Modélisation et validation formelle d'architectures logicielles basées sur les patrons de sécurité

Fadi Obeid, Philippe Dhaussy

Univ. Bretagne Loire
Lab-STICC
UMR CNRS 6285
ENSTA-Bretagne, Brest.

philippe.dhaussy@ensta-bretagne.fr

fichier : valid_ArchiSecu_AFADL_<data>.ppt

Modélisation et validation formelle d'architectures logicielles sécurisées

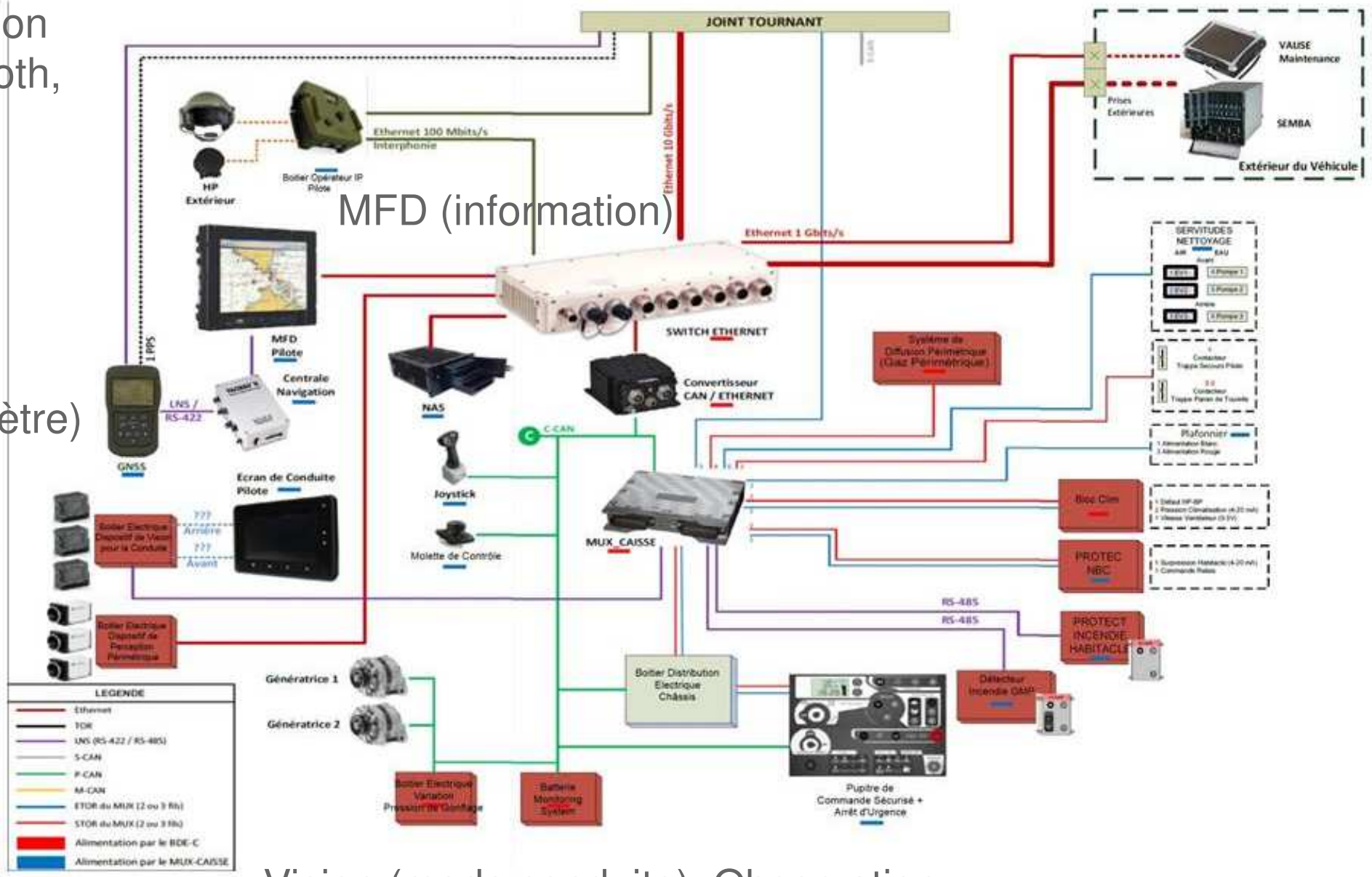
- Contexte, motivations
- Patterns de sécurité
- Formalisation
- Processus d'intégration dans une architecture et validation
- Perspectives

Architecture matérielle d'une vétronique (partielle)

communication (Wifi, Bluetooth, SIO).

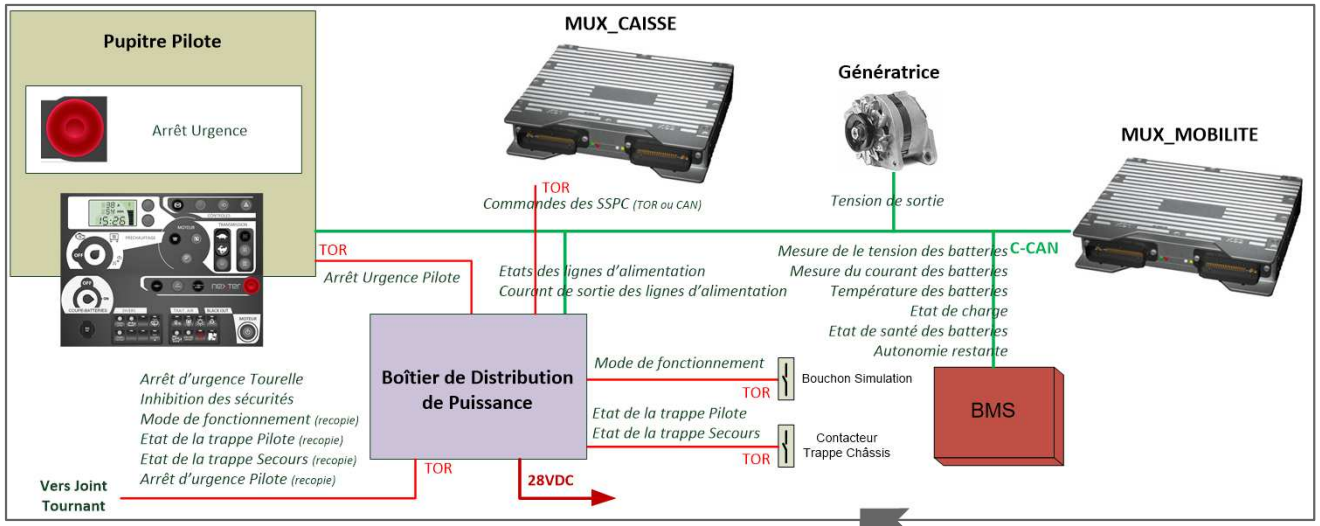
Localisation (GPS, odomètre)

Réseau CAN Equipement sur RS Capteurs / Effecteurs TOR



Vision (mode conduite), Observation

Analyse de sécurité centrée fonction : Un exemple

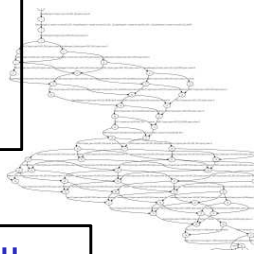


Evènements redoutés :

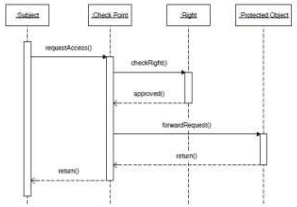
- Explosion des batteries due à un mauvais chargement
- Electrocution dû à une mauvaise connexion des équipements, défaut d'isolement.

Exigences de SdF et de sécurité

Analyse de sécurité



Simulation



Fonction Gestion de l'énergie

Analyse formelle (model-checking)

Scénarios d'attaque

Conception des protections

Propriétés de sécurité

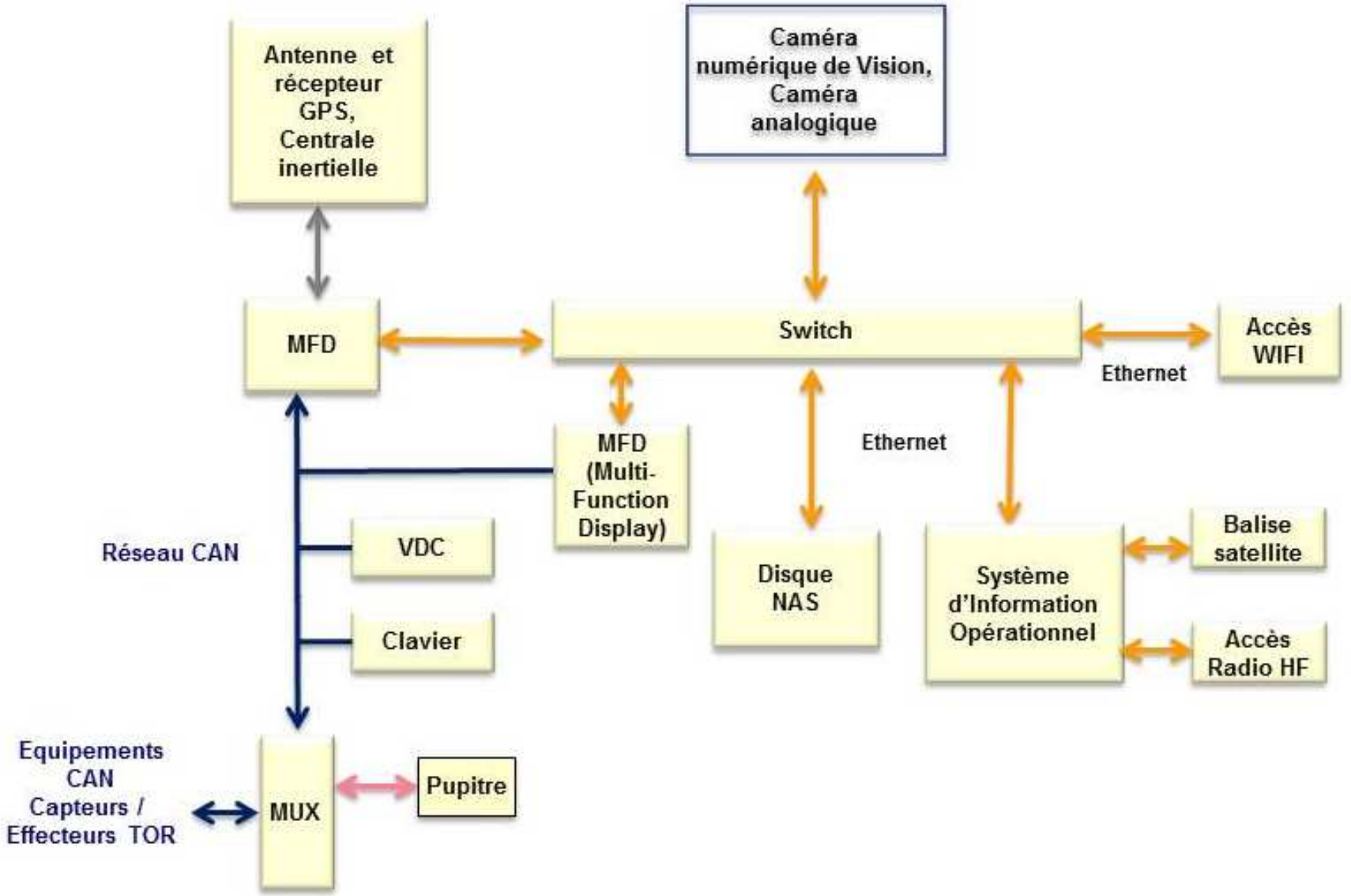
Intégrité : Pas d'altération ou de destruction (volontaire ou accidentelle) des données, lors de leur traitement, conservation ou transmission,.
Conservation du format permettant leur utilisation.

Confidentialité : non divulgation d'information aux entités non autorisées.

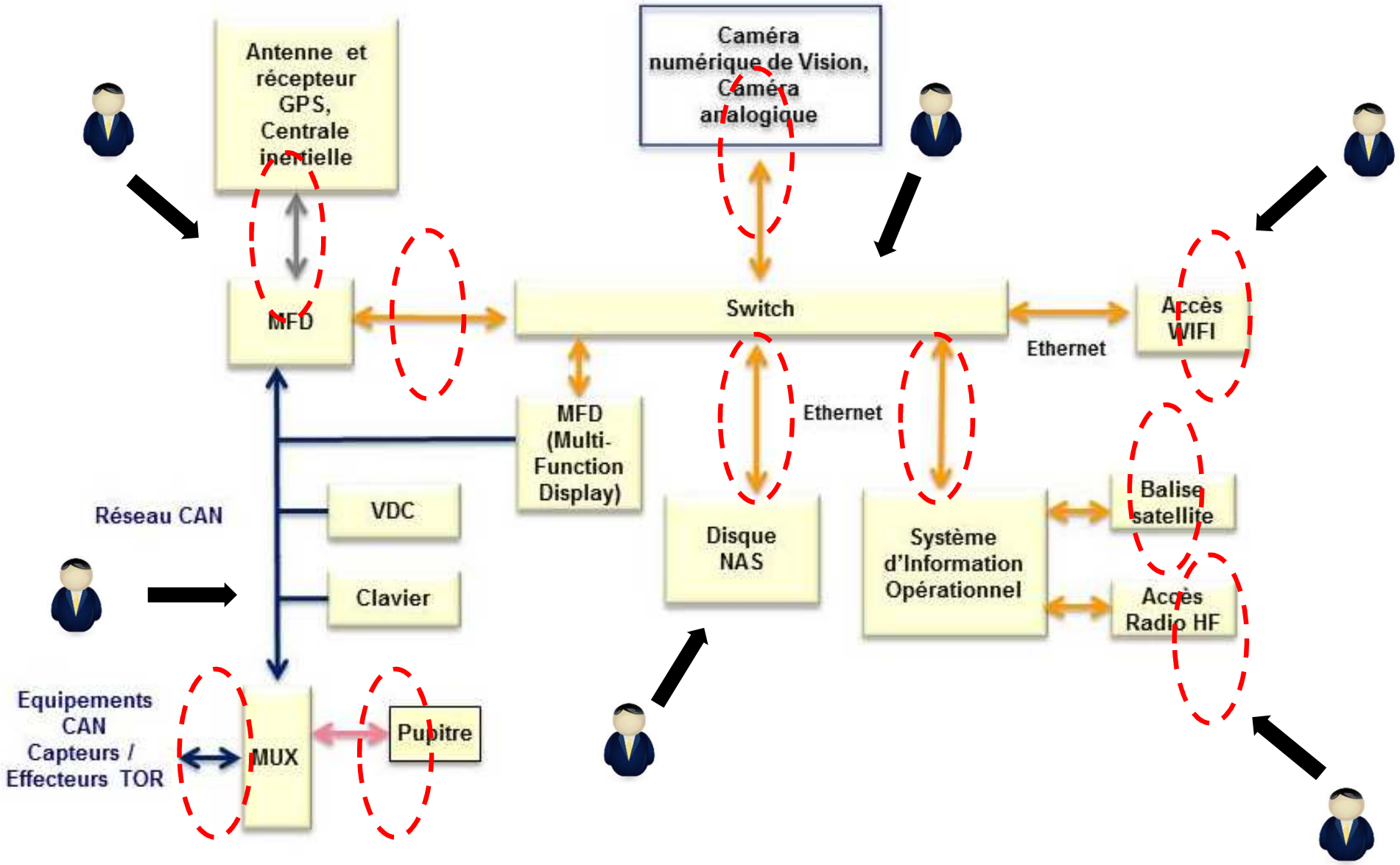
Disponibilité : « être prêt à l'utilisation »
Associé à la sûreté de fonctionnement . Liée au contexte et prend en compte les temps de réponses et les modèles de fautes (pannes franches, fautes d'omissions, temporelles, byzantines).

Autres propriétés : Intimité (privacy), Authenticité / non-répudiation, Responsabilité, Pérennité, Exclusivité, Protection de la propriété intellectuelle, … [TCSEC, 1985, ITSEC 1991, Bishop, 2003, Clemente 2010, Rouzaud Cornabas 2010].

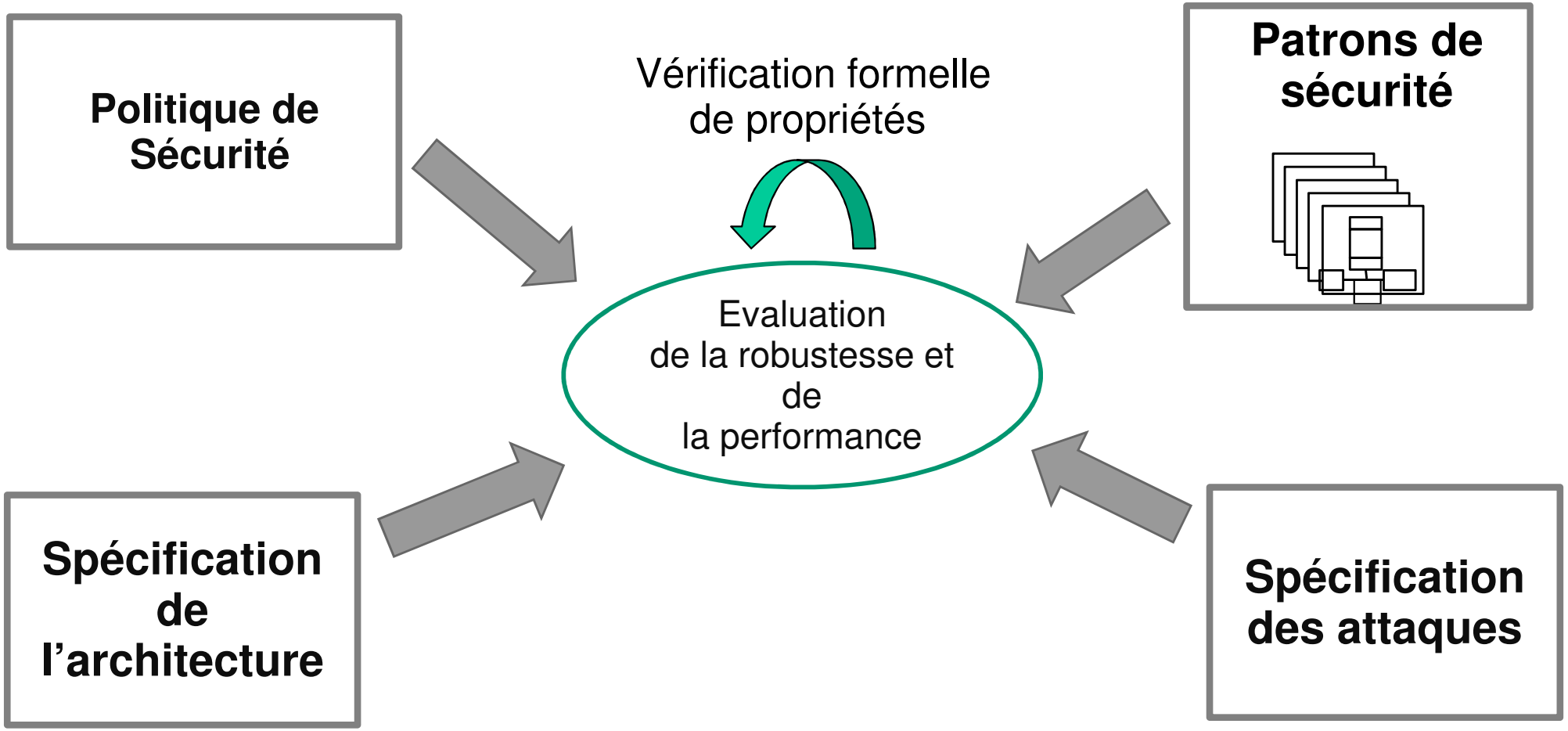
Modèle : abstraction de l'architecture



Modèle : abstraction de l'architecture



Processus de sécurisation et validation



Modélisation et validation formelle d'architectures logicielles sécurisées

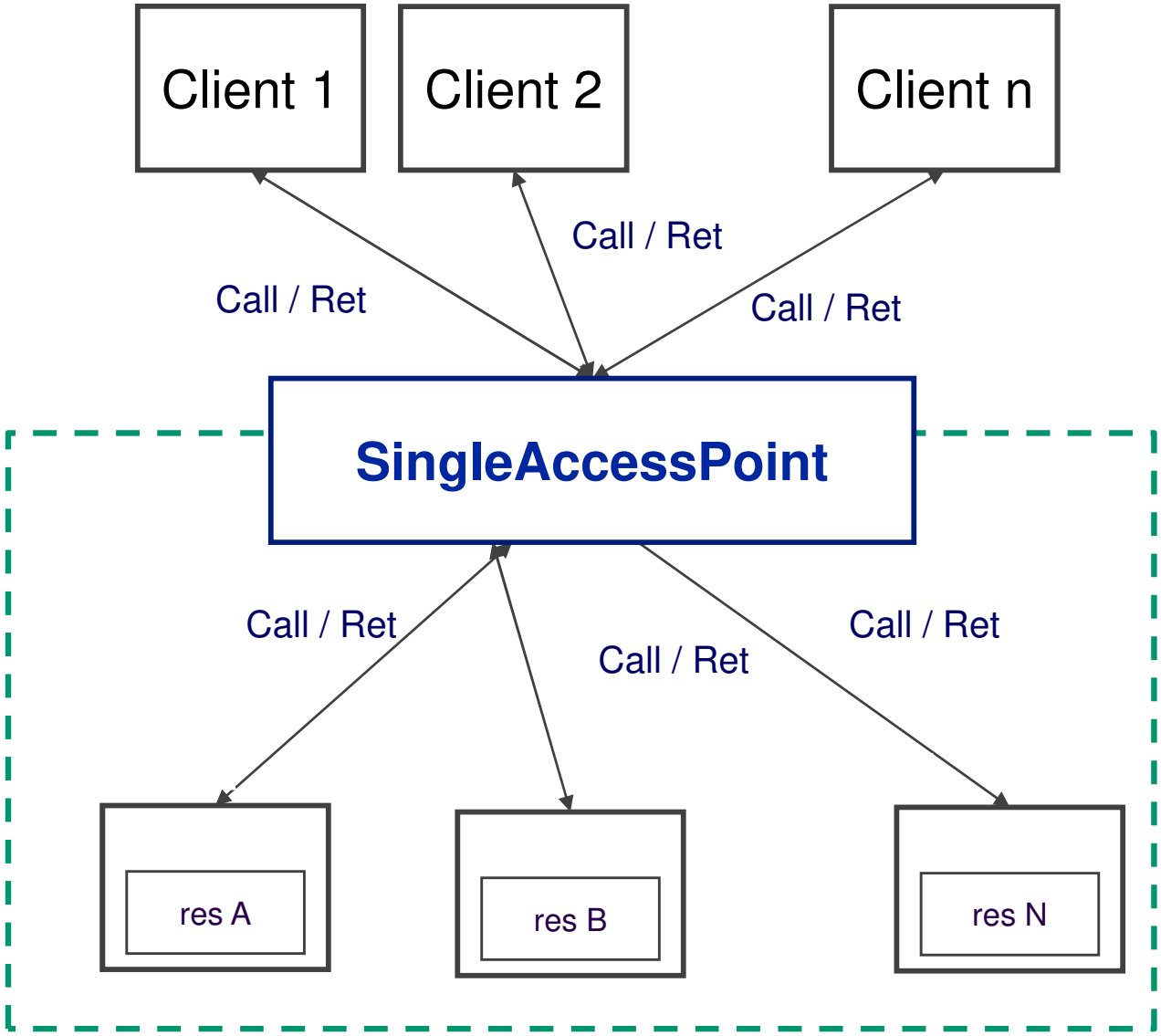
- Contexte, motivations
- Patterns de sécurité
- Formalisation
- Processus d'intégration dans une architecture et validation
- Perspectives

Patrons de sécurité

Solution générale pour des problèmes de sécurité répertoriés

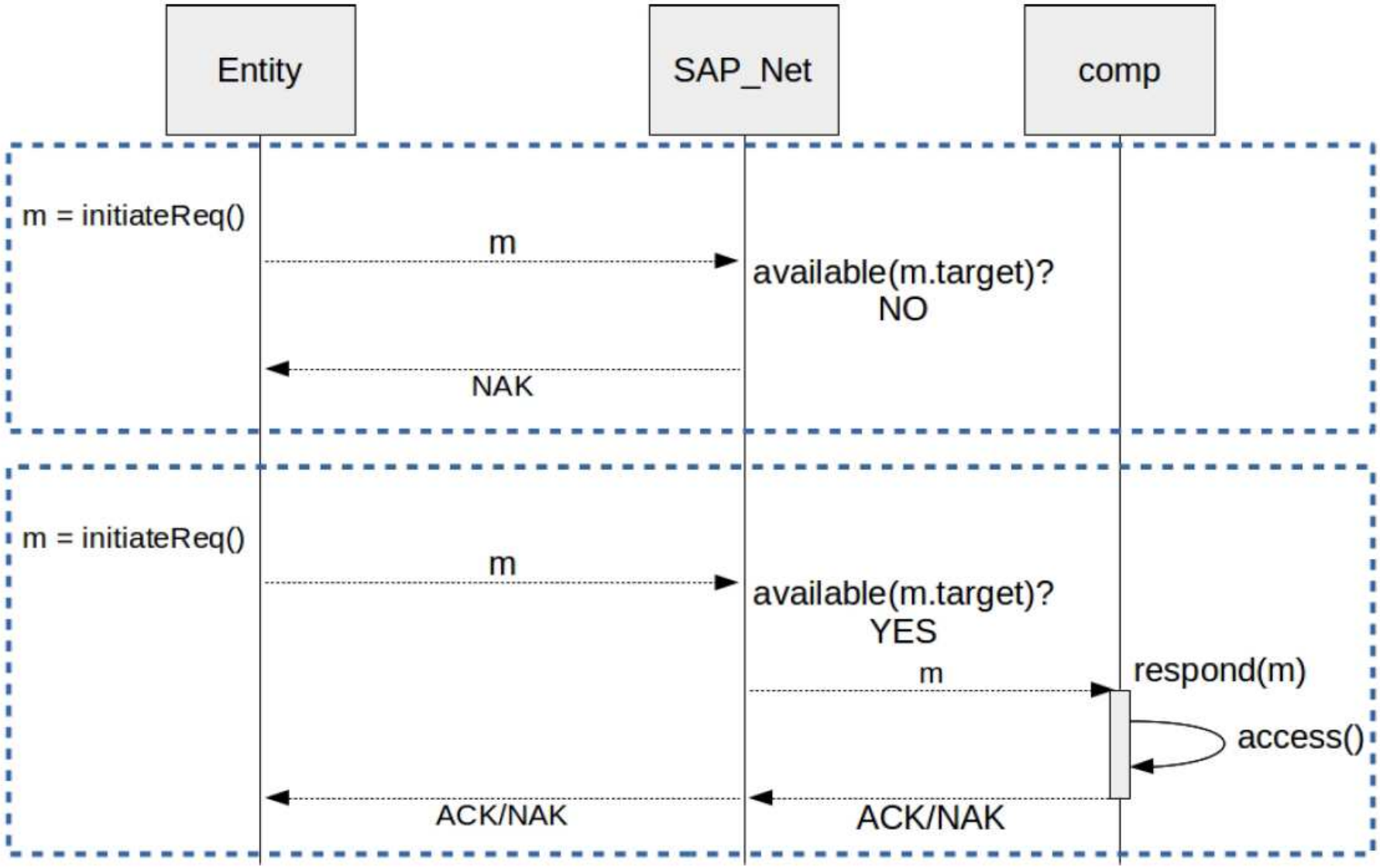
- [Yoder & Barcalow, Proc of 4th Pattern Language of programs, 1997]
- [Schumacher, Roedig, 2001]
- [Schumacher, Fernandez, Hybertson, Buschmann. Wiley & Sons, 2005]
- [Fernandez, 2006]
- [Heyman, Yskout, Scandariato, Joosen. Proc. of 3rd International Workshop on Software
- Engineering for Secure Systems, 2007]
- [Yoshioka, Washizaki, Maruyma. Progress in Informatics, 2008]
- https://en.wikipedia.org/wiki/Security_Patterns
- [Washizaki, Fernandez, Maruyama, Kubo,Yoshioka. Int Conf on Database and Expert Systems Applications, 2009.]
- [Hafiz, Adamczyk, Johnson. IEEE Software, 2007]
- Hafiz, Johnson. Tech report, 2006]
- http://www.munawarhafiz.com/securitypatterncatalog/index.php ...

Exemple : Single Access Point

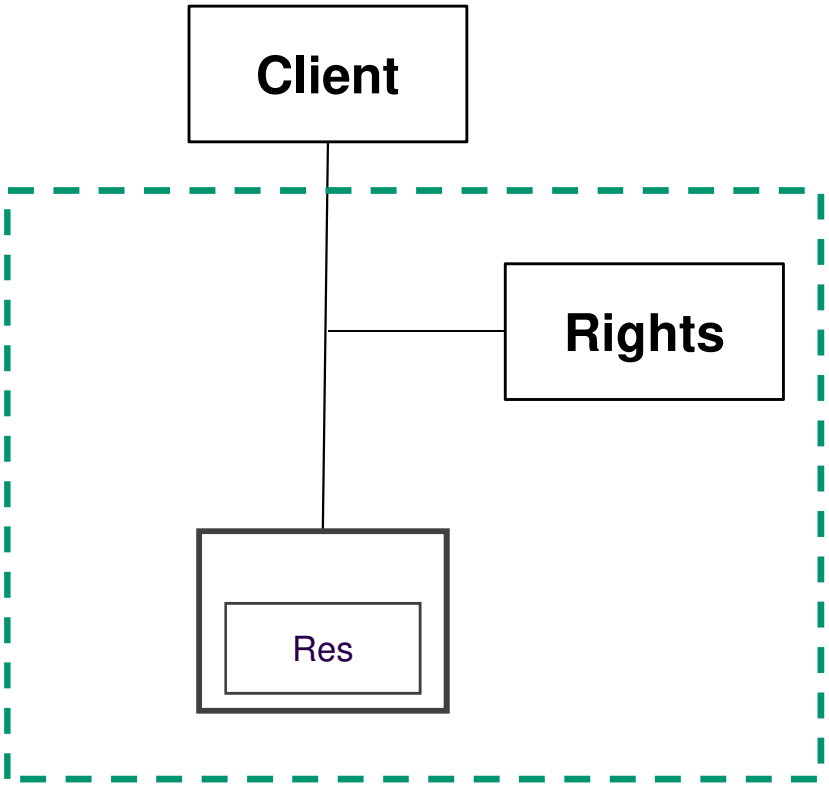


SingleAccess Point
Authorization_Information Access_log
access ()

Single Access Point : fonctionnalités



Exemple : Authorization



rights : Rights

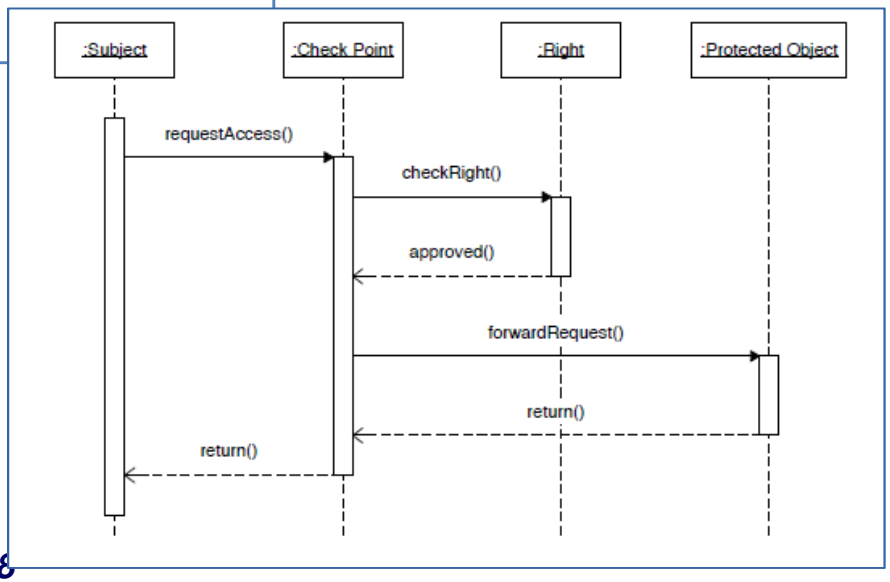
Access_type
Constraint
Transferable : bool
Protected_list
Allowed_list

checkRights ()
protected ()
allowed ()

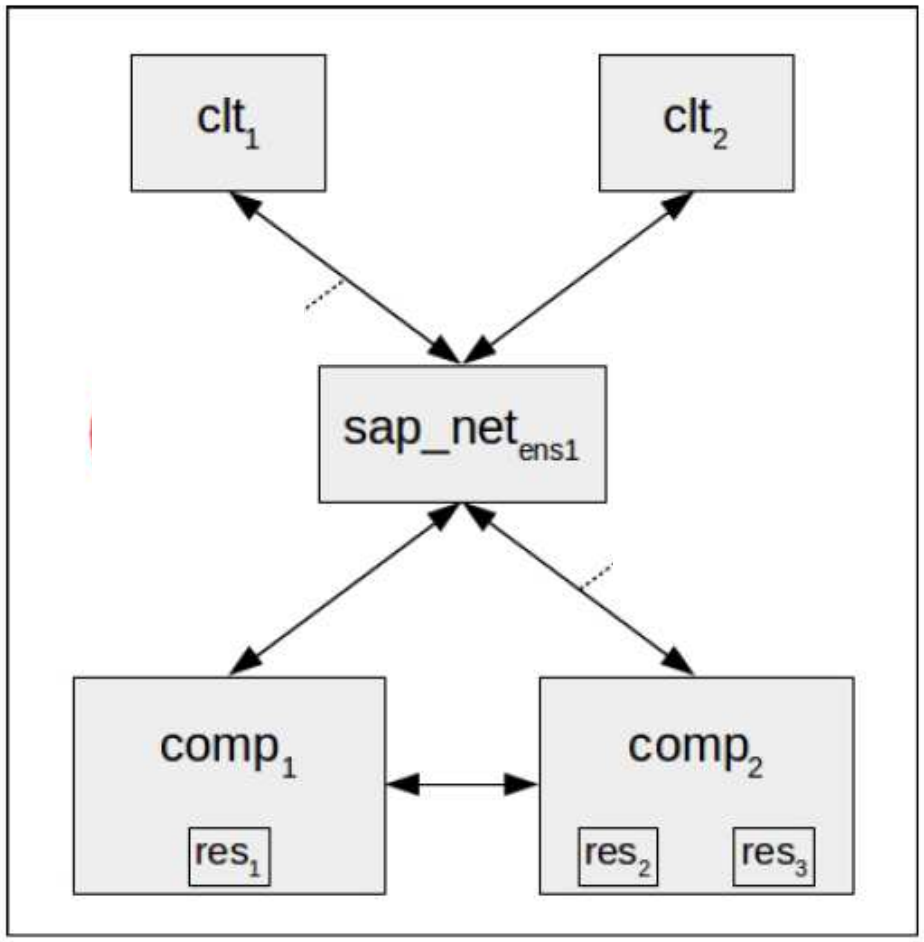
e : Entity

obj : type

access ()



SAP : exemple d'architecture

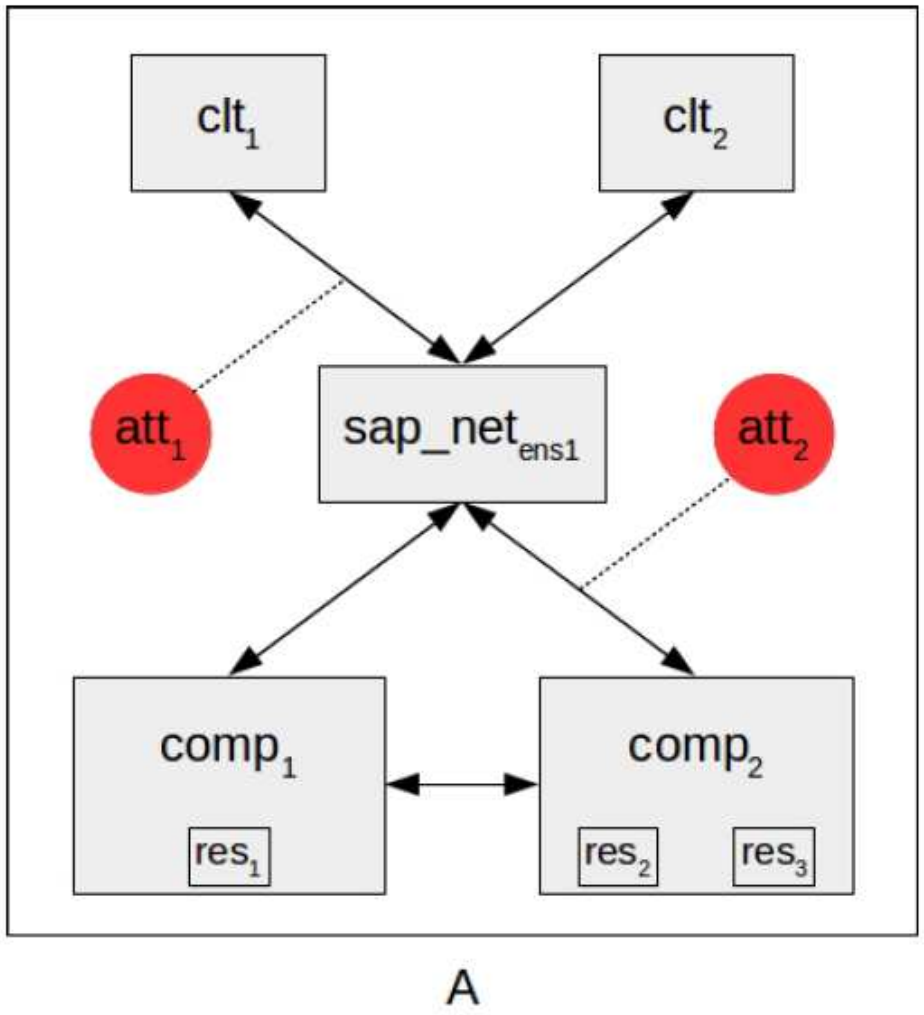


A

Hypothèses

- H1 : l'attaquant peut insérer des messages sur n'importe quel canal de communication.
- H2 : l'attaquant ne peut pas supprimer un message sur un canal.
- H3 : l'attaquant ne peut pas modifier un message signé par un SAP, ni un message ayant pour source une autre entité que lui.

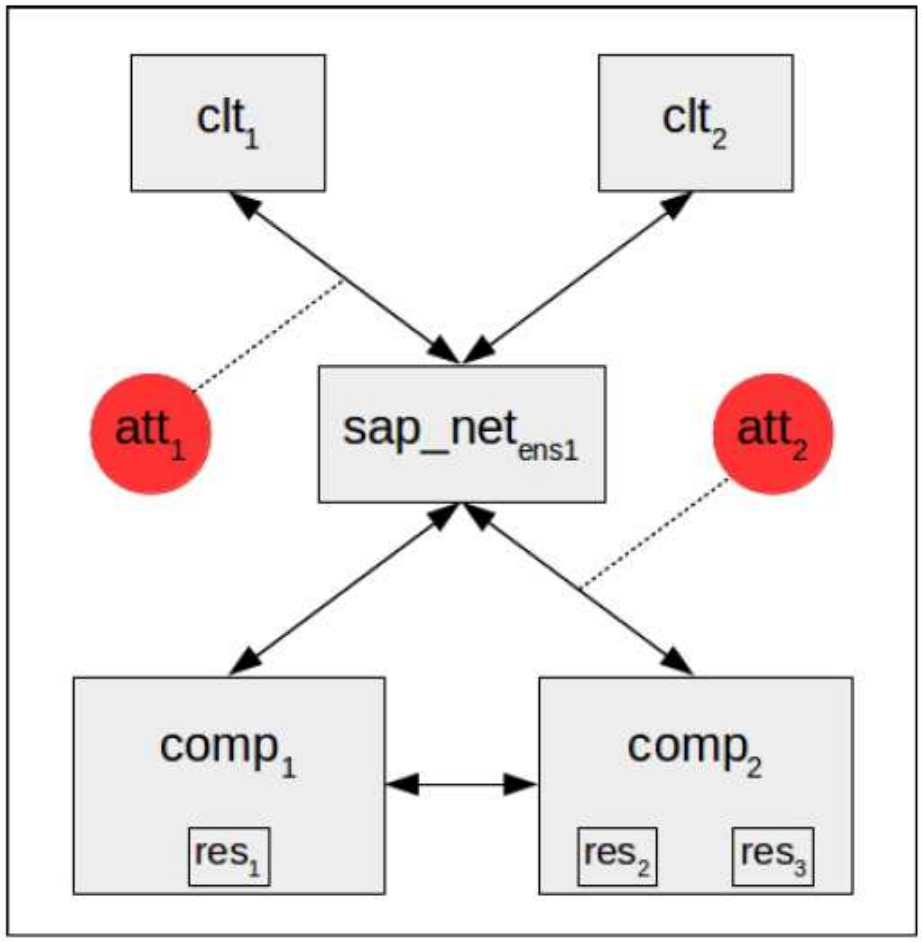
SAP : exemple d'architecture



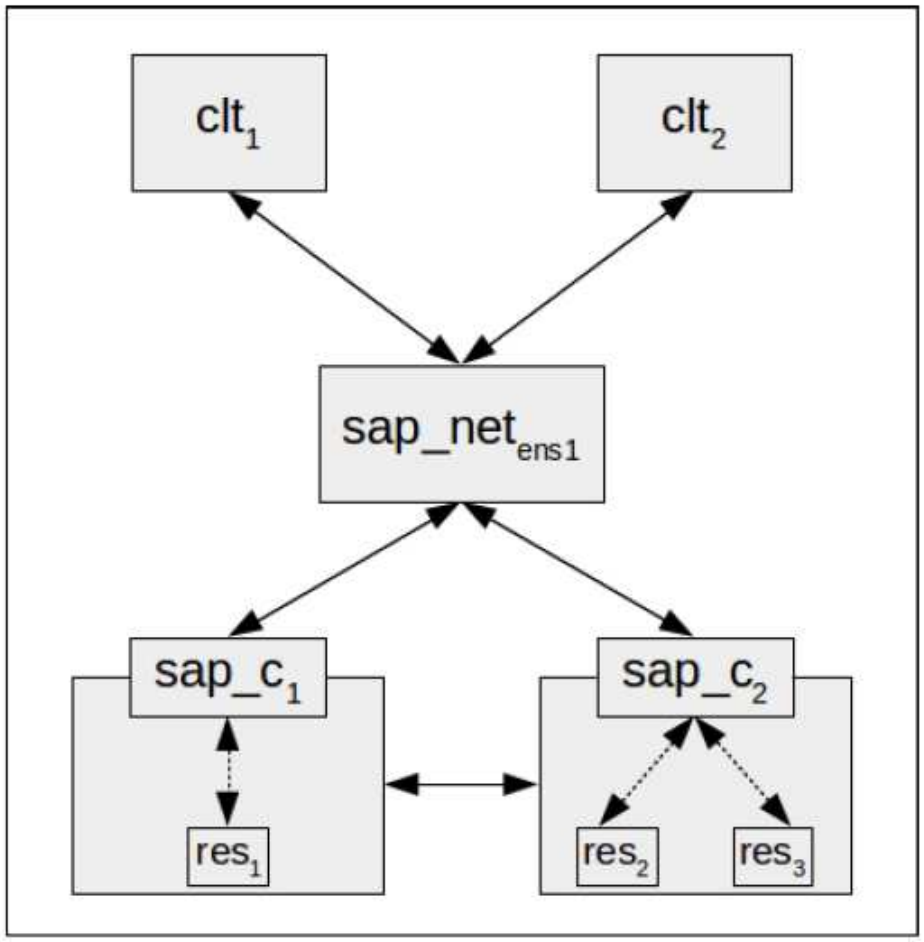
att1 : stopée par sap_net_ensi1

att2 : non stopée

SAP : exemple d'architecture



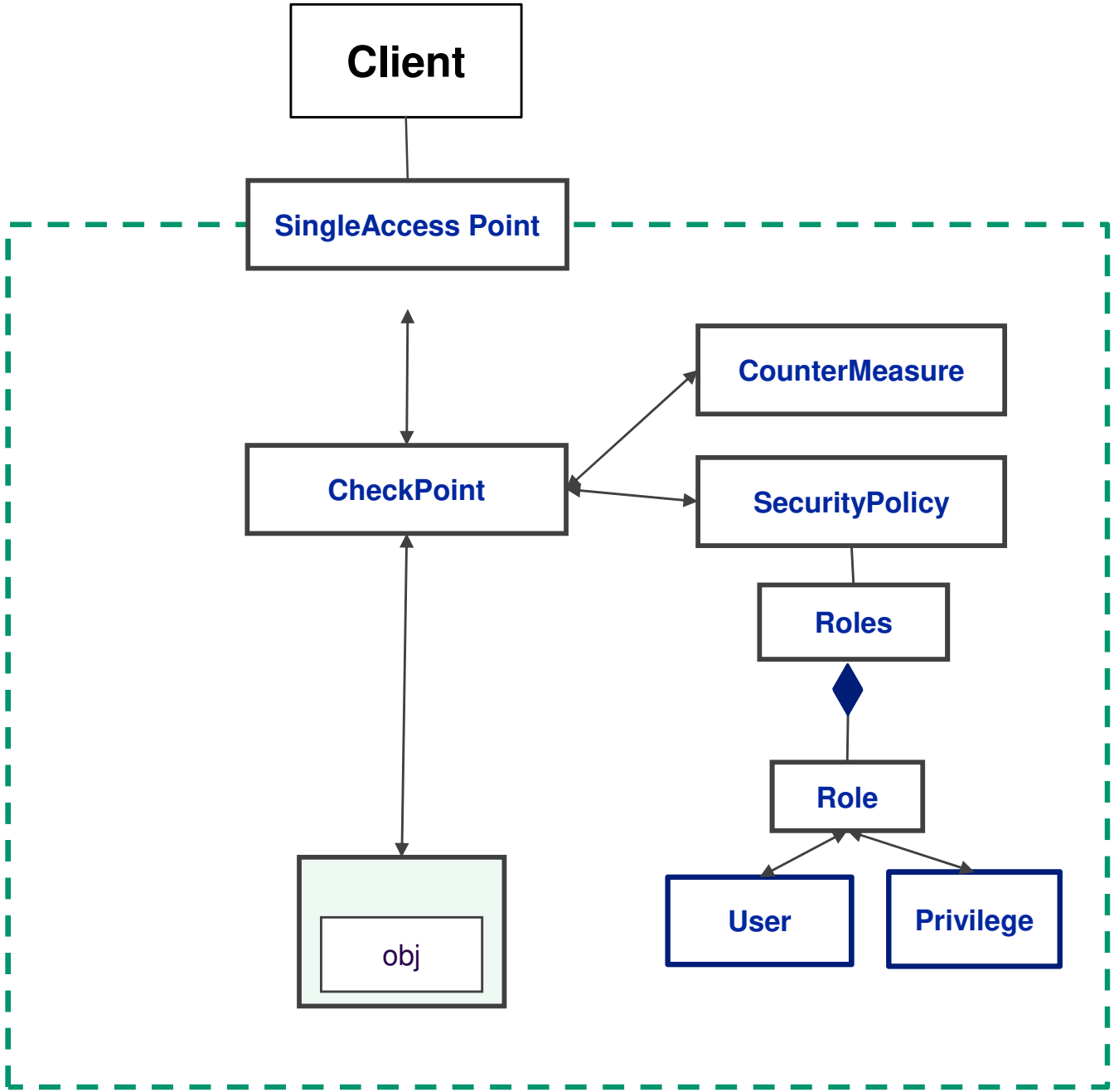
A



B

att2 : intégration de sap_c1 et sap_c2

Composition de patterns

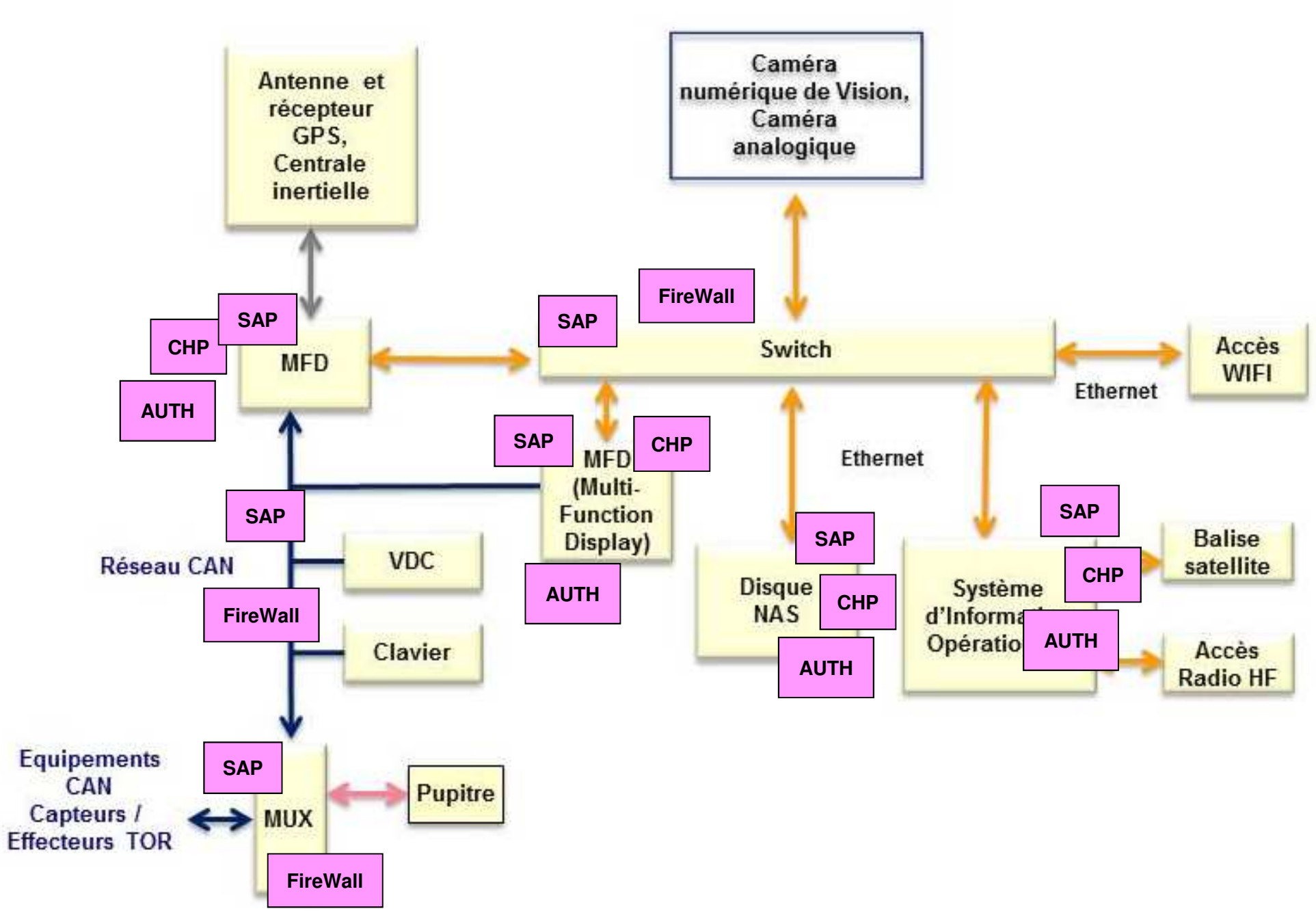


Role
privileges users
assignToUser () takeFromUser () addPrivilege () removePrivilege ()

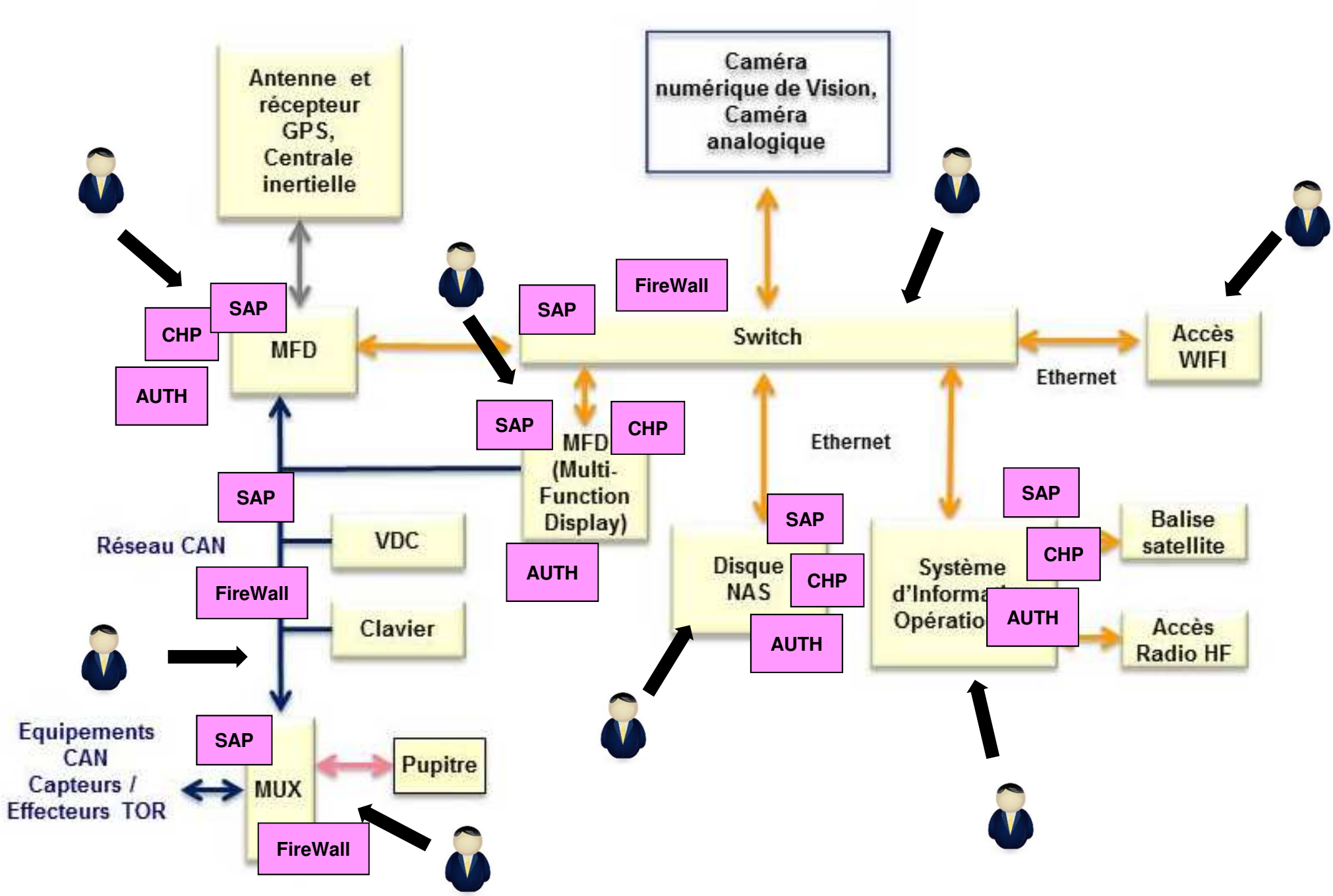
User
create () delete ()

Privilege
create () delete ()

Sécurisation de l'architecture



Etude de comportement face aux attaques



Questions de recherche abordées

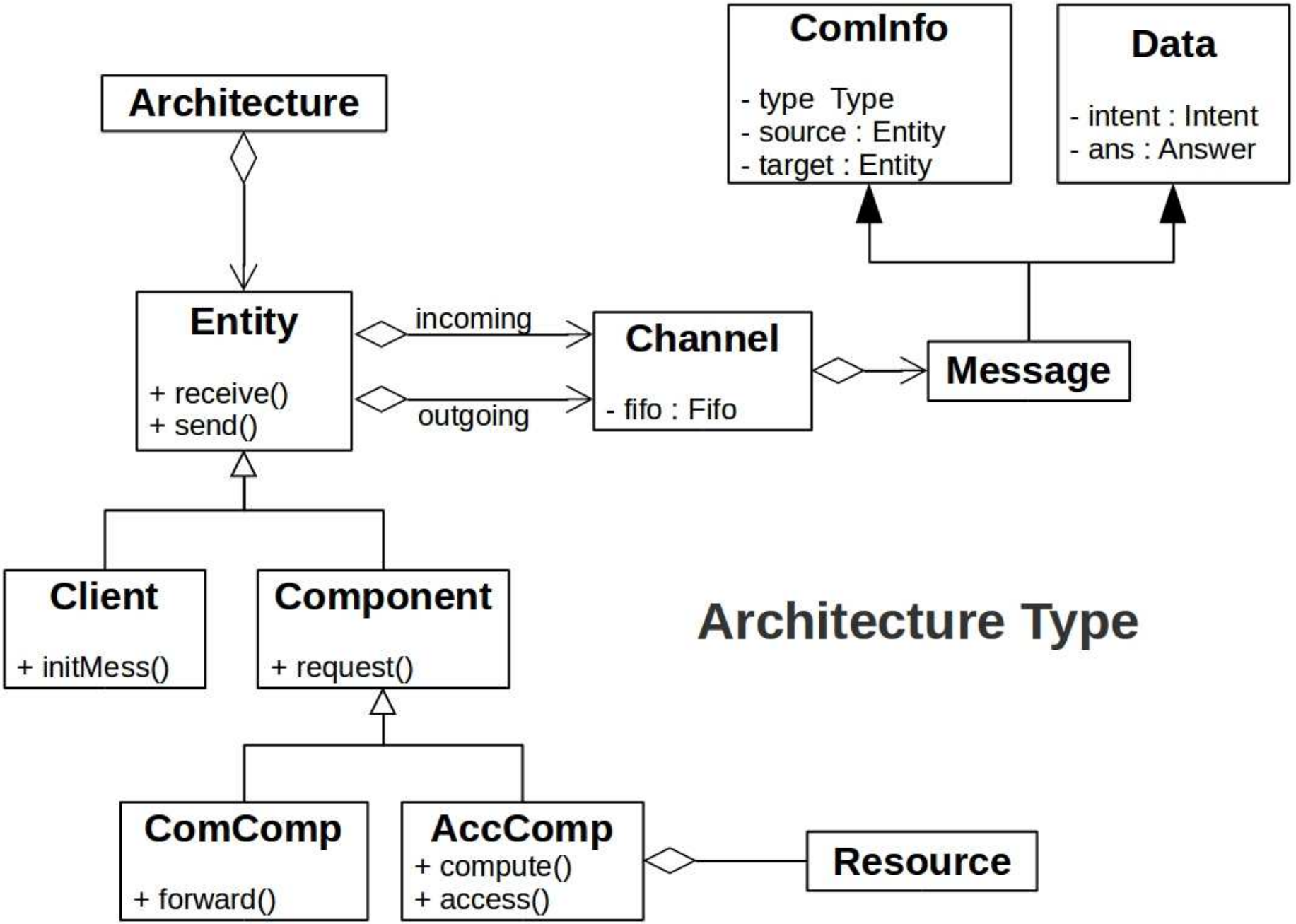
[Thèse Fadi Obeid, Lab-STICC, Ensta Bretagne, mai 2018]

1. Comment spécifier formellement les patrons de sécurité (conformance avec la politique de sécurité souhaitée) ?
2. Comment les intégrer dans un modèle d'architecture (composition) ?
3. Comment valider le modèle résultant sécurisé (vérifier des propriétés ?)

Modélisation et validation formelle d'architectures logicielles sécurisées

- Contexte, motivations
- Patterns de sécurité
- Formalisation
- Processus d'intégration dans une architecture et validation
- Perspectives

Modèles d'architecture



Formalisation des propriétés de sécurité (SAP)

Confidentialité :

Tout message échangé en interne d'un ensemble protégé de composants ne doit pas être vu à l'extérieur de cet ensemble.

prt_sap_net_4 :

$$\forall\, m \in \mathcal{M}ess,\ \forall\, e \in \mathcal{E}nt,\ \forall\, c_s \in \mathcal{S}ap\_Net,$$
$$\Box\, [evt\_receive\ (e, m) \wedge$$
$$(m.comInfo.source \in c_s.subs \wedge m.comInfo.target \in c_s.subs) \Rightarrow\ e \in c_s.subs] \qquad (3.10)$$

Formalisation des propriétés de sécurité (SAP)

Authenticité :

Tout message, provenant de l'extérieur d'un ensemble de composants protégés par un SAP, doit être contrôlé avant d'être transmis aux composants internes à l'ensemble.

$$\begin{aligned}&\textbf{prt\_sap\_net\_1.a}:\\&\forall\, m \in \mathcal{M}ess,\ \forall\, c_s \in \mathcal{S}ap\_Net,\ \forall\, c \in c_s.subs,\\&\Box\,[pre\_receive\ (c,\ m)\ \wedge\ m.comInfo.source \notin c_s.subs\ \Rightarrow\\&pre\_check\ (c_s,\ FrwReq(m))]\end{aligned} \tag{3.6}$$

Formalisation des propriétés de sécurité (SAP)

Disponibilité :

Tout requête de transfert de message par un SAP_NET, doit être contrôlée.

$$\begin{aligned}&\mathbf{prt\_sap\_net\_3 :}\\&\forall\, req \in \mathcal{F}rwReq,\ \forall\, c_s \in \mathcal{S}ap\_Net,\\&\Box\,[evt\_request\ (c_s,\ req) \Rightarrow\\&\Diamond\ evt\_check\ (c_s,\ req)]\end{aligned} \tag{3.9}$$

Autres patrons formalisés

- **CheckPoint**
- **Authorization**
- **Firewall**

Propriétés de sécurité (mécanisme de type SAP)

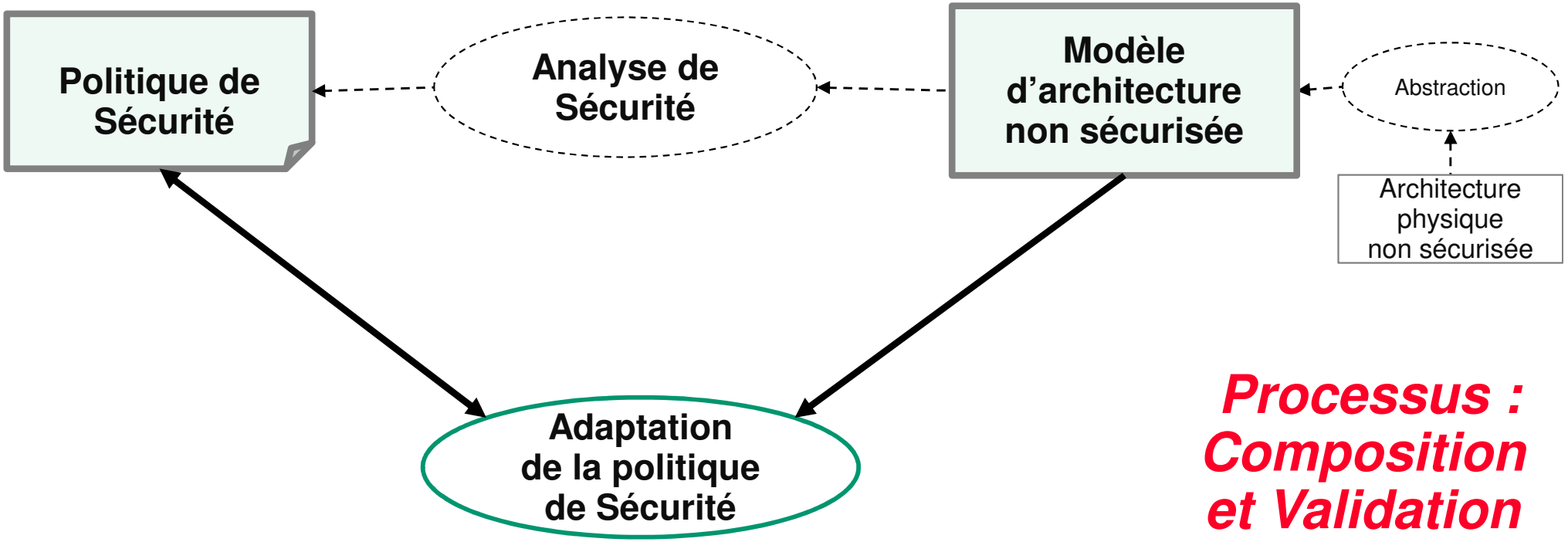
Table 5.7: Propriétés de sécurité vérifiées de type SAP.

Propriétés	Localisations	Types de propriétés
$prt\_sap\_1\_loc$	avec $loc \in \{gcs_i, net_i\ (i \in \{1,2\}),$ $plc_j\ (j \in \{1 \ldots 4\})\}$	Disponibilité (vivacité)
$prt\_sap\_net\_1.a\_loc$	avec $loc \in \{gcs_i, net_i\ (i \in \{1,2\})\}$	Authenticité (invariant)
$prt\_sap\_net\_1.b\_loc$	avec $loc \in \{gcs_i, net_i\ (i \in \{1,2\})\}$	Authenticité (invariant)
$prt\_sap\_net\_2\_loc$	avec $loc \in \{gcs_i, net_i\ (i \in \{1,2\})\}$	Disponibilité (vivacité)
$prt\_sap\_net\_3\_loc$	avec $loc \in \{gcs_i, net_i\ (i \in \{1,2\})\}$	Disponibilité (vivacité)
$prt\_sap\_net\_4\_loc$	avec $loc \in \{gcs_i, net_i\ (i \in \{1,2\})\}$	Confidentialité (invariant)
$prt\_sap\_c\_1\_loc$	avec $loc \in \{gcs_i\ (i \in \{1,2\}),$ $plc_j\ (j \in \{1 \ldots 4\})\}$	Authenticité (invariant)
$prt\_sap\_c\_2\_loc$	avec $loc \in \{gcs_i\ (i \in \{1,2\}),$ $plc_j\ (j \in \{1 \ldots 4\})\}$	Disponibilité (vivacité)
$prt\_sap\_c\_3\_loc$	avec $loc \in \{gcs_i\ (i \in \{1,2\}),$ $plc_j\ (j \in \{1 \ldots 4\})\}$	Disponibilité (vivacité)

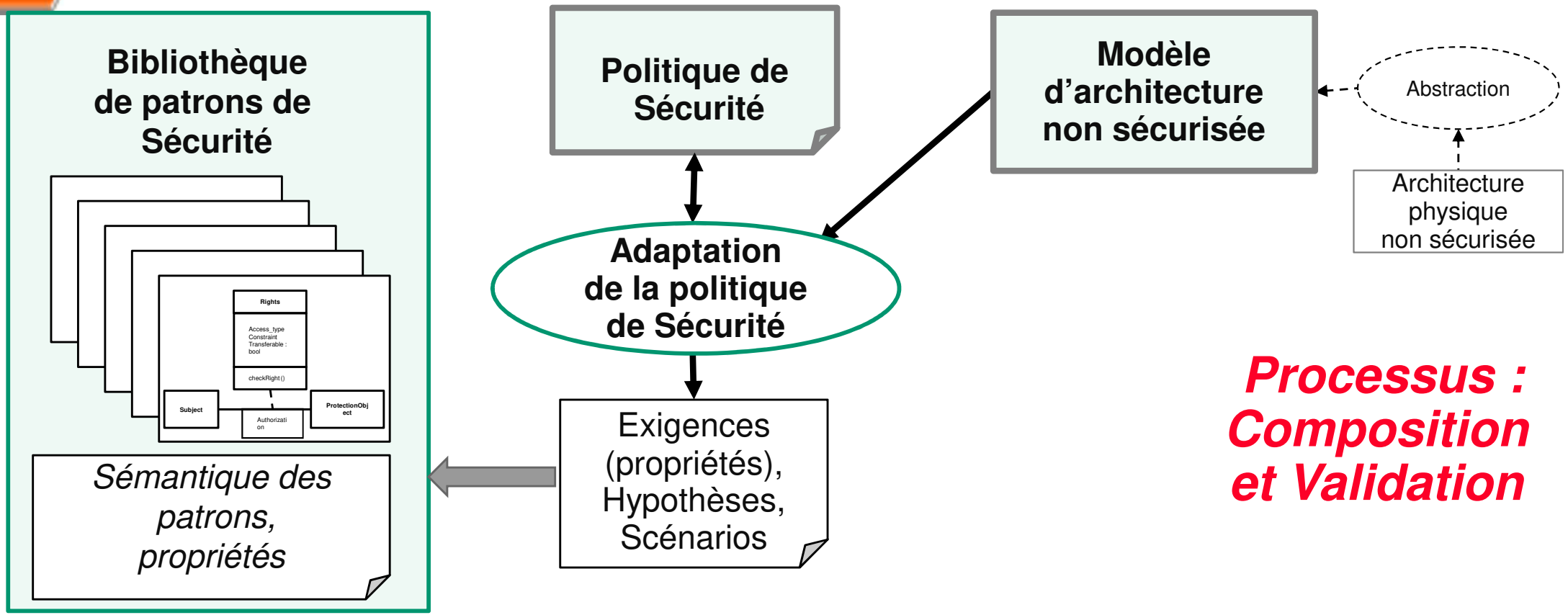
Modélisation et validation formelle d'architectures logicielles sécurisées

- Contexte, motivations
- Patterns de sécurité
- Formalisation
- Processus d'intégration dans une architecture et validation
- Perspectives

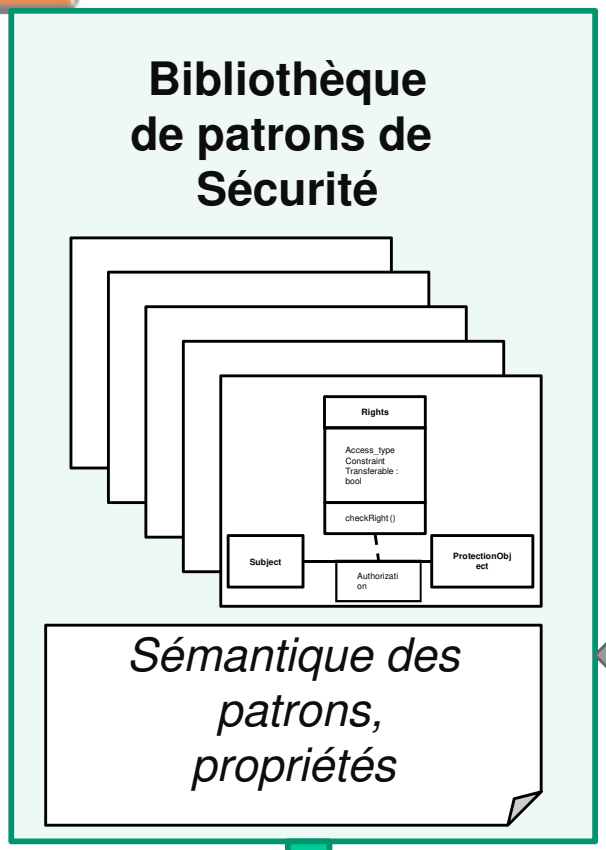
Processus : Composition et Validation

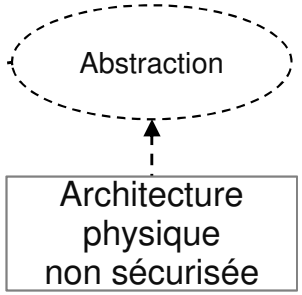


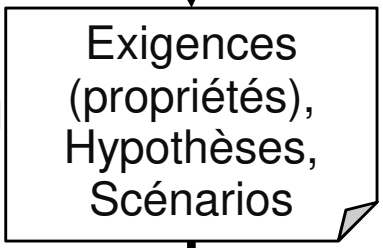
Processus : Composition et Validation

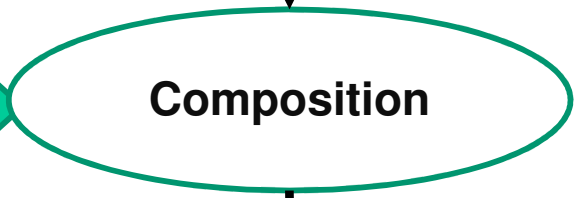


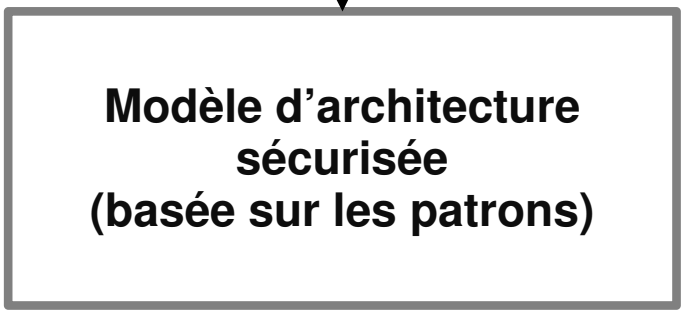
Processus : Composition et Validation



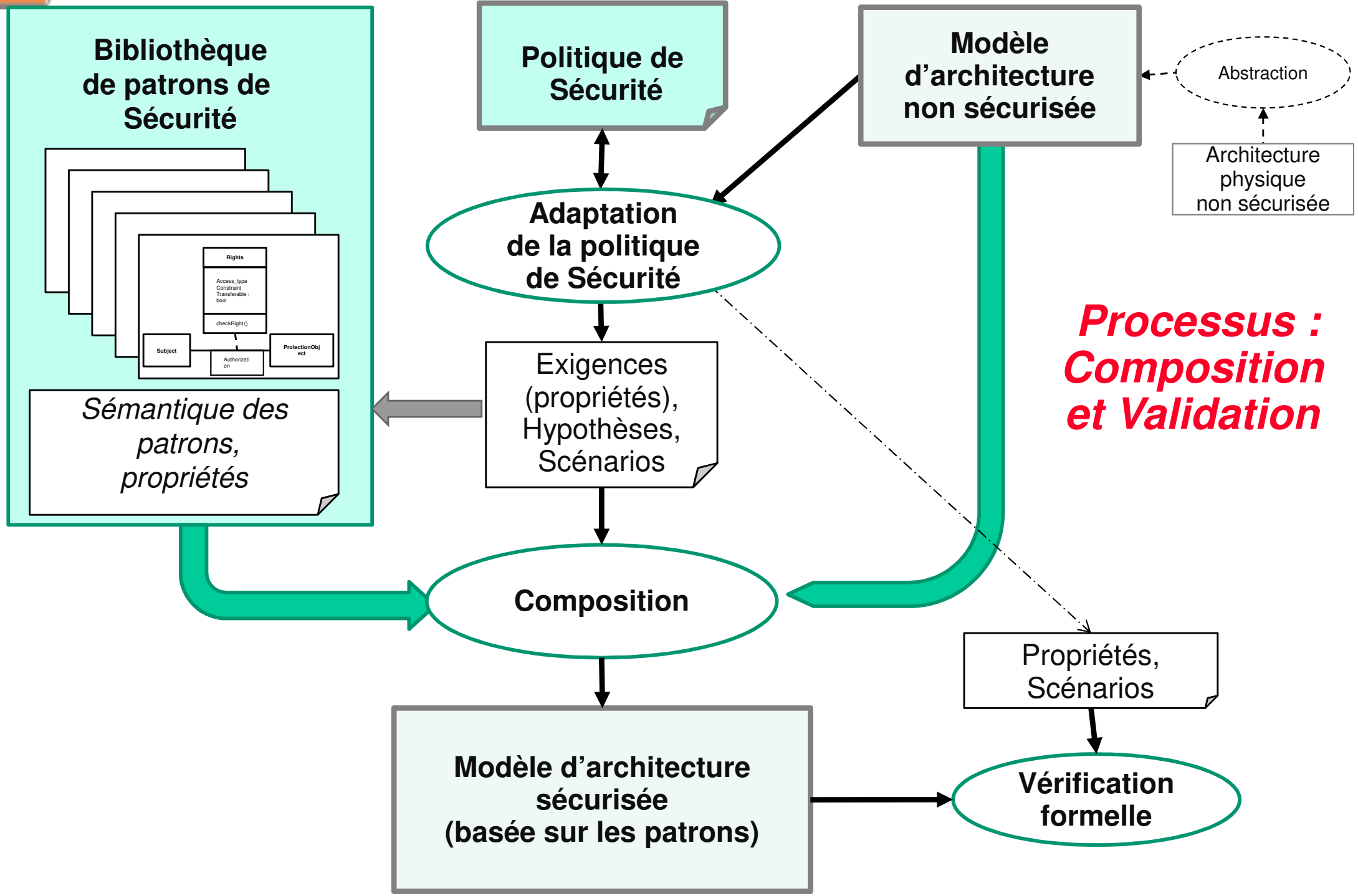




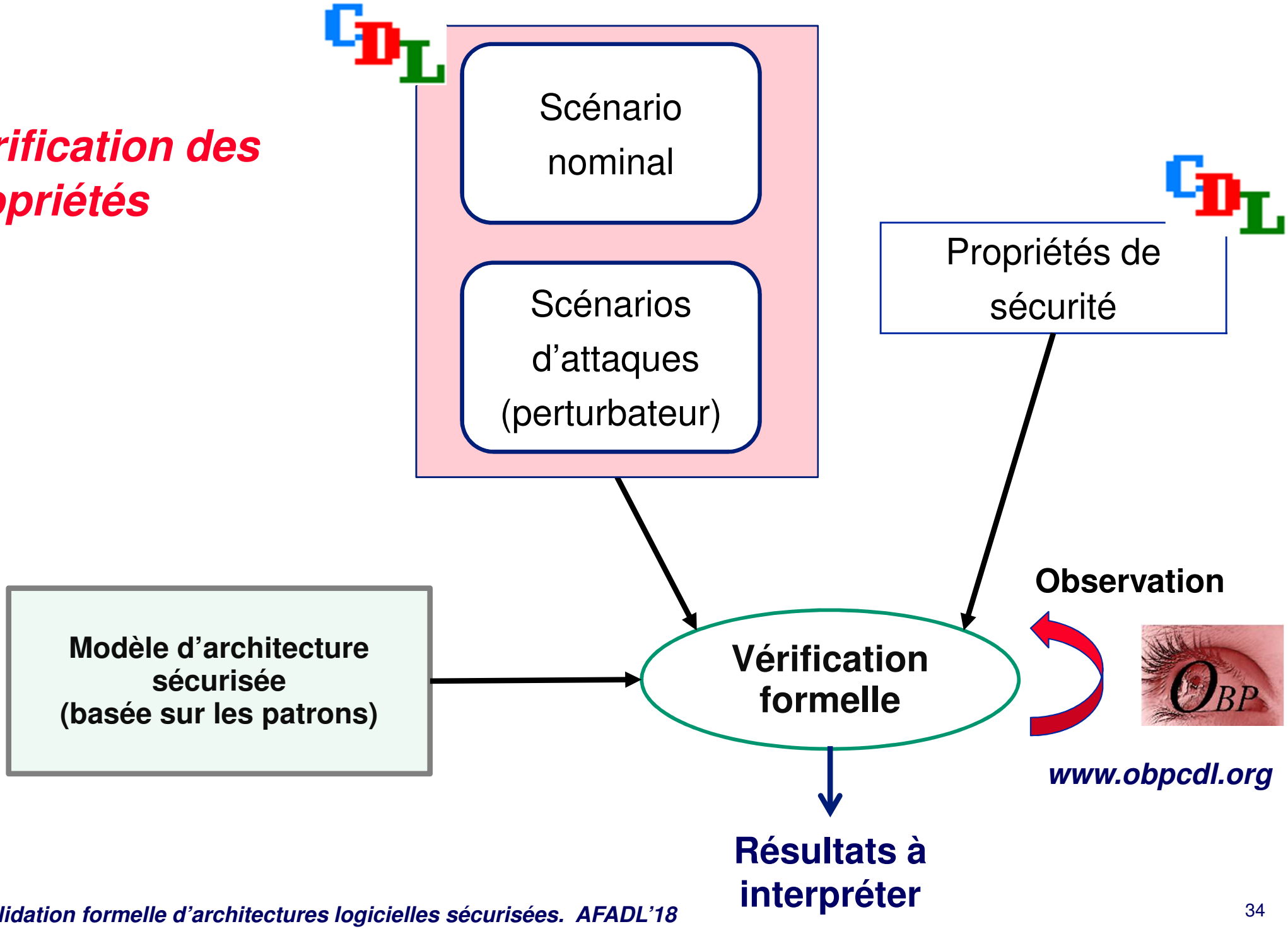




Processus : Composition et Validation


Bibliothèque de patrons de Sécurité
Rights
Access_type
Constraint
Transferable : bool
checkRight()
Subject
Authorization
ProtectionObject
Sémantique des patrons, propriétés
Politique de Sécurité
Modèle d'architecture non sécurisée
Abstraction
Architecture physique non sécurisée
Adaptation de la politique de Sécurité
Exigences (propriétés), Hypothèses, Scénarios
Processus : Composition et Validation
Composition
Propriétés, Scénarios
Modèle d'architecture sécurisée (basée sur les patrons)
Vérification formelle

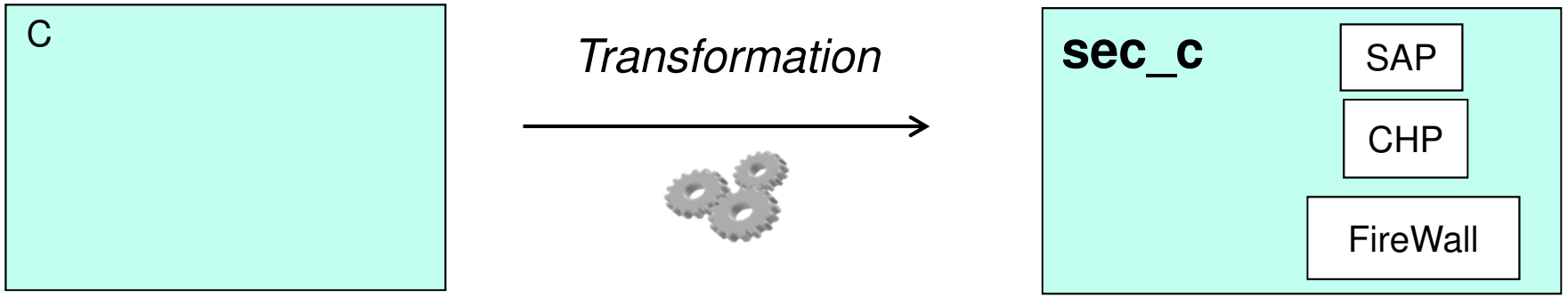
Vérification des propriétés



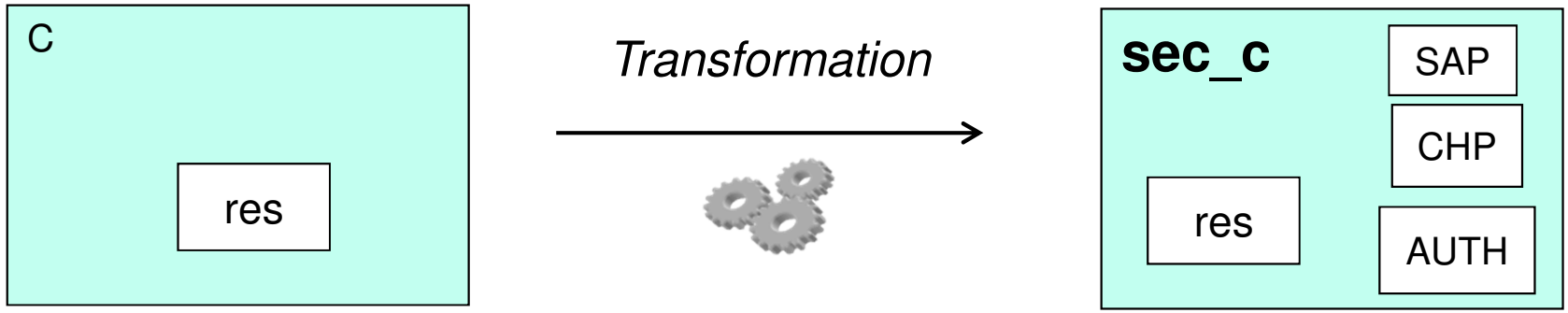
Une approche

Fonctionnalité de sécurité : intégrée dans un composant

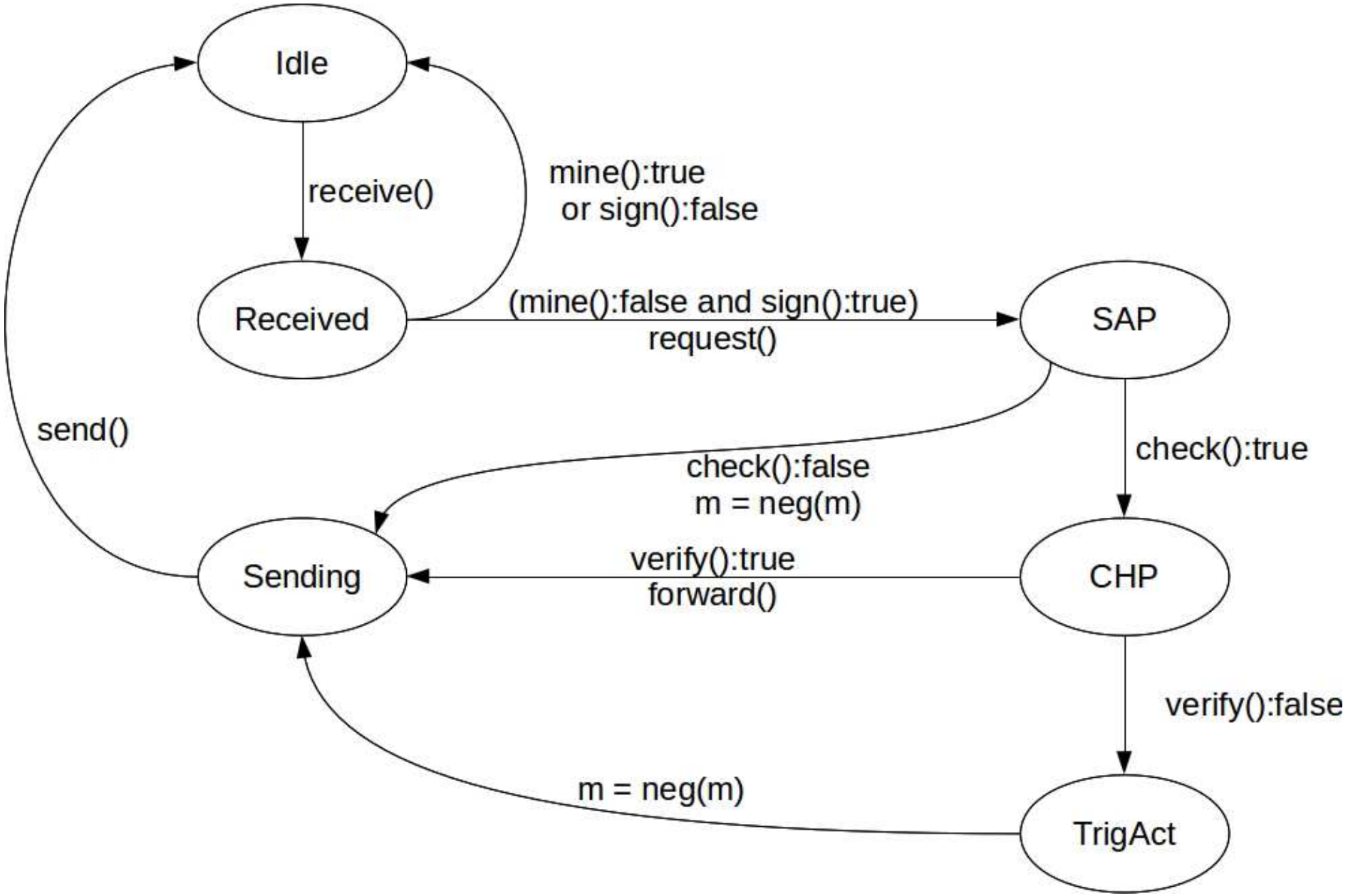
Type NET



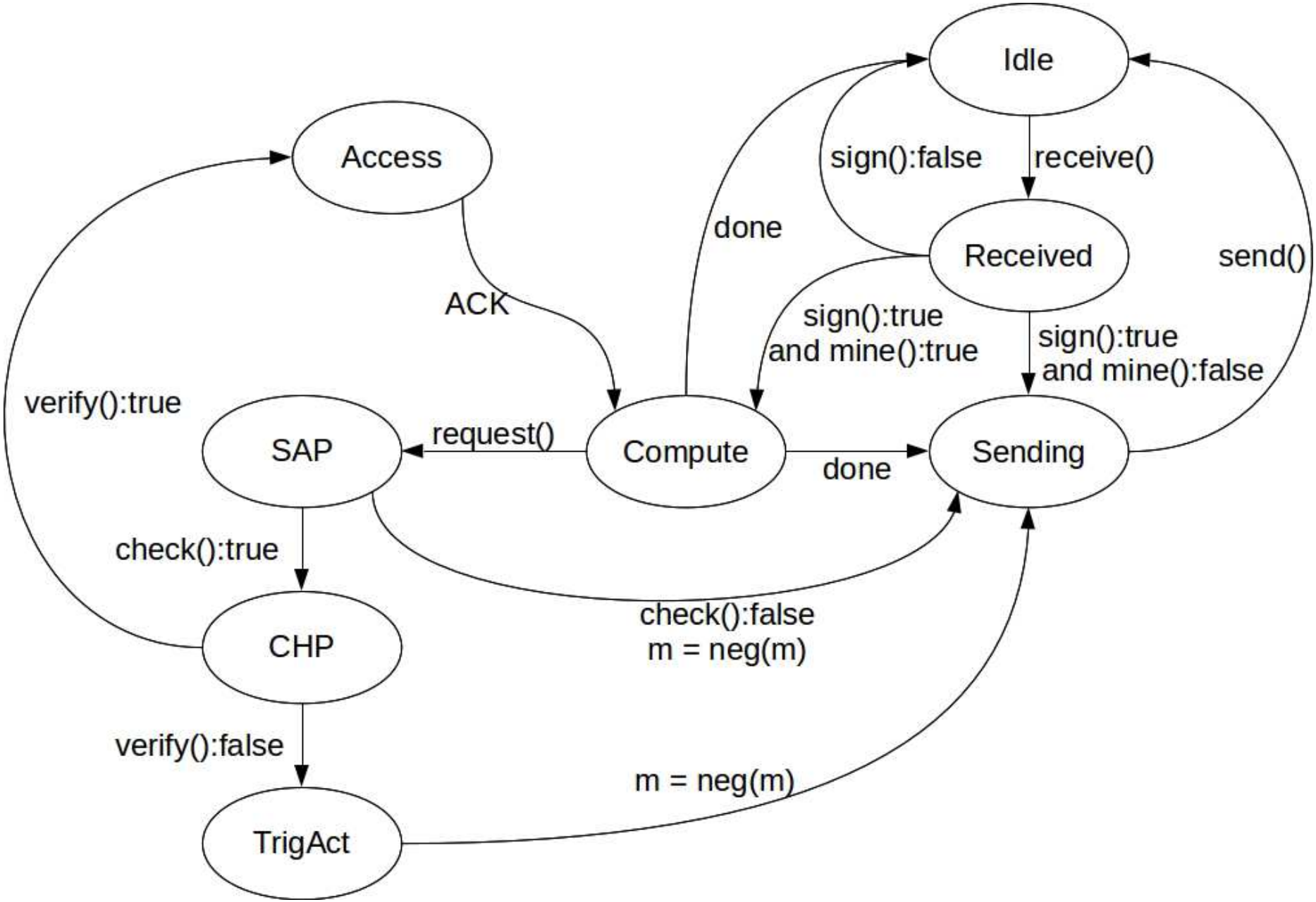
Type ACCESS



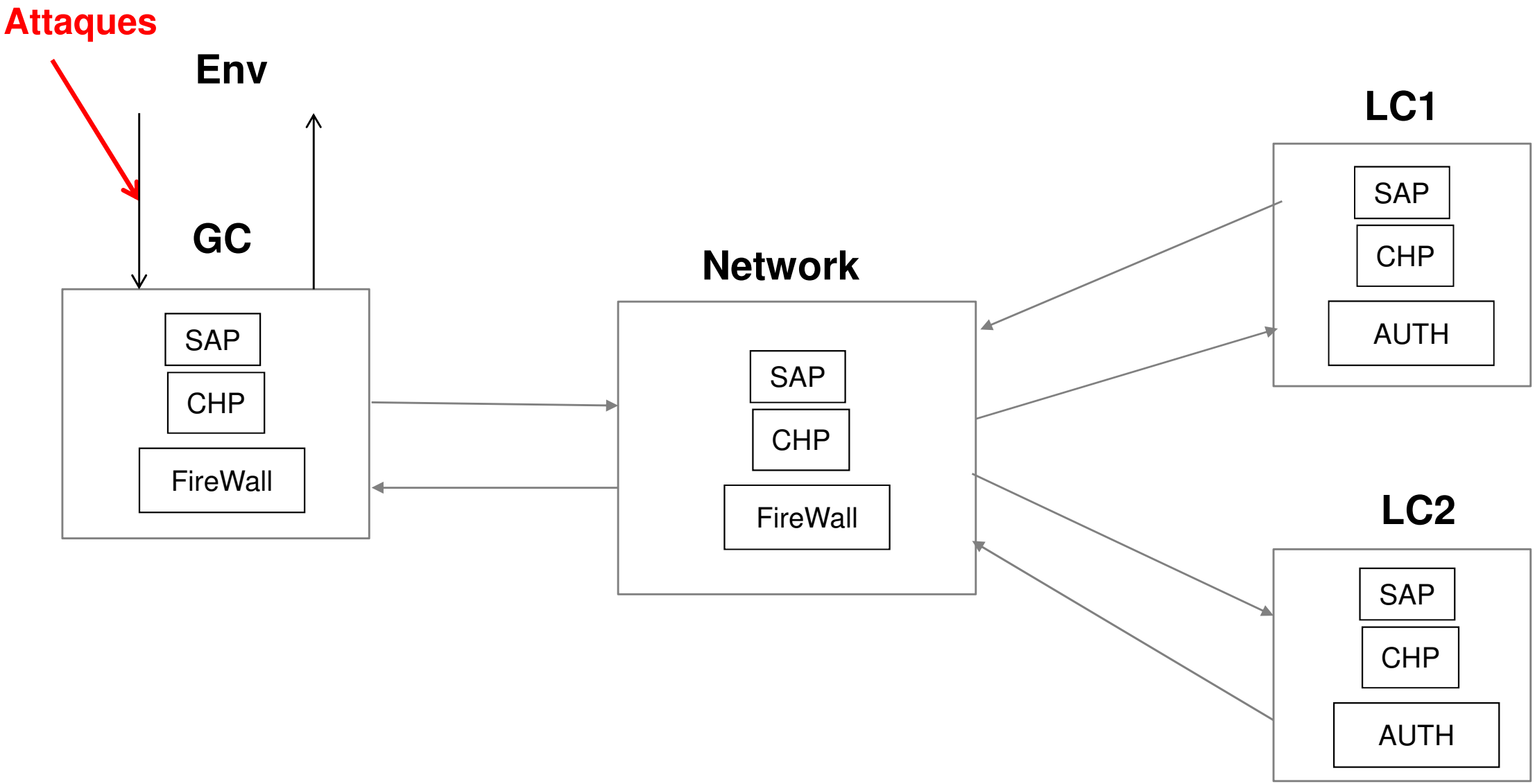
Approche : Automate d'une d'entité sécurisée : cas type NET



Approche : Automate d'une d'entité sécurisée : cas type ACCESS



Simulation OBP : mode attaque



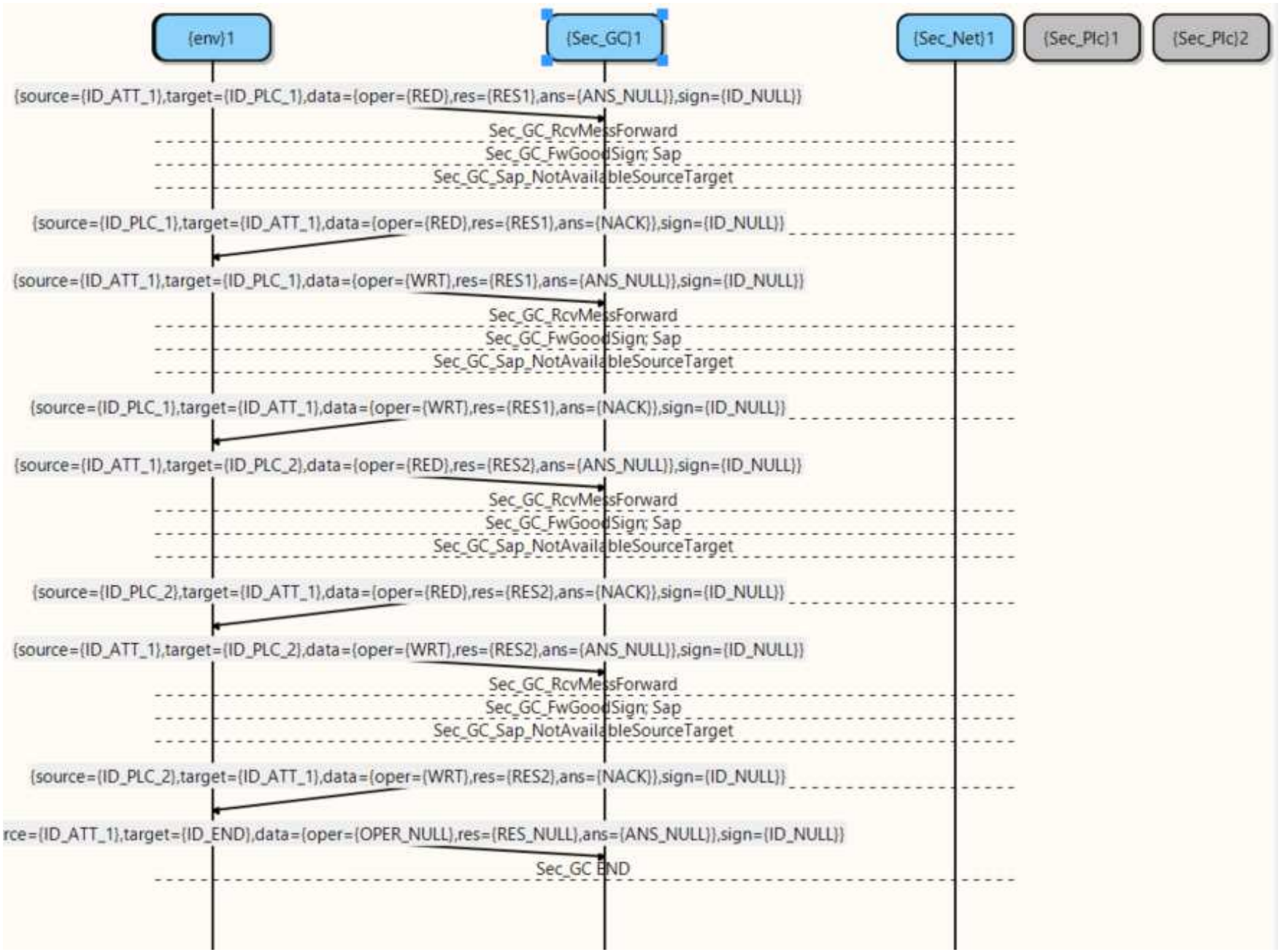
Simulation OBP : mode attaque

Attaque 1 →

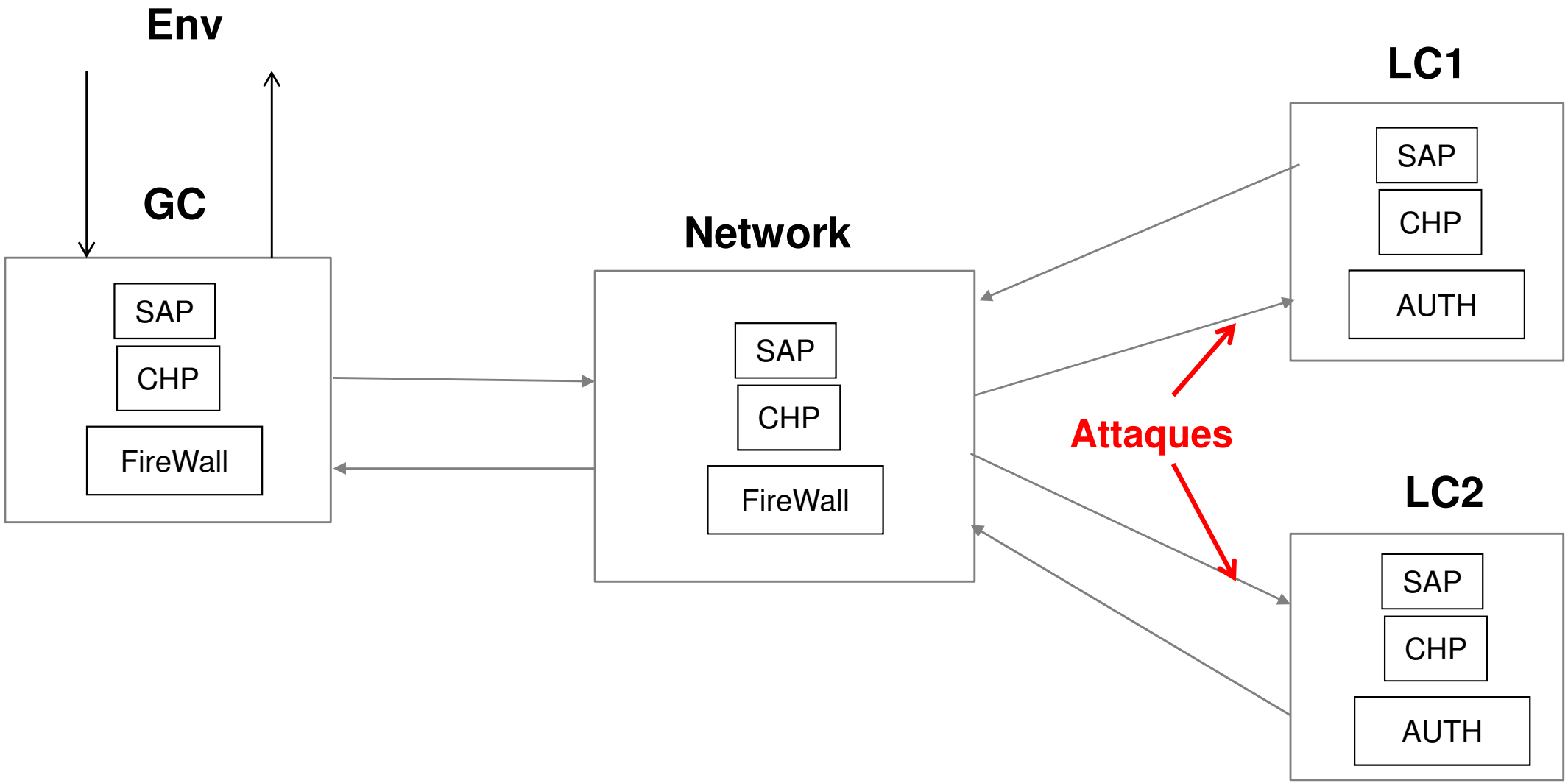
Attaque 2 →

Attaque 3 →

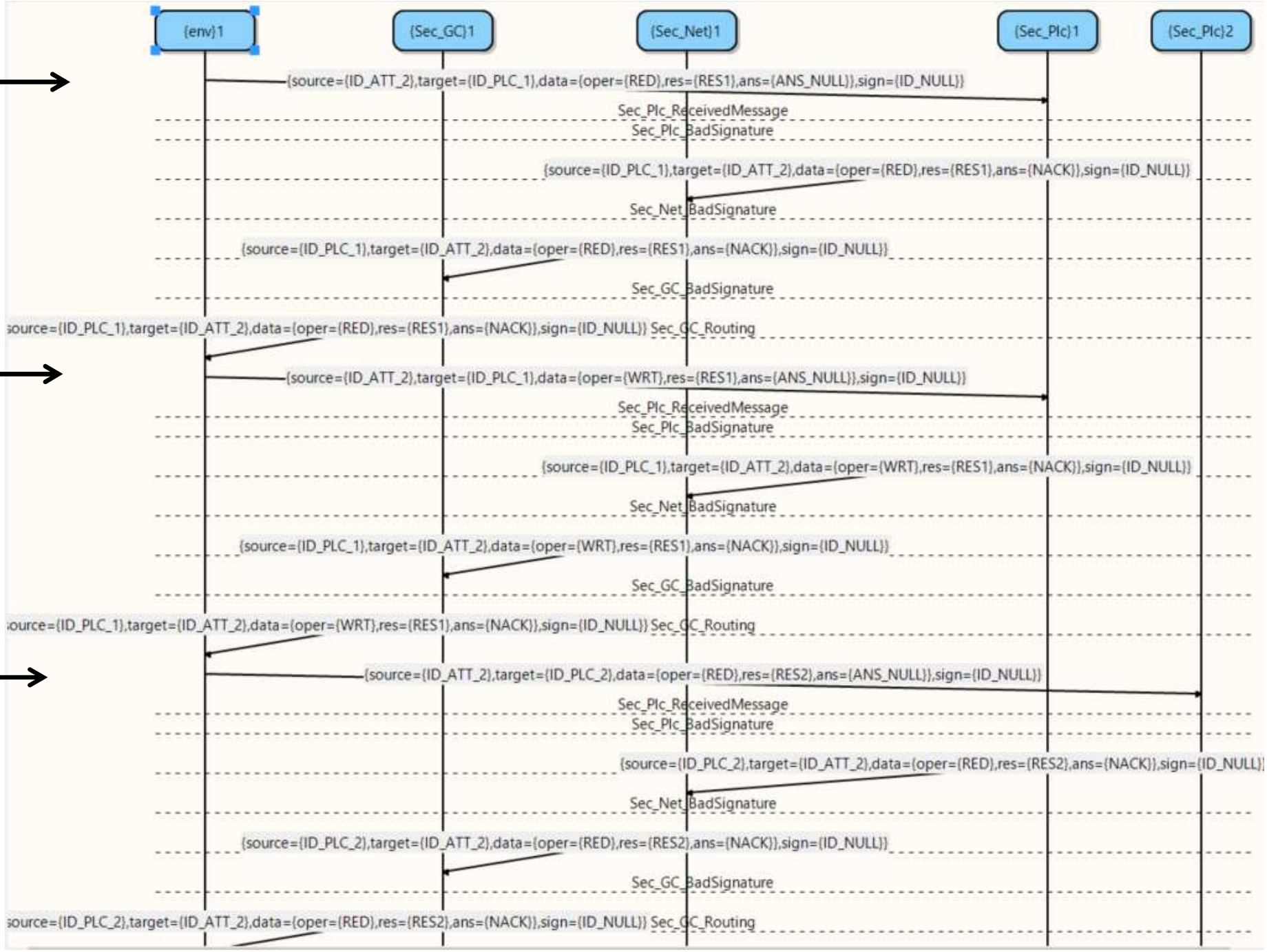
Attaque 4 →



Simulation OBP : mode attaque



Simulation OBP : mode attaque



Propriétés de sécurité (mécanisme de type SAP)

Sureté : Invariant

prt_sap_c_1 : $\forall\ c \in Sap\_C,\ \forall\ e \in Ent, \forall\ opRes \in OpRes,$

$[\,]\ [evt\_access\ (c, e, opRes)\ )\ \Rightarrow pre\_check\ (c, AccReq\ (e, opRes))]$

Vivacité : SE-LTL

prt_sap_c_3 : $\forall\ c \in Sap\_C,\ \forall\ req \in AccReq,$

$[\,]\ [\ evt\_request\ (c, req)\ \Rightarrow \Diamond\ evt\_check\ (c, req)\ ]$

Propriétés de sécurité (mécanisme de type SAP)
Expression en CDL

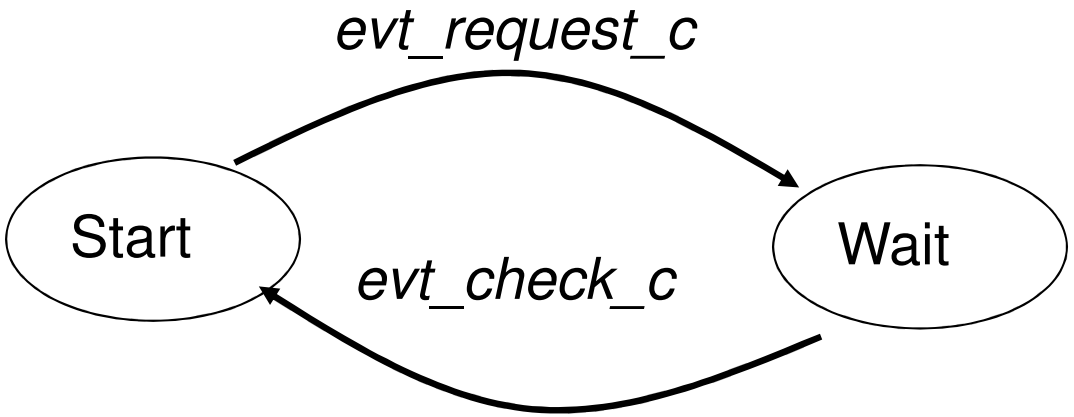
Sureté : Invariant ou observateur de rejet

prt_sap_c_1 : Invariant

assert [not *evt_access* (*c, e, opRes*)) **or** *pre_check* (*c, AccReq* (*e, opRes*))] **]**

Vivacité : Observateur

prt_sap_c_3 :



Vérification sous OBP

Sureté : Invariant : analyse d'atteignabilité

Vivacité :

- **Cas Traces finies :**

  Pour tous les états finaux du graphe :

  L'observateur ne reste pas dans Wait

- **Cas Traces non finies :**

  Extension d'OBP : Plug (model-checking LTL, …bientôt SE-LTL)

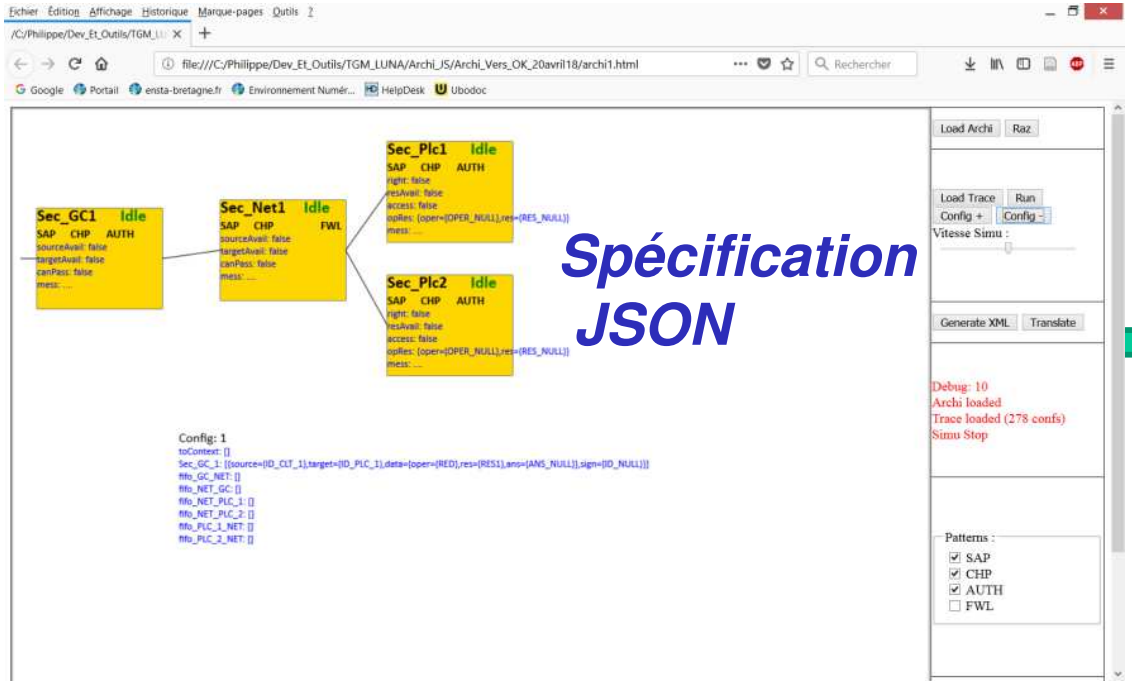
  [ ] [*Obs.Wait* => ◊ not *Obs.Wait*]

Modélisation et validation formelle d'architectures logicielles sécurisées

- Contexte, motivations
- Patterns de sécurité
- Formalisation
- Processus d'intégration dans une architecture et validation
- Perspectives

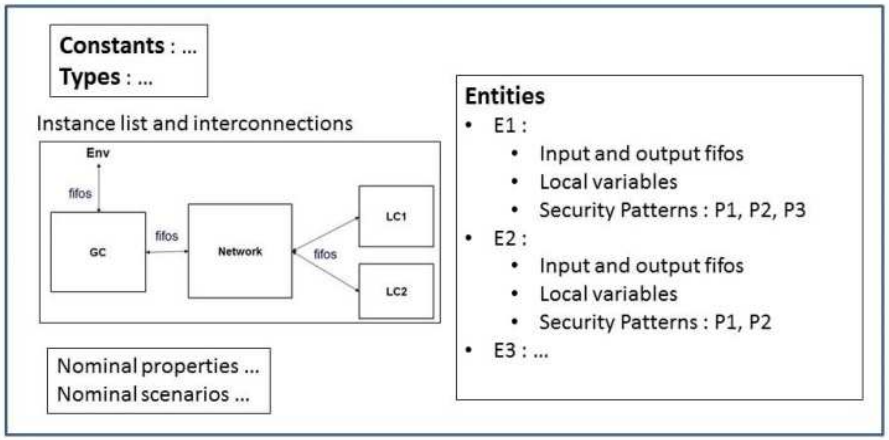
Processus

Prototype en cours de développement



Spécification JSON

Modèle XML



Génération

Modèles Architecture, Propriétés, scénarii

Processus

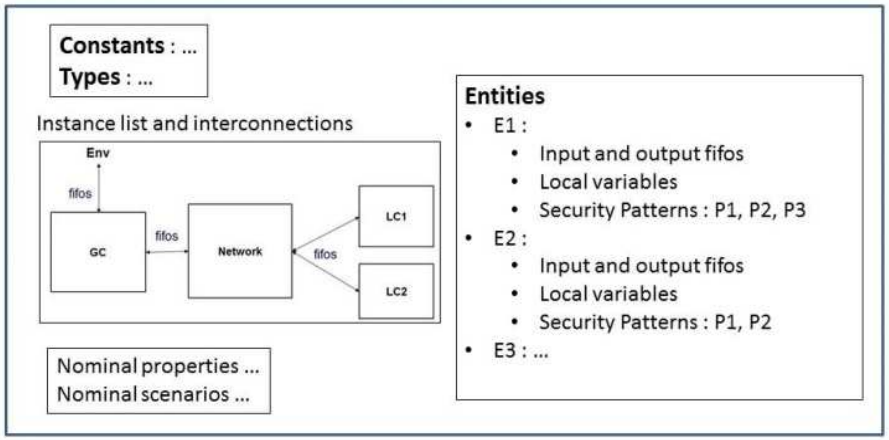
Prototype en cours de développement

Etat des invariants et observateurs

Contenu des fifos

Traces d'exécution

Modèle XML



Génération

Modèles Architecture, Propriétés, scénarii

Perspectives

- **Intégration des patrons**

  → Evaluation des stratégies (critères)

- **Politiques de sécurité complexes (dynamiques)**

  → Composition de patrons
  → génération des propriétés à vérifier

- **Prise en compte des types d'architecture**

  → communication synchrones, modèles temporisés, …

- **Composition de patrons**

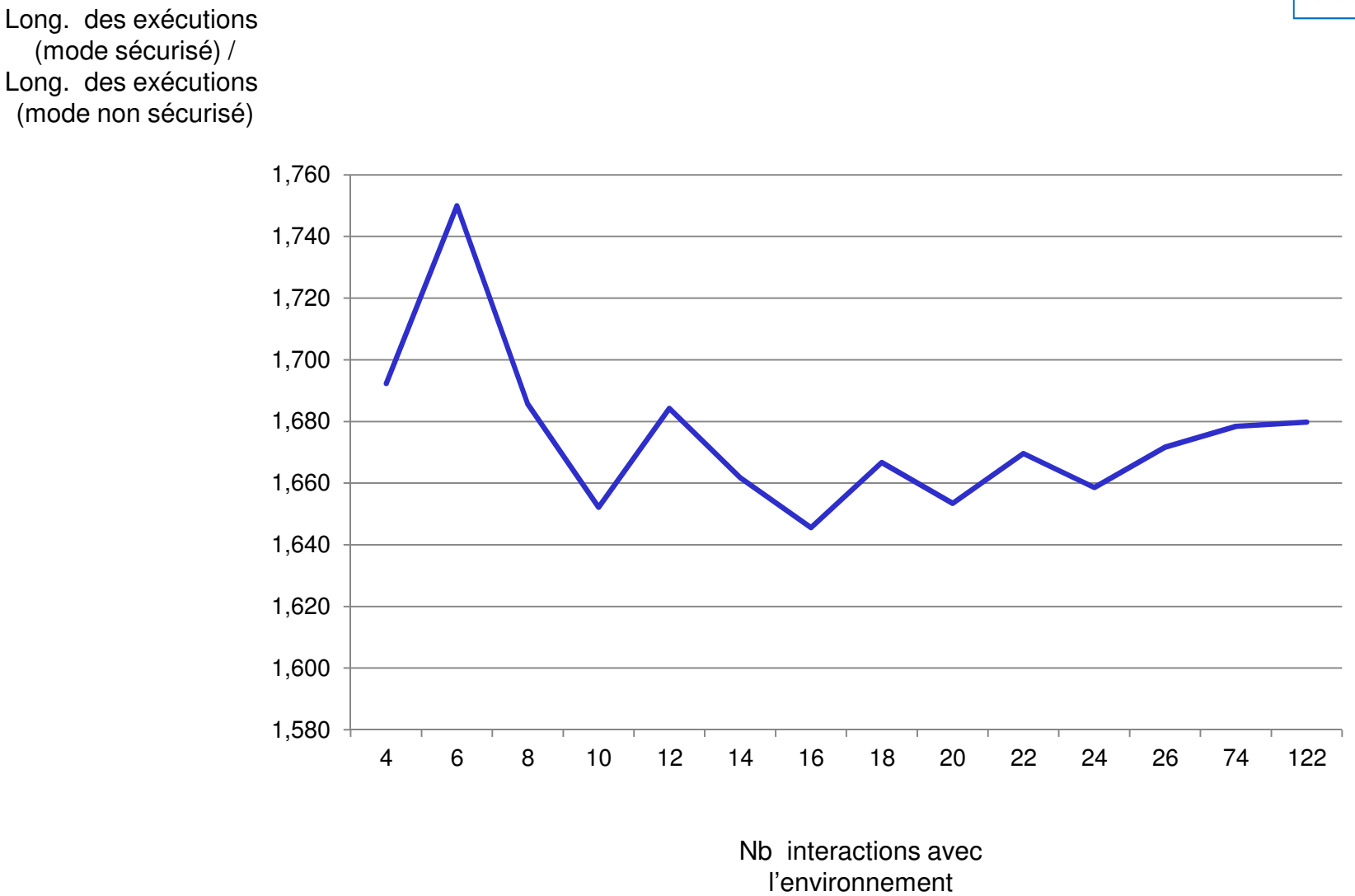
  → composition (incrémentale ?) d'automates
  → Preuves

Merci pour vos questions

http://www.obpcdl.org

Analyse de la complexité

Souhait : la complexité dédiée à la sécurité : non proportionnelle au traffic



Modèles d'architecture