Le TEE, nouvelle ligne de défense dans les mobiles

SSTIC 2013

Hervé Sibert

2013-06-07

Disclaimer

The contents of this document are subject to change without prior notice.

ST-Ericsson makes no representation or warranty of any nature whatsoever (neither expressed nor implied) with respect to the matters addressed in this document, including but not limited to warranties of merchantability or fitness for a particular purpose, interpretability or interoperability or, against infringement of third party intellectual property rights, and in no event shall ST-Ericsson be liable to any party for any direct, indirect, incidental and or consequential damages and or loss whatsoever (including but not limited to monetary losses or loss of data), that might arise from the use of this document or the information in it.

ST-Ericsson and the ST-Ericsson logo are trademarks of the ST-Ericsson group of companies or used under a license from STMicroelectronics NV or Telefonaktiebolaget LM Ericsson.

All other names are the property of their respective owners.

For more information on ST-Ericsson, visit www.stericsson.com

Plateforme mobile

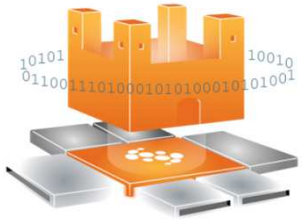
>>>>>

- Quasi-téléphone
  - matériel : PCB avec SoC, BB, NVM, DDR, ports, écran tactile…
  - système : OS Linux, Android

Késako TEE?

ST ERICSSON

Changement de paradigme



Peu déployé, restreint au mobile

Plusieurs fournisseurs, portage spécifique

[1990, 2007]

Framework applicatif propriétaire

"Apps"

JVM (J2ME)

Couche d'adaptation

RTOS propriétaire

Fonctions sensibles

HW de sécurité (secrets, crypto…)

[2007,?]

Native apps

Apps

Dalvik

Android framework

Linux

Fonctions sensibles ???

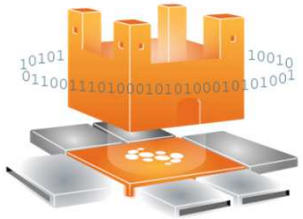
HW de sécurité (secrets, crypto…)

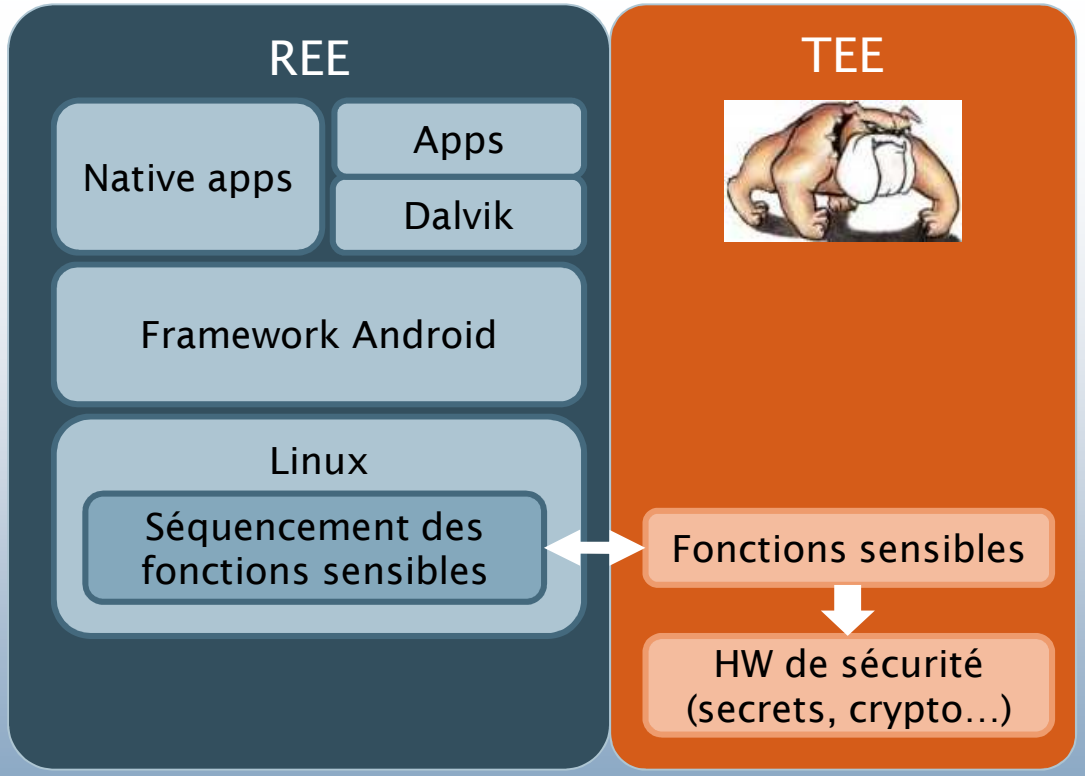
Pas de standards

Chaque fabricant de SoC a le sien, pas ou peu de suivi

Ouvert ou "ouvrable"…

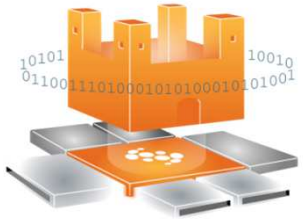
TEE = Trusted Execution Environment





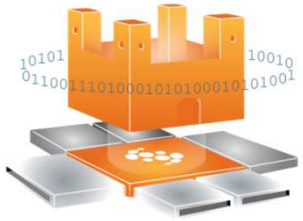
Les fonctionnalités sensibles sont déportées dans le TEE, mais leur gestion demeure extérieure au TEE.

Le TEE, ses fonctionnalités



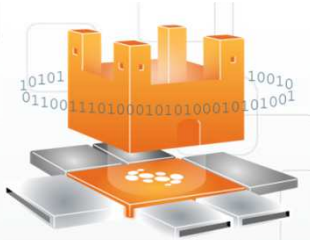
- Dédié à la gestion des fonctions sensibles et du code dont l'exécution dans l'OS normal exposerait la sécurité aux attaques
- Contrôle du boot sécurisé
- Fonctions de sécurité
  - Stockage sécurisé
  - IHM sécurisée (écran tactile, affichage)
  - Canaux sécurisés avec SIM, Secure Element
- Fonctions d'isolation
  - Isolation du code et des données (propres et applicatives) vs OS normal
  - Protection des clés sur lesquelles reposent les fonctions de sécurité
  - Isolation entre les différentes applications du TEE (code et données)

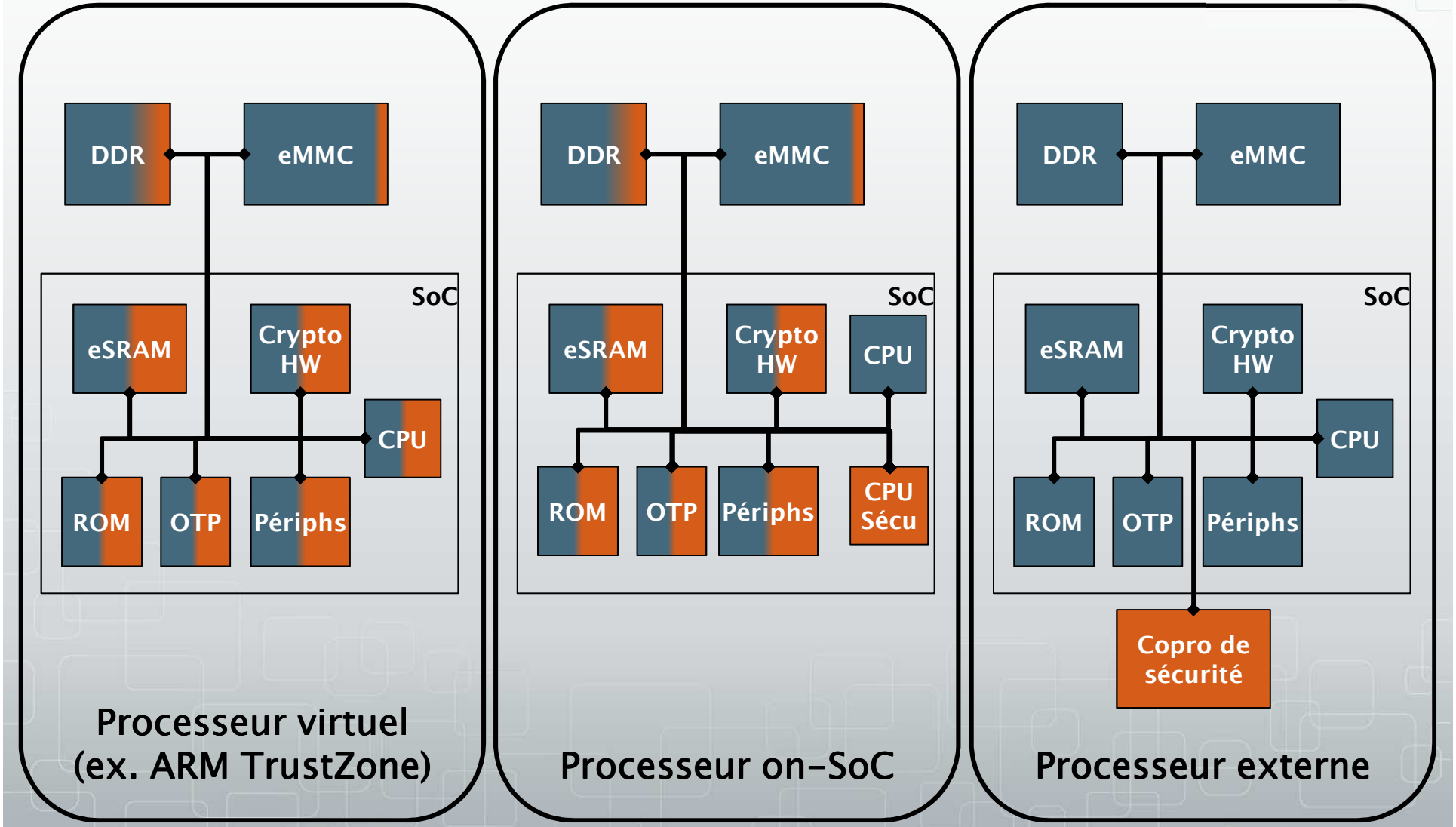
Le TEE GLOBALPLATFORM™



- GlobalPlatform est une alliance d'industriels du domaine de la carte à puce et du paiement, centrée sur le Secure Element, rassemblant fabricants de carte à puce, de SoCs, professionnels du paiement, opérateurs, OEMs, vendeurs de solutions de sécurité, labos…
- GlobalPlatform a commencé à définir un TEE standard il y a ~5 ans
  - Volonté de contrôler une migration potentielle de la sécurité vers le SoC
- Standardisation des interfaces
  - TEE Internal API : développement de "Trusted Applications" dans le TEE
  - TEE Client API : communication avec les TAs depuis l'OS
  - → Encourager le développement logiciel dans le TEE
  - → Simplifier la migration d'un TEE à l'autre pour les fabricants
- Programme de certification pour l'apport de garanties de sécurité
- Programme de conformité pour assurer l'interopérabilité des "GP TEEs"

Bases matérielles pour un TEE



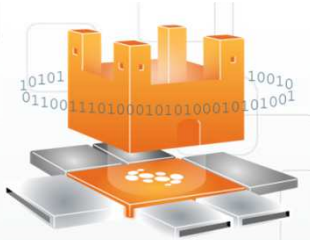


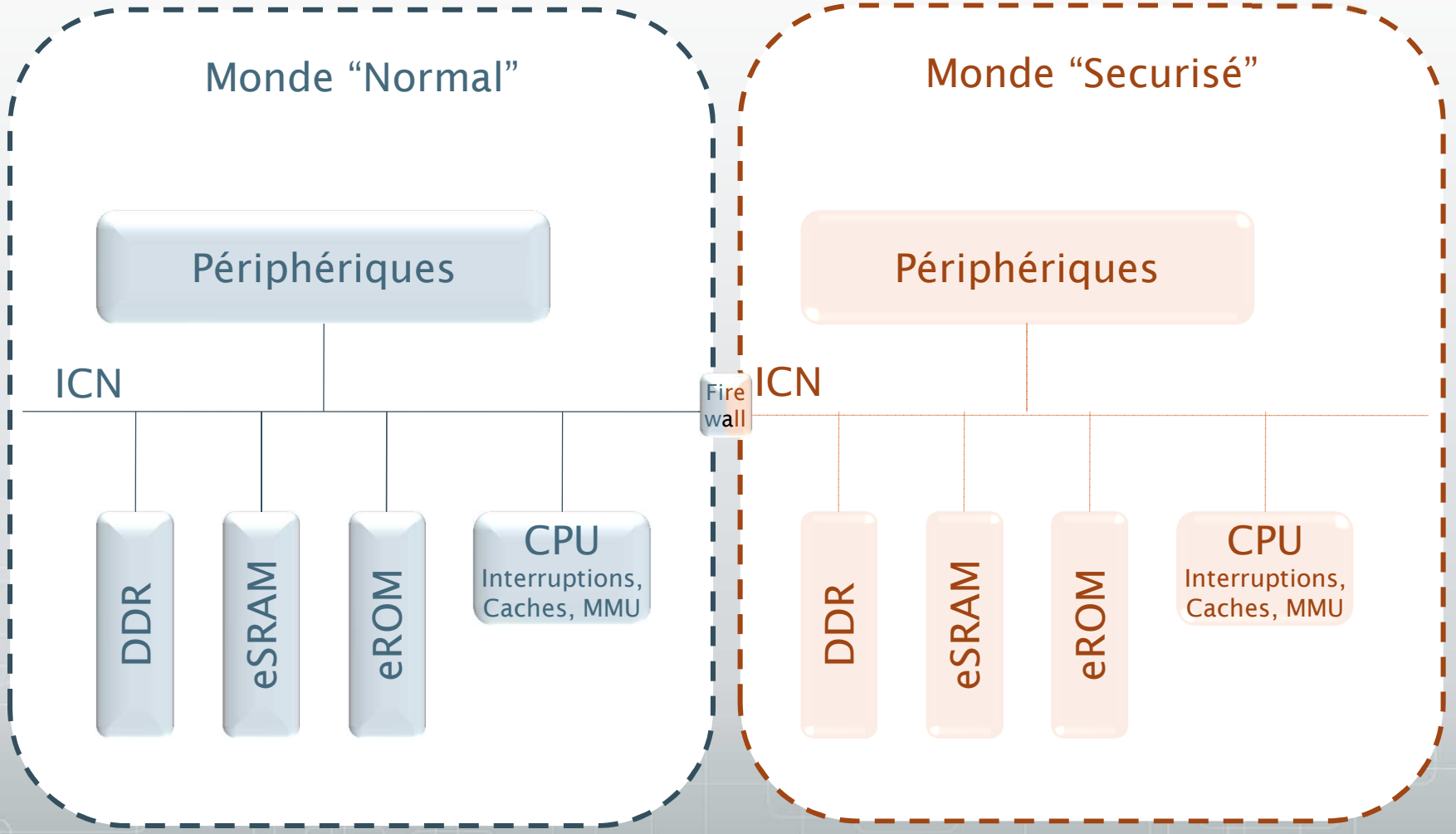
ST ERICSSON

Késako TrustZone?

ST ERICSSON

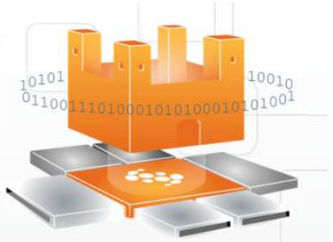
ARM TrustZone en bref



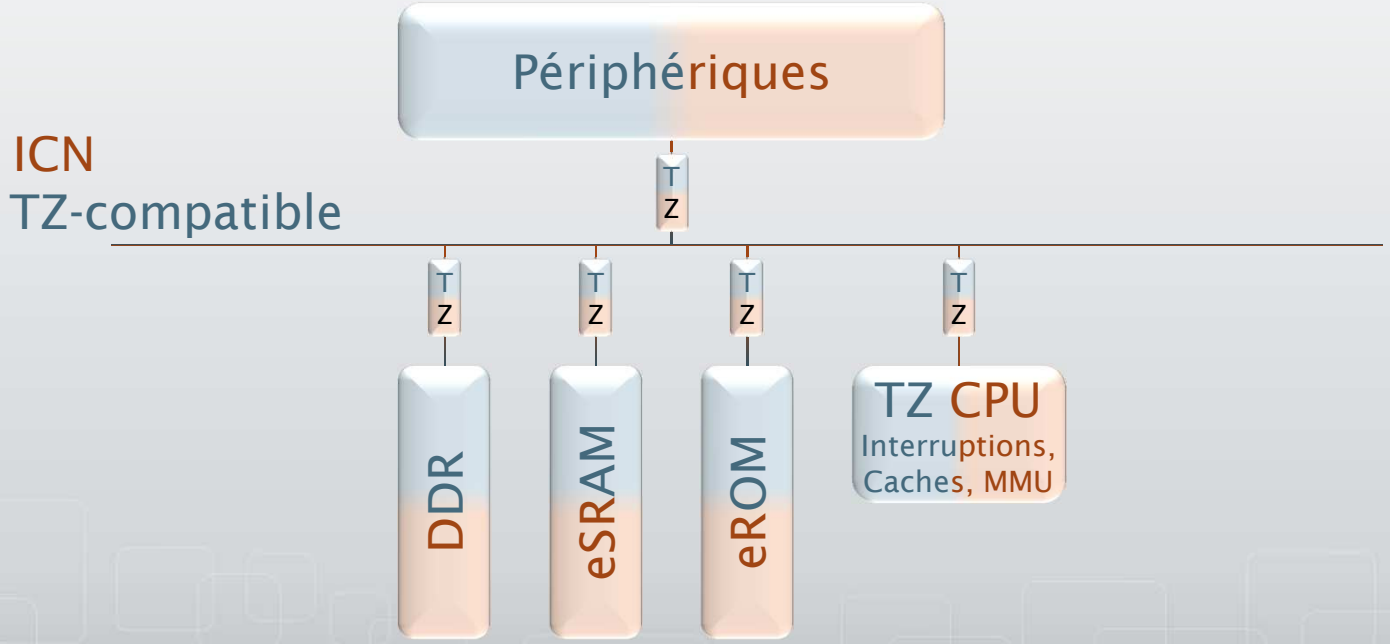


ST ERICSSON

ARM TrustZone en bref

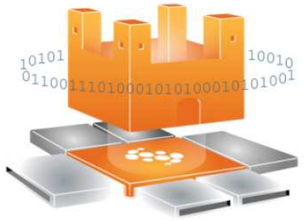


Les composants clés supportent état normal et état sécurisé

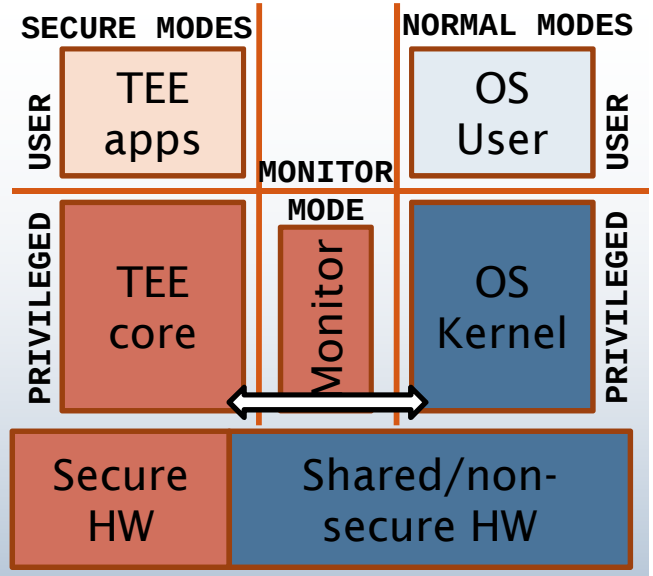


ST ERICSSON

Les fondamentaux de TrustZone

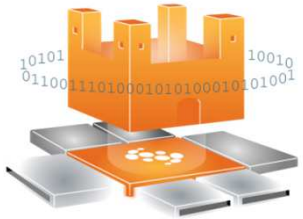


- Le bit NS: 0=Secure, 1=Non-Secure
  - Propagé sur le bus interne, il matérialise la sécurité des accès processeurs
- Isolation au sein du CPU
  - Nouveaux modes: secure privileged/user
    - Le mode moniteur, invoqué via l'instruction Secure Monitor Call ou une interruption sécurisée, héberge le code qui sauve/restaure les registres du mode lors de l'entrée/la sortie



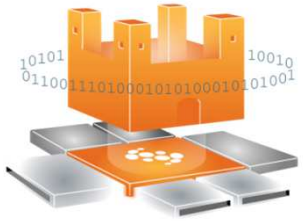
  - Chaque mode a ses tables MMU
    - Les pages peuvent être taguées secure/non-secure
    - Le bit NS propagé lors d'un accès CPU à la mémoire est à 0 si le processeur est en mode secure et si l'adresse cible est taguée secure
  - TZIC: chaque interruption (IRQ/FIQ) peut être secure/non-secure
  - Chaque ligne de cache est taguée suivant l'accès qui l'a mise en cache

Mais encore ?



- Isolation hors du CPU
  - La MMU autorise les accès suivant les descripteurs
    - Les descripteurs MMU du monde normal peuvent permettre l'accès à une page taguée "secure" du côté sécurisé, en l'absence d'autres protections
  - Protection matérielle des accès hors du CPU
    - Registres internes du SoC – l'accessibilité suivant la valeur du bit NS est définie lors de la conception
    - TrustZone Memory Wrapper – devant une mémoire, n'autorise que les accès avec NS=0 jusqu'à une frontière définie dans un registre (contrôlé par le monde sécurisé)
    - TrustZone Protection Controller – devant les (contrôleurs de) périphériques, permet de filtrer les accès suivant la valeur du bit NS
- Extensions
  - Des DMAs compatibles TZ permettent d'effectuer des transactions "pour le compte" du mode secure
  - Du matériel spécifique permet des filtrages plus fins que S/NS

TrustZone dans la poche



- Tous les CPU ARMs depuis ARM1176
- En particulier tous les Cortex Ax
- Mais…
  - …encore faut-il que le SoC permette un usage cohérent de TrustZone
    - Le CPU démarre en mode secure
    - Si TrustZone n'est pas utilisé, pas de raison d'en sortir…
    - Si le bit NS n'est pas utilisé au niveau des mémoires/périphériques, pas d'utilisation intéressante possible

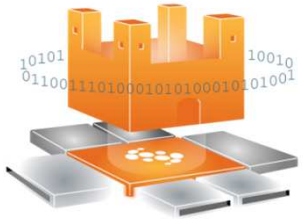
Bonne protection contre un rootkit TZ ☺

- Dans la pratique
  - Les SoCs de la plupart des smartphones récents sont pensés pour un usage cohérent de TrustZone…
  - …mais les garanties de sécurité sont presque inexistantes.

Une recette de TEE

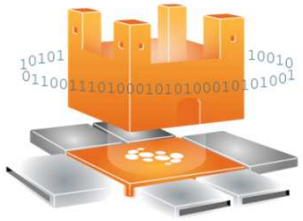
ST ERICSSON

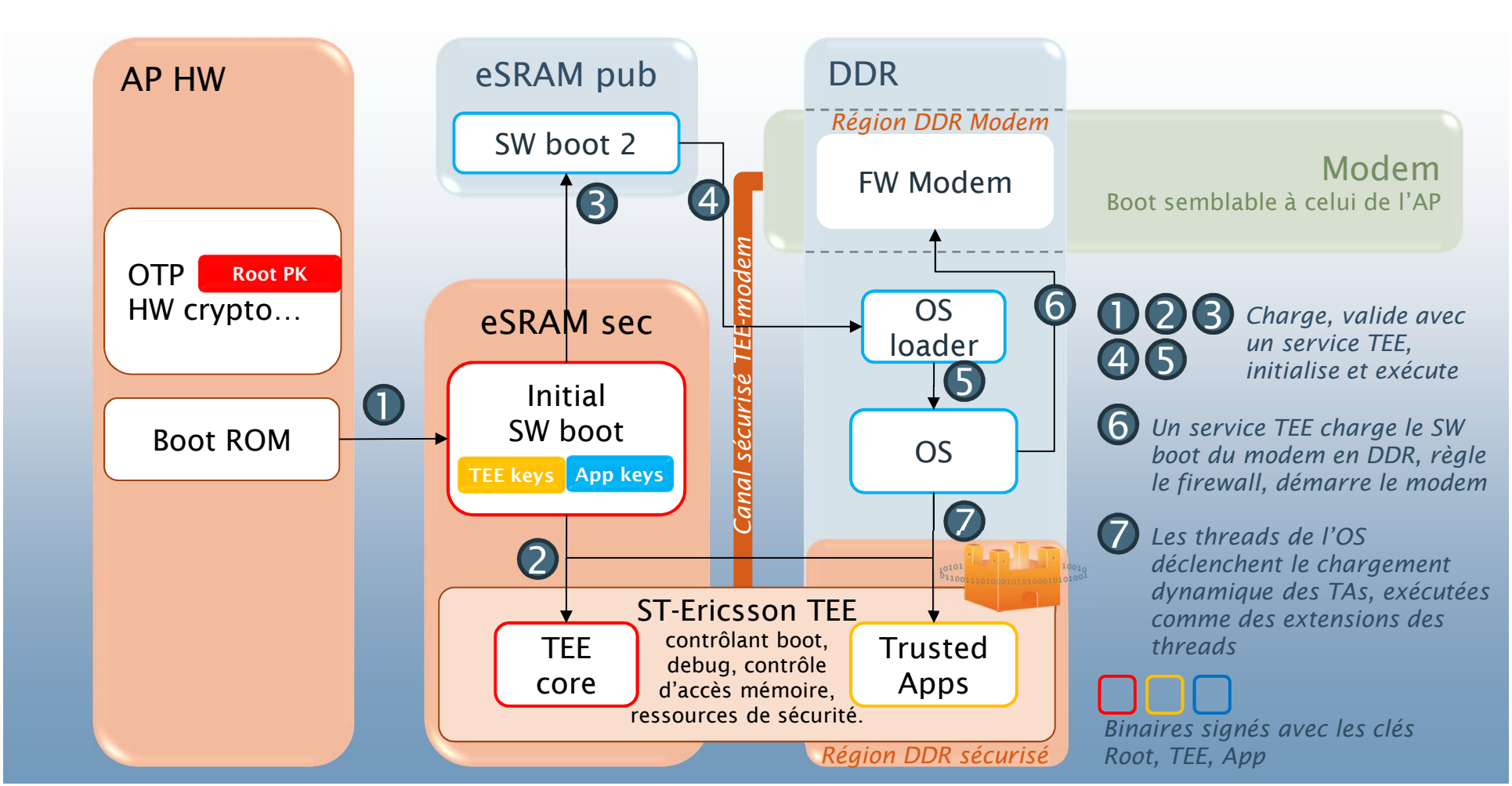
Premier ingrédient: un secure boot



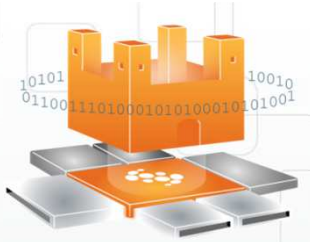
- Racine de confiance matérielle
  - OTP contenant au moins une clé publique, une clé symétrique unique, un identifiant public unique
  - Protection des accès au mode secure par test de chip et debug (JTAG)
- ROM boot
  - Initialisation limitée de la plateforme, en mode secure
  - Chargement en RAM on-chip, vérification et exécution du SW boot, depuis USB, UART, eMMC…
- SW boot
  - Initialisation complète de la plateforme (contrôleur DDR, power mgmt…)
  - Contrôle du mode de boot (provisioning/production/…)
  - Initialisation du TEE
  - Passage du CPU en mode “normal” pour le boot Android

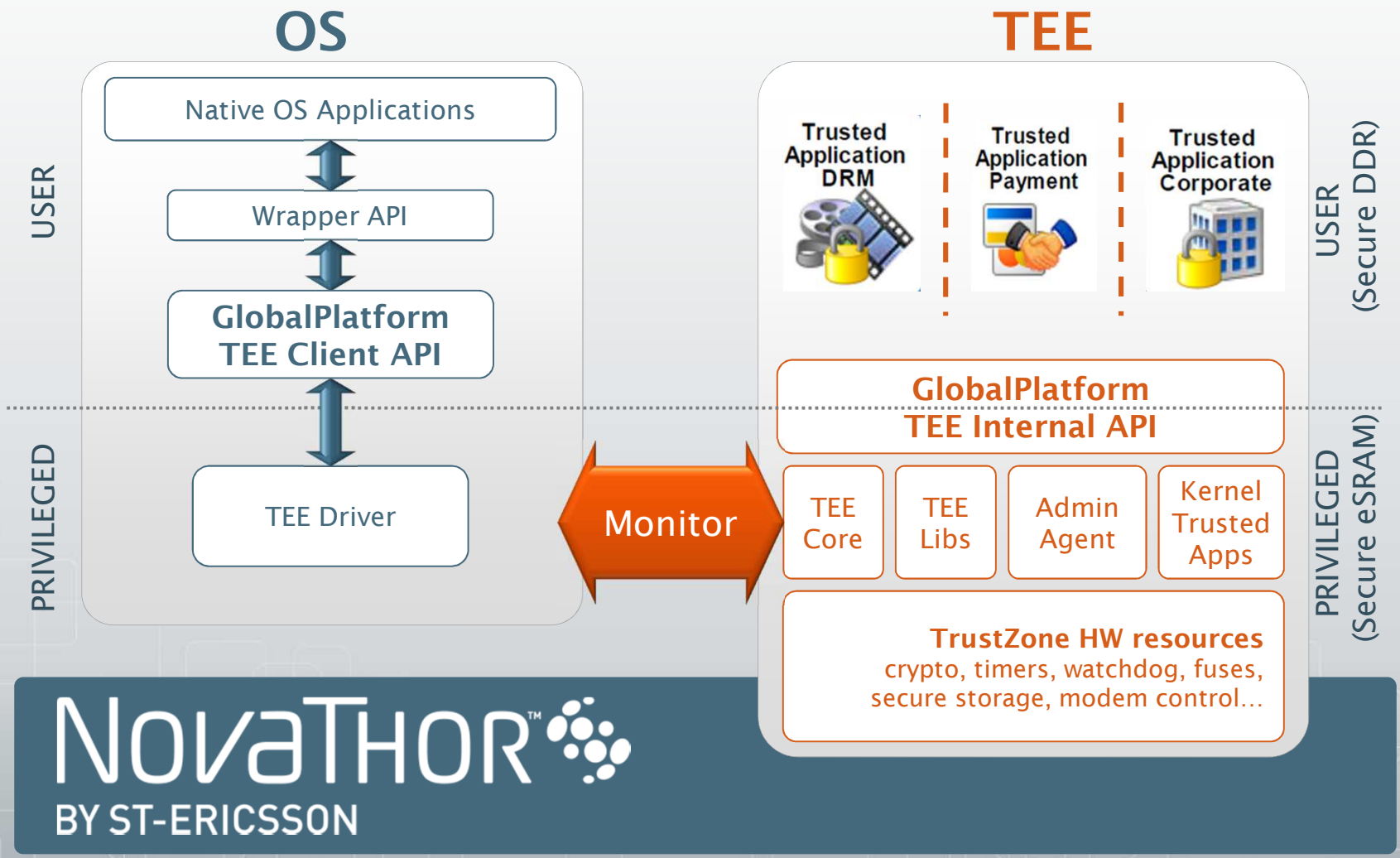
Chaîne de confiance du boot





Second ingrédient: un TEE…

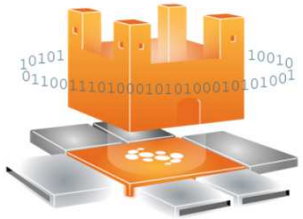




NovaThor™
BY ST-ERICSSON

ST ERICSSON

…disponible



Decision de concevoir le TEE ST-Ericsson fondée sur :
- L'utilisation du TEE comme coeur de la sécurité plateforme commune ST-Ericsson,
- Le besoin des clients de garder le contrôle sur la sécurité, excluant une solution commerciale

ST-Ericsson TEE

Etudes TEE/virtualisation

ST ERICSSON → TEE v1 → TEE v2

En déploiement, **conforme aux spécifications TEE v1.0**

Déployé sur NovaThor U8500, support de la TEE Client API

Standards

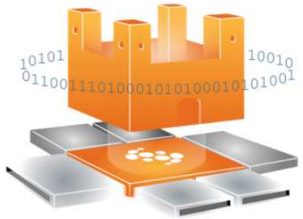
OMTP OPEN MOBILE TERMINAL PLATFORM — Premier standard TEE (pas de spécifications)

Contribution aux spécifications, aux tests de conformité, à la définition de la certification

GLOBALPLATFORM™ Device Committee
Spécifications TEE + conformité + certification

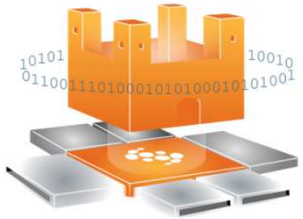
2007 2008 2009 2010 2011 2012 2013

…coopératif



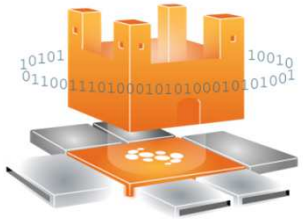
- Gestion de nombreuses fonctions critiques du système en collaboration avec l'OS
  - Multi-coeurs
  - Caches
  - Power state
  - Resets
- Contrôle de configuration du système de sécurité
  - Debug et tests
  - Power, clock
  - Mémoires
  - Filtres sur le bus

…joli (?)



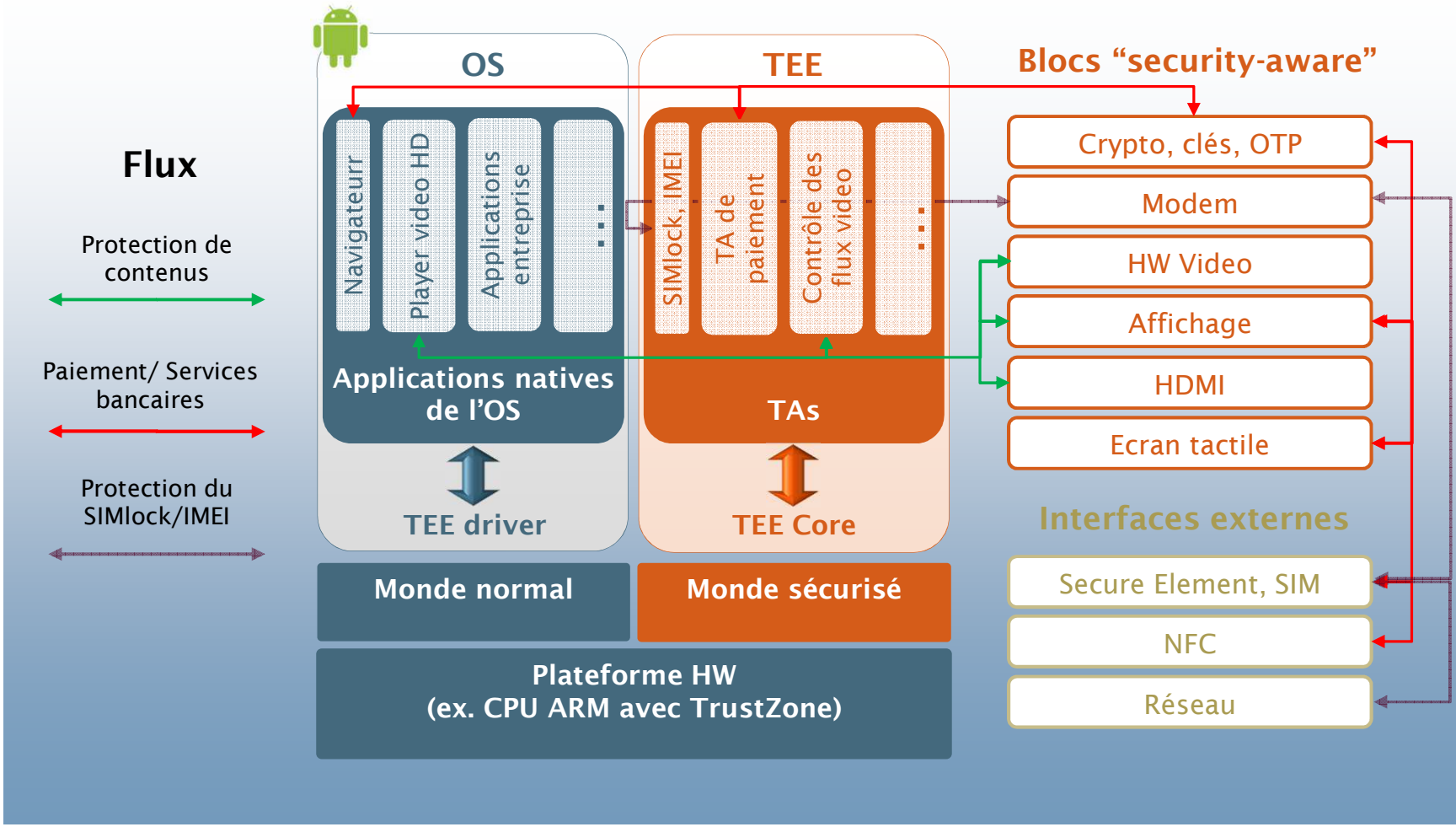
- Small is beautiful
  - Les Trusted Applications sont réduites au maximum
    - Limitation de la surface d'attaque dans le monde sécurisé
  - L'orchestration reste du ressort de l'OS
    - Le TEE hérite du scheduling de l'OS
  - Le besoin en mémoire est réduit
    - Le Paging-on-demand permet de conserver une exécution "on-chip" du mode "secure privileged"
- Chargement dynamique des Trusted Applications
  - Les binaires des TAs sont stockées dans un répertoire spécifique
  - Le driver TEE fournit le binaire TA demandé lorsqu'un client ouvre une session avec cette TA

…utilisable



- Build
  - Basé sur la toolchain Android (gcc, make)
    - Toolchain unique pour Android et le TEE
    - Outils de signature intégrés dans la toolchain
  - Gestion de conf avec git, revue de code avec gerrit
- Debug
  - printf, qui repasse la sortie au driver TEE,
  - JTAG et RAM dump, si le debug du mode secure est activé
- Développement de TAs
  - Trusted Applications SDK extrait du git TEE
  - Interface client côté Linux : GlobalPlatform TEE Client API v1.0c, interface interne pour les TAs : GlobalPlatform TEE Internal API v1.0
    - http://www.globalplatform.org/specificationsdevice.asp

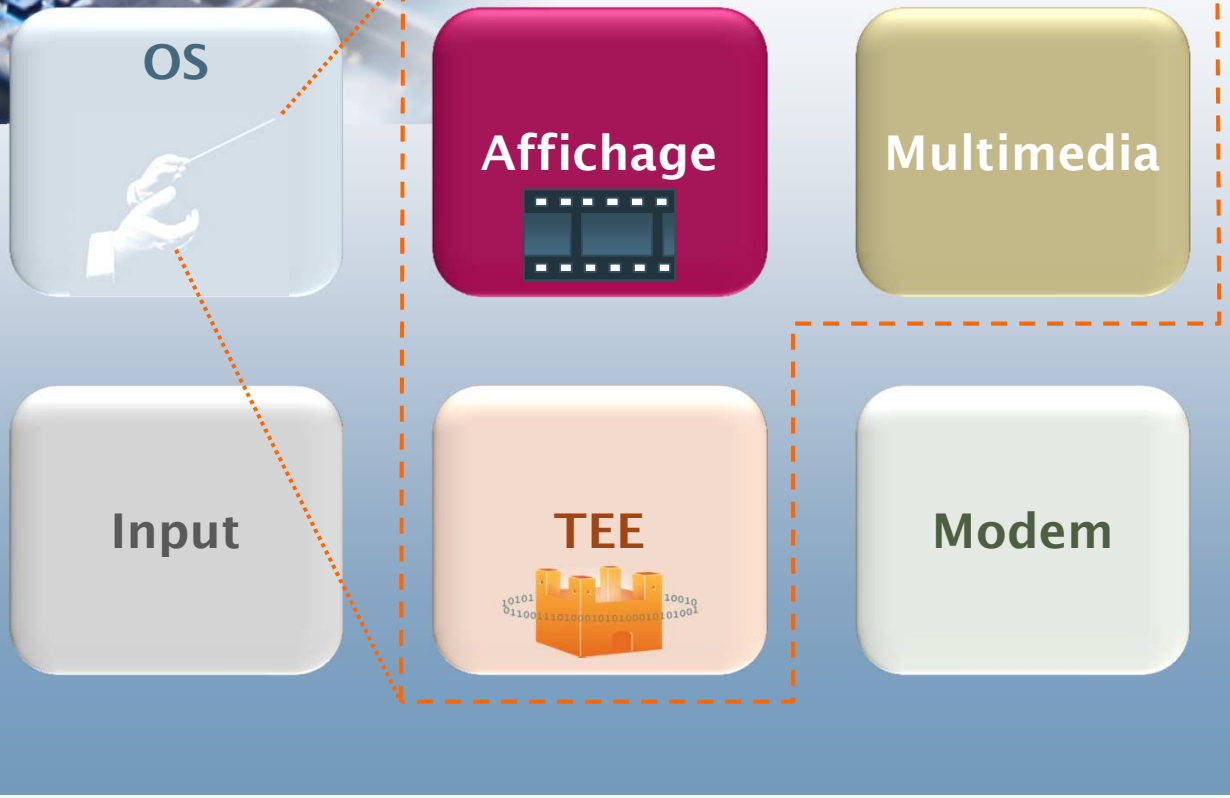
Troisième ingrédient : des applications



Exemples d'applications

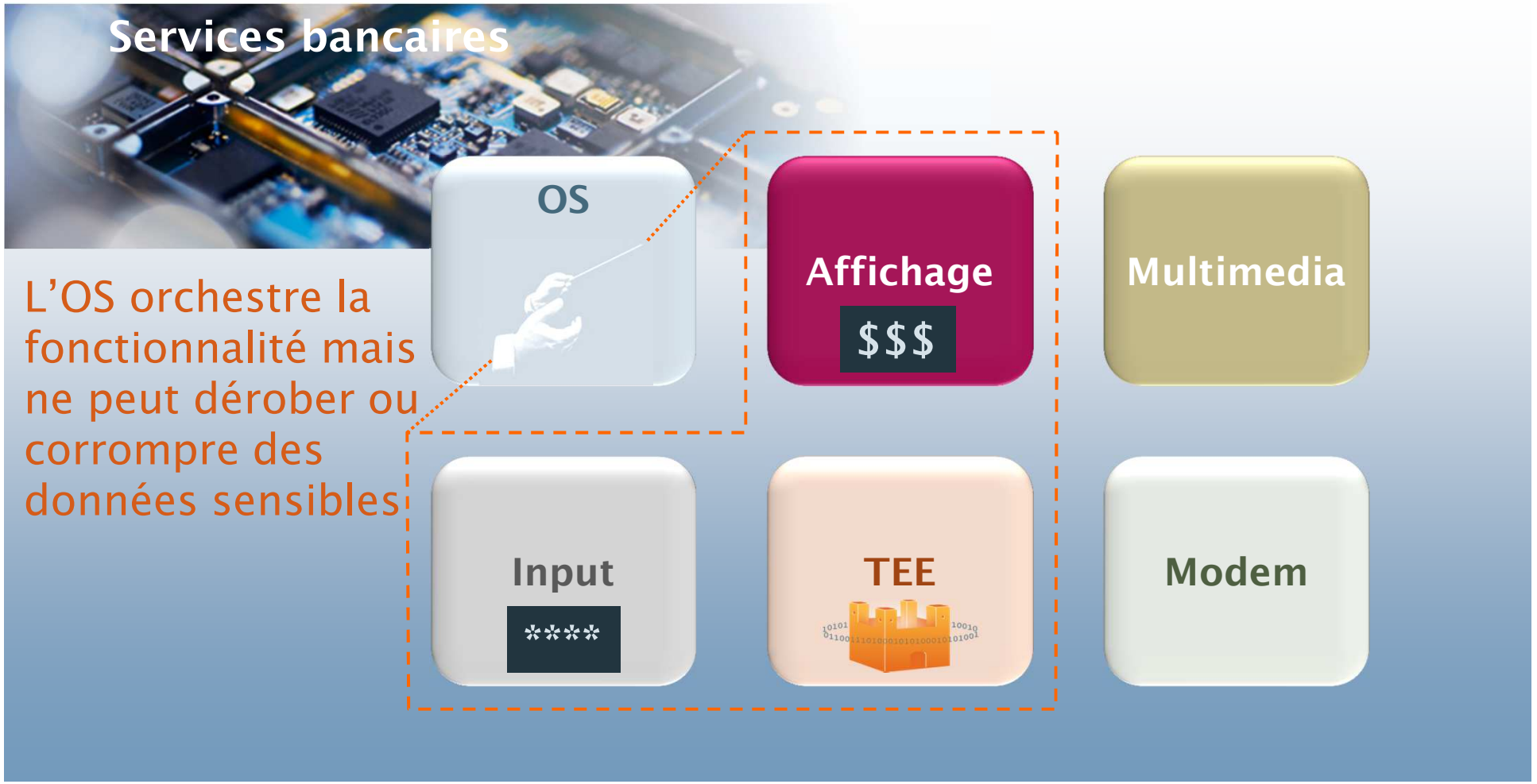
Protection de contenus

L'OS orchestre la fonctionnalité mais ne peut dérober ou corrompre des données sensibles



Exemples d'applications

Entrée sécurisée de code PIN



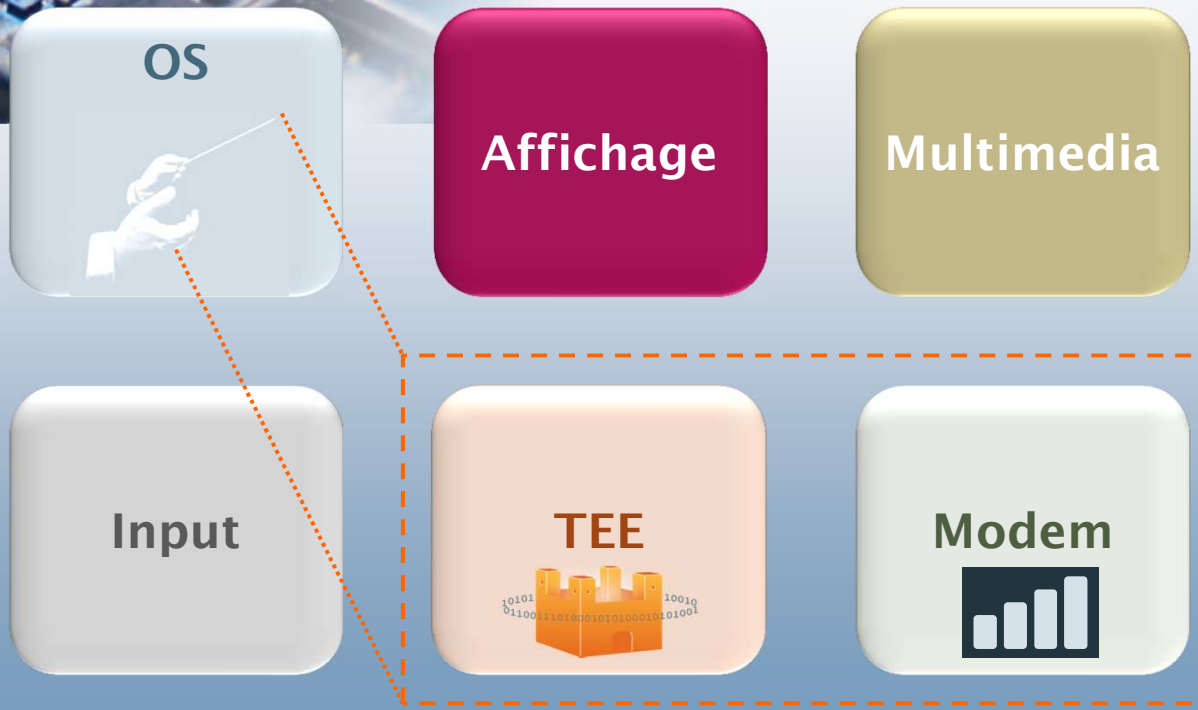
L'OS orchestre la fonctionnalité mais ne peut dérober ou corrompre des données sensibles

Exemples d'applications

Gestion de device

Subsidy Lock - SIMlock

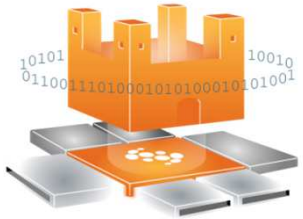
L'OS orchestre la fonctionnalité mais ne peut dérober ou corrompre des données sensibles



Et la sécurité ?

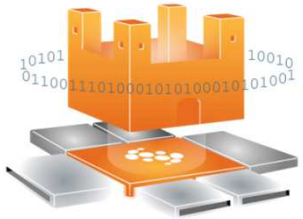
ST ERICSSON

Dans le TEE ST-Ericsson



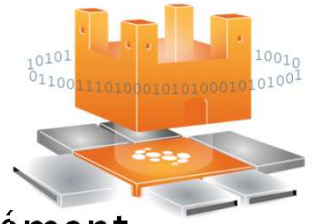
- Vérification du code
  - Le noyau du TEE est chargé et vérifié par le secure boot, on-chip
  - Les TAs sont signées et leur signature est vérifiée au chargement
- Isolation
  - Le noyau du TEE est exécuté on-chip, en mode Secure privileged
  - Les TAs tournent en DDR en mode Secure user
  - La MMU Secure (descripteurs en ROM et RAM on-chip protégée par HW) permet d'isoler les TAs du noyau et les TAs entre elles
- Stockage sécurisé
  - Les données des TAs sont protégées cryptographiquement et stockées dans un FS Linux
  - NVM dédiée TEE (RPMB de l'eMMC) pour la protection contre le rollback
- Autres protections
  - NX pour les user TAs et le noyau (partiellement), todo : ASLR

Cible de sécurité



- Un intermédiaire entre OS ouvert et Secure Element
  - Pas (ou peu) de résistance aux attaques matérielles
  - Beaucoup plus intégré au SoC qu'un SE → plus de fonctionnalités, plus de ressources, mais aussi plus d'interactions et de surface d'attaque
- Attaques sur TrustZone
  - Par canaux cachés (timing…)
  - Autres ? (bugs HW / absence de séparation physique)
- Attaques sur TEE
  - Dan Rosenberg, http://vulnfactory.org/blog/2013/04/08/motorola-bootloader-unlocking/
  - GlobalPlatform TEE Internal API = C…
- Une cible de choix
  - "Next generation mobile rootkits", Thomas Roth, Black Hat EU '13

Profil de Protection TEE

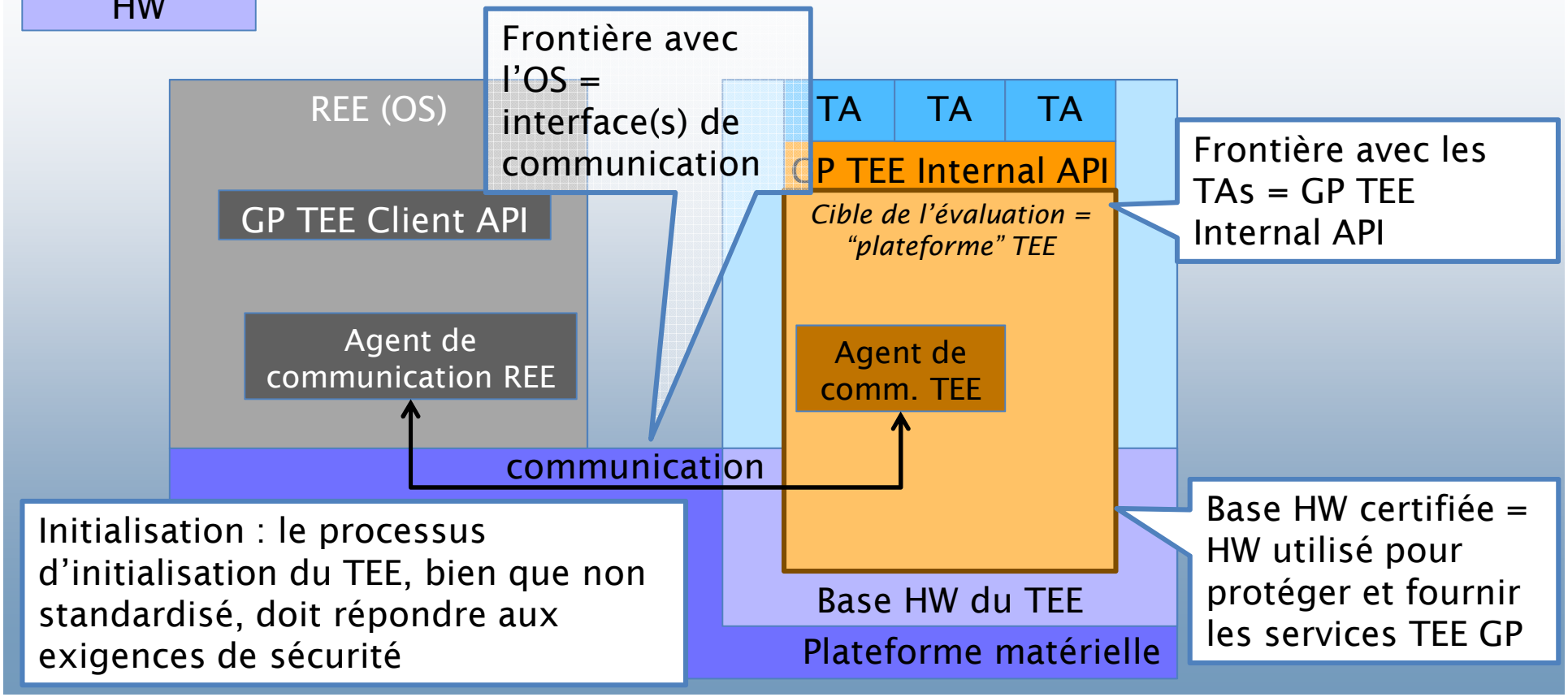


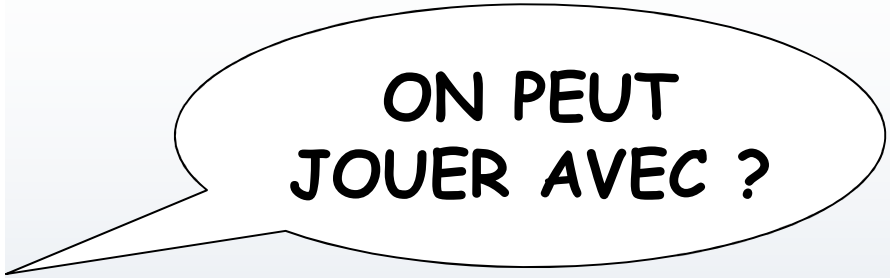
TEE complet

GP TEE

HW

Objectif = prévention contre les attaques reproductibles aisément
Modèle d'attaque = attaque SW/HW pour déterminer une vulnérabilité exploitable par logiciel uniquement, puis attaque logicielle
Niveau d'assurance EAL2+ (potentiel d'attaque augmenté)





Stay tuned!

herve.sibert@stericsson.com

ST ERICSSON