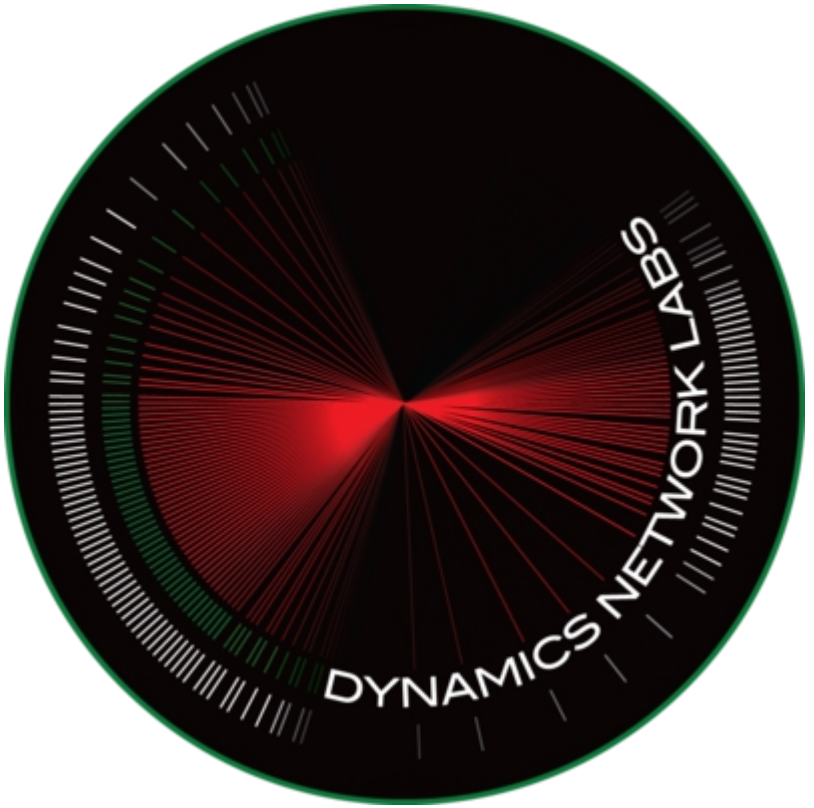
Linux Security Modules



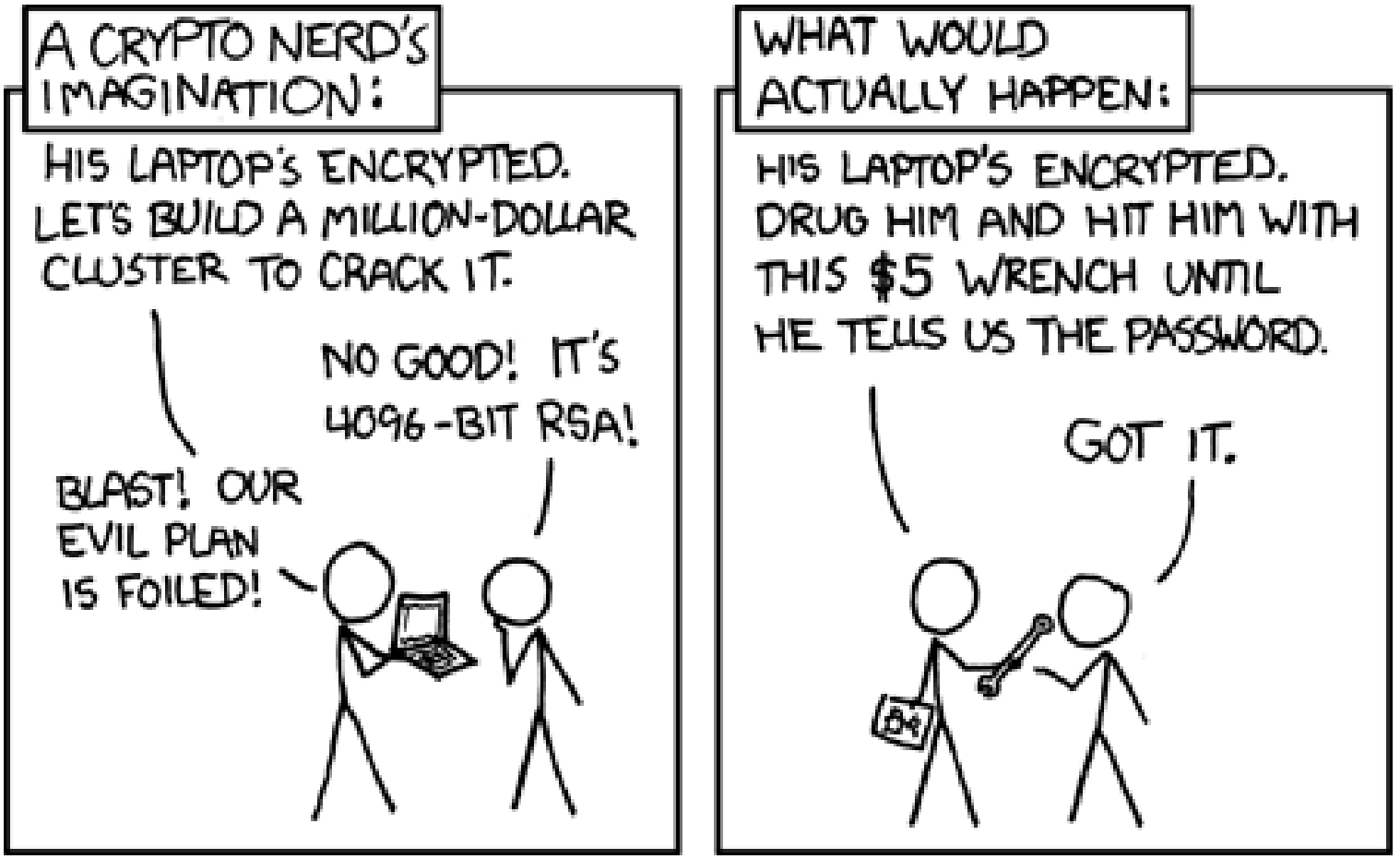
DYNAMICS NETWORK LABS

www.dynamics-network.com

Linux Security Modules

Security ?

Avant propos



Avant propos – la sécurité globalement

- La sécurité est assurée par un principe de couche, notre couche sera le contrôle d'accès

- Modèle formel
- Implémentation
- Vérification
  - Critère d'évaluation (orange book – DoD)
- Sécurité ou outils d'administration système et réseau

Avant propos – LSM

- LSM est un framework proposant des vérifications de sécurité
- Les implémentations actuelles reposent sur des modèles formels de type MAC (SELinux, Tomoyo, smack, apparmor)
- Sans une de ces implémentations chargées, le kernel utilise les capabilities

Approche

- Modèle formel – théorie
  - Contrôle d'accès
- Implémentation dans Linux
  - Infrastructure LSM
  - Fonctions partagées
- Pistes de réflexion, critique

Modèle formel

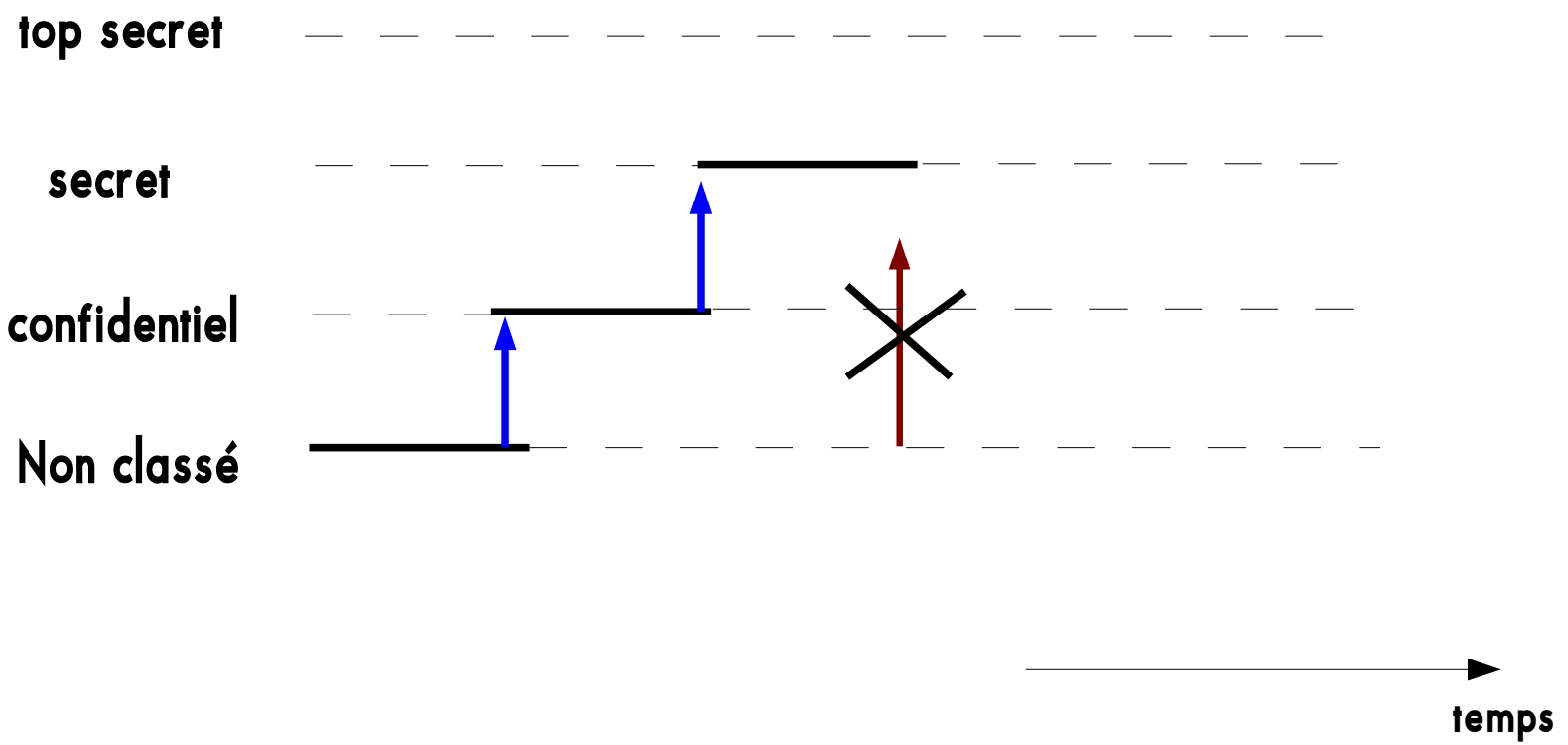
Modèle formel

- Un modèle formel décrit une méthode de spécification de la politique de sécurité d'un système
- En pratique, il repose sur une idée fondatrice ou une technique
- Le formaliste est introduit pour décrire l'idée fondatrice ou la technique

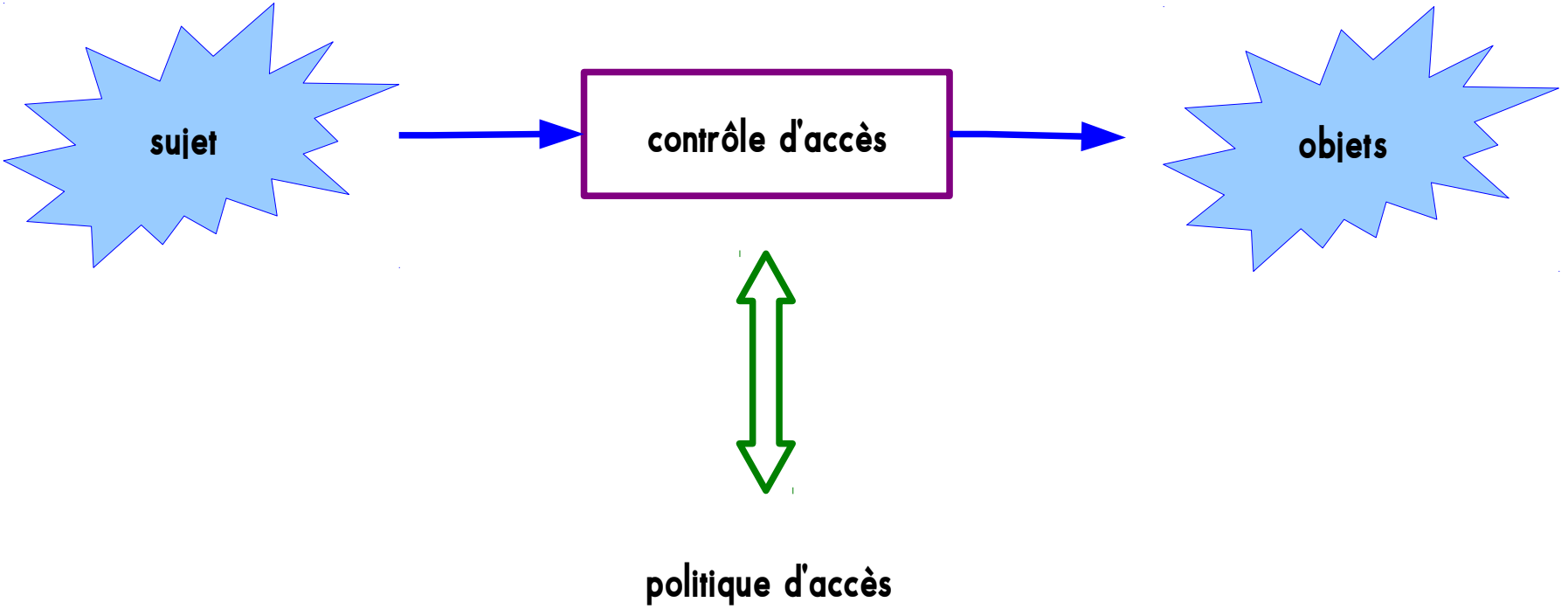
Modèle formel

- LOMAC : Low Water-Mark Mandatory Access Control – 2000
- Bell-La Padula (BLP) – 1973
- object-capability – 1981
- Take-grant – 1977
- Biba – 1977
- Matrice de contrôle d'accès – 1971
- ..

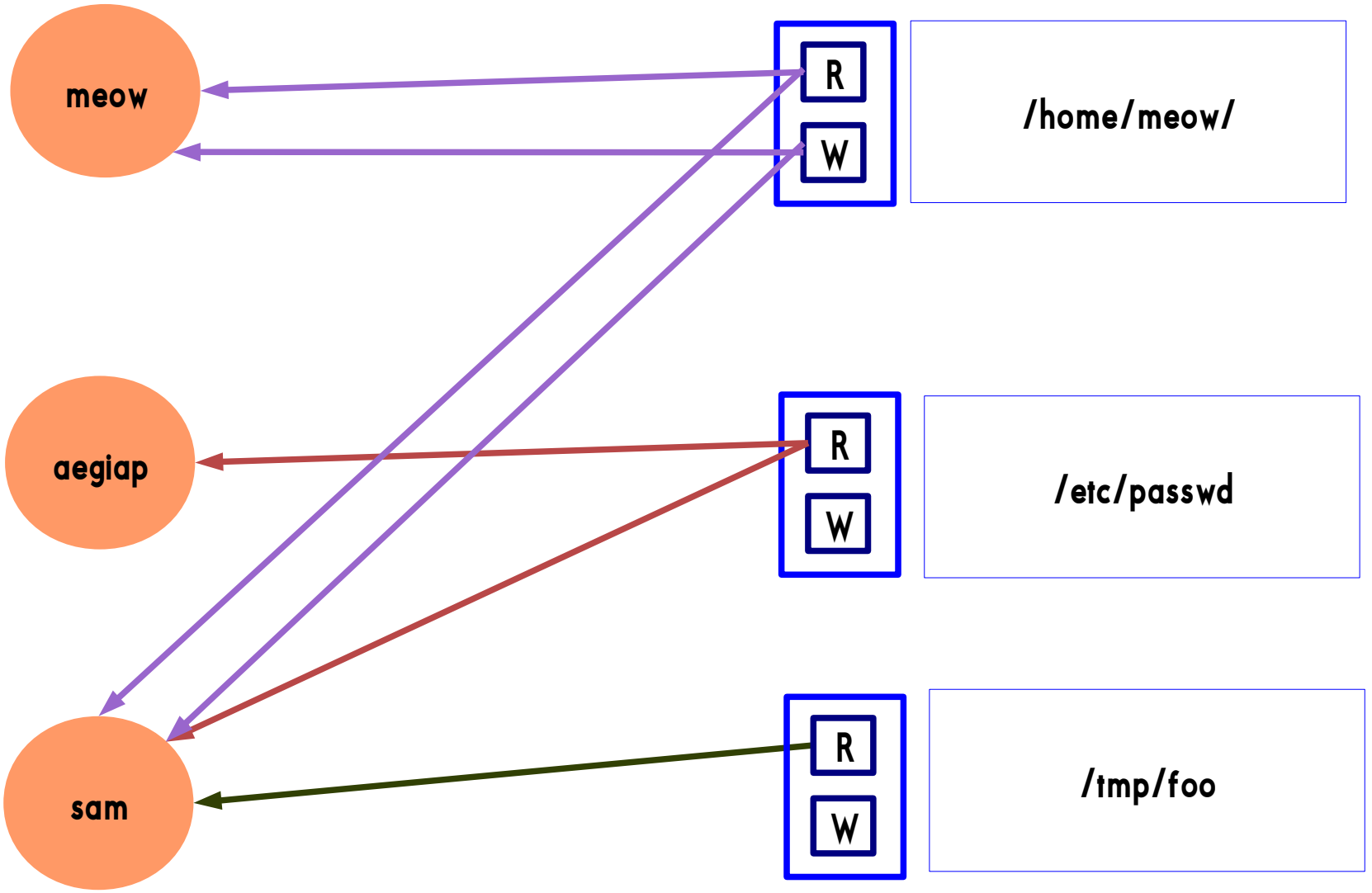
- Mécanisme de water marking d'objets



Matrice de contrôle d'accès : sujet et objet



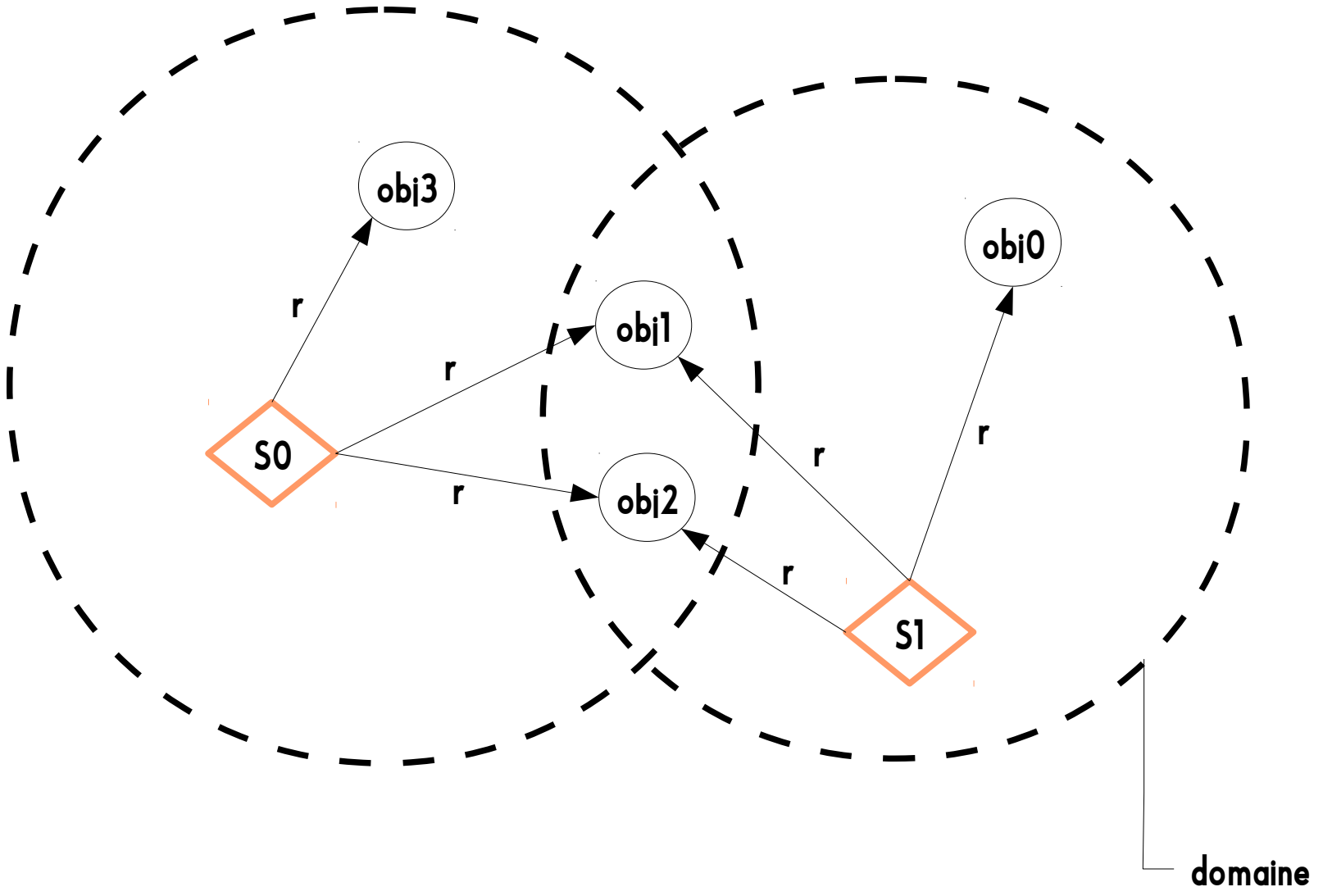
Matrice de contrôle d'accès



Matrice de contrôle d'accès

	/etc/passwd	/tmp/foo	/home/meow
meow	-	-	{ read, write }
aegiap	{ read }	-	-
sam	{ read }	{ read }	{ read, write }

Matrice de contrôle d'accès : domaine



Modèle par fonction de sécurité

- Intégrité : BIBA, Clark Wilson
- Confidentialité : HRU, Bell LaPadula, Take-Grant

- HRU : matrice de contrôle d'accès M(sujet x objet)
- Take-Grant : formalisme de graphe
- Bell LaPadula : classification des objets, habilitation des sujets

Implémentation dans Linux

Implémentation dans Linux – framework

- Infrastructure de hooks
  - Développement de modèle formel
- Des « fonctionnalités partagées »

Les hooks LSM

- include/linux/security.h
- struct security_operations foo_ops {

  ...

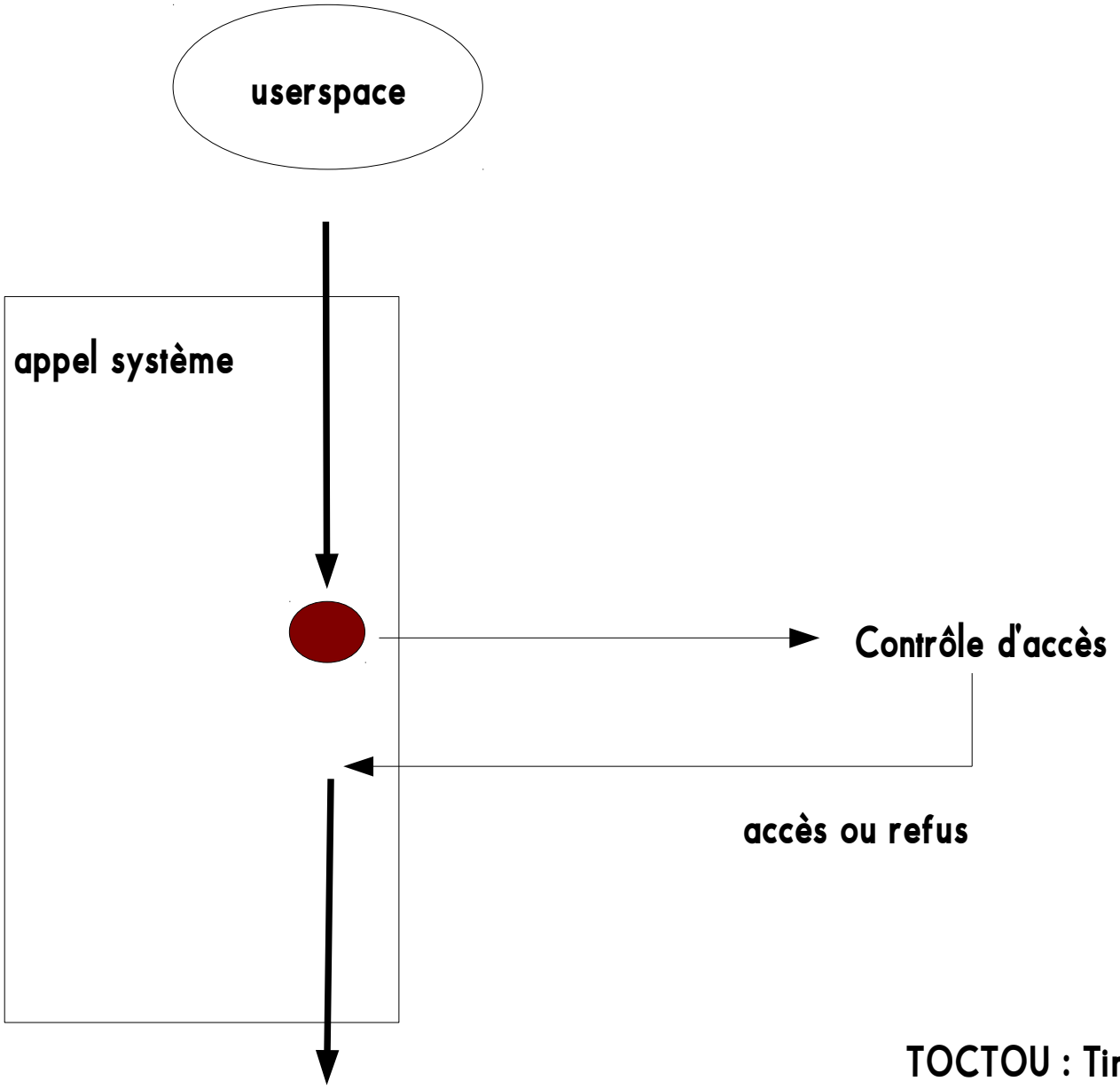
  - .inode_create = foo_inode_create,
  - .task_kill = foo_task_kill,

  ...

}

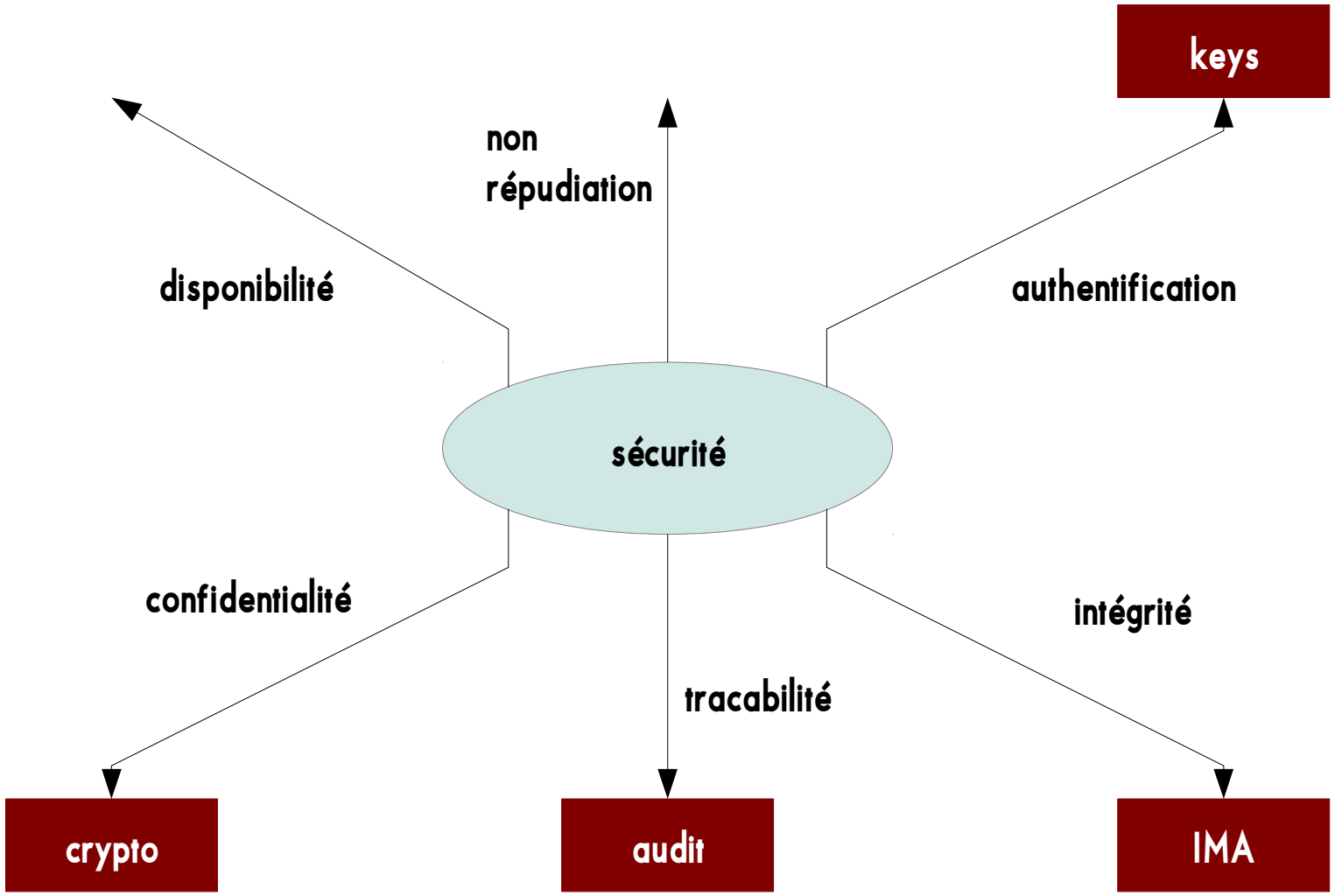
- smack, SELinux, apparmor, tomoyo

Les hooks



TOCTOU : Time Of Check – Time Of Use

Fonctionnalité partagée – critère de sécurité



Pistes de réflexion

Condition d'intégration des modules upstream

- Il faut pouvoir expliquer la capacité défensive du nouveau module.
- Pas compatible avec la gestion du projet upstream – « troll detected »
- Exemples : stacking de modules, intégration tomoyo, rejet de fonctionnalités « because it's not secure »

Piste de réflexion : modèle par fonction de sécurité

- Les modèles par thème (intégrité, confidentialité, etc) ne couvrent pas toutes les fonctionnalités
- Impossibilité du stacking LSM
- Utilisé un modèle protégeant l'intégrité ne protègera pas les autres fonctions, et il est impossible de charger les autres modèles
- Un mécanisme de hook, un seul module de sécurité ? (netfilter)

Piste de reflexion : document, livres..

- Les LSM sont ils considérés comme un mécanisme essentiel du kernel ? (irq, process management, vfs, etc)
- Les bouquins sur le kernel ne parlent pas des LSM

Le mécanisme de hook rend la création de module facile

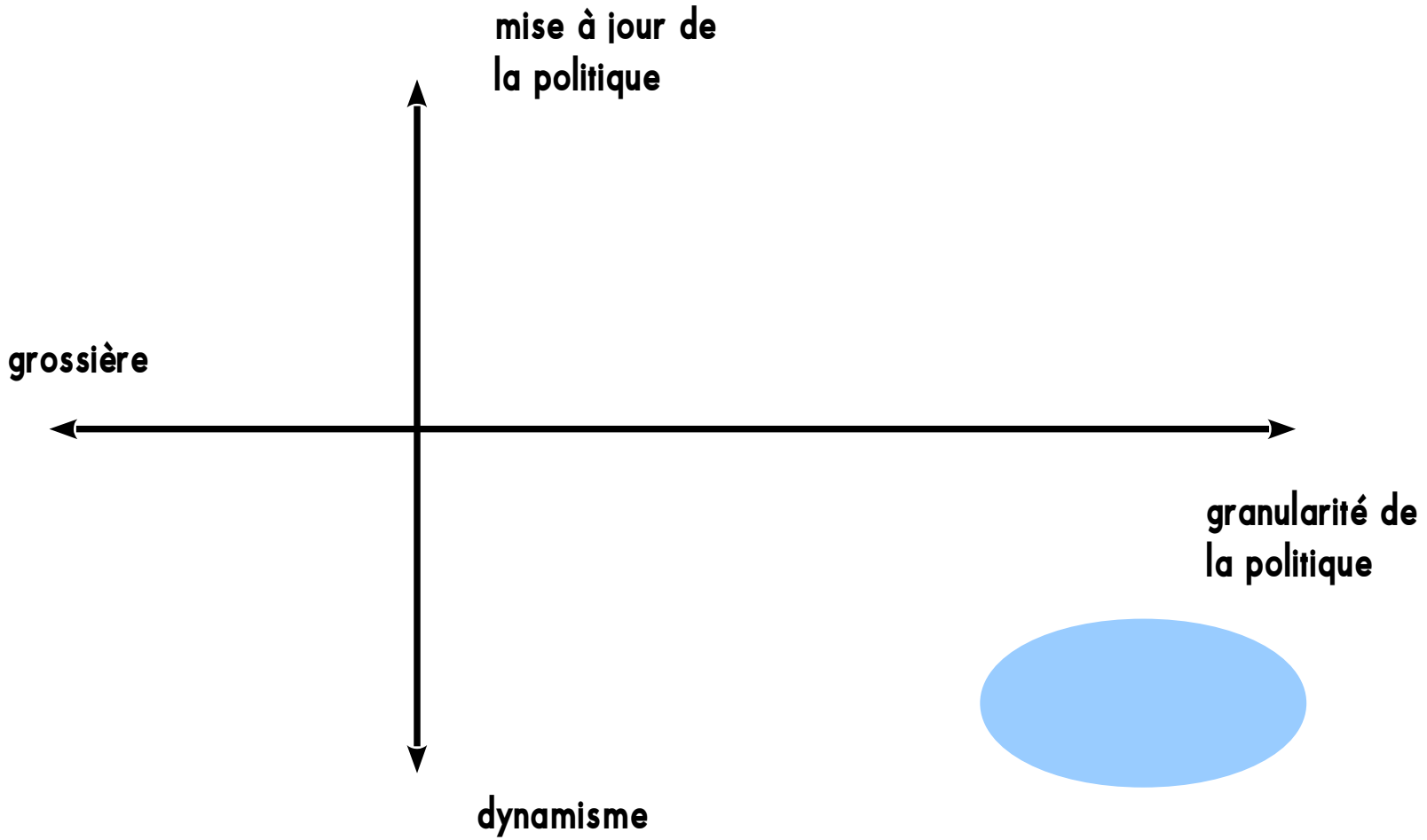
- Module formel au hack – regard sur les propositions
- Refus d'innovation sur les modèles formel, 30 ans

Points difficiles

- Politique monolithique implique configuration complexe – 350K+ lignes de conf, [application x fichier x action]

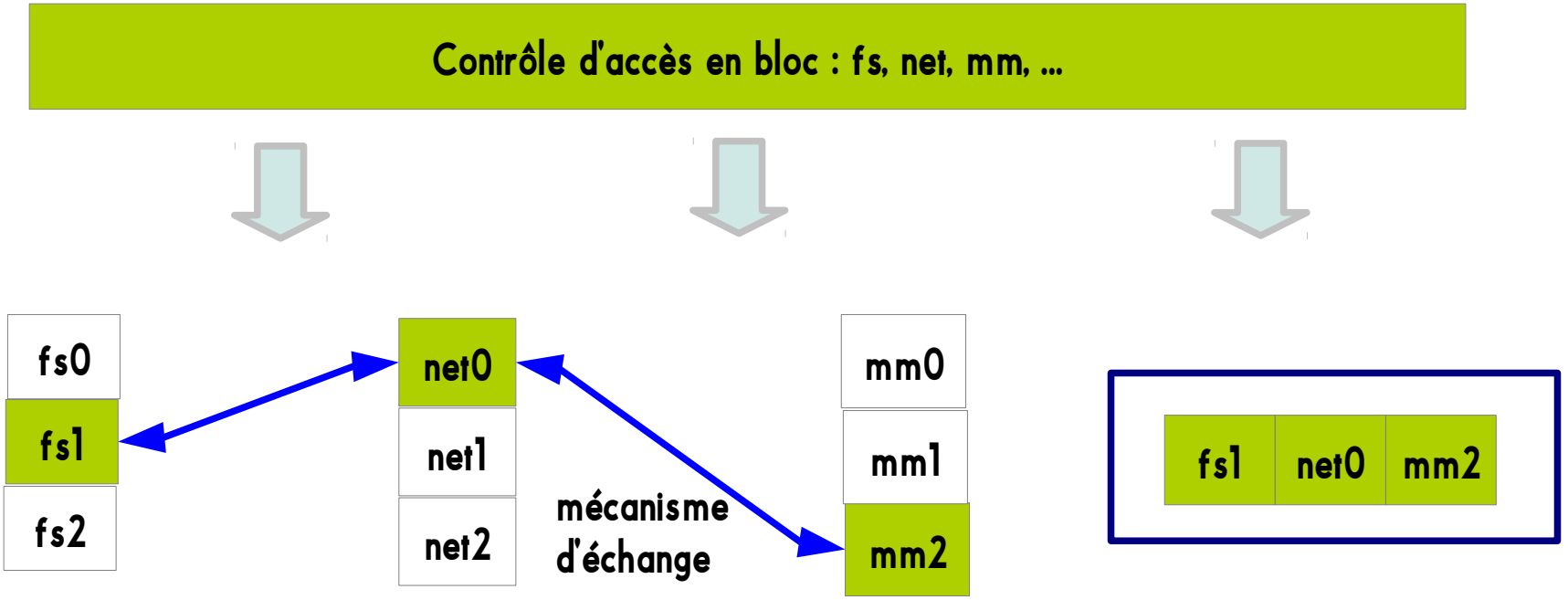
- Pouvons nous rendre le contrôle d'accès dynamique ?
- Pouvons nous conserver la granularité ?

Piste de réflexion – orthogonalité : granularité vs dynamisme

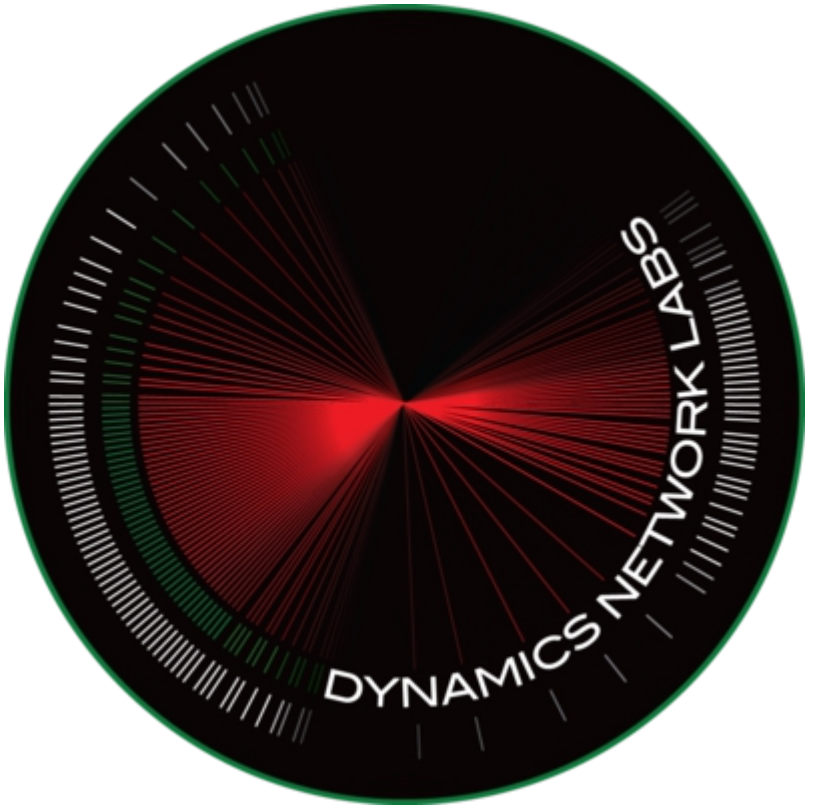


Points difficiles

- Modèle monolithique cependant des fonctions partagées
- Peut on séparer par thème ?



Linux Security Modules



DYNAMICS NETWORK LABS

www.dynamics-network.com