EURO-MILS, Certification, and Theorem Proving

Julien Schmaltz, OUNL / TU Eindhoven, julien.schmaltz@ou.nl

(with contributions from reek Verbeek, OUNL, Sergey Tverdyshev and Holger Blasum, SYSGO AG, and Burkhart Wolff, University of Paris Sud)

EURO-MILS No. 318353
Secure European Virtualisation for Trustworthy Applications in Critical Domains

*

About the speaker

- Background in EDA, EE, and formal methods
- Expertise in Interactive Theorem Proving (> 10 years)
- Used ACL2 and Isabelle/HOL on diverse projects
- Other expertise in model-checking, model-based testing, SAT/SMT
- Applications to hardware and software
- Recently (1 year) involved in a project about certification

3

Goals of this talk

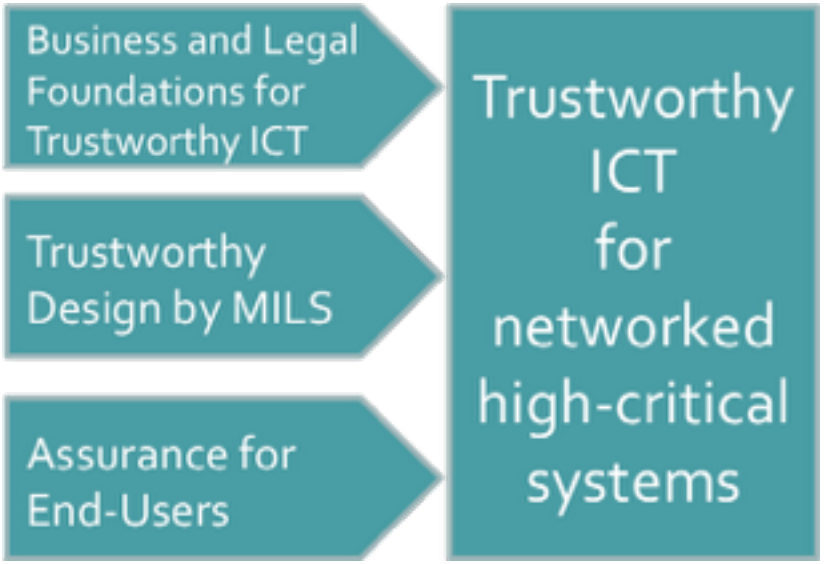
- Short overview of the EU project EURO-MILS
- Formal validation of the security policy using Isabelle/HOL
- Relation of EURO-MILS to some topics suggested for this workshop

3

EURO-MILS Objective and Strategy

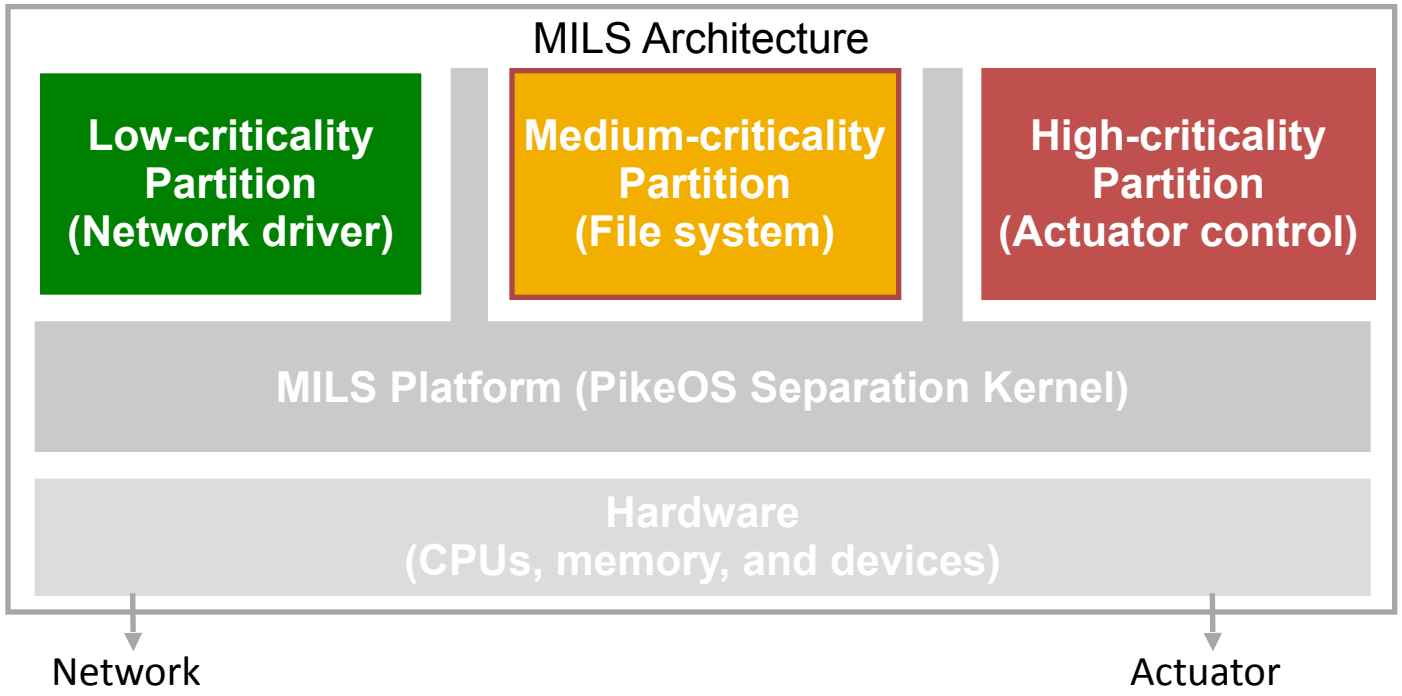
- **High-criticality networked cyber-physical systems**
  - Drivers are avionics and automotive
  - EURO-MILS delivers cross-domain solutions
- **Integration and networking requires trustworthy ICT**
- **MILS – Multiple Independent Levels of Security**
  - **High-assurance security architecture**
  - **Scalable and affordable security**
  - **Compositional**

  design, assurance, security



EURO-MILS: First pan-European MILS architecture and certifiable platform, for the SYSGO PikeOS separation kernel

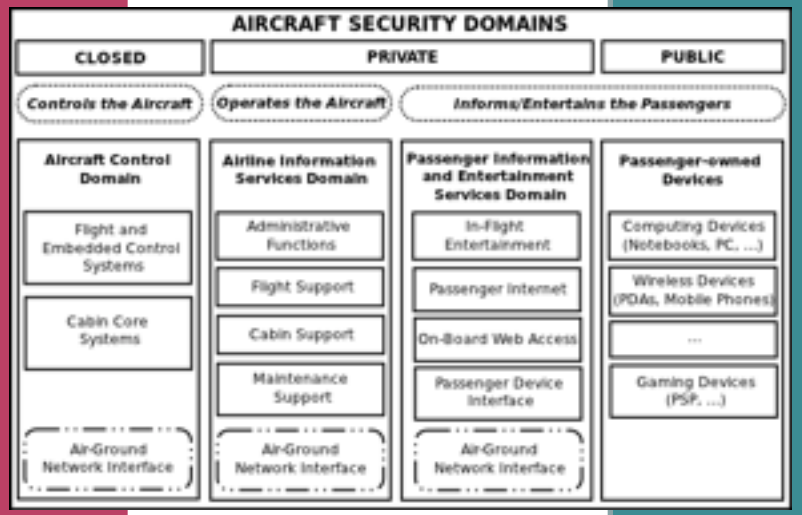
Separation Kernel Overview

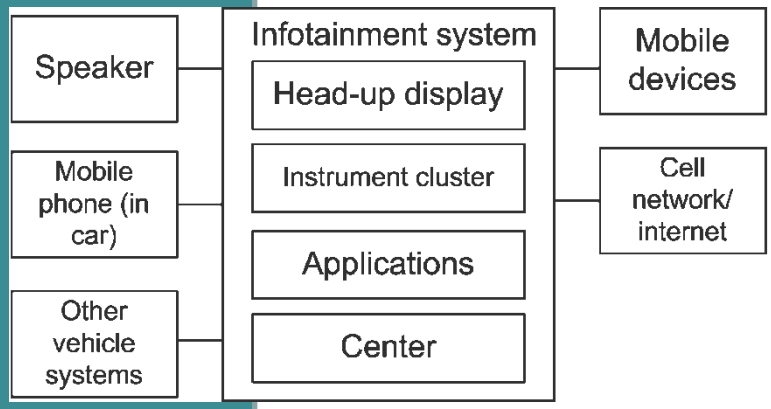


Two prototypes One MILS platform

Avionic

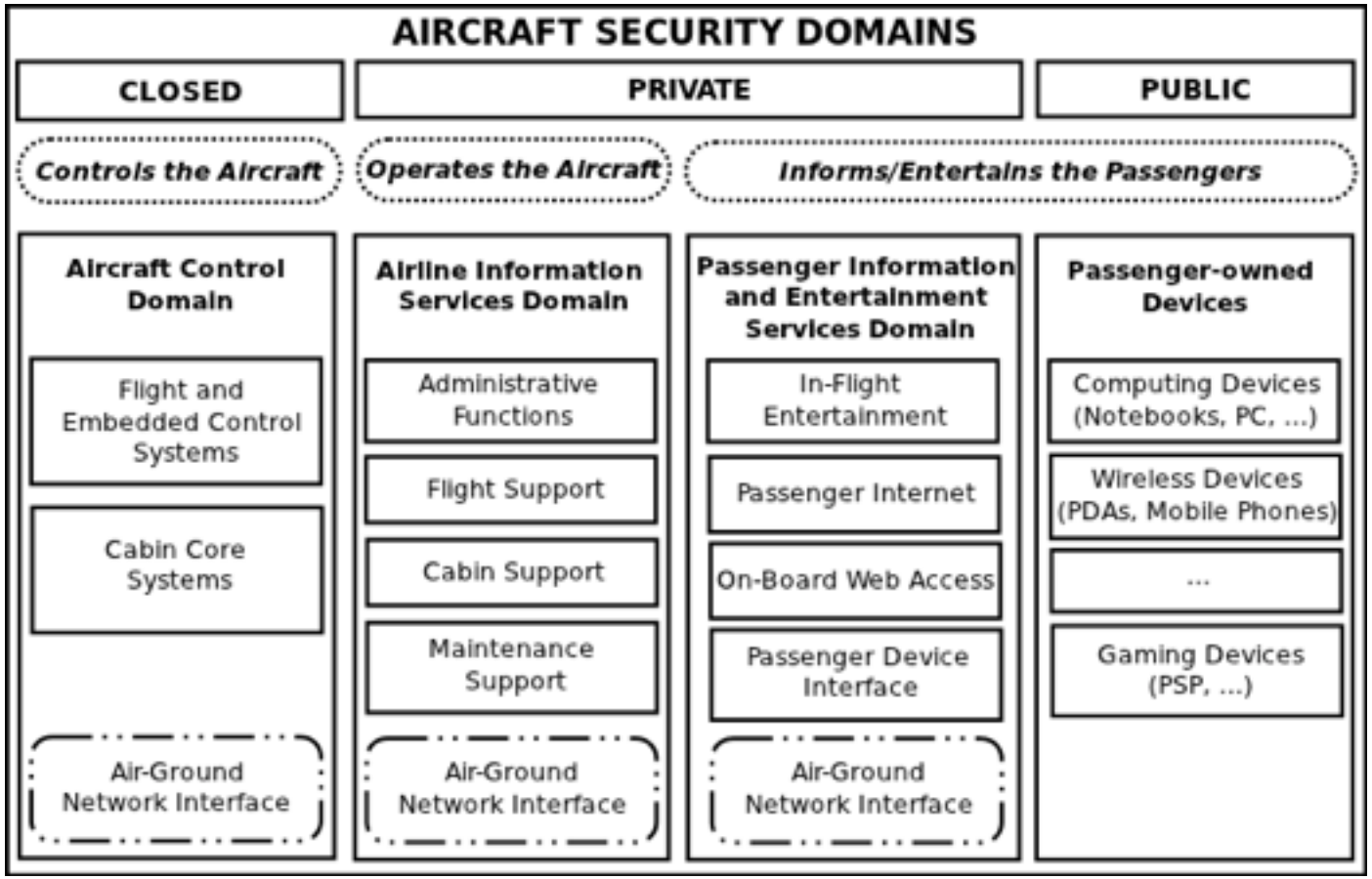
Automotive





Trustworthy ICT
for networked
high-critical systems

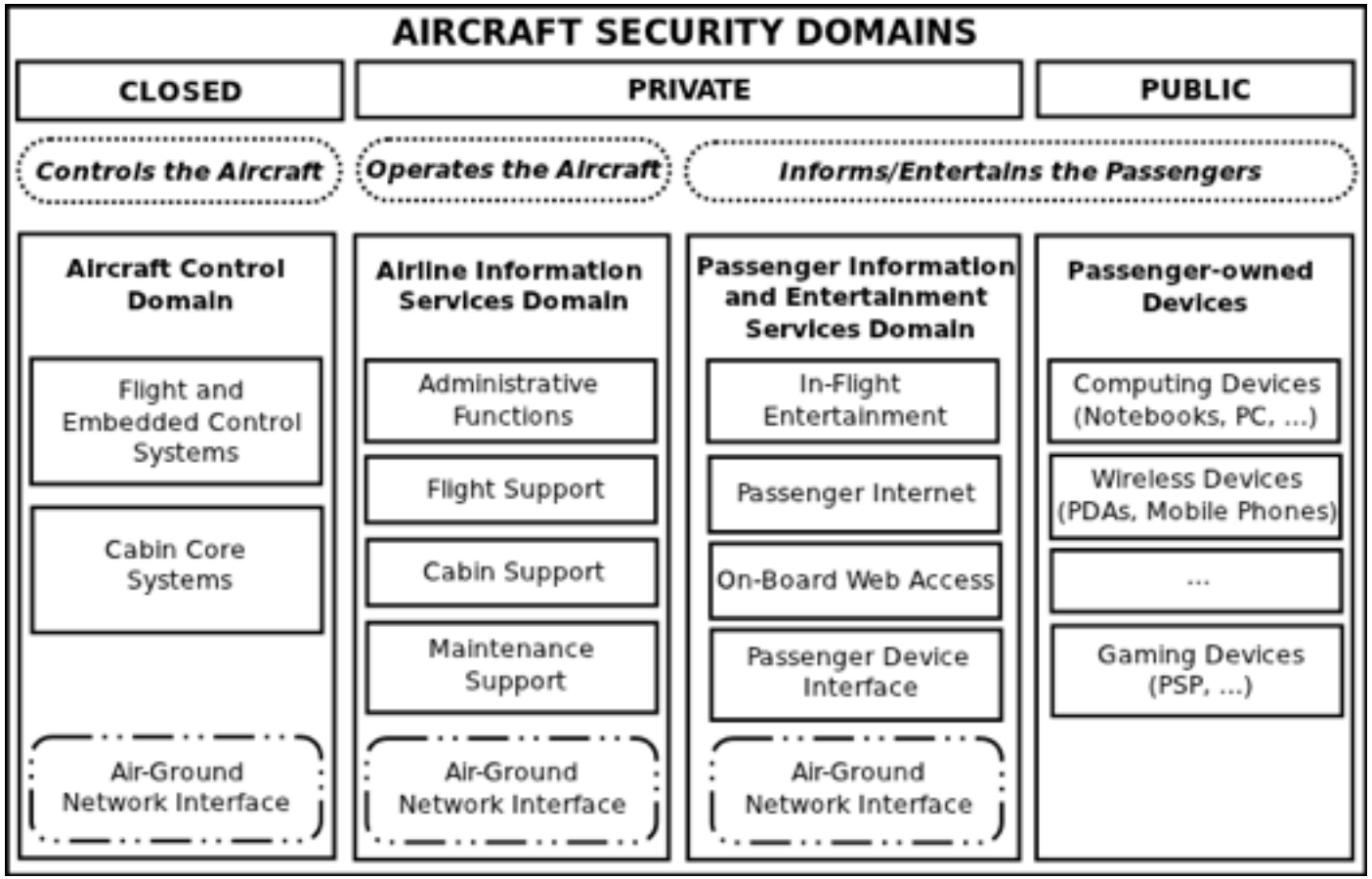
Example: Aircraft Security Domains



- Picture adapted from ARINC 811.
- Domains are defined In ARINC 664 Part 5.

5

Example: Aircraft Security Domains

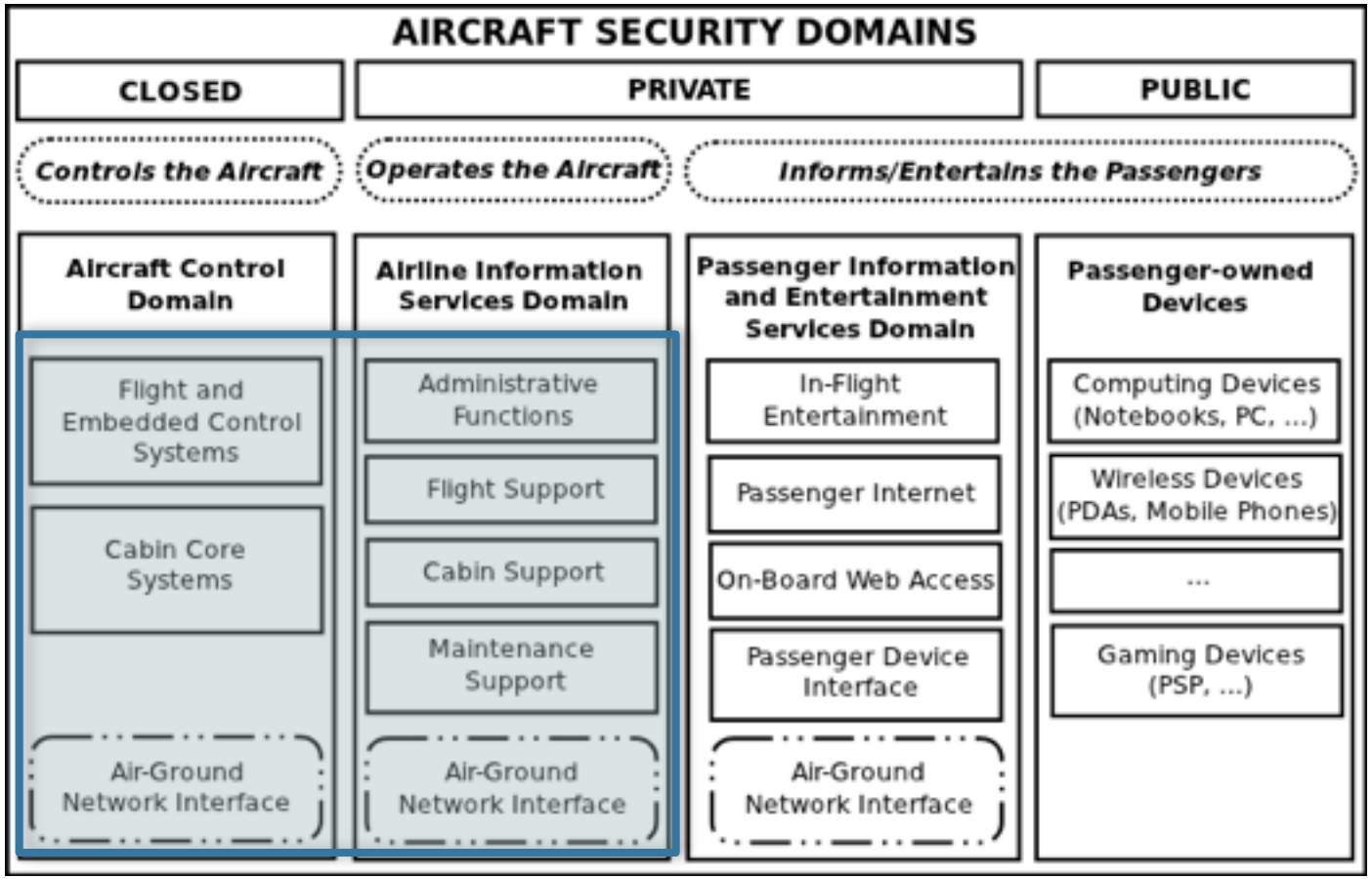


Perspective "User"
(not 100% accurate)

- Picture adapted from ARINC 811.
- Domains are defined In ARINC 664 Part 5.

5

Example: Aircraft Security Domains



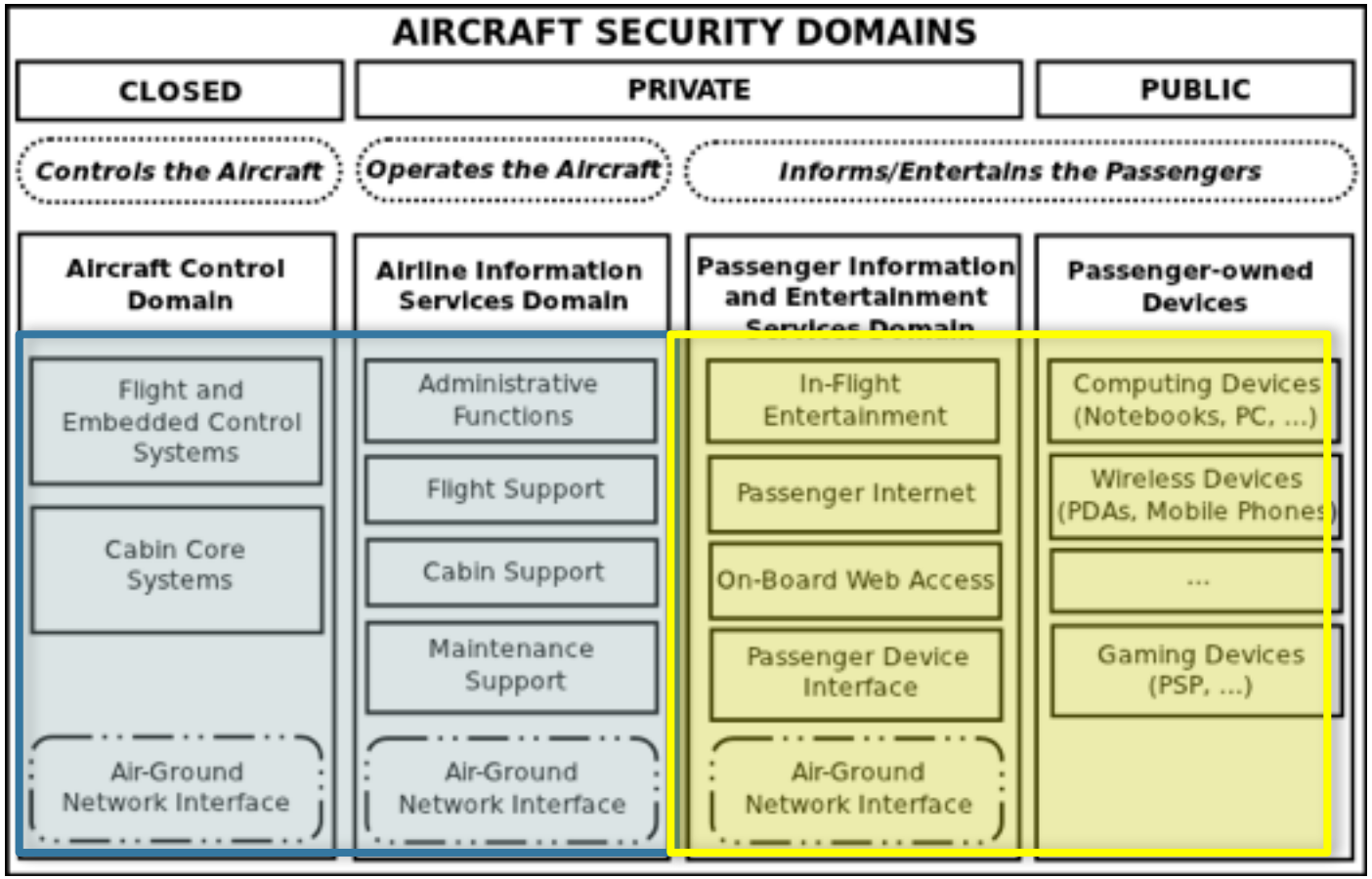
Perspective "User"
(not 100% accurate)

Crew

- Picture adapted from ARINC 811.
- Domains are defined In ARINC 664 Part 5.

5

Example: Aircraft Security Domains



Perspective "User"
(not 100% accurate)

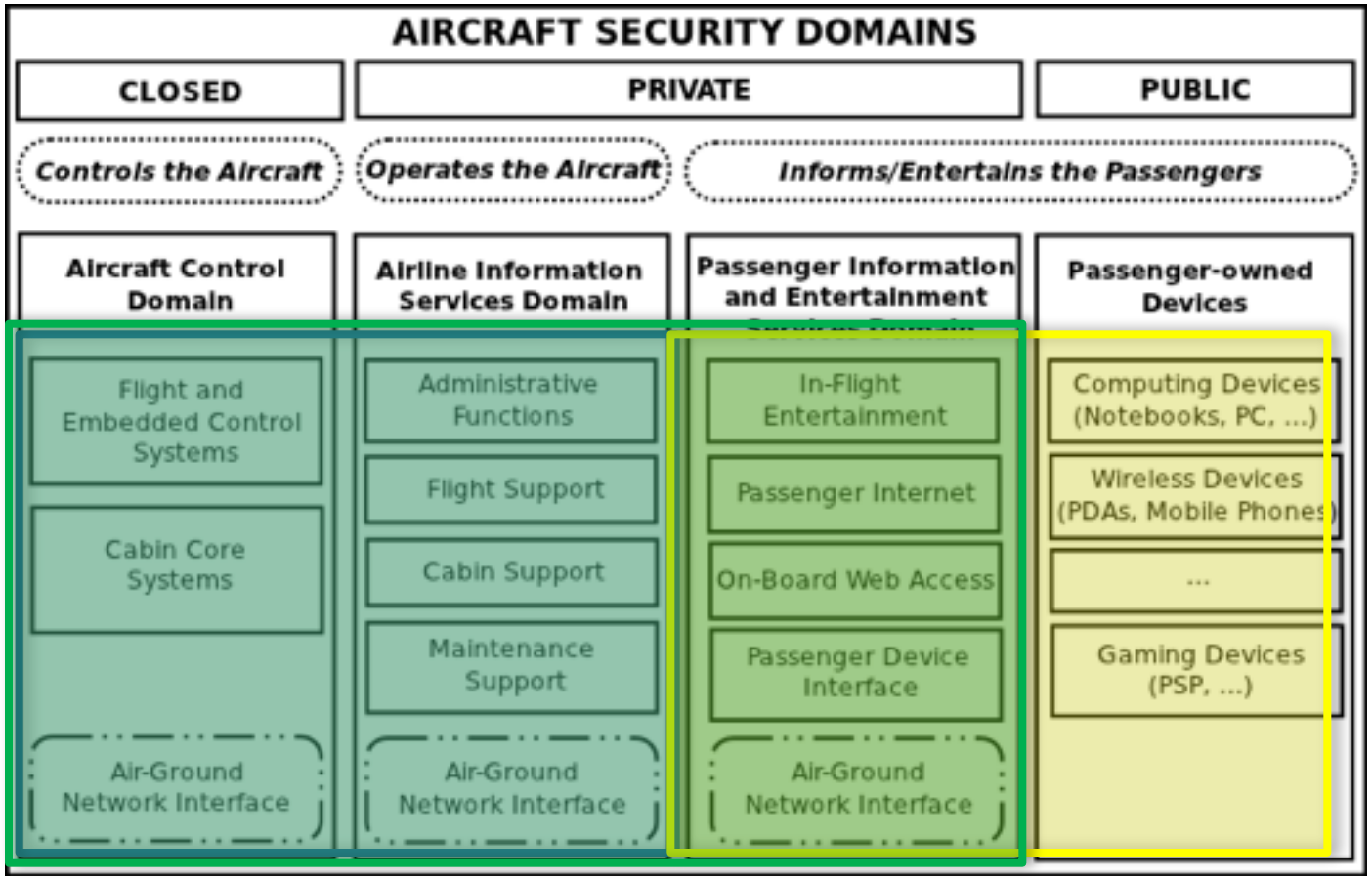
Crew

Passenger

- Picture adapted from ARINC 811.
- Domains are defined In ARINC 664 Part 5.

5

Example: Aircraft Security Domains



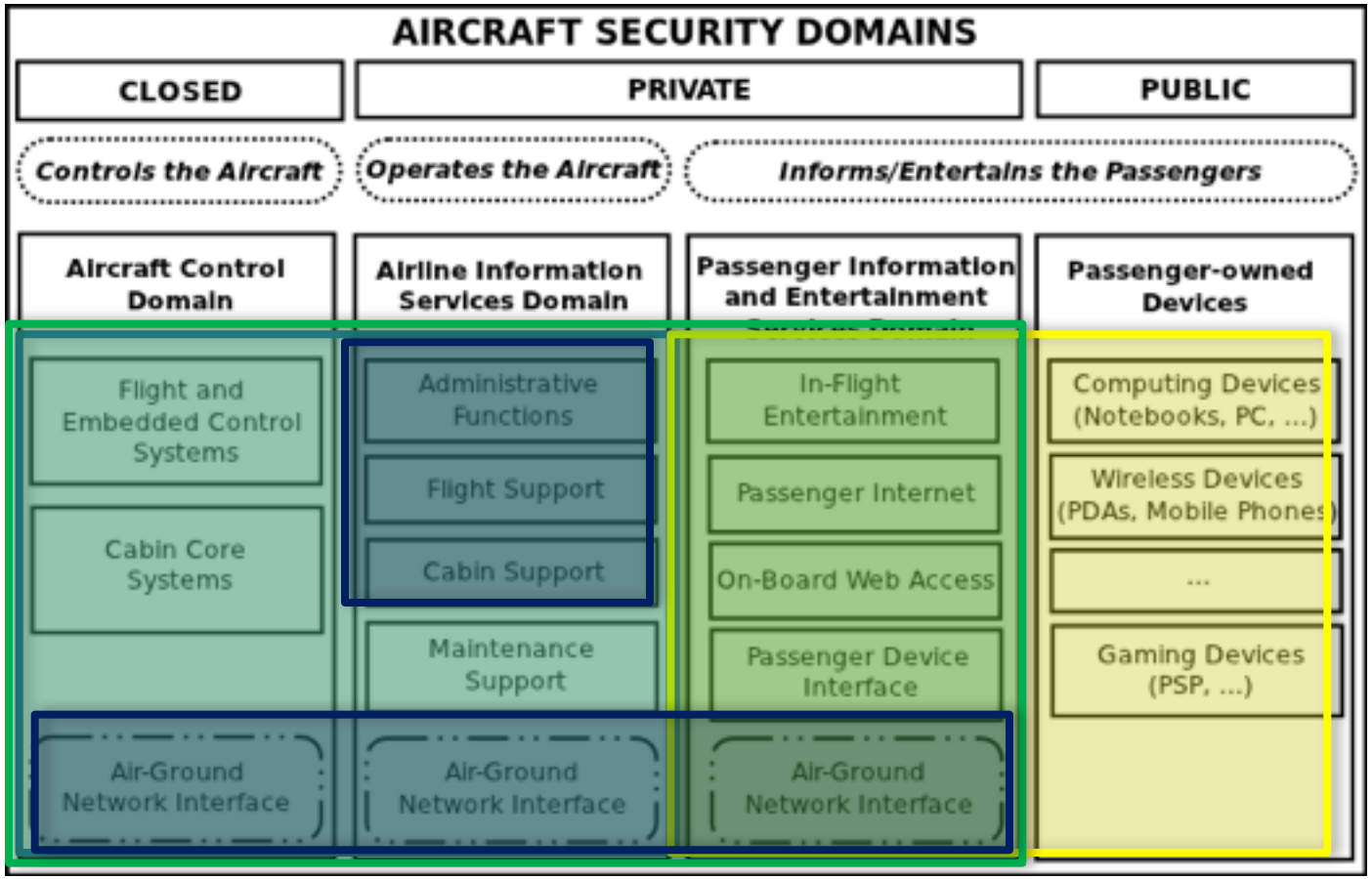
Perspective "User"
(not 100% accurate)

- Crew
- Maintenance (all types)
- Passenger

- Picture adapted from ARINC 811.
- Domains are defined In ARINC 664 Part 5.

5

Example: Aircraft Security Domains



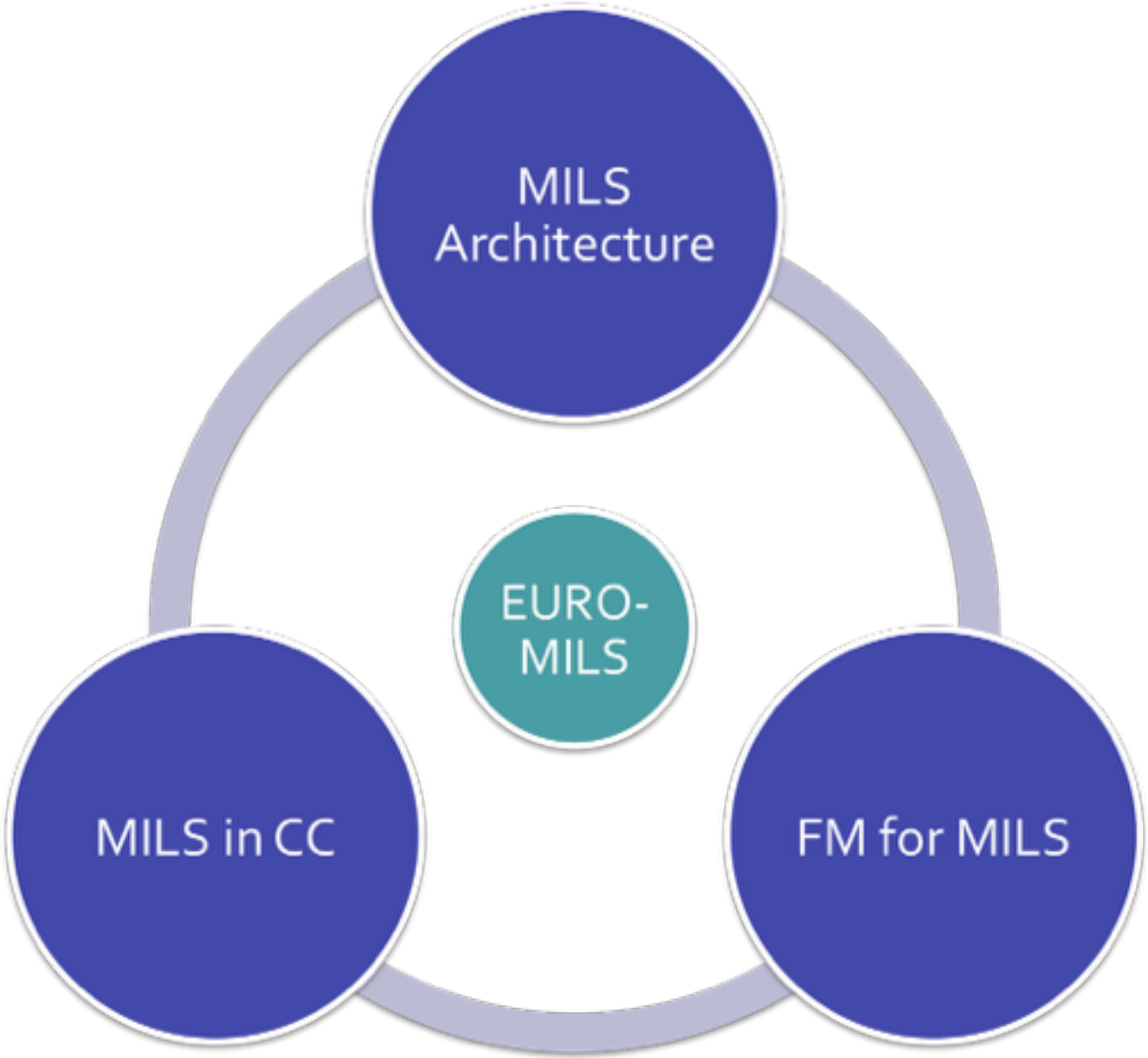
Perspective "User"
(not 100% accurate)

- Crew
- Maintenance (all types)
- Passenger
- Others (Air Traffic Control, Airline Services, Ground)

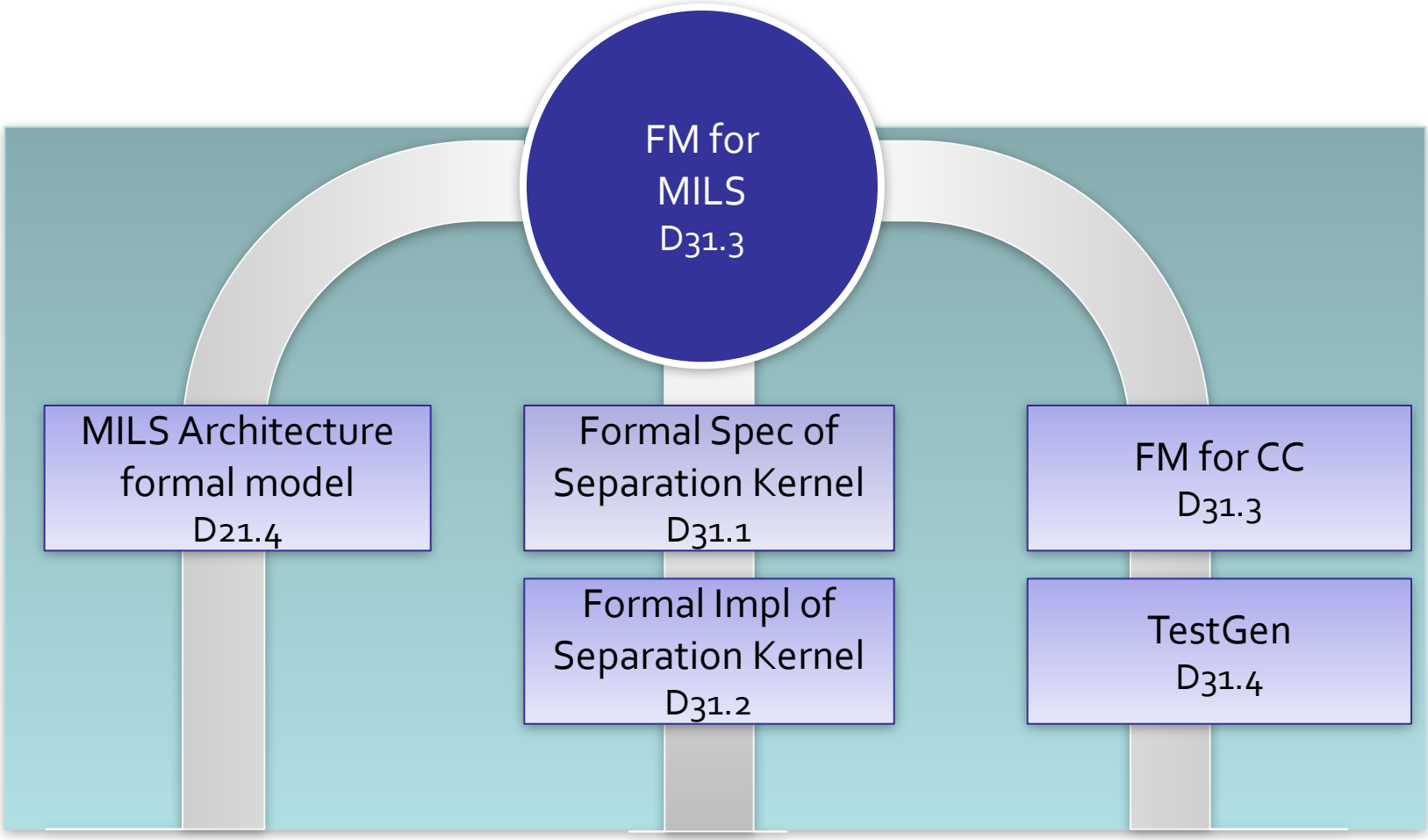
- Picture adapted from ARINC 811.
- Domains are defined In ARINC 664 Part 5.

5

EURO-MILS Focus Points



Focus: Formal Methods (Isabelle/HOL)



Formal validation of separation

- Formal proofs in Isabelle/HOL
- Formal methods ingredients
  - Formal model of the system
  - Formal model of the security policy
  - Formal definition of “secure”
- Goals of EURO-MILS
  - Proof itself and its application to PikeOS
  - *Methodologies to achieve the proof*

Isabelle in a Formal Certification Process

- Isabelle does not only support formal theory developments, it is a general system checking „Formal Content“

  - with text-features containing references to formal entities (antiquotations, which may be user-defined)
  - with code-generation helping to validate definitions
  - with generated code to be run against the implementation in testing scenarios
  - with a modern IDE (called PIDE) allowing to explore the formal content (tooltips on tooltips pointing to definitions).

Isabelle in a Formal Certification Process

EURO-MILS provides also documents for evaluators, comprising

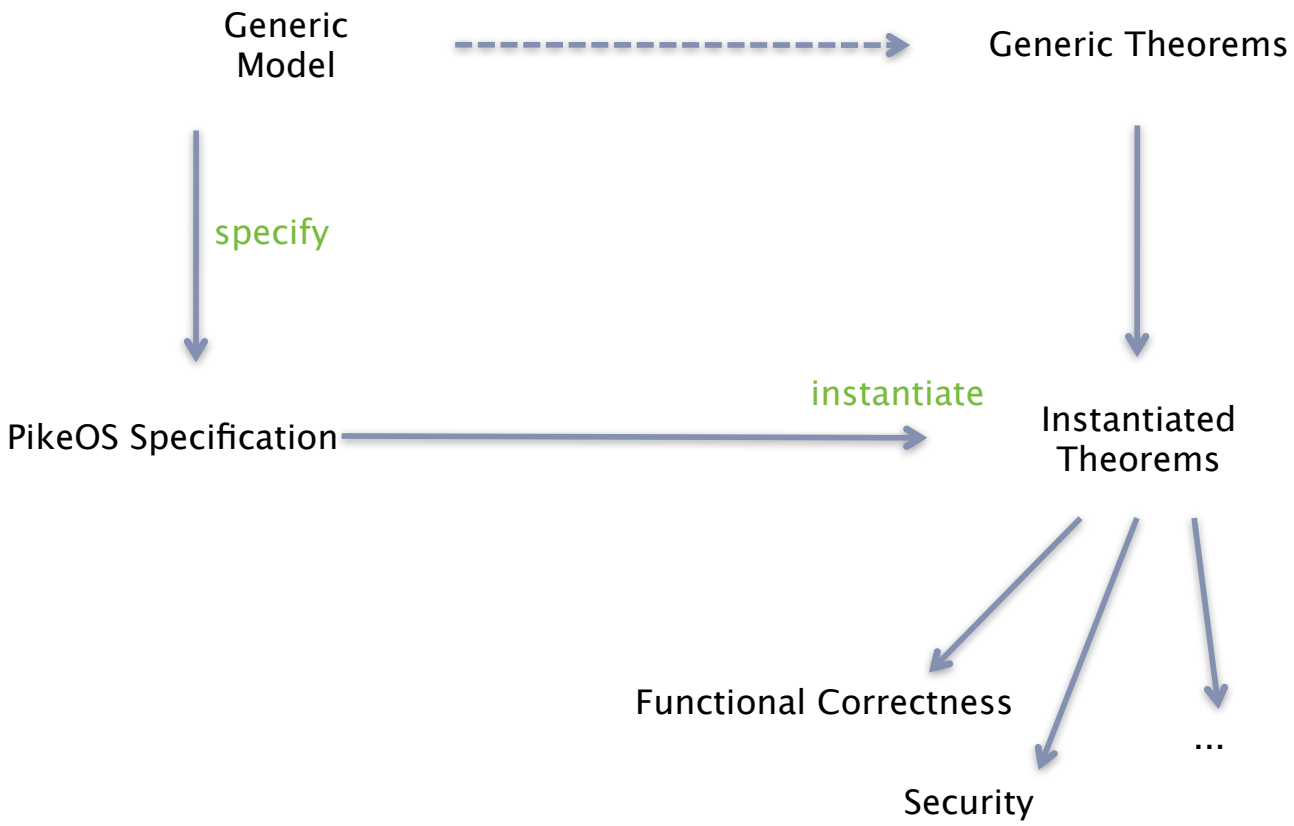
- ... a mandatory recommendation following a text by Eric Jaeger (ANSSI)[Remarques relatives à l'emploi des méthodes formelles déductives en sécurité des systèmes d'information, 2008, http://www.ssi.gouv.fr/IMG/pdf/ssi_formelle.pdf.]
  - Isabelle/HOL: Architecture, (why is it EAL7 ready ?)
  - Language and Methodology (how to use safely ?)
  - Extensions of Isabelle: Guidelines for the Evaluator
  - The dangers of bogus-proofs and - obfuscations
- ... a recommended StyleGuide specificially for certification documents is in preparation.

EURO-MILS and Certification

- **Relation between Security Target and Isabelle/HOL proofs**
  - relates to Subtopic 1.2
  - how to present theorem proving results to evaluation authorities
- **Isabelle Jaeger paper**
  - relates to Subtopic 2 (2.1 and 2.2)
  - how to check proper use of Isabelle/HOL
- **Guidelines about how to develop Isabelle/HOL theories**
  - … in such a way that they can be positively evaluated
- **Most of these points are work-in-progress … ETA 10-2015!**

3

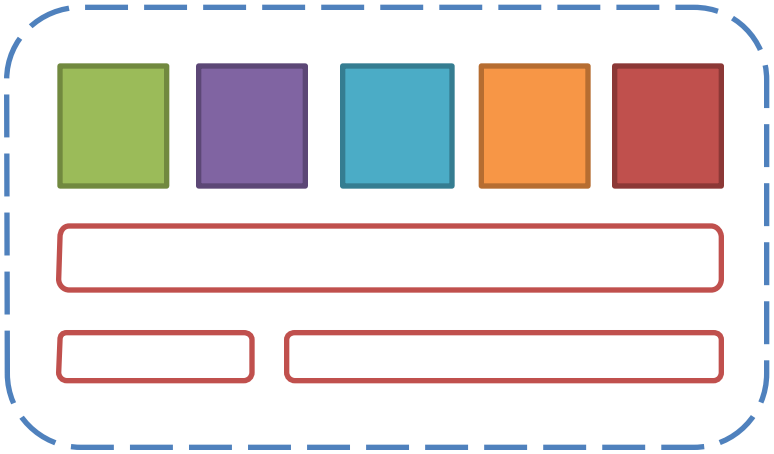
General Approach



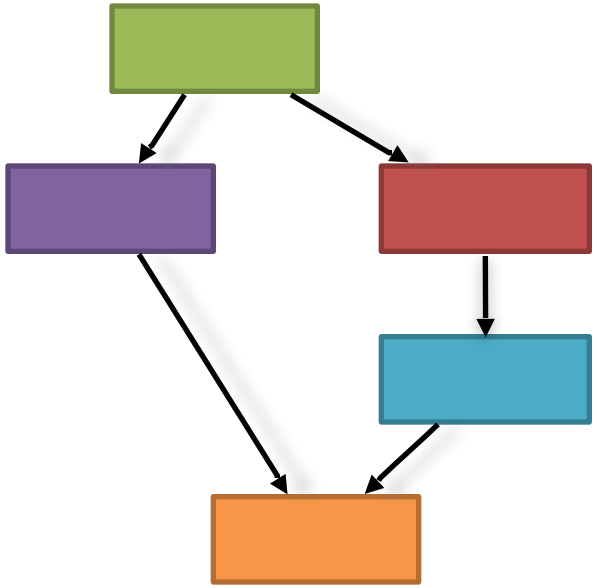
Generic model: CISK

- Controllable Interruptible Separation Kernel (CISK)
- Formal model of separation with control and interrupts
- Model closer to realistic systems, like PikeOS
- Deliverable D31.1 (PU)
- Related work: Rushby + Greve, Wilding, Vanfleet

MILS System



Security Policy



Types

'dom = {  , ...}

'action = {CPY, IPC, ...}

Functions

```
step ::
"'state ⇒ 'action ⇒ 'state"

ia ::
"'dom ⇒ 'dom ⇒ bool"

dom ::
"'action ⇒ 'dom"
```

Run

```
run :: "'state ⇒ 'action list ⇒ 'state"
where
"run s [] = s"
"run s (a#α) = run (step s a) α"
```

Purged run

```
purged_run :: "'state ⇒ 'action list ⇒ 'dom ⇒ 'state"
where
"purged_run s [] u = s"
"purged_run s (a#α) u =
      if ia (dom a) u
          then purged_run (step s a) α
          else purged_run s α"
```

Run

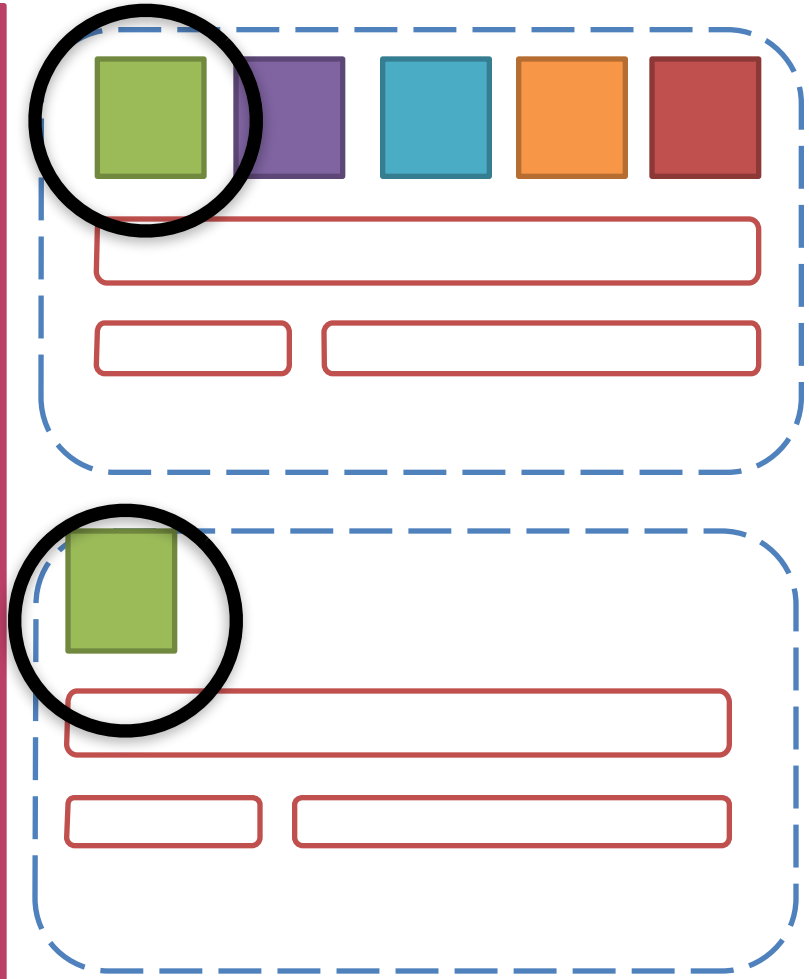
```
run :: "'state ⇒ 'action list ⇒ 'state"
where
"run s [] = s"
"run s (a#α) = run (step s a) α"
```

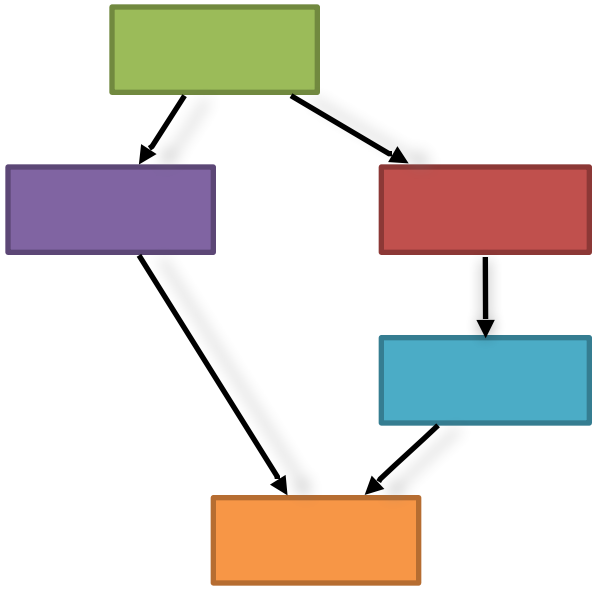
Security Definition

```
ia :: 'dom ⇒ 'dom ⇒ bool
step :: "'state ⇒ 'action ⇒ 'state"
dom :: "'action ⇒ 'dom"
```

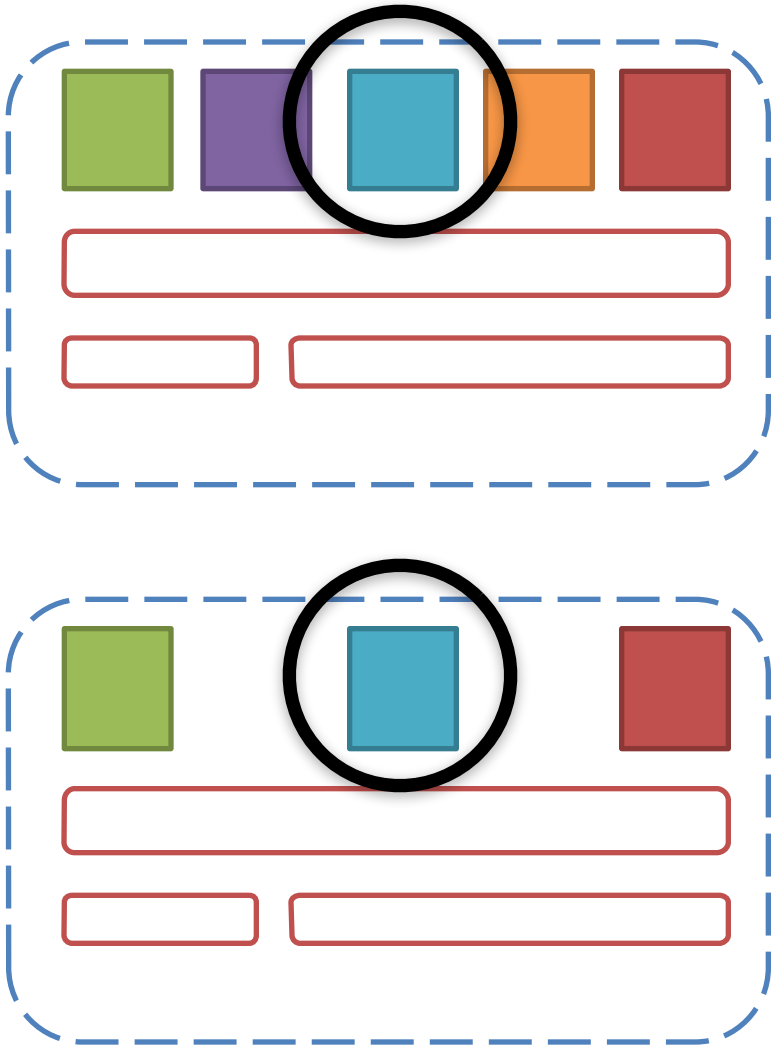
```
secure :: bool
where "secure ≡ ∀ α u.
                    u
    purged_run s0 α u = run s0 α"
```

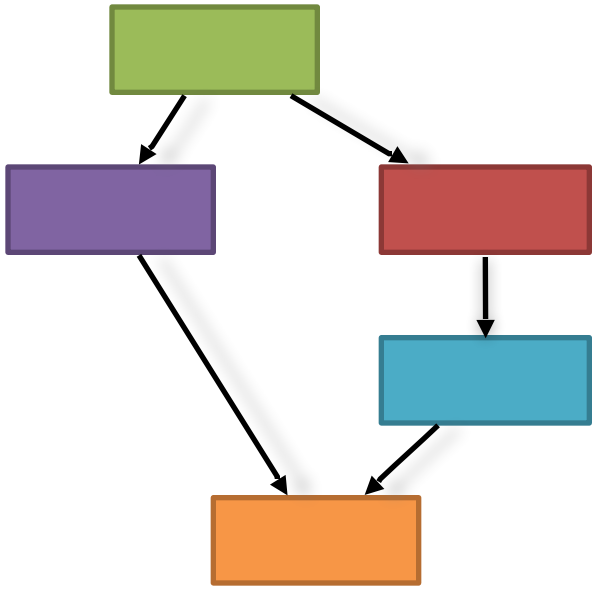
Security Policy





Security Policy





What's wrong with it?

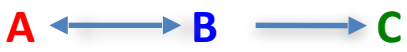
Rushby does not take into account:

1. Control
2. Interrupts
3. Aborted actions

$\alpha \in \{a_0, a_1, a_2, \ldots\}^*$

```
secure :: bool
where "secure ≡ ∀ α u.
    purged_run s0 α u = run s0 α"
```

Security Policy:



1. **Control**
2. Interrupts
3. Aborted actions

A = untrusted code that stops at terminate signal of B

B = downgrade information from A to C;
send terminate signal to A when dead

C = absorb downgraded information flow from A

Attack Surface

$\alpha \in \{a,b,c\}^* \; b \; \{c\}^*$

```
secure :: bool
where "secure ≡ ∀ α u.
    purged_run s0 α u = run s0 α"
```

1. Control
2. **Interrupts**
3. Aborted actions

CPY
IPC
SKIP

$\alpha \in \{CPY, IPC, SKIP\}^*$

```
secure :: bool
where "secure ≡ ∀ α u.
    purged_run s0 α u = run s0 α"
```

1. Control
2. **Interrupts**
3. Aborted actions

CPY = [CPY_INIT, CPY_CHECK, CPY_CPY]
IPC = [IPC_PREP, IPC_WAIT, IPC_BUF]
SKIP = [SKIP]

Attack Surface

α ∈{[CPY_INIT, CPY_CHECK, CPY_CPY], [IPC_PREP, IPC_WAIT, IPC_BUF], [SKIP]}*

```
secure :: bool
where "secure ≡ ∀ α u.
    purged_run s0 α u = run s0 α"
```

CPY = [CPY_INIT, CPY_CHECK, CPY_CPY]

We might not be able to prove security over traces such as:

α = CPY_CPY, CPY_CHECK, CPY_INIT, CPY_CHECK, CPY_CPY

CPY = [CPY_INIT, CPY_CHECK, CPY_CPY]

We need to be able to prove security over traces such as:

α = CPY_INIT, CPY_CHECK, CPY_INIT, CPY_CHECK

New Definition of Security

Security Policy:

secure1:

secure2:

CISK Summary

We have formalized:
1. A new generic model of separation kernels
2. A new definition of security
3. The proofs that any instance of the generic model is secure

The model includes:
1. Context switches + notion of currently active domain
2. Refinements from actions to action sequences
3. Interrupts
4. Possibility of aborting an action sequence

General conclusions
1. Model helps having a better understanding of the system
2. This is a win of using Theorem Proving rather than anything else

EURO-MILS CONTRACT N0: 318353

"The EURO-MILS project has received funding from the European Union's Seventh Framework Programme ([FP7/2007-2013]) under grant agreement number ICT-318353."

If you need further information, please contact the coordinator:

Technikon Forschungs- und Planungsgesellschaft mbH

Burgplatz 3a, 9500 Villach, AUSTRIA

Tel: +43 4242 233 55 Fax: +43 4242 233 55 77

E-Mail: coordination@euromils.eu