NGSCB : Mythes et Réalités

Bernard Ourghanlian
Directeur de la Technologie et de la Sécurité
bourghan@microsoft.com

Microsoft France

Microsoft

Un changement de nom

- Le 24 janvier dernier Microsoft a annoncé sa volonté de changer le nom de « Palladium » qui était jusqu'à présent le nom de code d'un projet interne à Microsoft
- « Palladium » s'appelle maintenant *Next-Generation Secure Computing Base* (NGSCB)

Sommaire

- Introduction et motivation
- Architecture
- Hardware et noyau de sécurité (Nexus)
- Applications
- Politique
- Résumé
- En guise de conclusion
- Questions - Réponses

Introduction et motivation

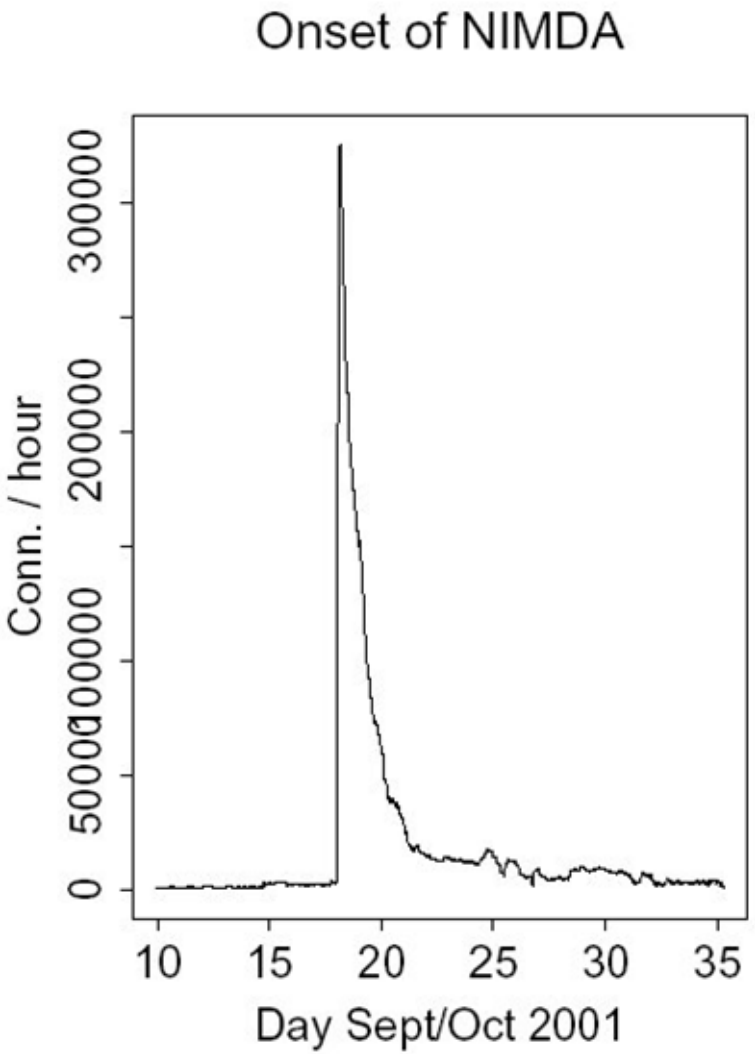
NIMDA et *Code Red*

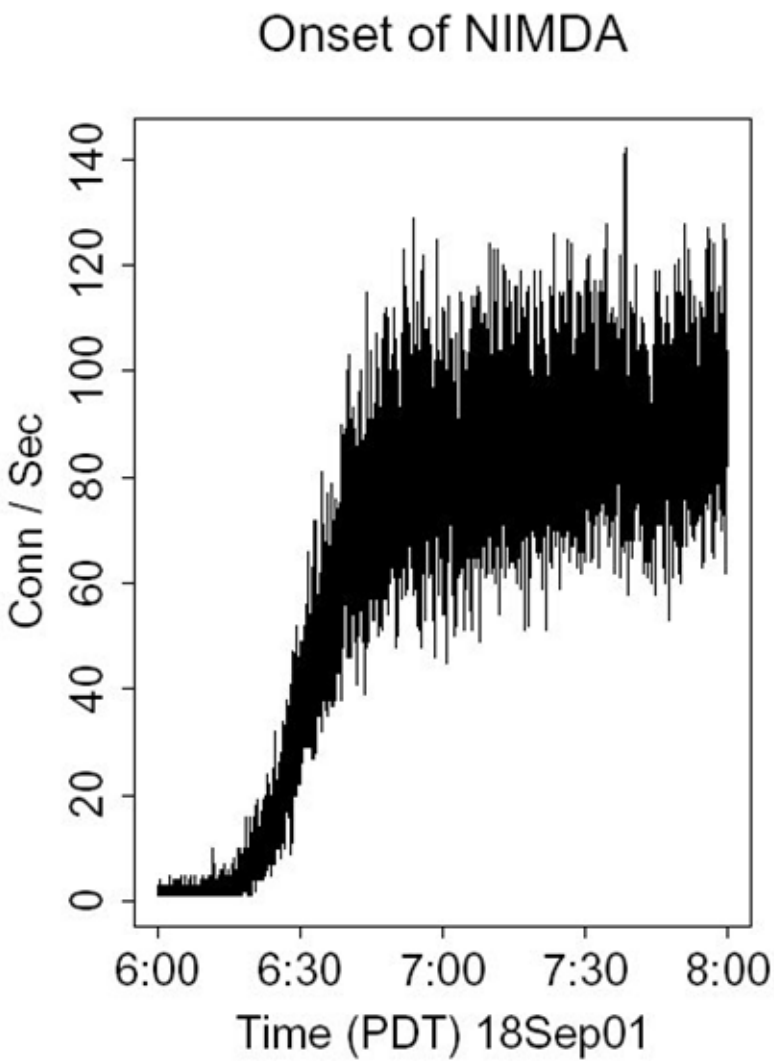
	Famille *Code Red*	Famille *NIMDA*
Estimation de la vitesse de propagation*	150 000 machines infectées en 14 h	2,2 millions de machines infectées en 24 h
Impact économique mondial estimé en 2001*	2,62 milliards de dollars	635 millions de dollars

***Computer economics Carslbad, CA**

- Dans certains cas, certaines sociétés ont été forcées de se couper temporairement de l'Internet
  - Exemples publics : Siemens, US Bancorp Piper Jaffray, Booz Allen & Hamilton et General Electric

La virulence de *NIMDA*



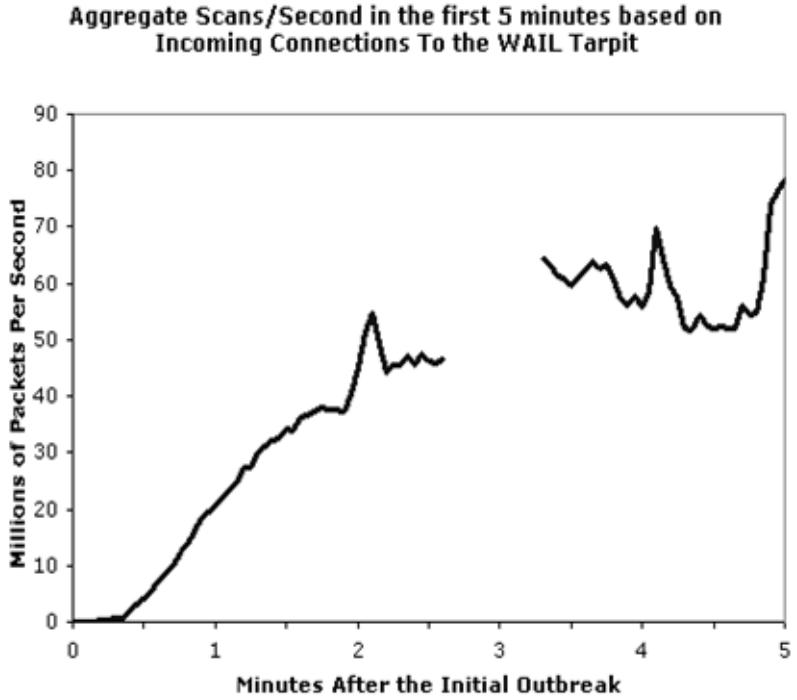


Source : Stanison, Paxson, Weaver. *How to 0wn the Internet in Your Spare Time.*

Microsoft

Slammer

- Un saut quantique dans la vitesse de propagation des virus
  - Au début de la propagation du virus sur Internet, doublement du nombre de systèmes parcourus toutes les 8,5 secondes
  - 90 % des systèmes vulnérables infectés en 10 minutes



Source : University of Winsconsin Advanced Internet Lab (WAIL)

Microsoft

Les risques se généralisent

Enquête *Information Security Breaches 2002*

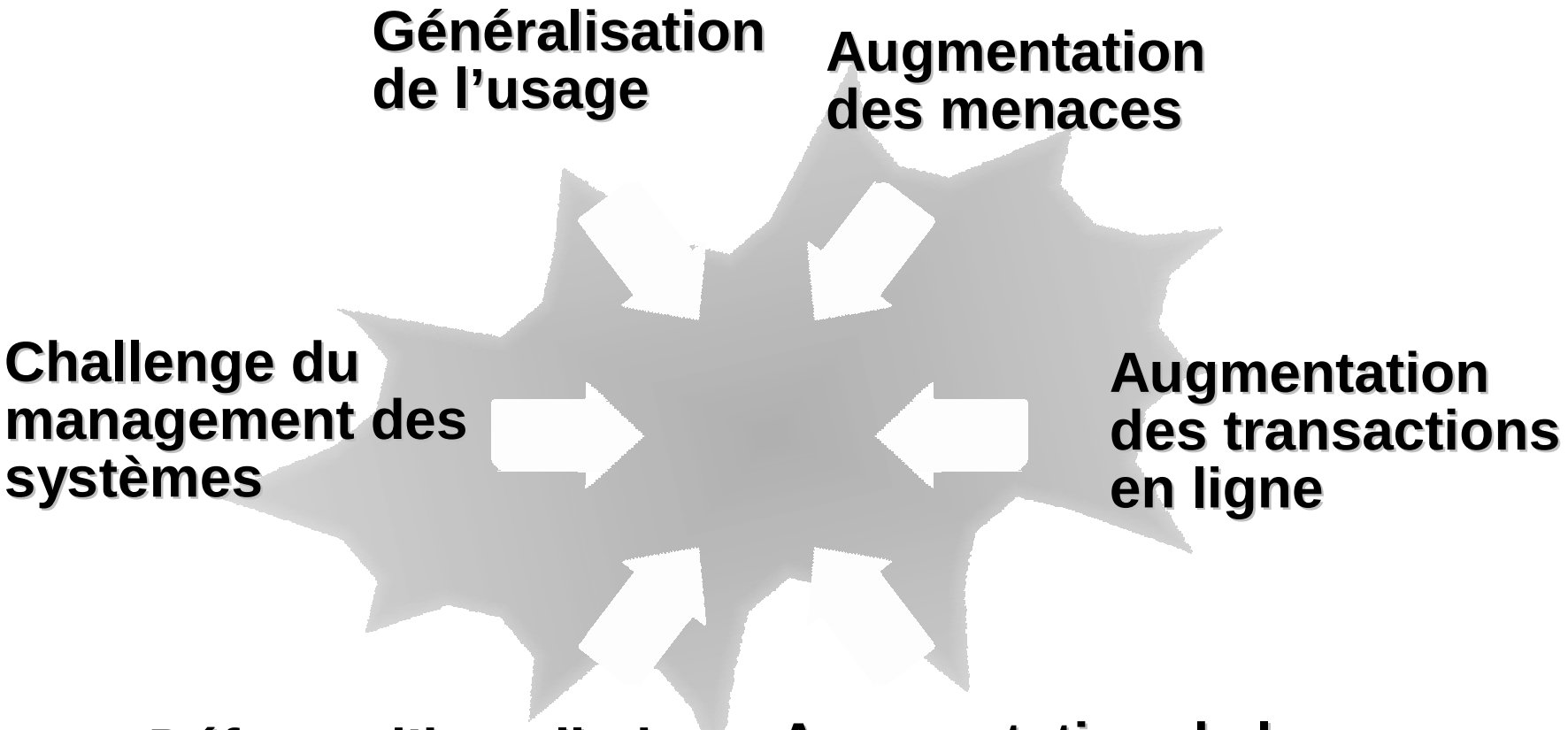
- 78 % des grandes entreprises en Angleterre ont souffert d'incidents de sécurité
- 44 % de l'ensemble des entreprises affectées
- 41 % de l'ensemble des entreprises ont souffert d'infections virales (triplement depuis 2000)

CERT

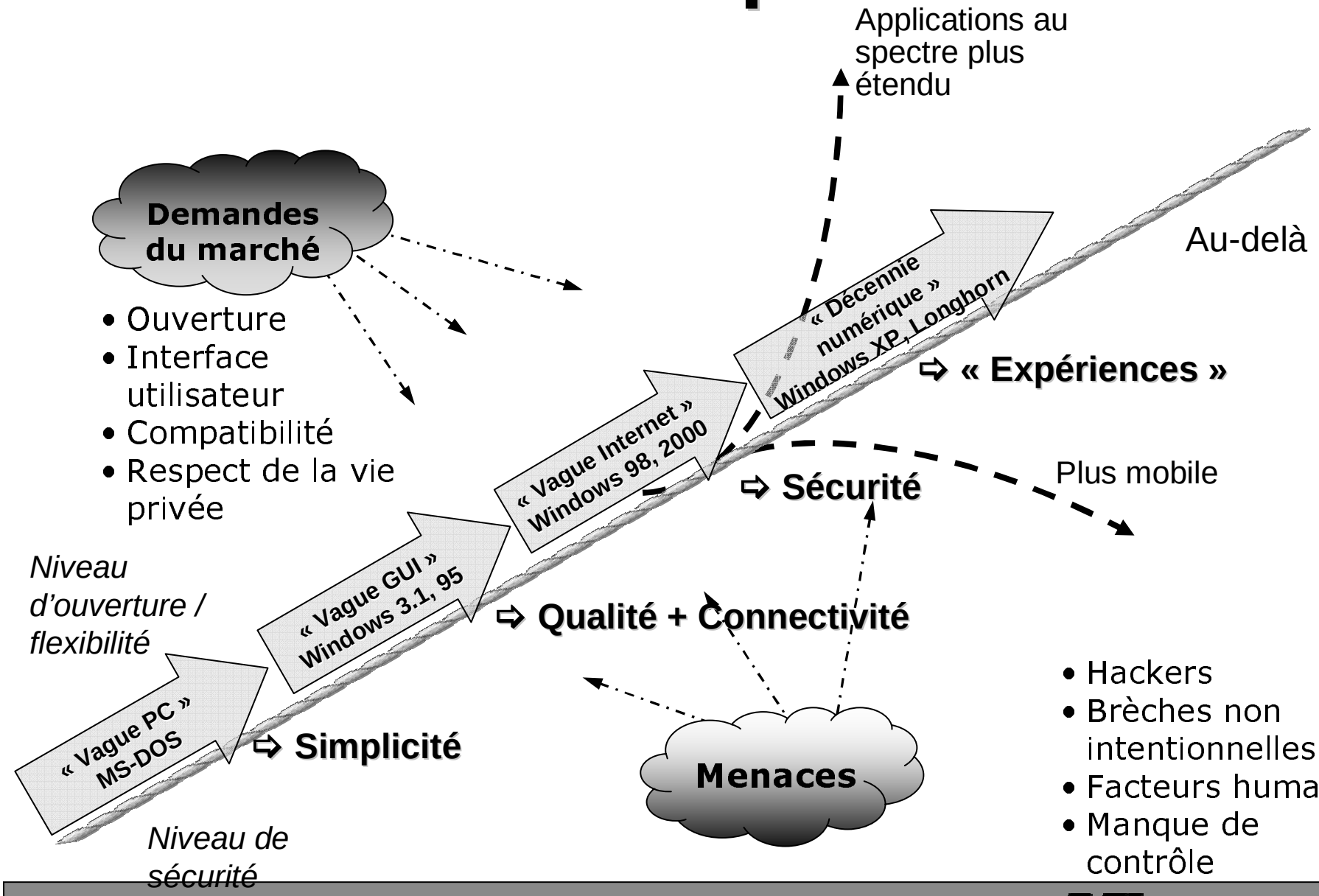
- 95 % de toutes les brèches de sécurité aux U.S sont dues à des problèmes de mauvaise configuration

Source: Information Security Breaches Survey 2002, PricewaterhouseCoopers
Source: CERT, 2002

Nous vivons une période de choc frontal

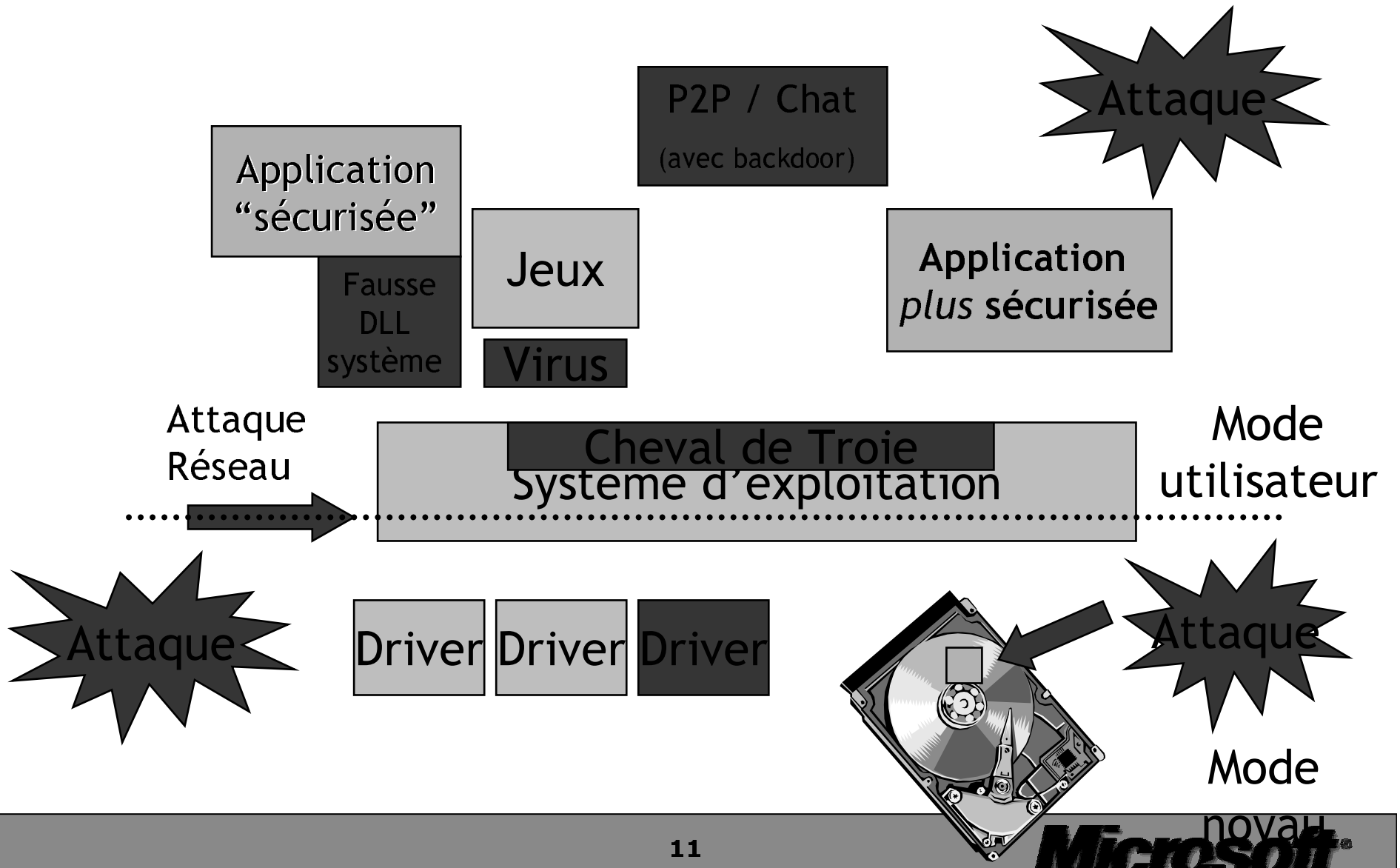


Pourquoi NGSCB?



Microsoft

Le PC aujourd'hui



Microsoft

Les buts de NGSCB

- Quelques façons de le dire…
  - « Protéger le logiciel du logiciel »
  - « Permettre d'enrichir la sécurité par des notions d'intégrité de la machine et du logiciel »
  - « Rendre le PC aussi sécurisé qu'il est flexible »
  - « Permettre aux applications Windows sur un PC d'être autant dignes de confiance que leurs consoeurs s'exécutant dans un autre environnement »
    - Sécurité du téléphone versus VoIP

Un logiciel digne de confiance…

- Une extension du concept d'authentification
  - Aujourd'hui, les *principals* peuvent prouver leur identité à travers le réseau en utilisant des mécanismes de chiffrement
    - Les gens, les machines,… peuvent signer des déclarations, s'authentifier mutuellement, déchiffrer des secrets…
    - … Grâce à des clés privées de signature et à des clés publiques de vérification de signature
  - Si l'on ajoute la possibilité pour les programmes de s'authentifier mutuellement, il devient impossible de mentir…
    - On peut encore utiliser le hardware que l'on veut, le logiciel que l'on veut…
    - …Mais on ne peut plus prouver ce qui n'est pas vrai

Un logiciel digne de confiance…

- Un programme comme un *principal…*
  - On peut distinguer un programme d'un autre au travers du réseau
  - On peut utiliser le hachage d'un programme comme son identité
  - L'identité du programme définit ses comportements possibles
  - On peut distinguer les « bons » programmes des « mauvais »
    - Ceci ne nous aide pas à écrire des « bons » programmes
    - Ceci ne nous aide pas non plus à décider quels programmes sont « bons »
  - On peut parler de groupes de programmes
  - On peut implémenter de nombreuses politiques (de sécurité et de respect de la vie privée) fondées sur l'identité du programme

Les principes fondateurs

1. NGSCB sera construit à partir des plus hauts standards en matière de sécurité et de respect de la vie privée.
2. Un PC NGSCB doit être capable de *booter* tout OS compatible NGSCB et d'exécuter des logiciels en provenance de tous les fournisseurs tout comme les PC d'aujourd'hui.
3. Le *Trusted Computing Base* (TCB) de Microsoft sera disponible (les sources) pour examen.
4. Un PC NGSCB doit être capable d'exécuter les applications et les *drivers* d'aujourd'hui.
5. Quiconque peut écrire des applications Windows pour un PC pourra écrire des applications tirant parti de NGSCB.
6. NGSCB n'arrêtera pas le piratage.
7. Il n'est pas nécessaire de disposer des informations sur l'utilisateur pour permettre à NGSCB de fonctionner.
8. NGSCB peut ne pas être capable de résister à des attaquants disposant d'un accès physique à une machine mais empêchera toute attaque de type BORE (*Break Once, Run Everywhere*).
9. NGSCB permettra la mise en place de tout type de politique de sécurité et de respect de la vie privée.
10. Les systèmes NGSCB donneront les moyens de protéger la vie privée mieux que tout système d'exploitation aujourd'hui.

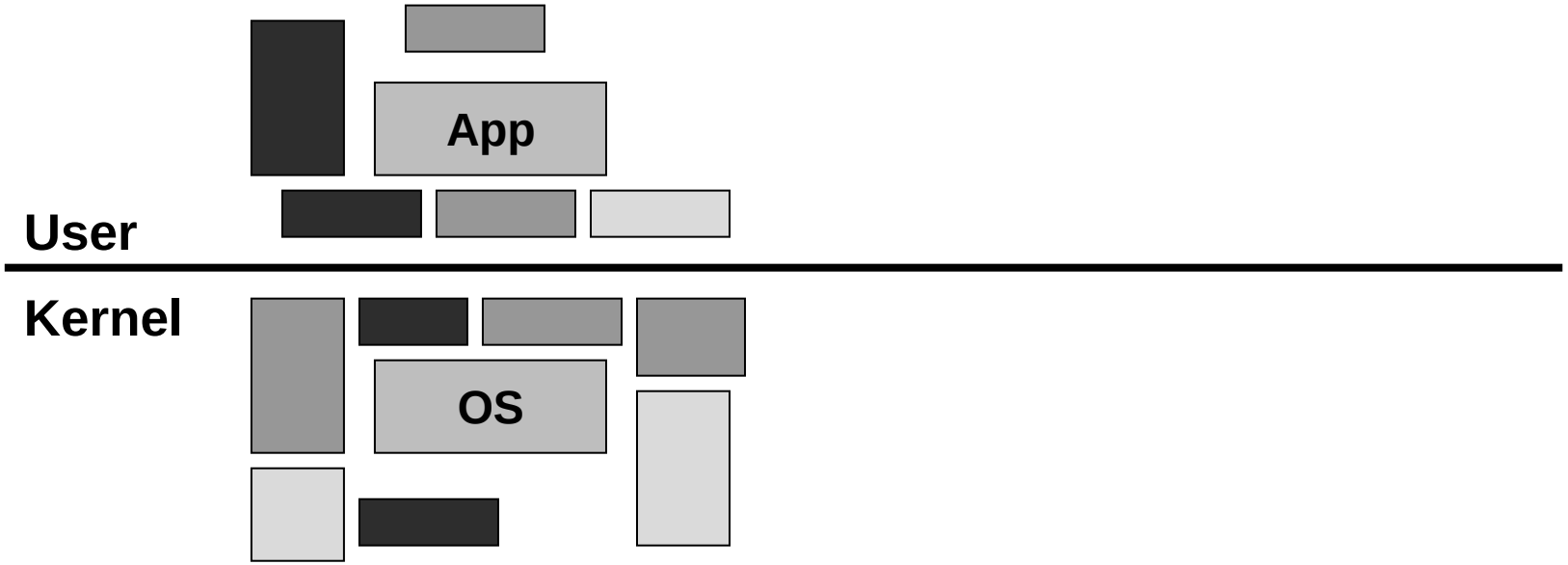
Qu'est-ce que NGSCB ?

- NGSCB est un ensemble de nouvelles fonctionnalités de sécurité « au sein » de Windows
  - Autorisées grâce à un nouveau hardware
  - Correspondant à un nouveau logiciel : *Trusted Security Kernel* (Nexus) et *Nexus Computing Agents*
- L'objectif est de « protéger le logiciel du logiciel »
  - Défense contre des codes malicieux qui s'exécutent eu sein du *Ring 0*
- Permet de mettre en service et de protéger un *Trusted Computing Base* (TCB) décentralisé dans le cadre de systèmes « ouverts »

Mécanismes : construire dynamiquement un périmètre de sécurité

- Ensemble de mécanismes
  - Isolation du logiciel (mémoire cloisonnée – établissement du TCB)
  - Authentification du logiciel (attestation - extension du TCB)
  - Secrets pour le logiciel (stockage scellé – persistance de l'état du TCB)
  - Entrées et sorties sécurisées (échanges « dignes de confiance » pour l'utilisateur)
- Assertions de sécurité, permissions et authentification fondées sur des références (*credentials*)
  - Lampson, Rivest, Abadi, etc.

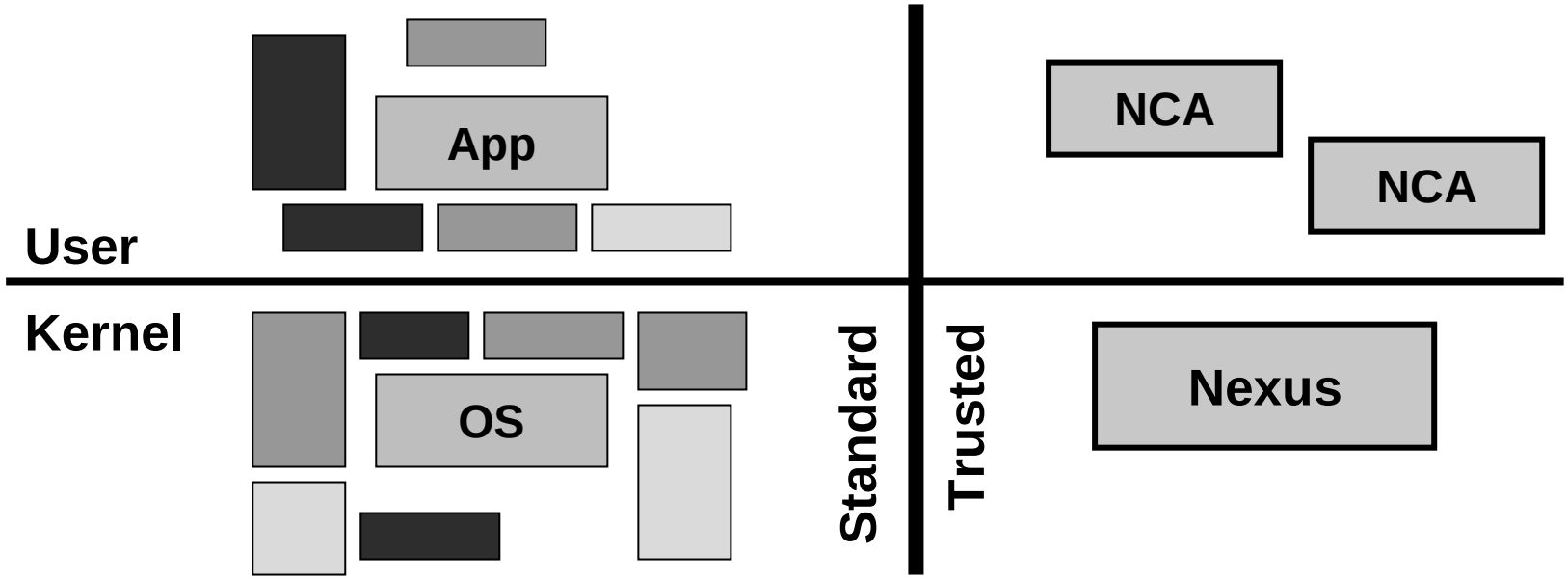
NGSCB « vu d'avion » (1/3)



- Comment préserver la flexibilité et l'extensibilité qui ont tant contribué à la richesse de l'écosystème du PC, tout en fournissant à l'utilisateur final un environnement sûr ?
- En particulier, comment peut-on garder quoi que ce soit de secret, quand des composants du noyau enfichables contrôlent la machine ?

NGSCB « vu d'avion » (2/3)

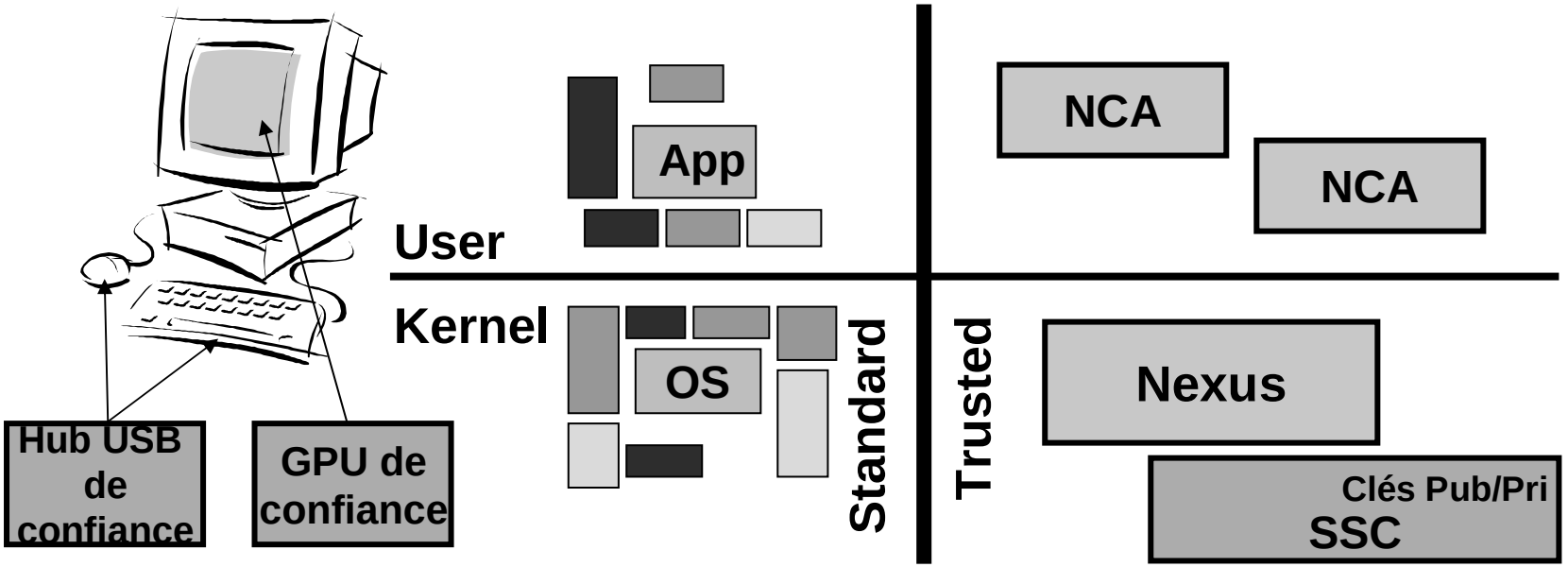
- La solution : subdiviser l'environnement d'exécution en ajoutant un nouveau mode au CPU



- Le CPU est soit en mode « standard » soit en mode « trusted »
- Les pages de la mémoire physique peuvent être marquées comme « trusted ». Les pages dites « trusted » ne peuvent être accédées que lorsque le CPU est en mode « trusted »

NGSCB « vu d'avion » (3/3)

- Les agents ont aussi besoin de laisser l'utilisateur entrer des secrets et d'afficher des secrets pour l'utilisateur



- Les entrées sont sécurisées par un « hub » USB de confiance pour le clavier et la souris qui permet de transporter une conversation sécurisée avec le *nexus*
- Les sorties sont sécurisées par un GPU de confiance qui transporte une conversation chiffrée avec le *nexus*
- Ceci permet une sécurité de bout en bout

Une mise en perspective (1/3)

- NGSCB propose une approche innovante de la sécurisation des PC : une cryptographie « intelligente » minimise le nombre des secrets intégrés au matériel afin que les concepteurs puissent se consacrer à la protection d'un petit ensemble de données
  - NGSCB repose sur des clés associées à la signature numérique de son *nexus* ; modifier le *nexus* revient à changer la signature et se traduit par la création d'un nouvel espace de clé, inutilisable si une relation de confiance n'est pas établie
  - NGSCB est une évolution de la spécification TCPA (*Trusted Computing Platform Alliance*) qui procure un point d'ancrage à partir duquel sont mises en place les fondations d'un environnement « digne de confiance »
    - Voir plus loin

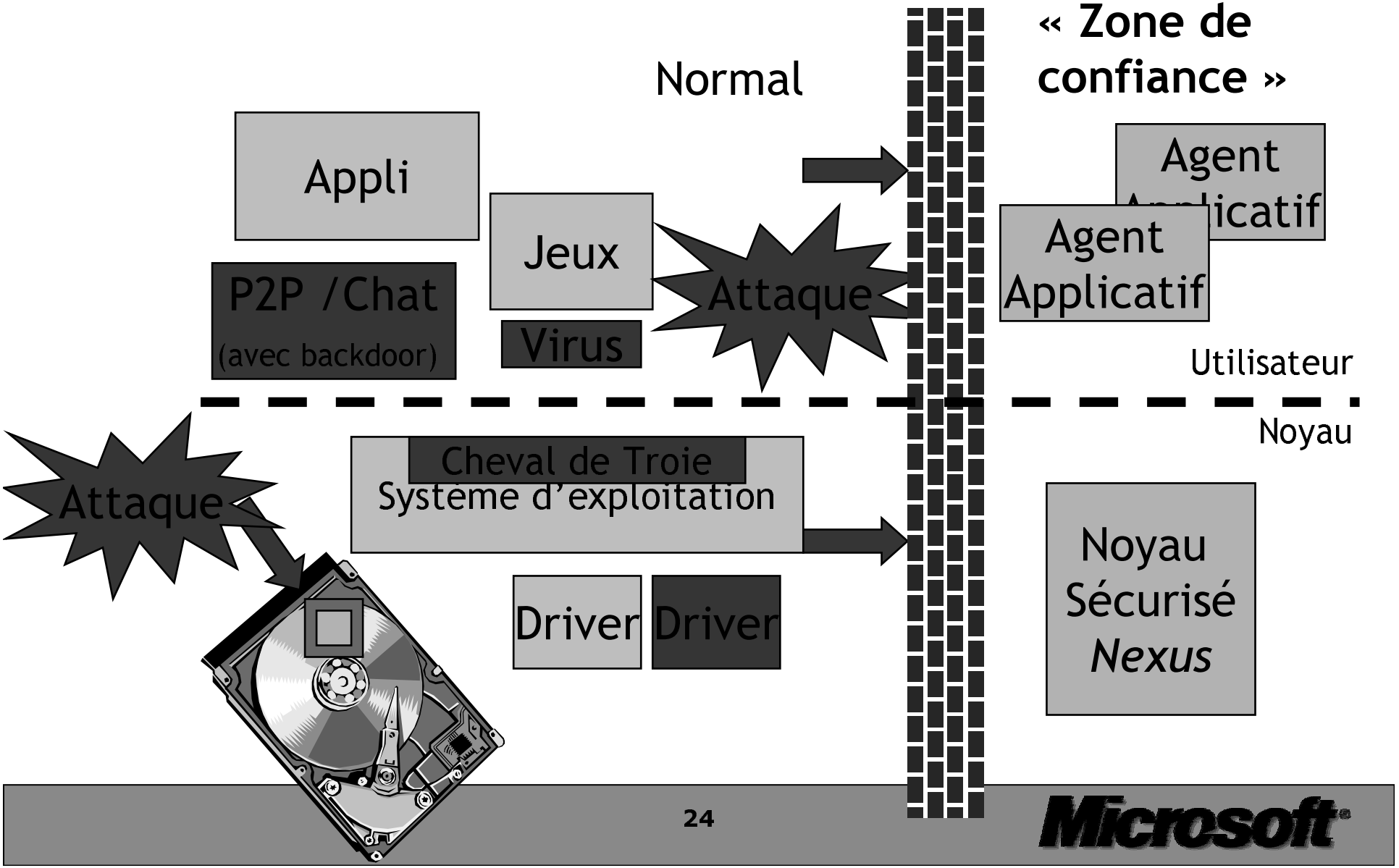
Une mise en perspective (2/3)

- L'architecture de NGSCB permet de résoudre le problème de la protection et de la sécurisation des 30 millions de lignes de code de Windows
  - Elle crée un environnement distinct, sécurisé dans lequel sont exécutés les composants de NGSCB ; ceux-ci sont liés à Windows par des interfaces de programmation (API) spécifiques, de manière à réduire encore le champ des opérations relatives à la sécurité
  - Le *nexus* de NGSCB peut être relativement petit et n'intègre que des fonctions de sécurité

Une mise en perspective (3/3)

- L'architecture présente également l'avantage de ne pas nécessiter une refonte du système d'exploitation Windows : le composant sécurité des applications sensibles peut être géré sans difficulté dans l'environnement NGSCB
  - Il n'y a donc aucune remise en cause des logiciels existants
  - Les sous-systèmes de sécurité peuvent évoluer à au rythme plus lent des logiciels sécurisés
- Le niveau de confiance dont bénéficie l'environnement Windows sans NGSCB reste cependant inchangé

Architecture NGSCB



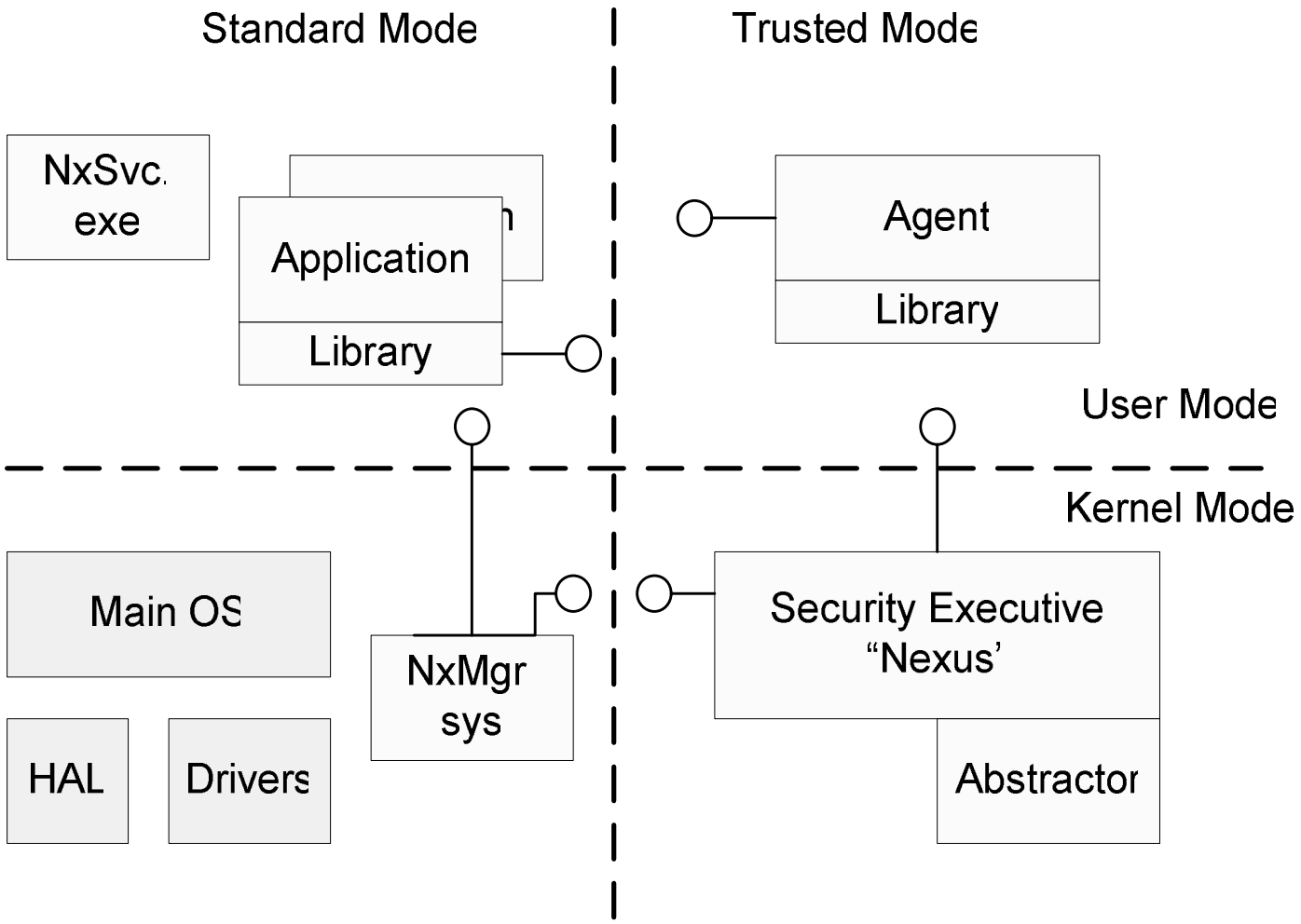
Microsoft

Architecture

Comment fonctionne NGSCB ?

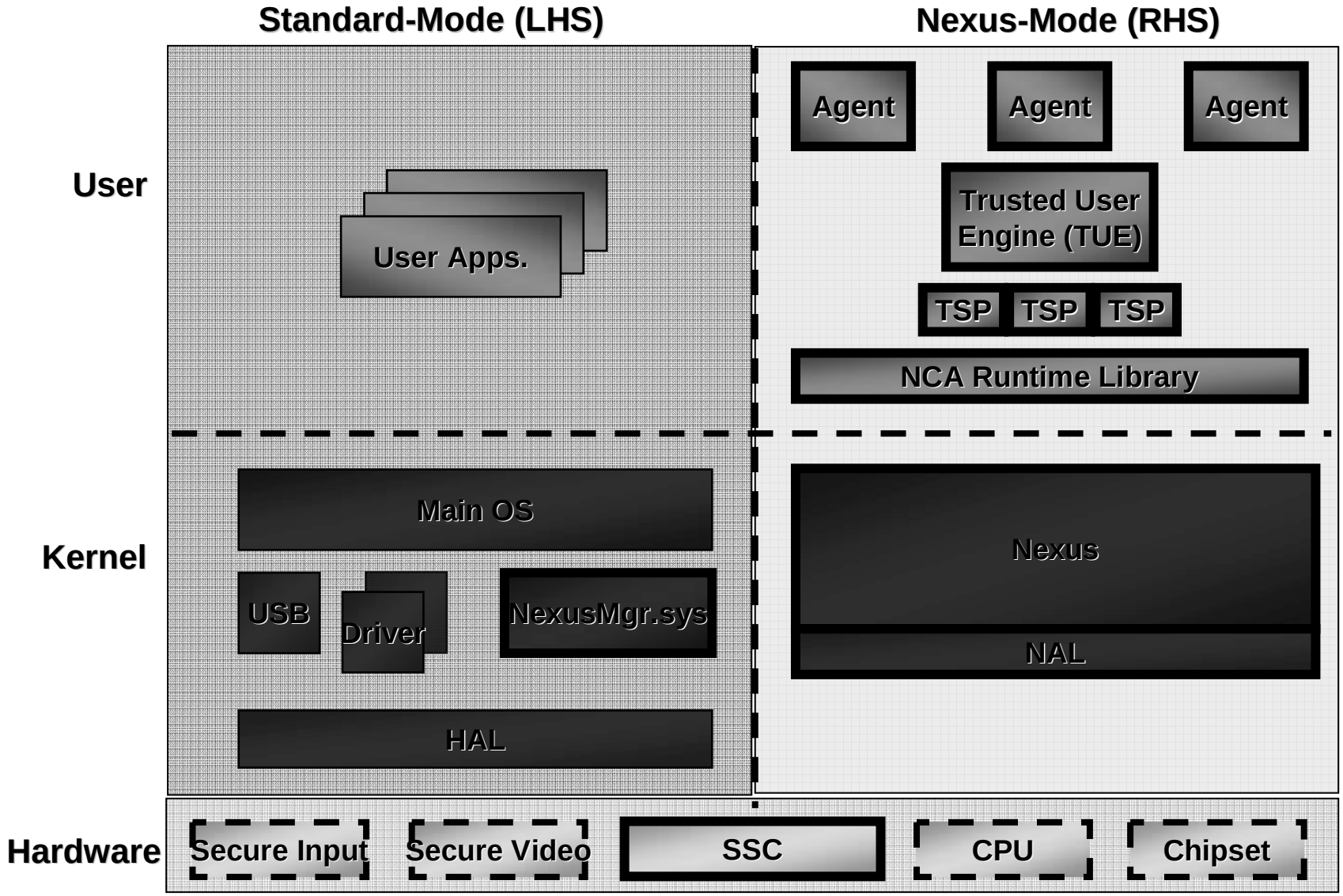
- Tire parti d'améliorations CPU (nouveaux modes et nouvelles instructions) pour cloisonner une zone de mémoire protégée
- Un petit noyau de sécurité (le « nexus ») permet d'abstraire le hardware et de fournir l'environnement de programmation
- Les composants logiciels qui utilisent les secrets s'exécutent « derrière le mur » (ce sont les « *Nexus Computing Agents* » ou NCA)
- Les secrets sont liés à l'identité du logiciel et à la plate-forme
- L'interaction avec l'utilisateur est sécurisée au travers de canaux sécurisés vers la vidéo, le clavier et la souris

Aperçu du système



Microsoft

De manière plus précise



Microsoft

Qu'est-ce que le n*exus* fournit à ses NCA ?

- Un environnement d'exécution séparé protégé pour les applications (NCA)
- Les NCA peuvent
  - Etre autonomes
  - Fournir des services aux applications
- L'utilisateur indique au *nexus* quelles NCA autorisées à s'exécuter (la politique d'exécution du *nexus* utilise les identités des NCA)
- Le *nexus* scelle les secrets pour tout NCA et peut « attester » de l'identité du NCA
- Fournit des services OS simples : *threads*, mémoire, IPC, E/S dignes de confiances, accès aux services de la partie gauche du schéma précédent (mode standard)

Les caractéristiques et les objectifs de NGSCB

- Les anciens OS fonctionnent sans modification (ou presque)
- Les anciens drivers fonctionnent sans modification (ou presque)
- Toutes les anciennes applications fonctionnent sans modification
  - Rien n'est « cassé » si l'on n'a aucune connaissance ou notion de NGSCB
- Les NCA doivent être développées pour NGSCB
  - Nous livrerons quelques agents autonomes
  - Nous livrerons quelques agents de type service système

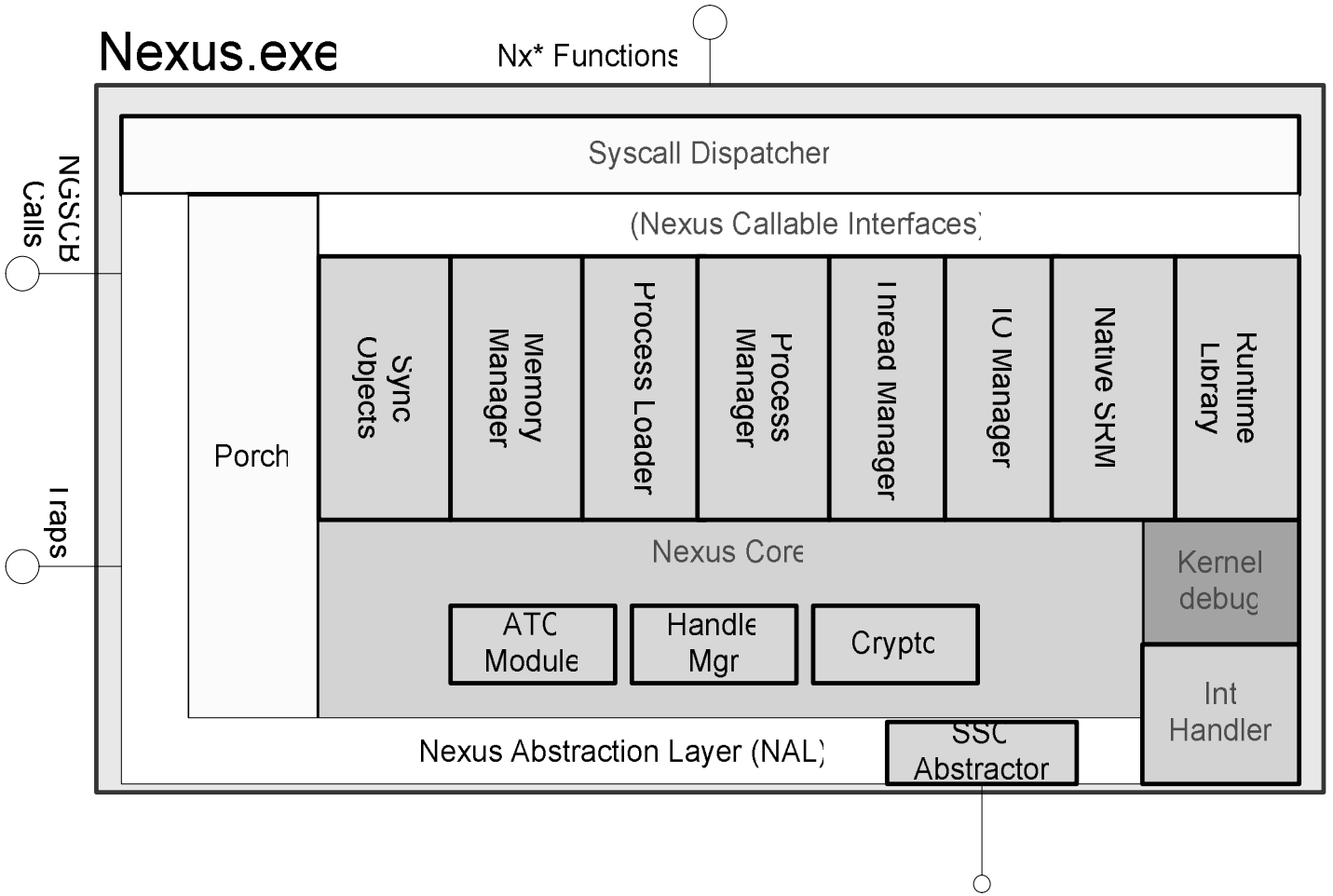
Le *nexus* comme système d'exploitation : ce qui est familier

- Espace d'adressage privé
- Contient les EXE
  - (peut éventuellement supporter des DLL)
- Notion de droit de propriété
- *Process-Control Block* normal
- Droits d'accès
- Création de *threads*, etc…

Le *nexus* comme système d'exploitation : ce qui est différent

- La séparation des processus est plus forte
  - Le système d'exploitation principal et les applications sont exclus les uns des autres de manière inconditionnelle
  - Le déboguage et l'inspection de la mémoire par le *nexus* et les agents est strictement contrôlé
- Le code qui peut être chargé dans un NCA est défini par la politique du NCA
- Les NCA ont un accès privilégié à une ou plusieurs clés de chiffrement (en fonction de l'identité du code)
  - La base de l'authentification et de l'autorisation
  - Egalement décentralisé

Une vue plus précise sur le Nexus



Microsoft

Le modèle de sécurité de NGSCB

- Les agents ont *moins* de privilèges que les applications (en general)
- Le fait d'être protégé lorsque son code s'exécute ne signifie pas que l'on soit protégé sur disque
- Le lien manquant est constitué par les *services de chiffrement*
  - Ce sujet sera couvert dans quelques instants …
- L'identité du code est un concept clé au sein de NGSCB

L'identité du code pour le *nexus* (1/4)

- Le hachage de la séquence d'instructions composant le *nexus* correspond à l'identité de ce *nexus* pour NGSCB (utilisation d'un algorithme de type SHA-1 renvoyant une valeur sur 160 bits)
  - Les utilisateurs spécifient les *nexuses* qui peuvent s'exécuter par leur signature numérique (*hash*)
  - Le *Nexus ID* est utilisé dans les opérations de scellement, de descellement et d'attestation
- La hachage suit une logique, à la fois dans le processeur, le *chipset* et le SSC

L'identité du code pour le *nexus* (2/4)

- La signature du *nexus* peut être générée par condensé du code en cours de chargement dans la section de mémoire sécurisée ou par hachage sur place du code après chargement
- Amorcer un *nexus* (noyau de sécurité) demande au SSC de calculer le hachage du *nexus* et de stocker sa valeur (sur 160 bits) dans un registre en lecture seule du SSC (PCR – *Platform Configuration Register*)
  - Changer le *nexus* change son identité
  - Le PCR détermine quelles sont les clés auxquelles a accès le *nexus*
- Aucune signature n'est nécessaire pour exécuter du code
  - Si l'utilisateur indique que le code correspondant à une valeur de hachage est autorisé à s'exécuter, celui-ci s'exécutera

L'identité du code pour le *nexus* (3/4)

- Le composant SSC peut émettre un certificat unique, authentifiant la version du *nexus* selon une authentification de bas niveau du PC
  - Par conséquent, un tiers est en mesure de vérifier que le PC exécute une version du *nexus* connue et digne de confiance
    - Le *nexus* est nécessairement intact puisque tout changement est répercuté au niveau du condensé
  - Cette valeur peut également servir, par l'entremise d'une clé privée stockée dans le SSC, à déchiffrer les clés locales au *nexus* et à ses agents de confiance
  - Un autre condensé (produit par une autre version du *nexus*) générera un certificat différent qui invalidera toutes les clés correspondant aux diverses autres versions du *nexus*
- Cette approche établit une distinction efficace entre authentification et identification

L'identité du code pour le *nexus* (4/4)

- Ce processus donne quelques indications sur les attaques matérielles susceptibles de compromettre la sécurité du système
  - Un pirate cherchera à charger un *nexus* modifié dans l'espace mémoire, tout en présentant un signature valable au SSC
  - Ce *nexus* serait alors en mesure de lui transmettre n'importe quelle information confidentielle et d'émuler toutes les identifications produites
- Ce type d'attaque est redoutable mais nécessite un accès physique au PC

Imiter la signature du *Nexus* (1/2)

- Imaginons que les plates-formes NGSCB donnent le contrôle total de l'environnement *nexus* à un *nexus* modifié, écrit par un pirate, et susceptible d'inclure des commandes de débogage de base autorisant un utilisateur à accéder aux secrets de NGSCB…
- La signature de ce *nexus* étant différente, toutes les clés sont transférées dans un autres espace non reconnu par l'environnement NGSCB initialement établi : le système n'est plus digne de confiance et les clés n'ont aucune valeur
  - Le pirate va donc chercher à charger un *nexus* modifié, tout en présentant une signature acceptée par le SSC

Imiter la signature du *Nexus* (2/2)

- Les bus situés de part et d'autres du contrôleur système sont vulnérables à ce type d'attaque
  - Le bus mémoire est difficile à intercepter en raison du nombre de signaux acheminés et du haut débit de données
  - En revanche, la connexion au SSC s'effectue via le bus LPC (*Low Pin Count*) qui est relativement étroit et lent
    - Il est cependant aisé d'enfouir ce bus à l'intérieur de la carte, afin d'en rendre l'accès difficile pour le pirate amateur
    - Il est d'ailleurs possible qu'à terme les transactions soient cryptées sur ce bus
- En revanche, les intrusions au niveau des puces électroniques de NGSCB sont à la portée des gouvernements et des entreprises désireuses d'investir dans cette technologie

Manifestes : identification du code pour les applications

- Objectif : permettre la « manageabilité » (éventuellement au détriment de la simplicité)
  - Création d'identités par hachage pour des « familles de code »
  - Pas nécessairement un simple condensé de séquences d'instructions
  - Spécifie la mise à jour et le déboguage
- Manifeste : nomme les « familles de code » en utilisant des preuves cryptographiques (hachages, clés de signature, chaînes de certificats) :
  - « Agent de paiement par carte de crédit Visa certifié »
  - « Agent MS-Money + mise à jour »
- Le « condensé de manifeste » peut être utilisé dans le stockage fourni par le *nexus* et par les fonctions d'attestation

Les détails matériels

Les changements du matériel

- Changements du CPU
- Changements de la MMU
- Les changements du *Southbridge* (interface du bus LPC)
- Le *Security Support Component* (SSC)
  - Nouveau composant sur la carte mère (bus LPC)
- Le *hub* de confiance USB
  - Peut être sur la carte mère, dans le clavier ou n'importe où entre les deux
- GPU digne de confiance

SSC : *Security Support Component*

- Penser à « une carte à puce soudée sur la carte mère »
- Bon marché, à périmètre fixe
- Contient
  - Au moins une clé AES et une paire de clés RSA
  - En réalité, il pourra contenir deux ou trois clés AES et deux ou trois paires de clés RSA
    - Ces clés AES & les clés privées RSA ne quittent jamais le chip
  - Des registres : le PCR (*platform configuration register*) qui contient le condensé du *nexus* en cours d'exécution
    - Le *nexus* n'a pas besoin d'être chargé lors du *boot* ; il peut être chargé plus tard
      - Tout ce dont a besoin NGSCB est le condensé du *nexus*
    - Ceci peut survenir à n'importe quel moment après le *boot* de la machine ; donc, quand on *boote* le *nexus* on hache le *nexus* et on place la valeur de ce hash dans le PCR

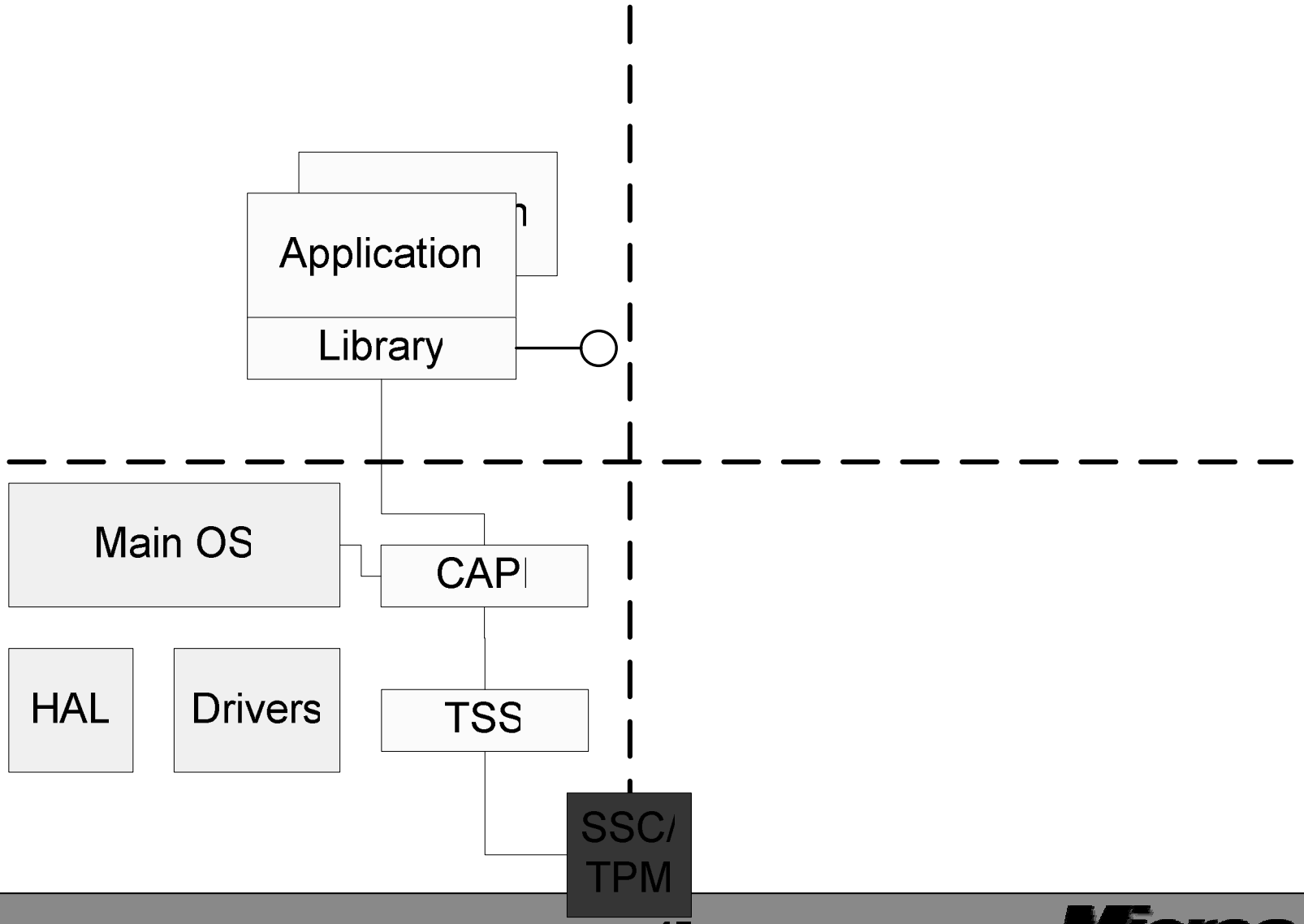
SSC : *Security Support Component*

- Doit être proche du *chipset* (c'est-à-dire pas dans une véritable carte à puce) car il est impliqué dans l'initialisation du *nexus*
- SSC sera peut-être intégré et supporté dans la version 1.2 du TPM de TCPA (nous l'espérons)

NGSCB et TCPA

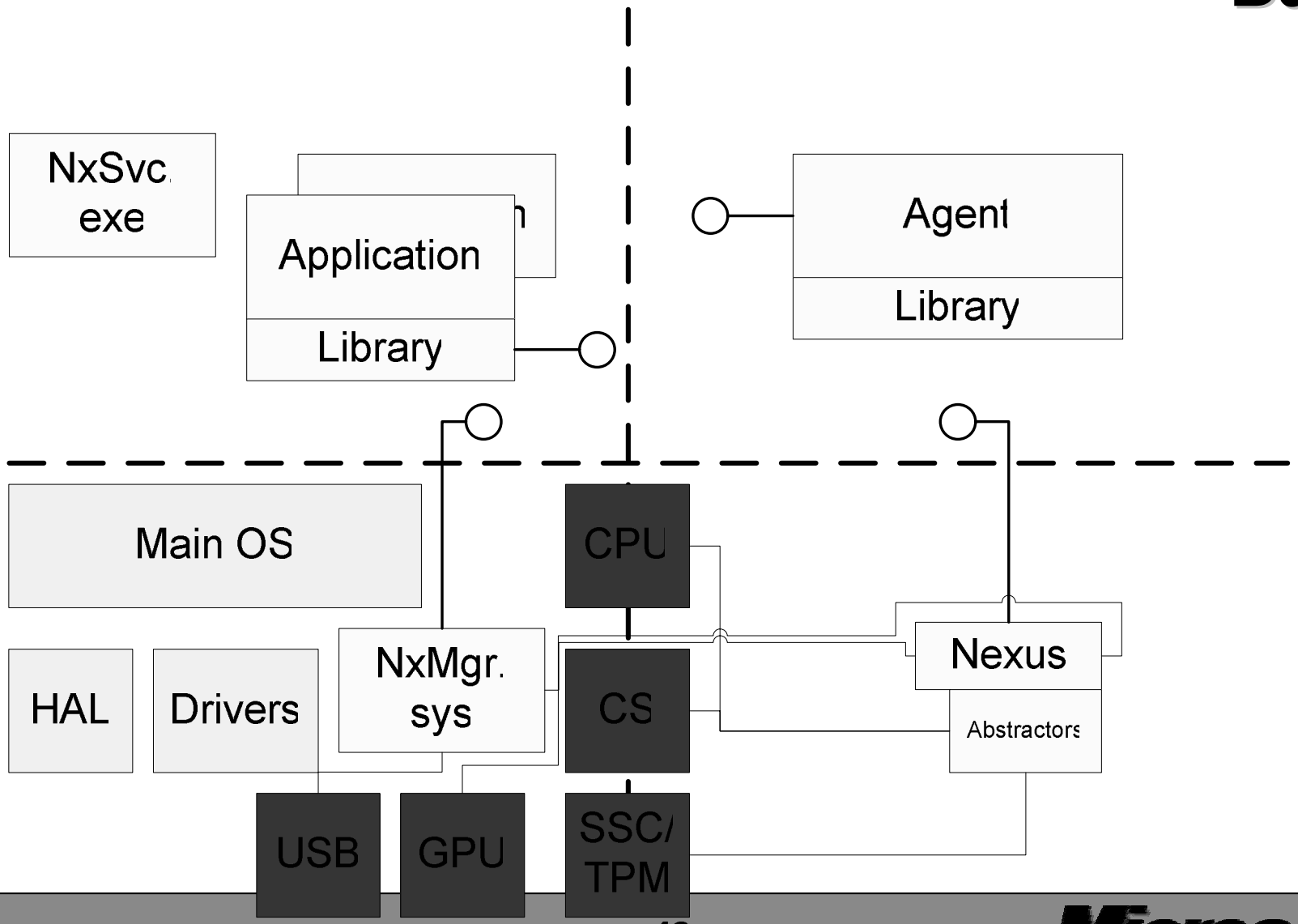
- Les fonctionnalités actuelles de TCPA constituent un sous-ensemble de NGSCB
  - Avec quelques petites différences clé
- Les spécifications 1.2 du TPM de TCPA pourront peut être constituer une solution pour les besoins du SSC
  - Les spécifications actuelles du TPM ne sont pas suffisantes pour NGSCB
  - On peut penser NGSCB comme « une maison que l'on cherche à construire, les spécifications TCPA du TPM étant le verrou que l'on peut utiliser pour la porte »
- Microsoft, membre fondateur de TCPA, travaille activement sur les spécifications et les recommandations

TCPA



Microsoft

Next-Generation Secure Computing Base



Microsoft

TCPA et TCG

- TCG = Tructed Computing Group (http://www.trustedcomputinggroup.org)
- Organisation à but non lucratif dont les membres fondateurs sont AMD, Hewlett Packard, IBM, Intel, Microsoft
  - Controbuteurs : Atmel, Gemplus, Infineon, Phoenix Technologies, Phillips, National Semiconductor, Nokia, nVidia, STMicroelectronics, Sony, VeriSign, Wave Systems
- TCG a adopté les spécifications de TCPA comme point de départ
- Les groupes de travail TCPA ont arrêté leurs travaux ; les groupes de travail TCG ont commencé les leurs
- Les membres de TCPA ont été invités à rejoindre TCG

TCPA et TCG

- Pourquoi un nouvel organisme ?
  - Changement du nom pour bien marquer la différence avec TCPA
  - Améliorations des structures afin d'accélérer le développement des spécifications
    - Modèle de gouvernance comparable aux autres organismes de normalisation (vote à la majorité,…)
    - Périmètre de la plate-forme plus large : PC, serveurs, PDA, téléphones mobiles
    - Périmètre logiciel plus large : TSS (*TPM Software Stack*)
    - Politique d'échange réciproque de propriété intellectuelle (RAND)
    - Programme de « logo » pour permettre de donner confiance aux utilisateurs
  - Augmentation sensible des ressources
    - TCG supporté par tous les membres
    - Amélioration du développement et de la communication

Algorithmes de chiffrement

- Le SSC ne pourra pas être mis à jour sur site
  - En effet, le *firmware* permettant de réaliser des mises à jour des éléments de sécurité de manière sûre reposerait sur les mêmes algorithmes
- Le SSC admettra des numéros de versions
- La version 1 du SSC nécessite
  - RSA 2048 + PKCS #1 V2.1
  - AES-128
  - SHA-1
  - HMAC (*Keyed-Hashing for Message Authentication*)
  - Un générateur de nombre aléatoires FIPS 140-2
  - Un petit nombre de compteurs monotones
- Le SSC contient au moins une paire de clés RSA, une clé AES-128 et une clé HMAC

Initialisation du *nexus*

- L'amorçage du *nexus* est initié par le système d'exploitation principal
  - Techniquement, c'est un chargement, pas un amorçage
  - Le code du *nexus* est chargé en mémoire physique
  - A souvent besoin du support du *chipset* afin de permettre l'exclusion des maîtres du bus
- Etapes pour initialiser un *nexus* :
  - *Reset* du CPU
  - Protection du *nexus* des maîtres du bus (y compris les DMA)
  - Détermination de l'identité du *nexus*
    - Calcul du hachage de la séquence d'instructions du *nexus*
  - Passage du contrôle au *nexus*

Les services du hardware pour le *nexus*

- Le hardware fournit au *nexus* les fonctions suivantes :
  - Isolation forte des processus
  - Fonctions de chiffrement et déchiffrement par *nexus* pour la protection des secrets persistants
  - Chemin sécurisé de et vers l'utilisateur
  - Attestation
- Les mécanismes d'attestation sont originaux
  - Les faits à propos des différents « éléments » (logiciels, utilisateurs, machines, services) peuvent être prouvés à (et crûs par) des entités distantes
  - Utilisation de preuves de type *zero knowledge*
- Le *nexus* rend service à ses NCA
  - Les services du *nexus* pour les NCA peuvent être un peu plus riches

Stockage scellé
(permettre au code de garder des secrets)

- Fonction de scellement et de descellement du SSC
  - Seal(secret, condensé du *nexus*, *nexus* cible) -> Blob
    - Demande « chiffre moi ce secret de telle manière que seul le *nexus* cible puisse le retrouver »
  - UnSeal(Blob) -> secret (or error)
    - Si le *nexus* courant est celui qui est nommé dans le blob :
      - Retourner le secret
    - Autrement
      - Retourner une erreur
  - Implémentation : AES en utilisant la clé AES-128 du SSC
    - Entropie pour protéger contre la reconnaissance du même secret

Algorithme de scellement du SSC

Fonctionnement de l'algorithme de scellement :

1. *Generate a 128-bit random number R*
2. *Let D0 be the current value of the PCR[0], D1 = PCR[1]*
3. *DIGEST M = HMAC[K_M](R || S || Target(s) || D0 || D1)*
4. *C = AES[K_S](R || S || Target(s) || D0 || D1 || M)*
5. *Return SSC_SUCCESS error code and ciphertext C*

Stockage sécurisé (scellé)

- Et les données sur le disque dur ?
- Le stockage scellé est une technologie de chiffrement
  - Le *nexus* obtient une clé du SSC et fournit des clés dérivées aux applications
  - Microsoft Money, Outlook, Windows Media Player obtiennent des clés différentes
  - Chaque application stocke ses données en les chiffrant avec sa propre clé
- Correspond virtuellement à un disque dur local chiffré, différent pour chaque application
- Les processus (système ou applicatif) malicieux ne peuvent accéder à ces données … **quelle que soit leur identité de sécurité**

Attestation

- L'attestation permet à un client distant de savoir quel logiciel s'exécute y compris potentiellement :
  - Système d'exploitation / *Nexus*
  - Application
  - Politique Client (contrôle virus, accès administrateur, etc.)
- L'attestation est une technologie d'authentification
  - Mais plus complète qu'une « simple signature »
  - Permet, en effet, l'authentification d'une configuration logicielle complète (*nexus*, application, processus)

Attestation
(Comment le code s'authentifie lui-même)

- Fonction « Quote » du SSC
  - Quote (string) -> Sign[string | Nexus-digest]
- Bloc de base du protocole :
  - Serveur/pair :
    - Contrôle la signature
    - Contrôle les certificats sur la clé de signature
    - Contrôle le condensé du *nexus*
  - Permet d'affirmer « *Nexus* Microsoft sur la plate-forme de confiance Lambda »
- Implémentation : RSA en utilisant la paire de clés du SSC

Attestation au niveau du SSC

- Il n'y a pas d'anonymat à ce bas niveau mais on ne peut utilisé la clé hardware qu'une seule fois lors de la mise sous tension de la machine
- Pour préserver l'anonymat, il est possible d'utiliser la clé hardware pour créer des pseudo-identités afin de fournir indirection/anonymat tout en préservant la capacité d'attestation de la plate-forme
  - La capacité de pouvoir créer des pseudo-identités nécessite la présence de tiers de confiance fournissant ce genre de service, ce qui n'est pas le cas aujourd'hui
    - Microsoft cherche à encourager l'émergence de ce type de marchés

Contrôle de l'état de sécurité

- ReadMonotonicCounter
  - Input: counter id
  - Output: counter value
- IncrementMonotonicCounter
  - Input: counter id
- Remarque : pas d'horloge temps réel sécurisée
  - Tout ce dont nous disposons sont des compteurs monotones
  - Permet de détecter un éventuel retour en arrière
- Autres fonctions : générateur de nombre aléatoire

La chaîne de certificats

- NGSCB s'appuie sur une chaîne de signatures numériques pour valider l'intégrité de la plate-forme
- Une plate-forme NGSCB inclura les signatures de la plupart ou de la totalité des acteurs de la phase de production, notamment
  - Un tiers de confiance, certifiant le fabricant du SSC
  - Le fabricant du SSC, certifiant que le SSC est digne de confiance
  - L'assembleur de la carte mère, certifiant l'assemblage du SSC sur une carte mère autorisée
  - Le fabricant OEM du système, certifiant que la chaîne d'assemblage de la carte mère est intacte
- Ces certificats seront ajoutés à la plate-forme via du matériel sécurisé (lui-même probablement fondé sur NGSCB)
  - Les tiers auront ainsi la certitude d'avoir affaire à un véritable système NGSCB dans lequel ils peuvent avoir toute confiance, et non à une imitation

NGSCB et confidentialité

- NGSCB présente l'avantage de séparer l'authentification et l'identification en deux processus totalement distincts
  - La chaîne de certificats établit l'authenticité de la plate-forme vis-à-vis des tiers sans qu'il soit possible d'identifier la plate-forme ou l'utilisateur spécifique
  - Le tiers a toute latitude pour accorder sa confiance à une plate-forme authentifiée
  - Si le propriétaire de la plate-forme accepte de fournir des données d'identification au tiers, celui-ci est authentifié par la plate-forme sécurisée
- Tout le processus repose sur un principe de confiance anonyme

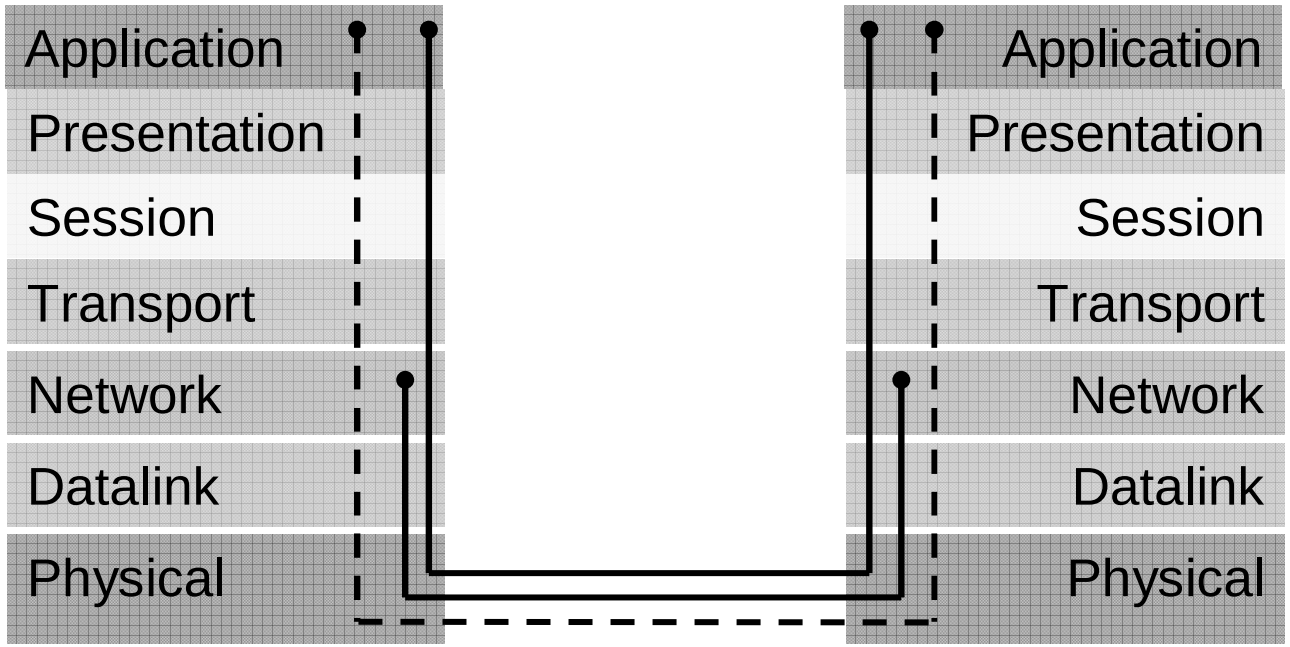
Applications

Applications

- Gestion du système
  - Amorçage sécurisé
  - Administration
    - Installation, changement de version et gestion de mise à jour
    - Login, gestion des clés / mots de passe, dispositif de chiffrement
    - Contrôle de la santé d'une machine, y compris le contrôle de virus
- Applications hautement sécurisées
  - Banques, transactions sécurisées
  - Gestion de l'information privée
- Ressources partagées
  - Kiosques
  - Utilisation d'applications d'entreprise (par les hommes et par les machines)

Un scénario en entreprise

- Connexion à distance



Application habituelle d'aujourd'hui : vulnérable

VPN : adéquat pour la couche réseau et en dessous ; tout ce qui est au-dessus est vulnérable

Application NGSCB : étend la passerelle au-delà de ce que permet le VPN ; maintenant toutes les couches sont protégées

Microsoft

Applications

- Applications collaboratives
  - Jeux multi-joueurs
  - Négociations
  - Enchères
- Contrôle d'accès décentralisé
  - Web Services
  - Authentification et autorisation inter-domaines
- DRM
  - Pour l'entreprise
  - Pour la vie privée du consommateur
    - Identité, données relatives à la santé et aux finances personnelles
  - Contenu numérique pour le marché de masse
    - Livres, films, audio, vidéo

Le sujet du DRM pour le marché de grande consommation est sujet à controverses

- Il y a des garde-fous dans le modèle NGSCB
  - Protection de la vie privée par rapport à l'utilisation potentielle d'identificateurs de plate-forme
  - Neutralité absolue par rapport à la politique mise en place
  - L'utilisateur a toujours le contrôle du code qui s'exécute
- Le potentiel pour la mise en place de politiques extrêmement restrictives existe
- Le succès du DRM requerra un effort des deux parties :
  - Les compagnies qui fournissent les contenus
  - Les opposants acharnés à ces compagnies
- Les politiques ne sont pas mises en place par les fournisseurs de technologie
  - Equilibre négocié

Les problèmes de politique

Problèmes de politique

- Certains des problèmes techniques que nous avons à résoudre pour faire de NGSCB un succès ont aussi des problèmes politiques qui lui sont attachés. Par exemple:
  - Comment, en pratique, construire un TCB « attestable » ?
    - « Attestable » == ouvert, auditable, compréhensible et que l'on puisse prouver à un tiers distant
  - Puisque la paire de clés RSA de NGSCB est unique à la plate-forme, quelles mesures doivent être prises pour défendre la vie privées de l'utilisateur (analyse éventuelle du comportement de l'utilisateur à travers l'analyse du trafic) ?

Politique pour le *nexus*

- Tout ce qui tourne aujourd'hui tournera du les systèmes NGSCB
- La plate-forme pourra exécuter tout *nexus*
  - L'utilisateur aura la responsabilité de choisir le nexus qu'il voudra utiliser
- Le *nexus* Microsoft pourra exécuter toute application
  - L'utilisateur aura la responsabilité de choisir les applications qu'il voudra exécuter
- Le *nexus* Microsoft interopérera avec tout fournisseur de service réseau
- Le code source du *nexus* Microsoft sera rendu disponible pour analyse détaillée

Respect de l'identité des machines

- Le problème : NGSCB utilise au moins deux ensembles de clés hardware uniques (une clé AES, une paire de clés RSA) :
  - Ces clés sont essentiellement équivalentes à des identificateurs uniques de machines
  - Et l'utilisation de ces clés est la seule manière de protéger votre environnement personnel !
- Stockage scellé :
  - Utilisé une clé unique AES mais les algorithmes sont :
    - Optionnels (l'utilisateur désigne quel logiciel peut accéder aux fonctions)
    - Aléatoires (on ne peut déterminer si deux textes chiffrés ont été créés ou non sur la même machine)
- Attestation :
  - Utilise une clé unique RSA mais est *conçue* pour authentifier la plate-forme
    - Optionnel (l'utilisateur désigne quel logiciel peut accéder aux fonctions)
    - Nous contrôlons strictement la divulgation de la clé d'authentification hardware
- Le hardware a des garde-fous très stricts par rapport au respect de la vie privée
  - L'accès aux composants liés à la clé publique RSA est strictement limité
  - Dans la conception actuelle de NGSCB, un seul export de la clé publique RSA est autorisé par démarrage électrique du PC

Quelques idées fausses « classiques » sur NGSCB

- NGSCB pourra censurer ou interdire du contenu sans la permission de l'utilisateur
  - La mise en place d'une telle politique, tel qu'est conçu NGSCB, n'est absolument pas possible
    - En effet, les applications compatibles NGSCB ne peuvent pas interférer avec d'autres applications NGSCB (de par la conception même de NGSCB) ni avec d'autres applications Windows traditionnelles
- NGSCB mettra hors du marché les fournisseurs non approuvés par Microsoft
  - Il n'y a aucune signature Microsoft requise pour utiliser NGSCB
- NGSCB n'est pas contrôlé par l'utilisateur
  - Tous les programmes NGSCB ne pourront s'exécuter que si ils sont autorisés par l'utilisateur

Quelques idées fausses « classiques » sur NGSCB

- NGSCB est un « super » moyen de répandre des virus
  - Les applications NGSCB ne s'exécutent pas avec des privilèges élevés
- Un NCA de NGSCB n'est pas déboguable
  - Si. Un drapeau dans le manifeste permet de mettre en service le déboguage
- VMware ne pourra pas s'exécuter en environnement NGSCB
  - VMware s'exécutera sans problème en environnement NGSCB et les applications compatibles NGSCB pourront être déboguées comme n'importe quelle autre application ; cependant, si une application stocke un secret qu'elle ne veut pas rendre visible à une autre application (ou à Windows), ce secret ne sera pas visible pour VMware ou pour un quelconque débogueur

Quelques idées fausses « classiques » sur NGSCB

- NGSCB surveillera l'utilisation de votre ordinateur et en informera le fournisseur et/ou Microsoft
  - Un des principaux objectifs de conception de NGSCB est d'interdire à quiconque (y compris Microsoft) d'espionner votre machine et de capturer les séquences de caractères entrées au clavier
- D'autres idées fausses en http://www.microsoft.com/france/securite/entreprises/palladium/

Résumé (1/2)

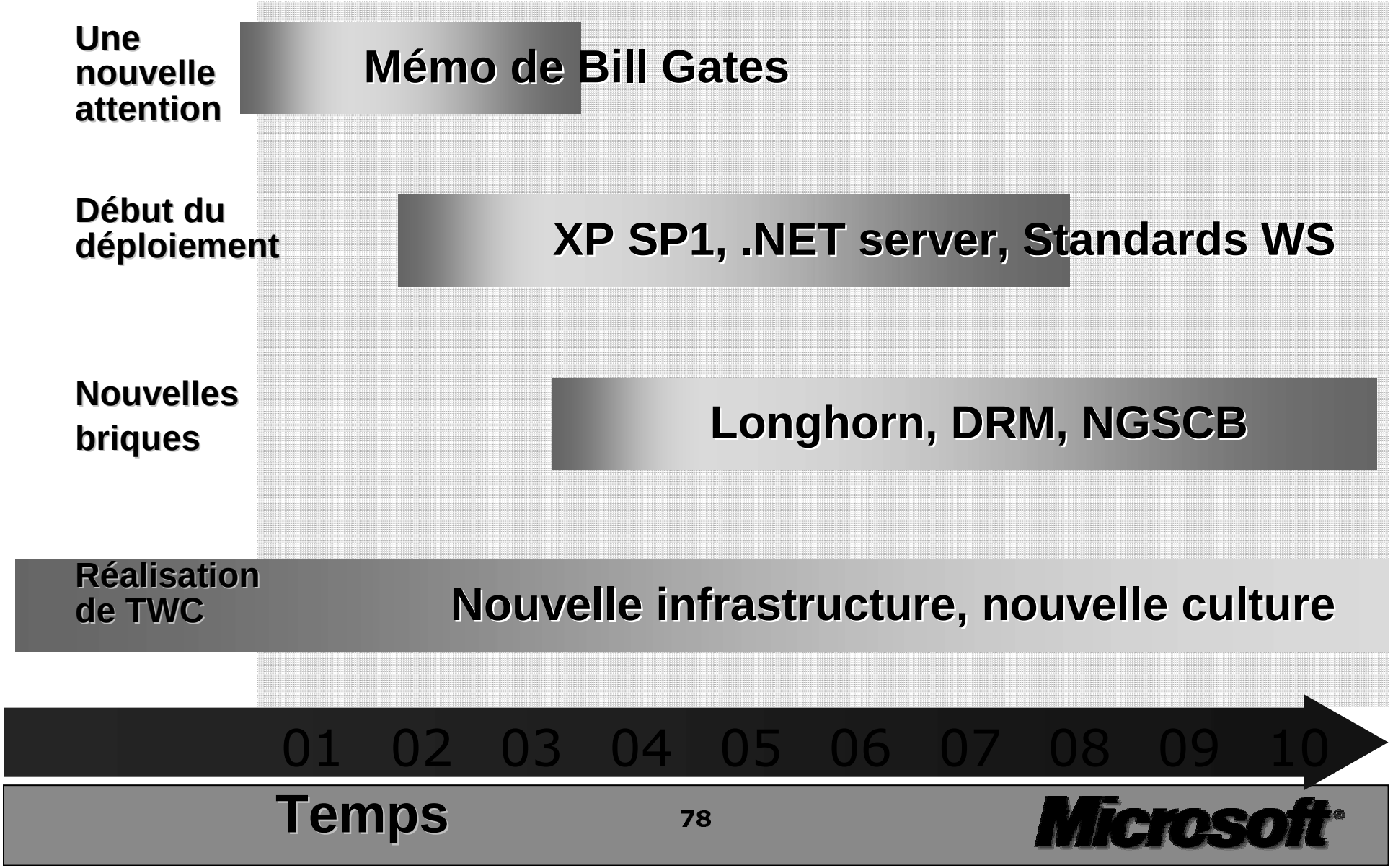
- NGSCB est un environnement d'exécution sécurisé fondé sur de nouvelles propriétés du hardware
  - Les processus NGSCB sont isolés les uns des autres par le hardware
  - Les processus NGSCB peuvent stocker et retrouver les secrets de manière sécurisée (en fonction de leur valeur de hachage)
- Le *nexus* fournit un environnement d'exécution et des services de sécurité et de chiffrements aux agents qui sont hébergés
  - Le hardware fournis des services de chiffrement au *nexus*
  - De même, le *nexus* fournit ces mêmes services aux agents qui s'exécutent au-dessus de lui

Résumé (2/2)

- NGSCB fournit quatre fonctionnalités clé :
  - Isolation forte des processus
  - Protection des secrets
  - Chemin sécurisé de et vers l'utilisateur
  - Attestation
- Les trois premières de ces fonctionnalités permettent de se protéger contre du code malicieux (virus, chevaux de troie, etc.)
- L'attestation permet de prouver à des entités distantes des faits à propos des logiciels, des utilisateurs, des machines, des services, etc. que ces entités distantes pourront croire en confiance

En guise de conclusion

Un long voyage



Microsoft

Plus précisément

- Bientôt : documentation du SDK NGSCB
- NGSCB 1.0 (Client) : introduit NGSCB, focalise sur une cible d'utilisateurs particulière
  - Plate-forme pour les premiers utilisateurs
  - Gouvernement, finance, assurance, santé, secteur juridique
- NGSCB 2.0 (Client, Serveur)
  - Déploiements plus larges et disponibilité d'un portefeuille applicatif plus étendu

Il reste encore beaucoup de chemin à parcourir…

- Nous avons besoin de nos partenaires et de nos clients pour arriver au bout
- Nous ne connaissons pas les réponses à toutes les questions : nous sommes intéressés par tous les feedbacks, questions, préoccupations,…
  - Sans œillères ni certitudes…

Sécurité et protection de la vie privée

- La tension fondamentale qui existe entre **exigence de sécurité** - laquelle tendrait, dans sa version optimale, à préconiser l'identification de chaque utilisateur -, et **protection de la vie privée** - qui exige au contraire la protection totale de l'anonymat-, est au cœur de toutes les préoccupations et de toutes les inquiétudes

Sécurité et protection de la vie privée

- Ces questions de respect de la vie privée et de sécurité informatique appellent naturellement un **débat de société** et, peut-être, de nouvelles règles du jeu.
- C'est précisément pour cette raison que nous avons fait le choix de la **transparence** pour le projet NGSCB, rendu public des années avant que l'architecture n'en soit aboutie, afin de recueillir l'avis de la communauté scientifique et de générer un débat que nous appelons de nos vœux.

Sécurité et protection de la vie privée

- **Ce qui est certain, c'est qu'il n'appartient certainement pas à Microsoft de faire ces choix et de se substituer à la souveraineté des gouvernements et au libre arbitre des citoyens**.
  - NGSCB laisse à l'utilisateur son entier libre arbitre quant à sa mise en fonction et, une fois en fonction, lui laisse la possibilité pleine et entière de contrôler le niveau de sécurité et de respect de sa vie privée.

Questions ?

Appendices

Changement de version & Migration

Changement de version et migration

- On voit apparaître un nouveau problème :
  - L'information est liée de manière cryptographique à une machine et/ou un système d'exploitation et/ou une application
- Il y a des secrets que l'on peut migrer et des secrets que l'on ne peut migrer
  - Composants applicatifs
  - Le *nexus*
  - Le hardware

Sauvegarder les secrets

- Sauvegarder le disque est tout à fait approprié pour cela tant que la carte mère ne meurt pas
- Les secrets des utilisateurs sont sauvegardés sous le mot de passe de l'utilisateur (en utilisant un chemin d'échange de données sécurisé)
- Secrets tiers ou détenus conjointement
  - Ont besoin d'un service réseau/distant, comme tout autre secret partagé
- Les clés partagées entre les machines peuvent permettre la récupération en cas de désastre, la sauvegarde et faciliter le management
- Migration des secrets
  - La même que pour la sauvegarde, excepté le fait qu'il faut utiliser PKSeal/PKUnseal pour réaliser la migration vers un nouveau *nexus* ou une nouvelle clé hardware

Changement de version du *nexus*

- Habituellement, on effectue une opération du type sceller(secret, soi-même)
- Mais on peut aussi réaliser une opération du type sceller(secret, autre *nexus* connu)
- Les nouveaux *nexuses* sont livrés avec un « certificat de changement de version »
- L'ancien *nexus* divulgue tous (certains ?) secrets au nouveau *nexus* (dépendant de la politique choisie)
- La mise à jour d'une version d'un agent utilise un processus similaire
  - Mais nous pensons que les fonctionnalités de « manifeste » pour les agents rendront rare ce cas de mise à jour

Mise à jour du *nexus*

1. L'ancien *nexus* reçoit une demande de mise à jour de l'utilisateur
2. L'ancien *nexus* vérifie le certificat de mise à jour du nouveau *nexus*
3. L'ancien *nexus* demande au HW de desceller son secret racine
4. L'ancien *nexus* demande au HW de sceller son secret racine pour le nouveau *nexus*
5. Le nouveau *nexus* est installé
6. Le nouveau *nexus* demande au HW de desceller le secret racine de l'ancien *nexus*
7. Le nouveau *nexus* génère un nouveau secret racine et scelle tous les secrets migrés en son sein

Variation : mise à jour du hardware

1. L'ancien *nexus* reçoit une demande de mise à jour de l'utilisateur
2. L'ancien *nexus* vérifie le certificat de mise à jour du nouveau *nexus* et les créances du HW de la machine cible
3. L'ancien *nexus* demande au HW de desceller son secret racine
4. L'ancien *nexus* effectue un *PKSeal* sur son secret racine vers le nouveau *nexus* sur le nouveau HW
5. L'ancien *nexus* demande au HW d'effectuer une opération de *Quote* sur le résultat de *PKSeal*
6. Le nouveau *nexus* effectue *PKVerifyQuote* sur les données résultant de l'opération de Quote.
7. Le nouveau *nexus* demande au HW de faire un *PKUnseal* sur le secret racine de l'ancien nexus
8. Le nouveau *nexus* génére un nouveau secret racine et scelle tous les secret migrés en son sein

Informations complémentaires sur les algorithmes

Algorithme de descellement du SSC

L'opération **Unseal** doit implémenter les étapes suivantes :

1. *$M = AES^{-1}[K_S](SealedBlob)$.*
2. *Interpret M as (BITS[128] R || SECRET S1 || DIGEST Target0 || DIGEST Target1 || DIGEST Sealer0 || DIGEST Sealer1 || DIGEST N).*
3. *DIGEST D = HMAC[KM](R || S1 || Target0 || Target1 || Sealer0 || Sealer1).*
4. *If (Target0 != PCR[0] || Target1 != PCR[1]) return SSC_UNSEAL_ERROR with S, Source set to zero.*
5. *If D!=N return SSC_UNSEAL_ERROR with S, Source set to zero.*
6. *Else return SSC_SUCCESS with S set to S1 and Source set to {Sealer0, Sealer1}.*

Stockage scellé vers une entité distante

- PKSeal *(peut être effectué par n'importe quel parti)*
  - Input: secret, target ProgId
  - Output: crypto blob b = PKEncrypt(secret, target)
- PKUnseal *(implémenté par le SSC)*
  - Input: crypto blob b
  - Output: secret or error
  - (secret,target) = PKDecrypt(b)
  - IF target == PCR THEN return s
  - OTHERWISE error

Compléments d'informations

- Site Web Microsoft France
  http://www.microsoft.com/france/securite/entreprises/palladium/
  - Livre Blanc
  - FAQ

Remerciements

- Les personnes suivantes ont été des contributeurs clé de l'initiative « Palladium » au sein de Microsoft :
  - Peter Biddle, John DeTreville, Paul England, Butler Lampson, John Manferdelli, Marcus Peinado, Bryan Willman, Blair Dillaway, Brian LaMacchia
- Conseillers, etc. :
  - Josh Benaloh, Roger Needham, Dan Simon, Chuck Thacker