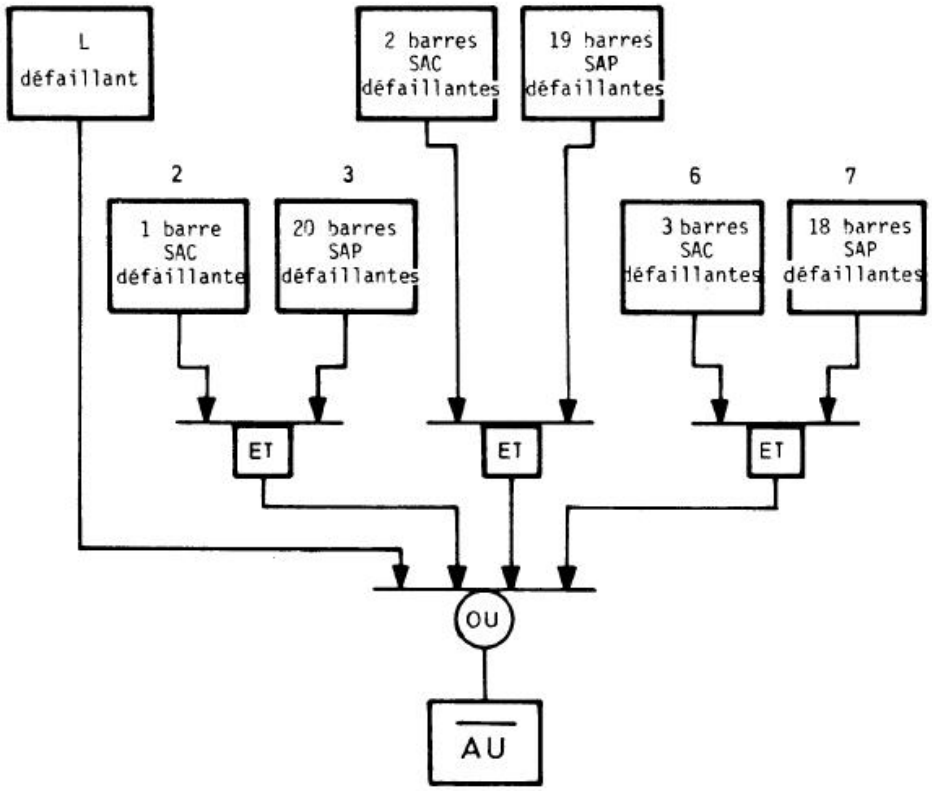
Validation



Implementation

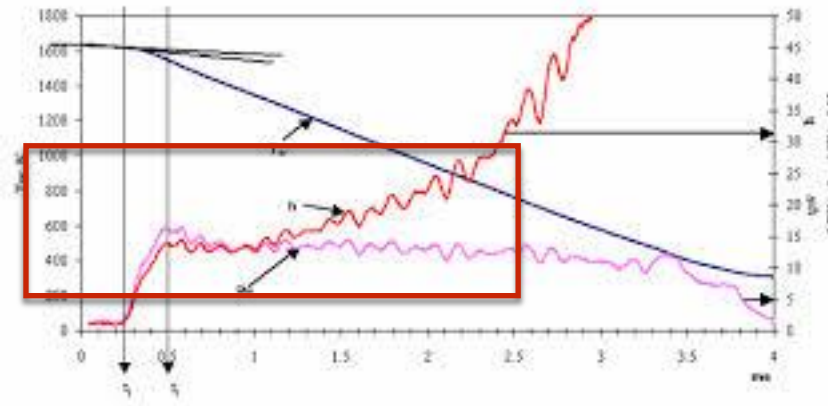


DESIGN | EXPECTED PROPERTIES



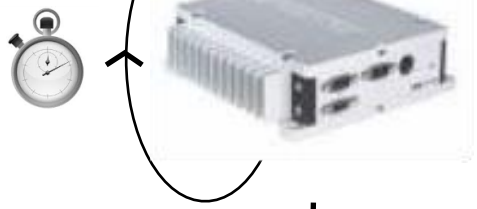
DESIGN | EXPECTED PROPERTIES

ASSUME



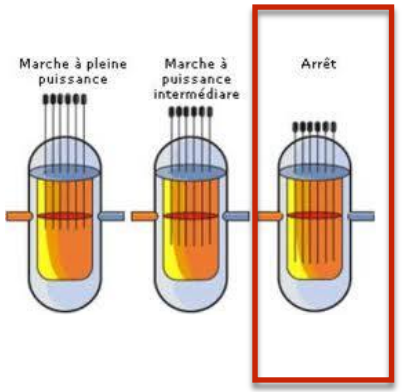
Sensors

Computations

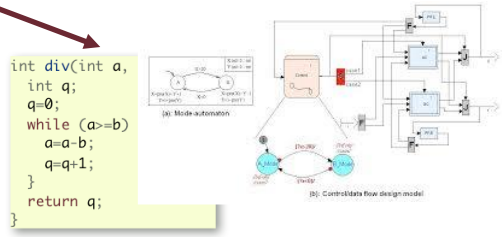


Actuators

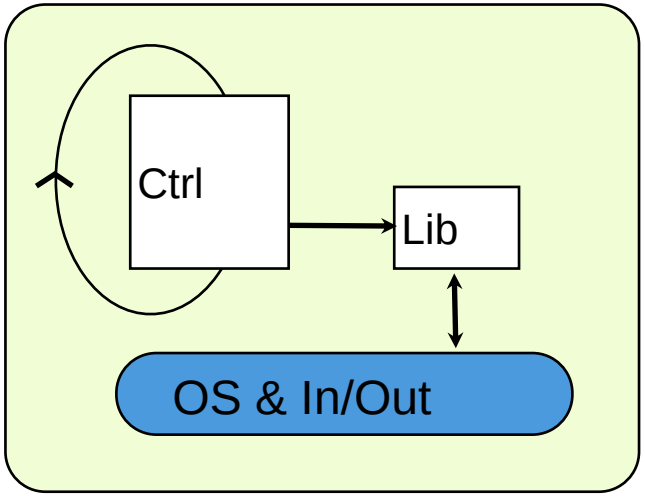
ASSERT



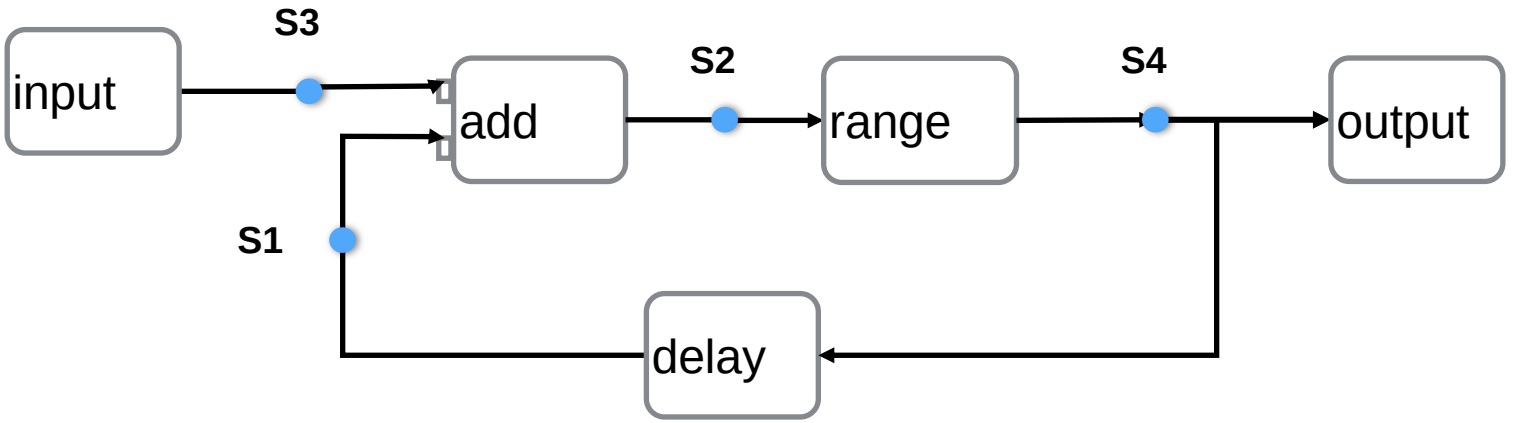
DESIGN | WHAT KIND OF CODE?



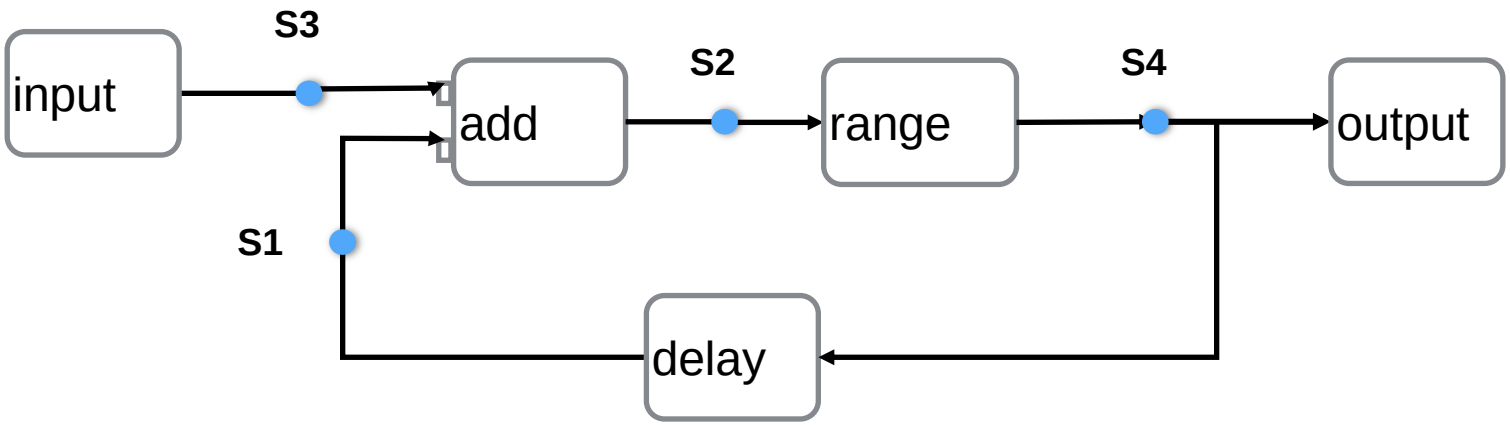
```
int div(int a,
  int q;
  q=0;
  while (a>=b)
    a=a-b;
    q=q+1;
  }
  return q;
}
```



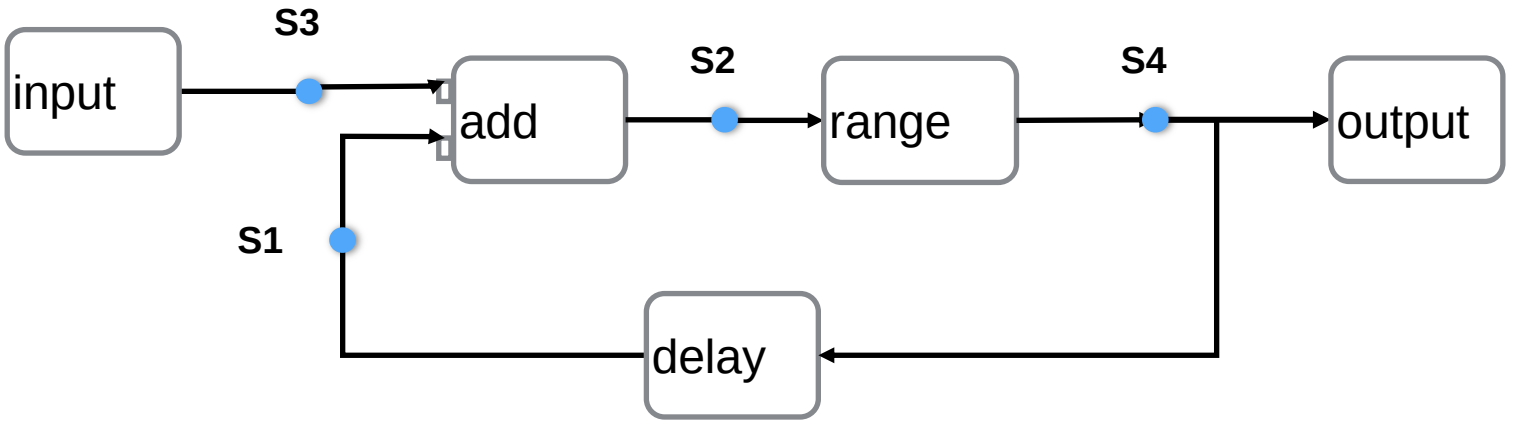
DESIGN | DIAGRAMS



DESIGN | DIAGRAMS: AN ENGINEER'S CODE…



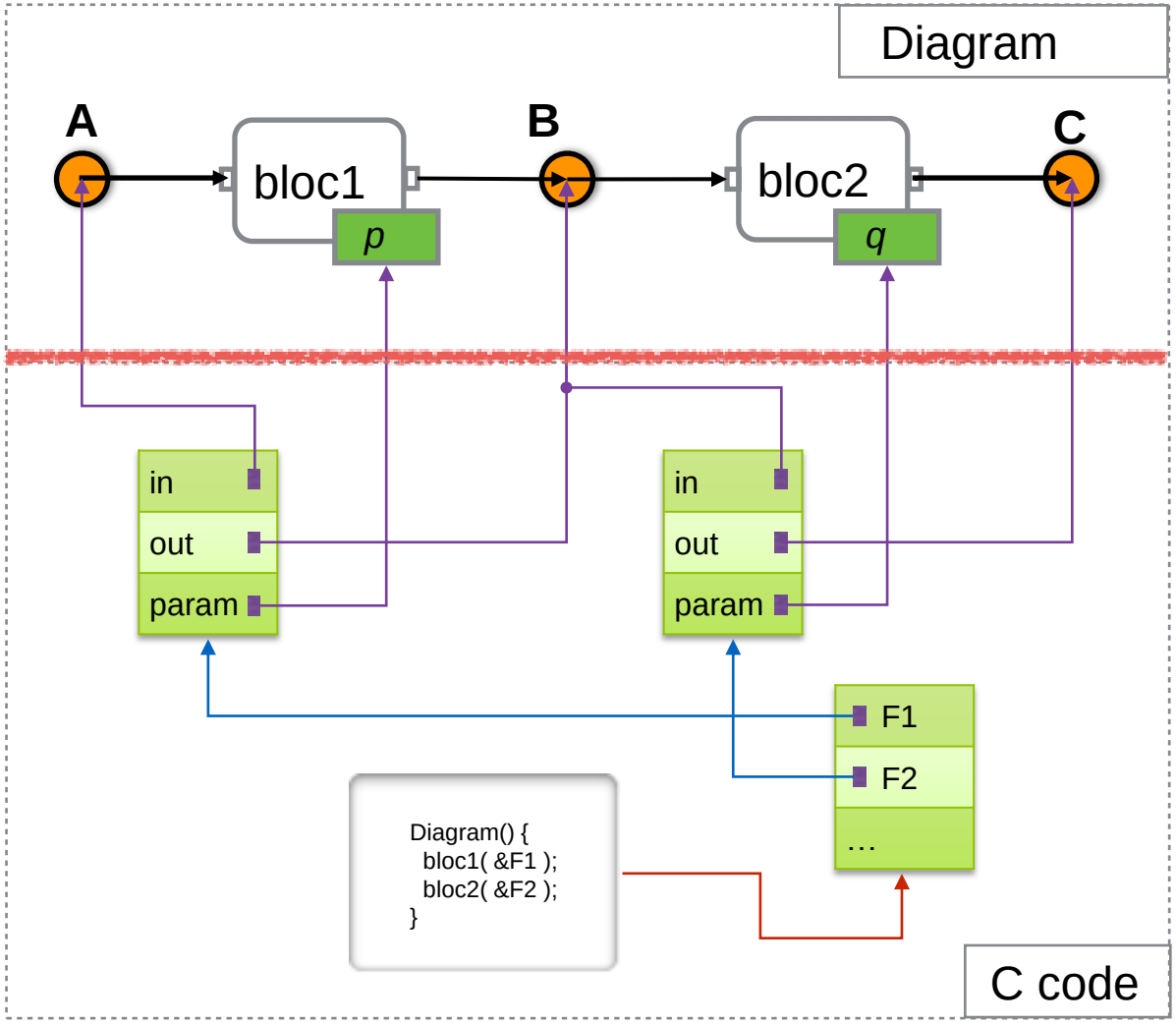
DESIGN | … IS A COMPUTER SCIENTIST'S MODEL



DESIGN | DIAGRAM INVOCATION

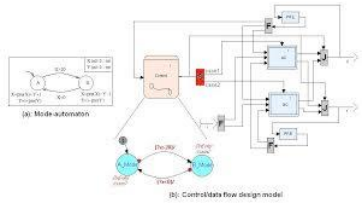
```
main()
{
    mode = Init ;
    Diagram(); // Initialization phase
    mode = Param ;
    Diagram(); // Parametrization phase ——— φ
    mode = Run ;
    loop {
        input();   // Sensor reads
        Diagram(); // Computational cycle ——— M
        output();  // Actuator writes
    }
}
```

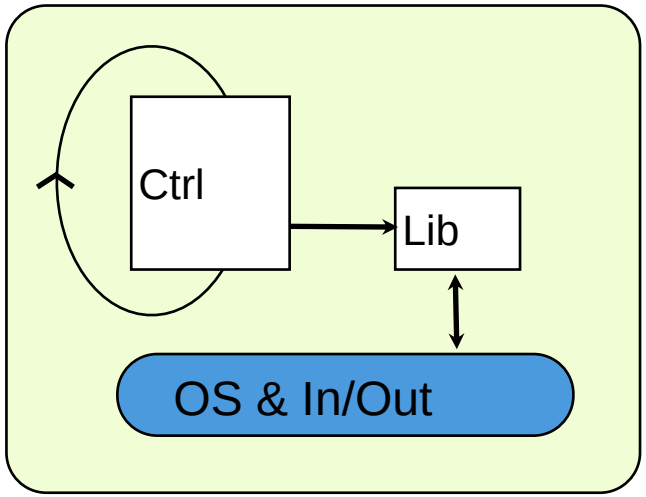
DESIGN | DIAGRAM DECOMPOSITION



WHAT KIND OF VERIFICATION?

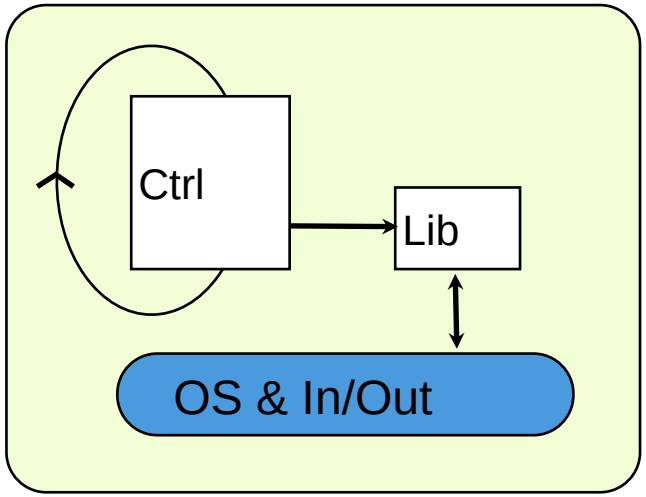
```
int div(int a, int b) {
  int q;
  q=0;
  while (a>=b)
    a=a-b;
    q=q+1;
  }
  return q;
}
```





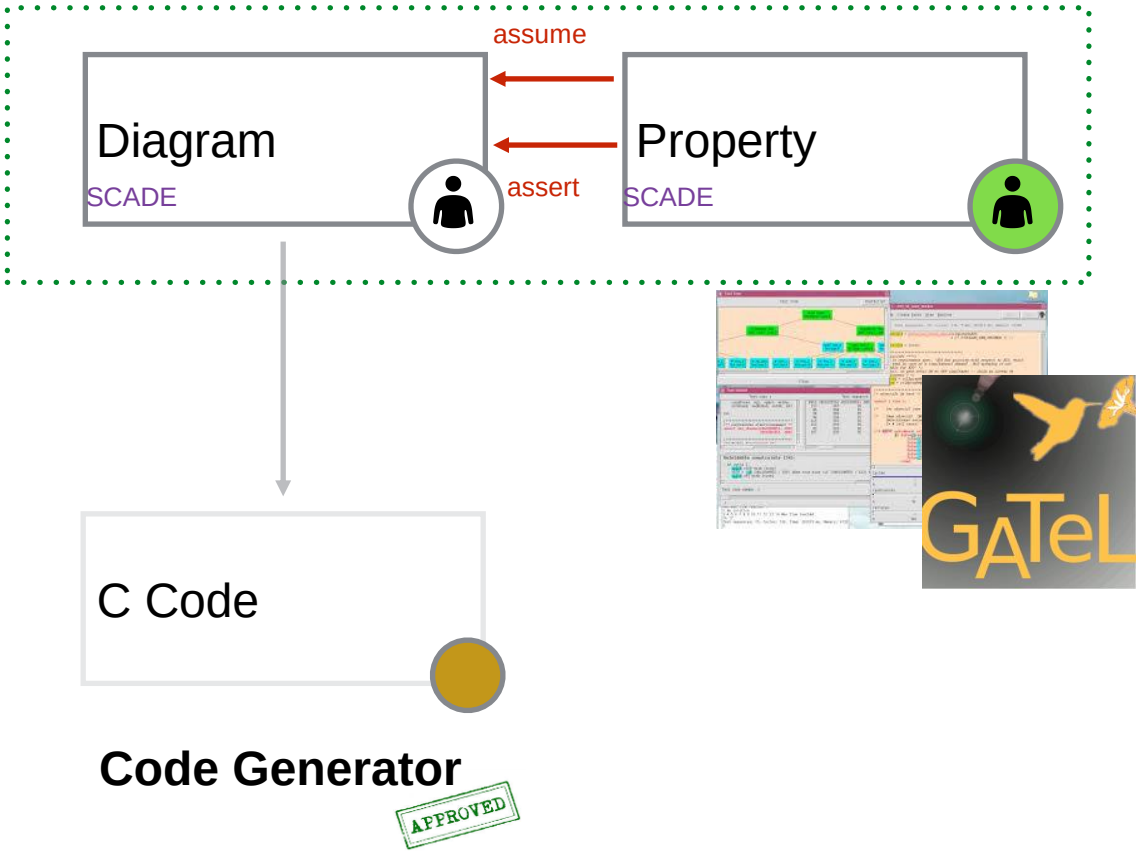
CAN WE ATTEMPT MODEL-LESS VERIFICATION?

```
int div(int a, int b) {
  int q;
  q=0;
  while (a>=b) {
    a=a-b;
    q=q+1;
  }
  return q;
}
```

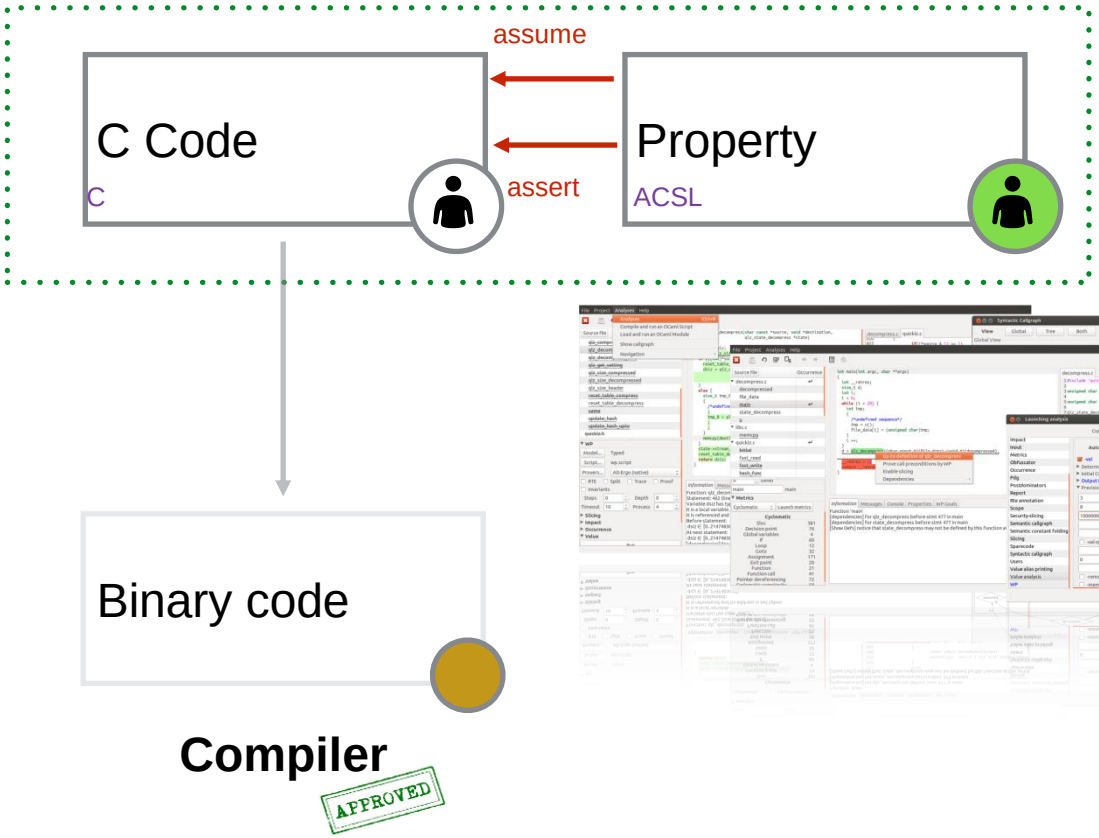


{

VERIFICATION | CEA LIST TOOLS



VERIFICATION | CEA LIST TOOLS



frama C
Software Analyzers

Development
Generation
Validation

VERIFICATION | CORE FRAMA-C COMPONENTS

```
/*@ ensures \result >= x && \result >= y;
    ensures \result == x || \result == y;
*/
int max (int x, int y) { return (x > y) ? x : y; }
```

Kernel

Frama-C
Open Source
Distribution

Path-Crawler
Plug-In

Impact & Slicing
Plug-In

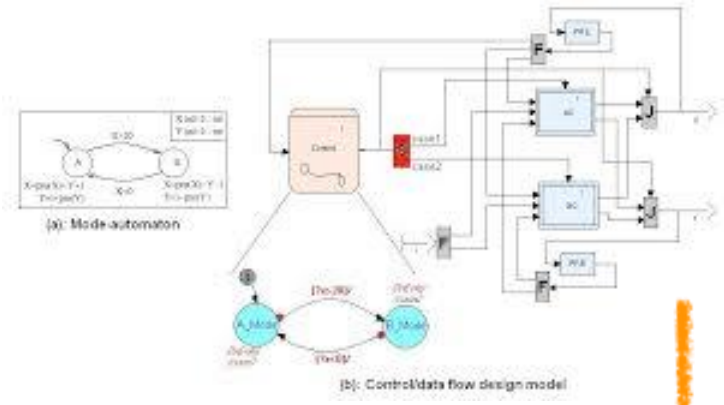
Value
Plug-In

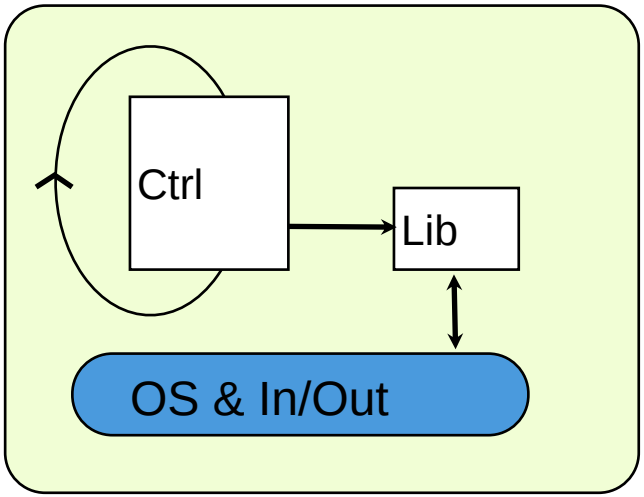
WP
Plug-In

Aoraï
Plug-In

Coq, Alt-Ergo
Why

VERIFICATION | STATE OF PRACTICE





+250 klocs

~10' analysis runtime

~200 remaining alarms

frama C
Software Analyzers

Value

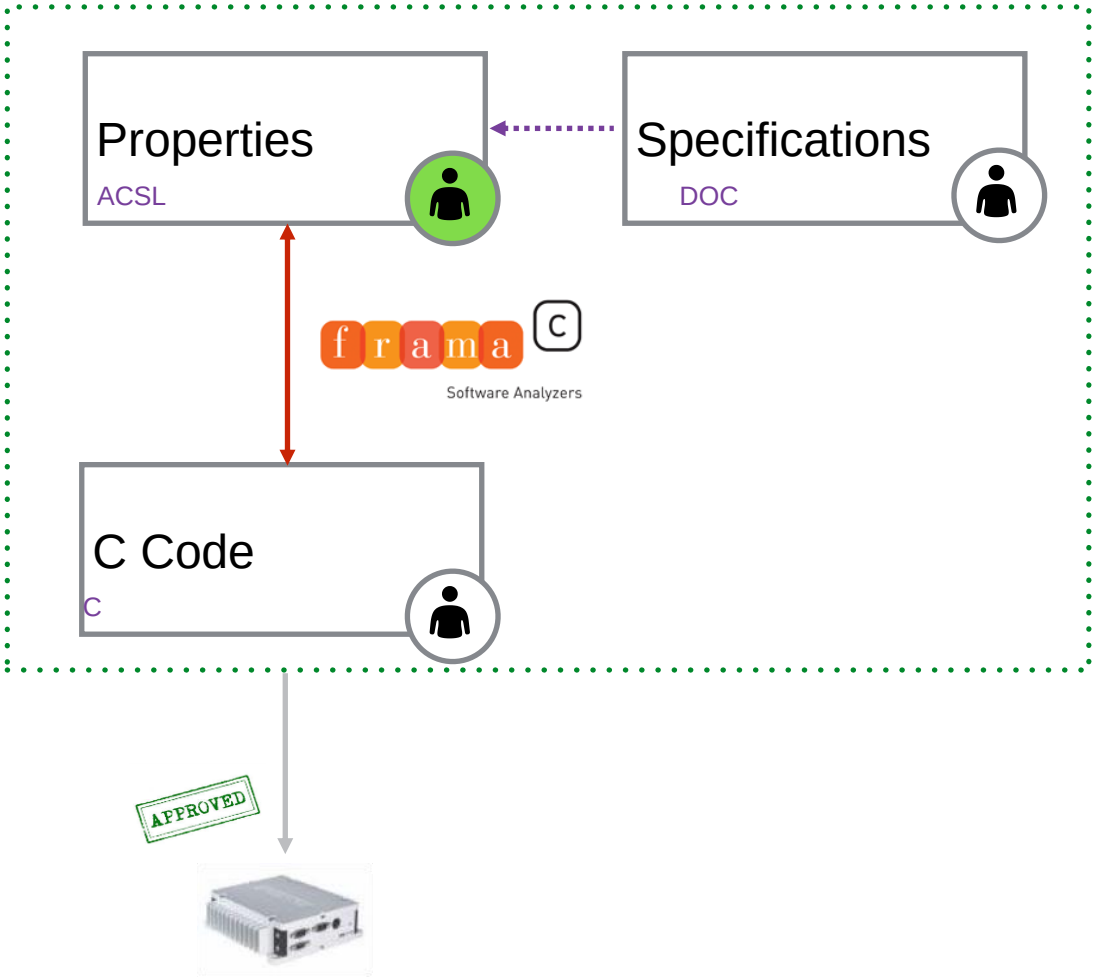
Absence of RTE

OS/Ctrl+Lib separation

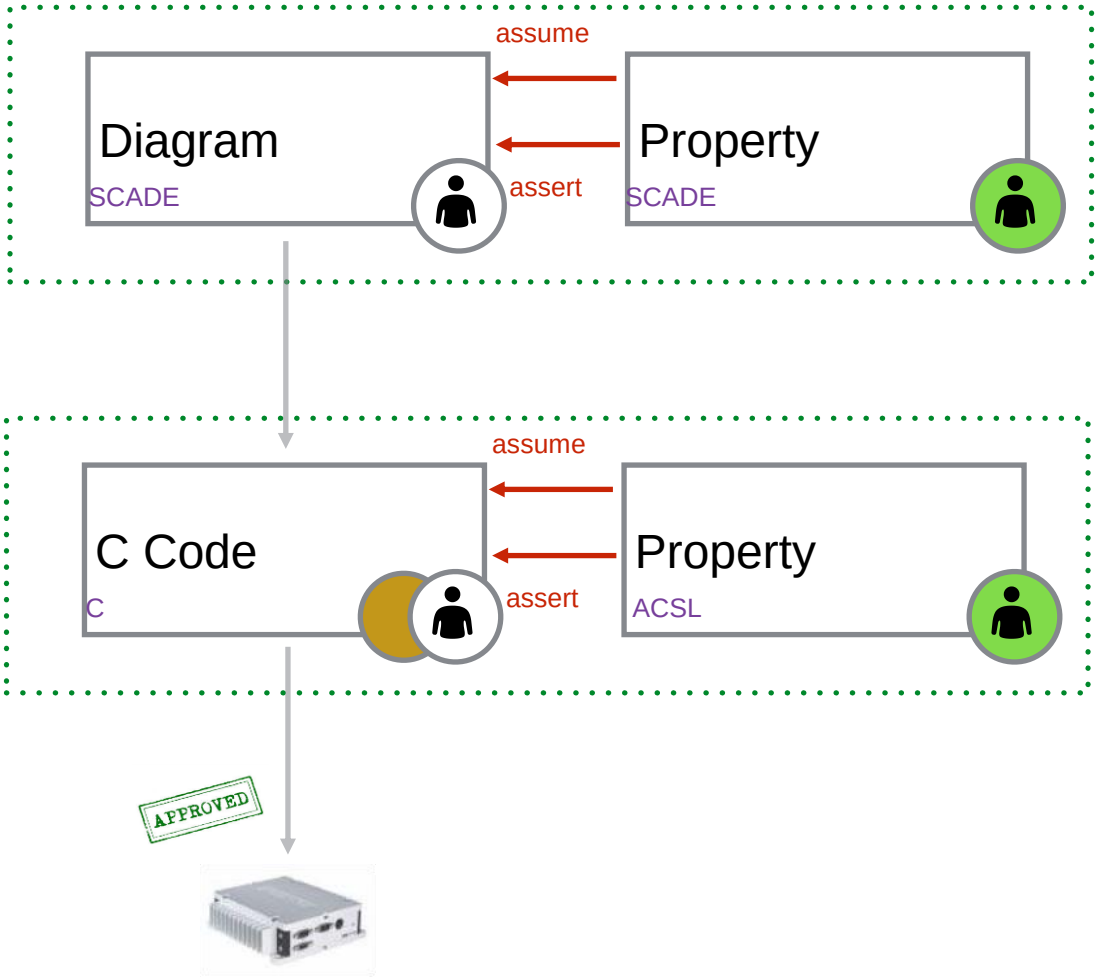
WP

Local proofs

VERIFICATION | STATE OF PRACTICE – LOCAL PROOFS



VERIFICATION | MULTIPLE PROOF TECHNIQUES



GATeL

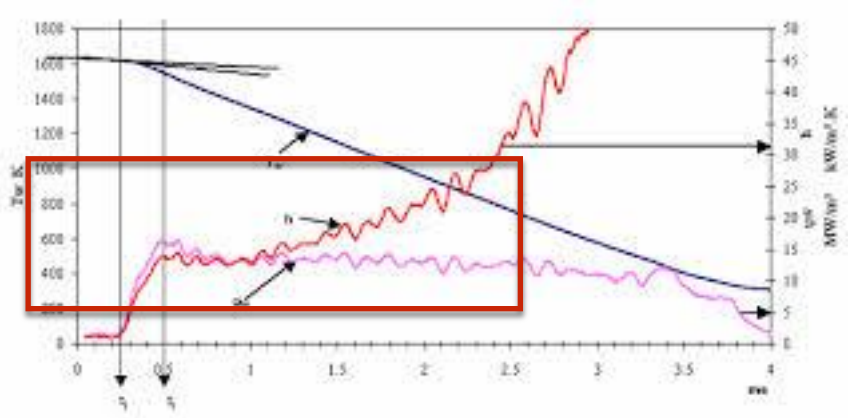
frama C
Software Analyzers

Development
Generation
Validation

}

A SPECIAL KIND OF PROPERTY

ASSUME

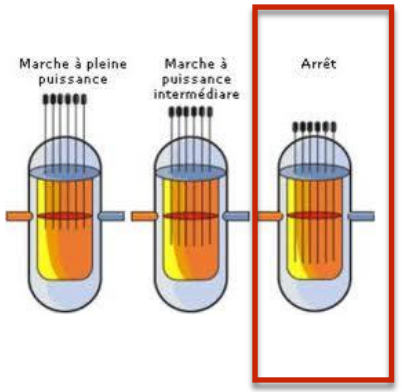


Sensors

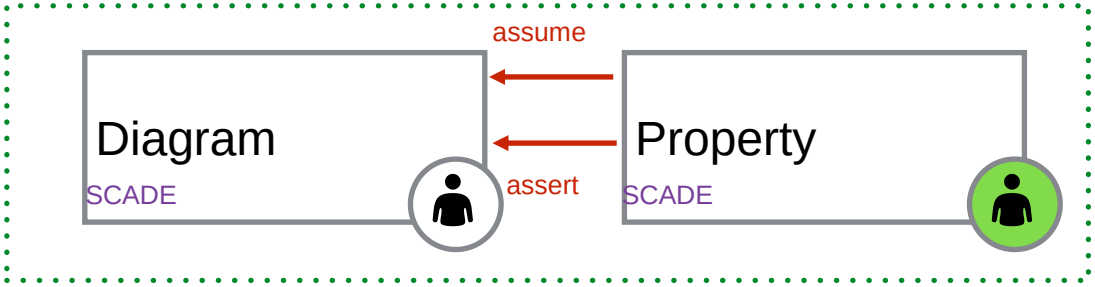
Computations

Actuators

ASSERT



VERIFICATION | DIAGRAM-LEVEL PROPERTIES

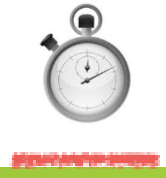


$$\overline{y}_t = \varphi(\overline{m}_t, \overline{x}_t)$$

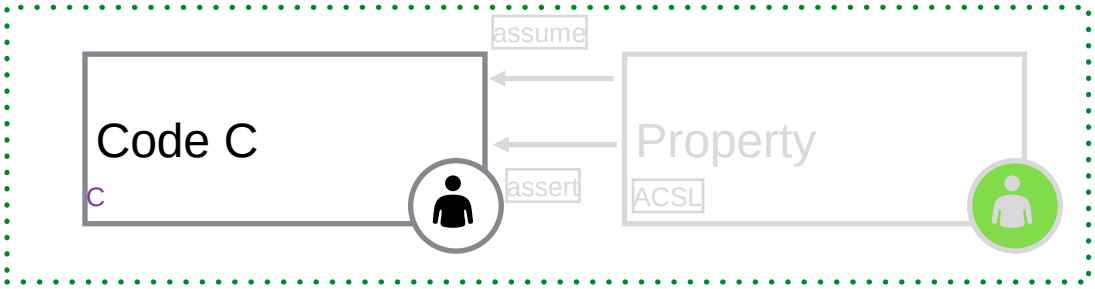
$$\overline{m}_{(t+1)} = M(\overline{m}_t, \overline{x}_t) \qquad \overline{a}_{(t+1)} = A(\overline{a}_t, \overline{x}_t, \overline{y}_t)$$

$$\forall (t \leq T),\ P(\overline{a}_t)$$

(φ, M) **= 1 diagram (abstract level)**



VERIFICATION | CODE-LEVEL PROPERTIES



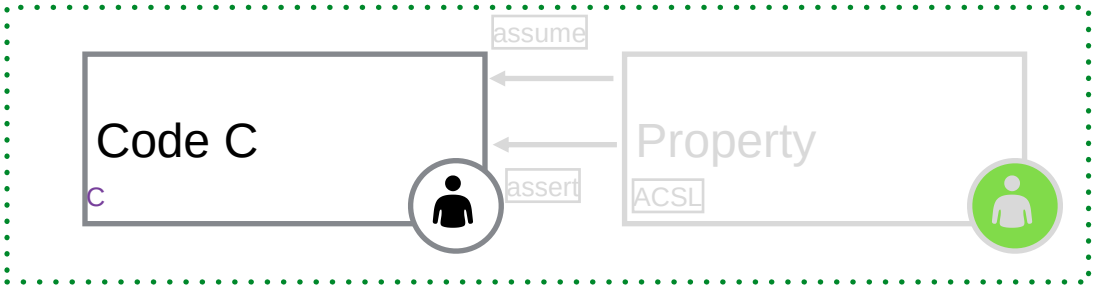
frama C
Software Analyzers

$$\overline{y}_t = \varphi(\overline{m}_t, \overline{x}_t)$$

$$\overline{m}_{(t+1)} = M(\overline{m}_t, \overline{x}_t)$$

(φ, M) **= 1 C function (low level details)**

VERIFICATION | CODE-LEVEL PROPERTIES



frama C
Software Analyzers

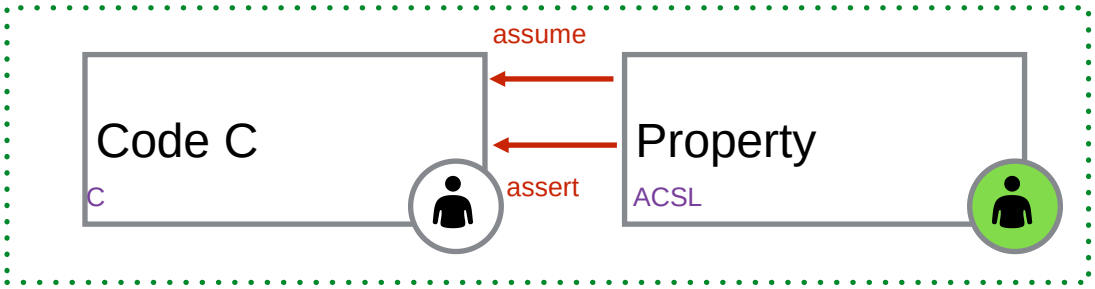
$$\overline{y}_t = \varphi(\overline{m}_t, \overline{x}_t)$$

$$\overline{m}_{(t+1)} = M(\overline{m}_t, \overline{x}_t)$$

$$\forall t, \quad (m^i)_t \in D^i$$

(φ, M) = **main function (low level details)**

VERIFICATION | CODE-LEVEL PROPERTIES



frama C
Software Analyzers

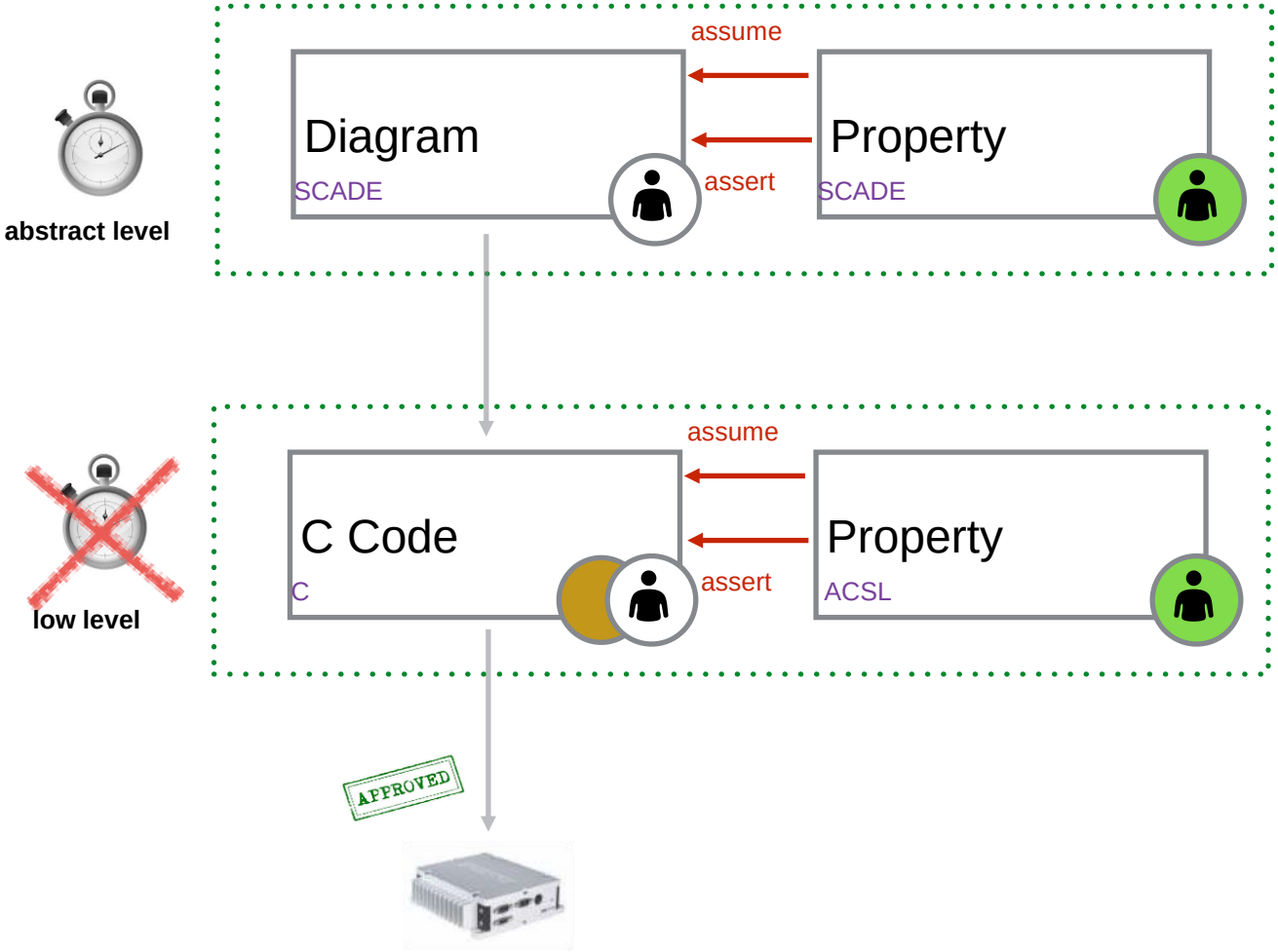
$$\overline{y}_t = \varphi(\overline{m}_t, \overline{x}_t) \qquad H(\overline{m}_t, \overline{x}_t)$$

$$\overline{m}_{(t+1)} = M(\overline{m}_t, \overline{x}_t) \qquad G(\overline{m}_{(t+1)}, \overline{x}_t, \overline{y}_t)$$

$$\forall t, \quad H(\overline{m}_t, \overline{x}_t) \implies G(\overline{m}_{(t+1)}, \overline{x}_t, \overline{y}_t)$$

(φ, M) **= 1 C function (low level details)**

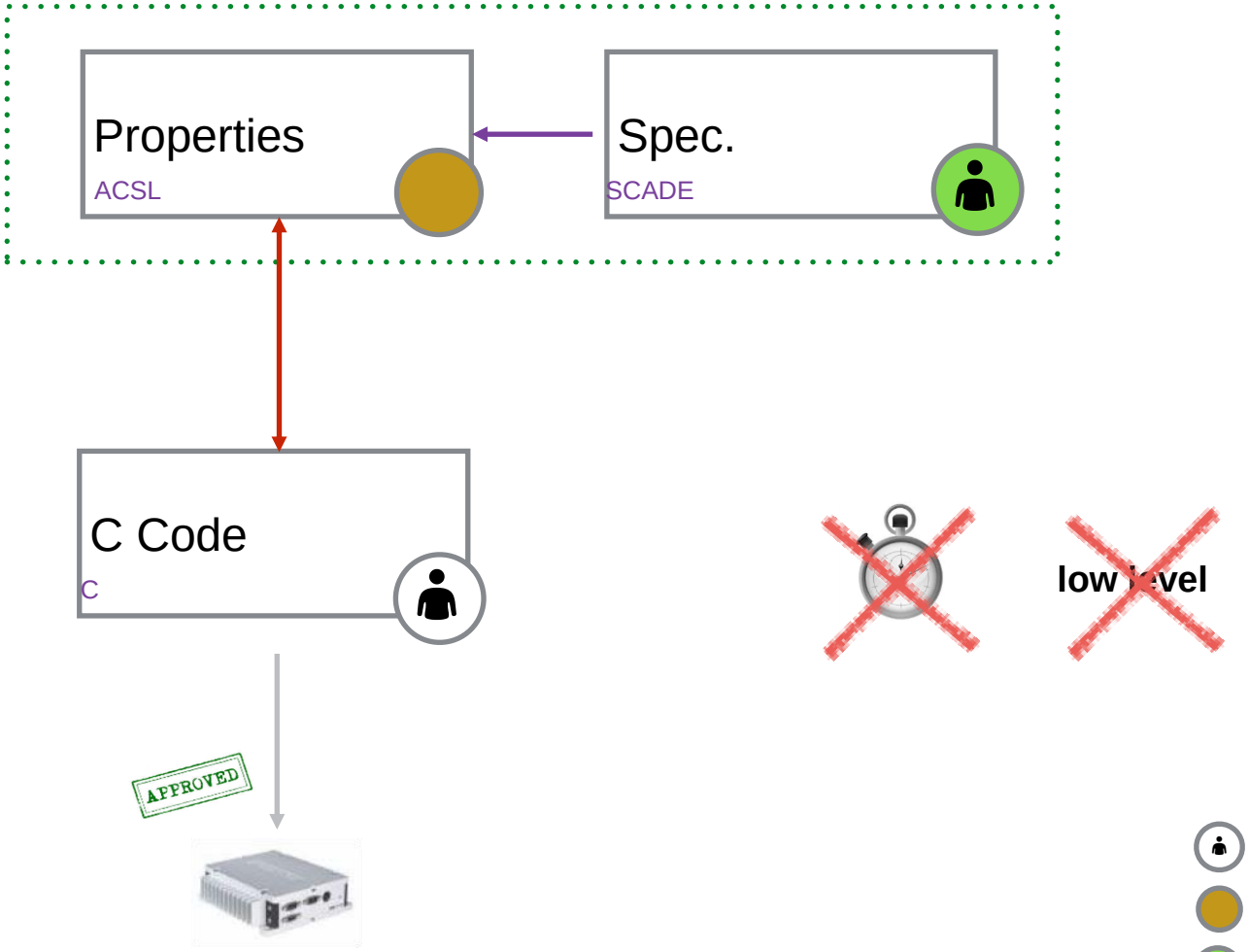
VERIFICATION | DO MORE / DO BETTER?



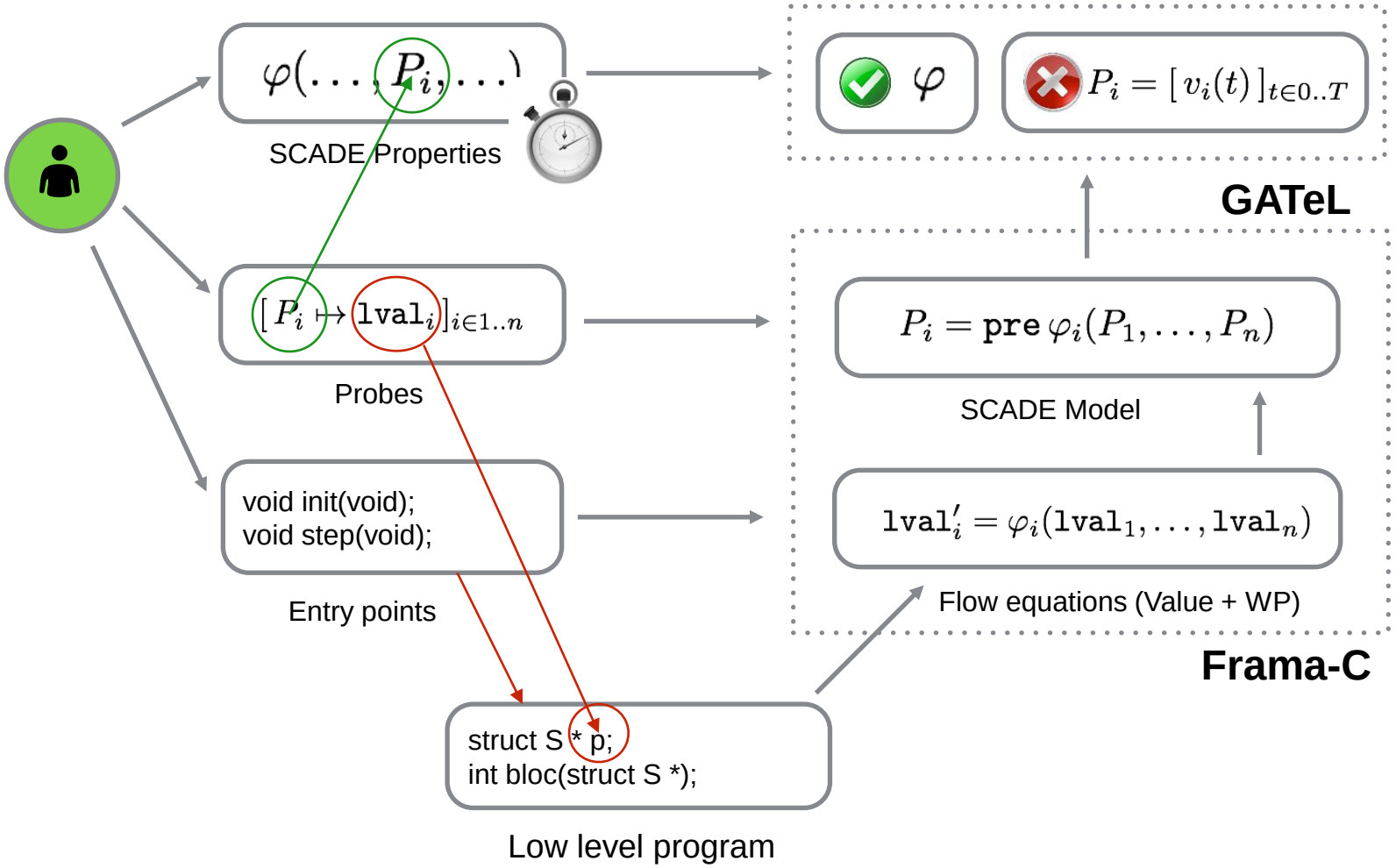
GATeL

frama C
Software Analyzers

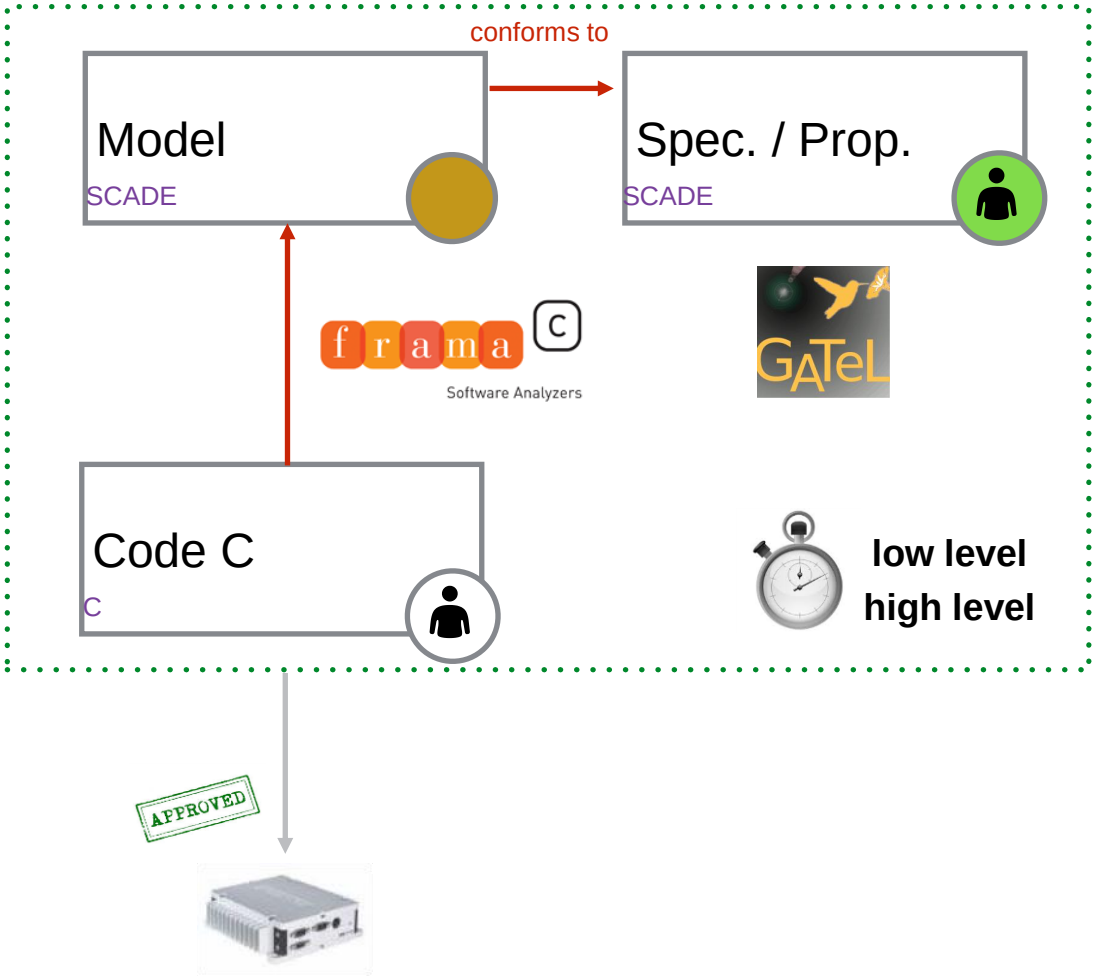
VERIFICATION | PROOFS FROM GENERATED PROPERTIES



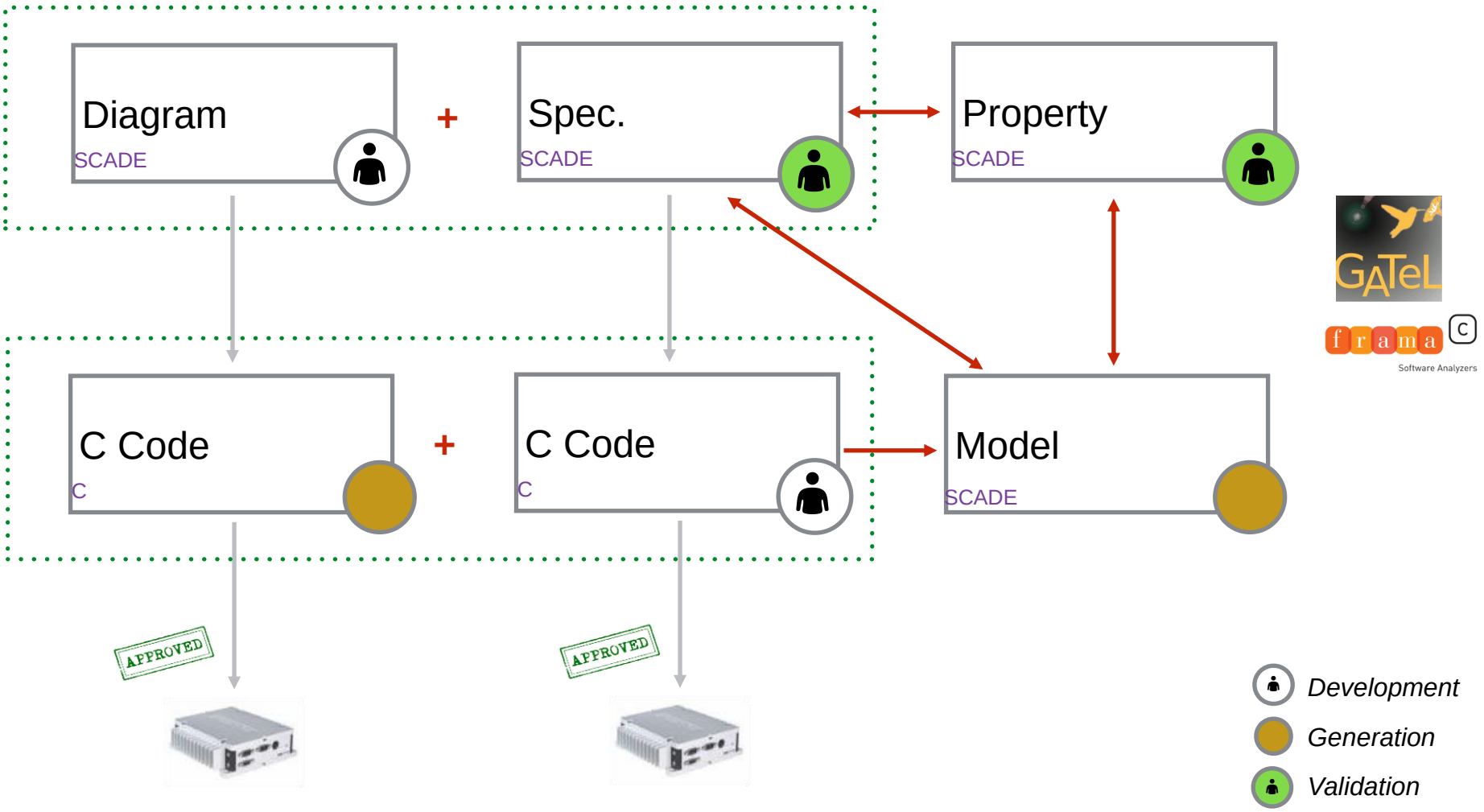
VERIFICATION | ANOTHER APPROACH



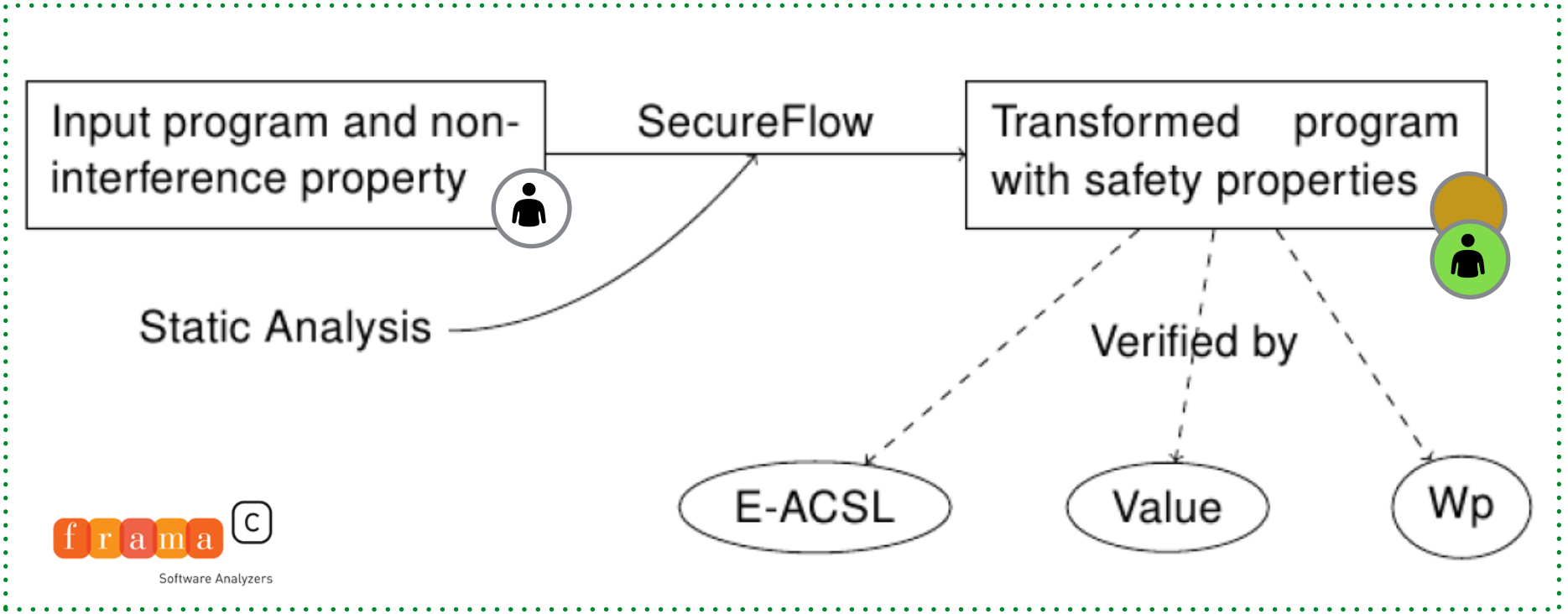
VERIFICATION | ANOTHER APPROACH – 2014



CONCLUSION | COMPLEMENTARY NEW APPROACHES

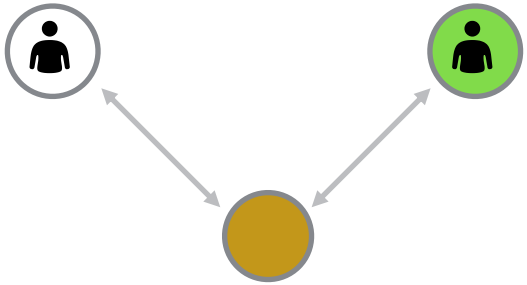


CONCLUSION | A DETOUR THROUGH CYBERSECURITY



CONCLUSION | A NEW DISCIPLINE

Systems engineering || **Proof engineering**



Software Security Laboratory
Software & Systems Engineering Department
CEA LIST

Florent Kirchner
florent.kirchner@cea.fr