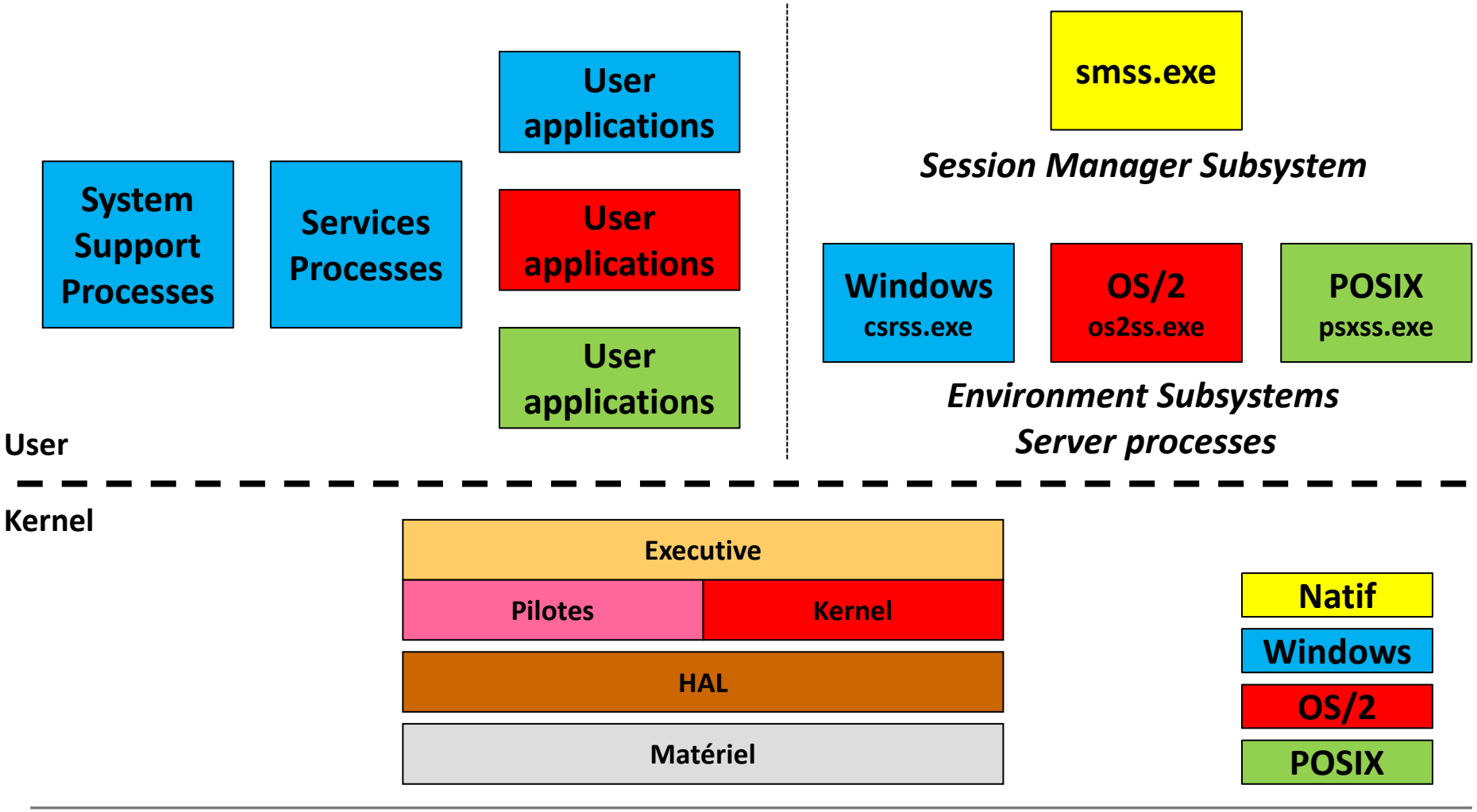
Architecture du système

AB – v1.93 (08/09/2021)

Architecture générale du système



Sous-systèmes
(*Environment Subsystems*)

- L'exécutive NT devait initialement pouvoir exécuter 3 types de processus :
  - **Win32** : issue de l'évolution de Win16
  - **OS/2** : base de Windows NT et pour succéder à ce système
  - **Posix** : imposé par les autorités américaines pour l'accès à certains marchés
- L'architecture sous forme de sous-systèmes permet de fournir un environnement d'exécution à ces trois types de processus en adaptant l'exécutive NT et son API native
- Un sous-système est composé :
  - d'un processus jouant le rôle de « serveur »
  - d'un ensemble de bibliothèques dynamiques offrant l'API aux processus

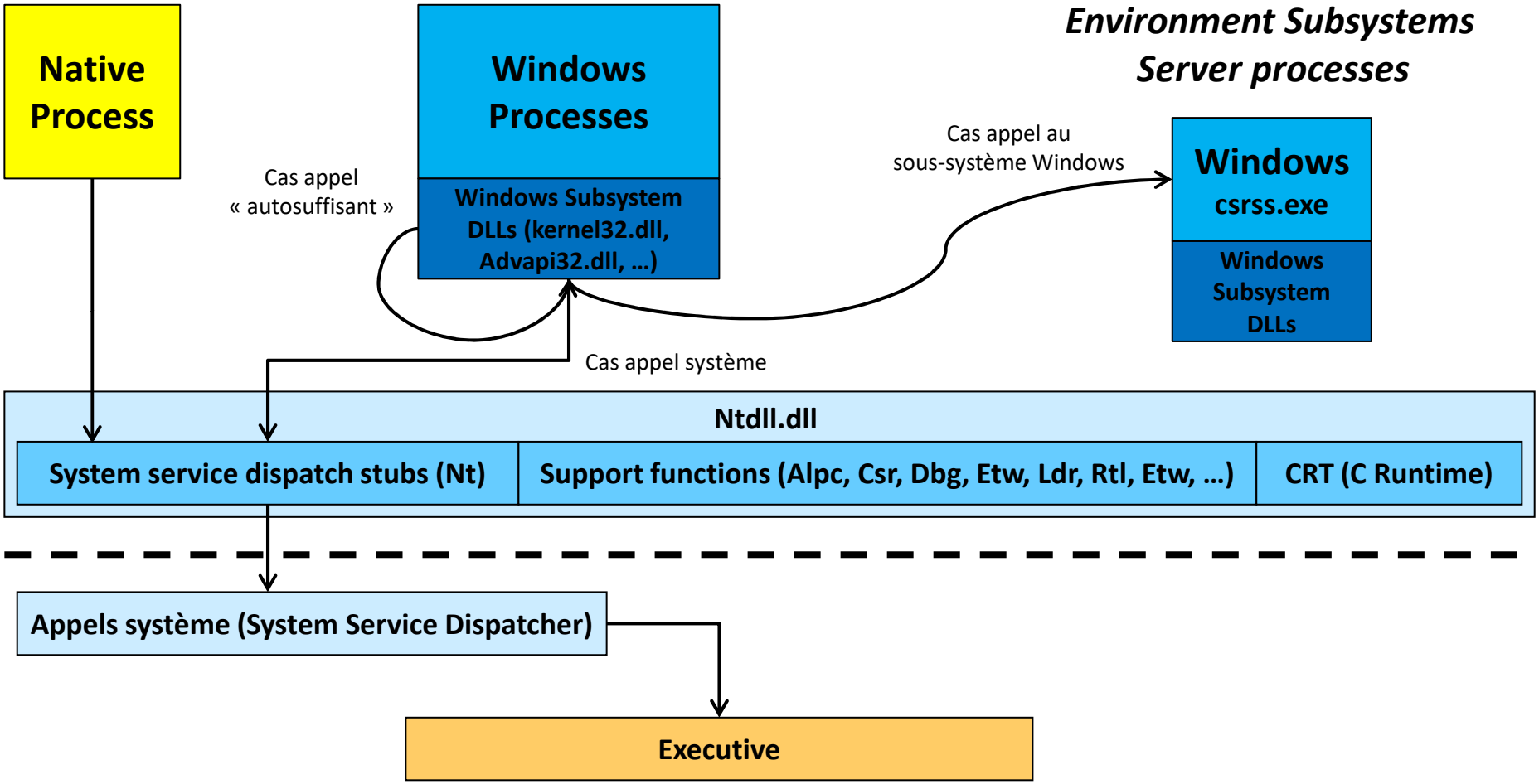
Sous-systèmes (*Environment Subsystems*)

- **Windows** (`csrss.exe`) :
  - Systématiquement démarré (obligatoire)
  - Permet l'exécution de programmes de type :
    - Windows 32/64 bits
    - 16 bits via VDM (*Virtual DOS Machine*) (32 bits seulement)
  - Serveurs hébergés :
    - csrsrv.dll : initialisation et fonctions diverses
    - basesrv.dll : *thread*, *process*, VDM
    - winsrv.dll : console, user services
    - sxssrv.dll : Side-by-Side
    - (consrv.dll : console - Transféré dans `ConHost.exe` depuis Windows 7)

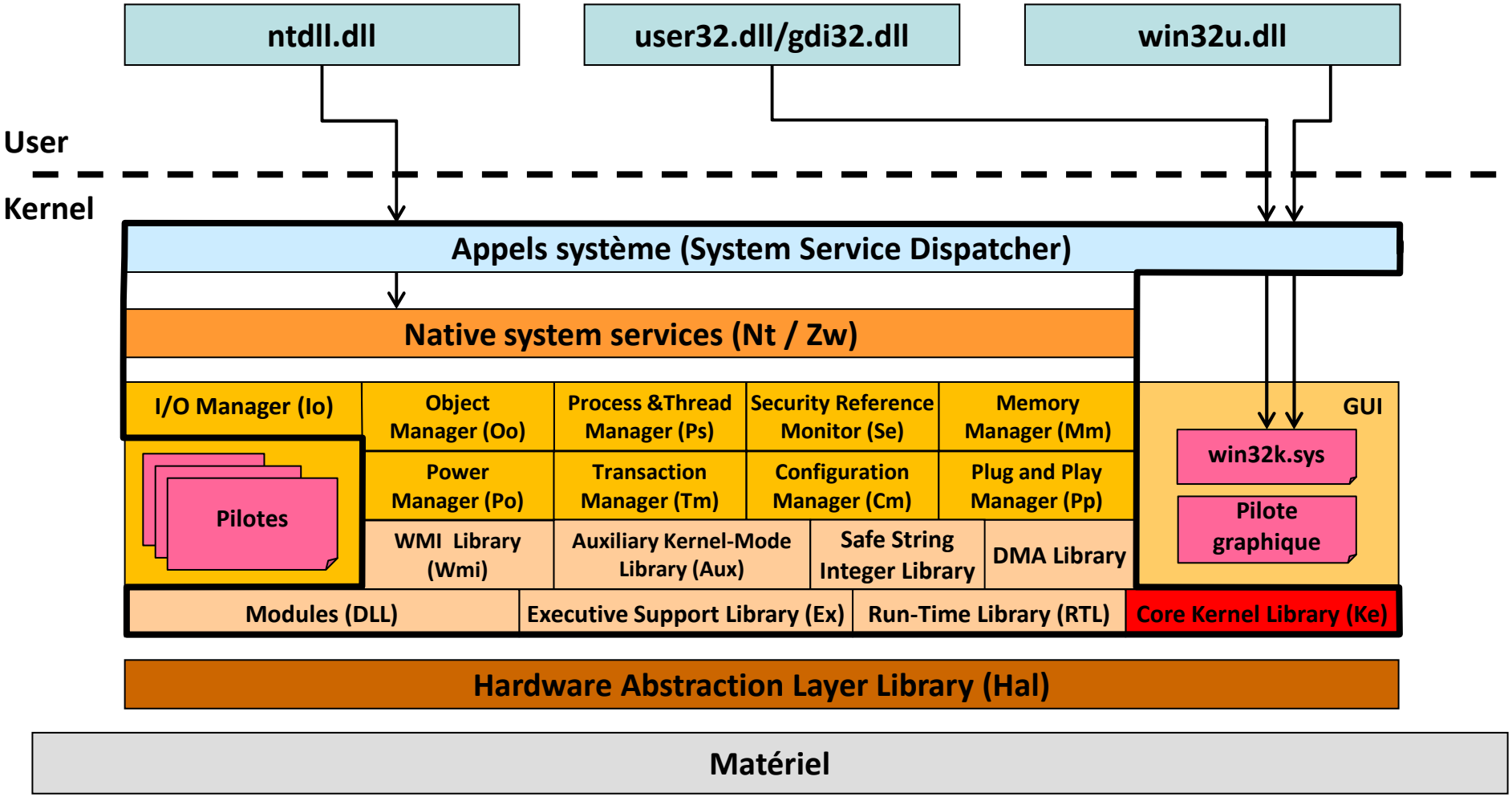
Sous-systèmes (*Environment Subsystems*)

- **POSIX** (`psxss.exe`) :
  - Permet initialement l'exécution de programmes POSIX 32 bits
  - XP/2003 : supprimé et substitué par *Windows Services for UNIX* (SFU)
  - Vista : remplacé par *Windows Subsystems for Unix-based Applications* (SUA) (ajout du support 64 bits)
  - Windows 8/2012 : suppression du sous-système
  - Windows 10 : remplacé par ***Windows Subsystem for Linux*** (WSL)
    - Permet l'exécution de binaire Linux ELF64
  - Windows 10 2004 : WSL 2 (basé sur la virtualisation)
- **OS/2** (`os2ss.exe`) :
  - Support uniquement de programmes OS/2 en invite de commande
  - Supprimé à partir de XP/2003

Relations et appels



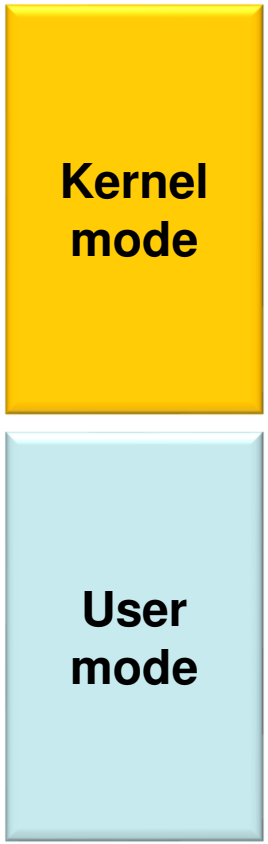
Noyau et *Executive*



Principaux composants du noyau

- Composants intégrés à Ntoskrnl.exe :
  - **L'*executive*** : fonctions « haut-niveau »
    - Fonctions appelables depuis le mode utilisateur via un appel système (Nt)
    - Fonctions de support pour les pilotes appelables depuis le noyau (Zw)
    - *Managers* et bibliothèques statiques (Se, Io, Ps, Cm, etc.)
  - **Le noyau** : fonctions « bas-niveau » (Ke)
- Bibliothèques dynamiques : fichiers .dll
- Pilotes (*device drivers*) : fichiers .sys
- Système graphique (Win32k.sys) : fonctions USER et GDI
  - Windows 10 : win32kbase.sys et win32kfull.sys

Mode utilisateur / noyau
Répartition des zones mémoire



FF…

00…

- **32 bits (x86) :**
  - `00000000` – `7FFFFFFF` : user
  - `80000000` – `FFFFFFFF` : kernel
- **32 bits avec /3GB :**
  Nécessite le flag IMAGE_FILE_LARGE_ADDRESS_AWARE
  - `00000000` – `BFFFFFFF` : user
  - `C0000000` – `FFFFFFFF` : kernel
- **64 bits (x64) :** 48-bit *Canonical form addresses*
  + Limite de Windows à 44 bits (supprimée depuis Windows 8.1)
  - `00000000`\`00000000 – `00007FF`\`FFFFFFFF : user
  - `FFFFF800`\`00000000 – `FFFFFFFF`\`FFFFFFFF : kernel
- **64 bits (IA64)**

Sessions Windows

Sessions Windows

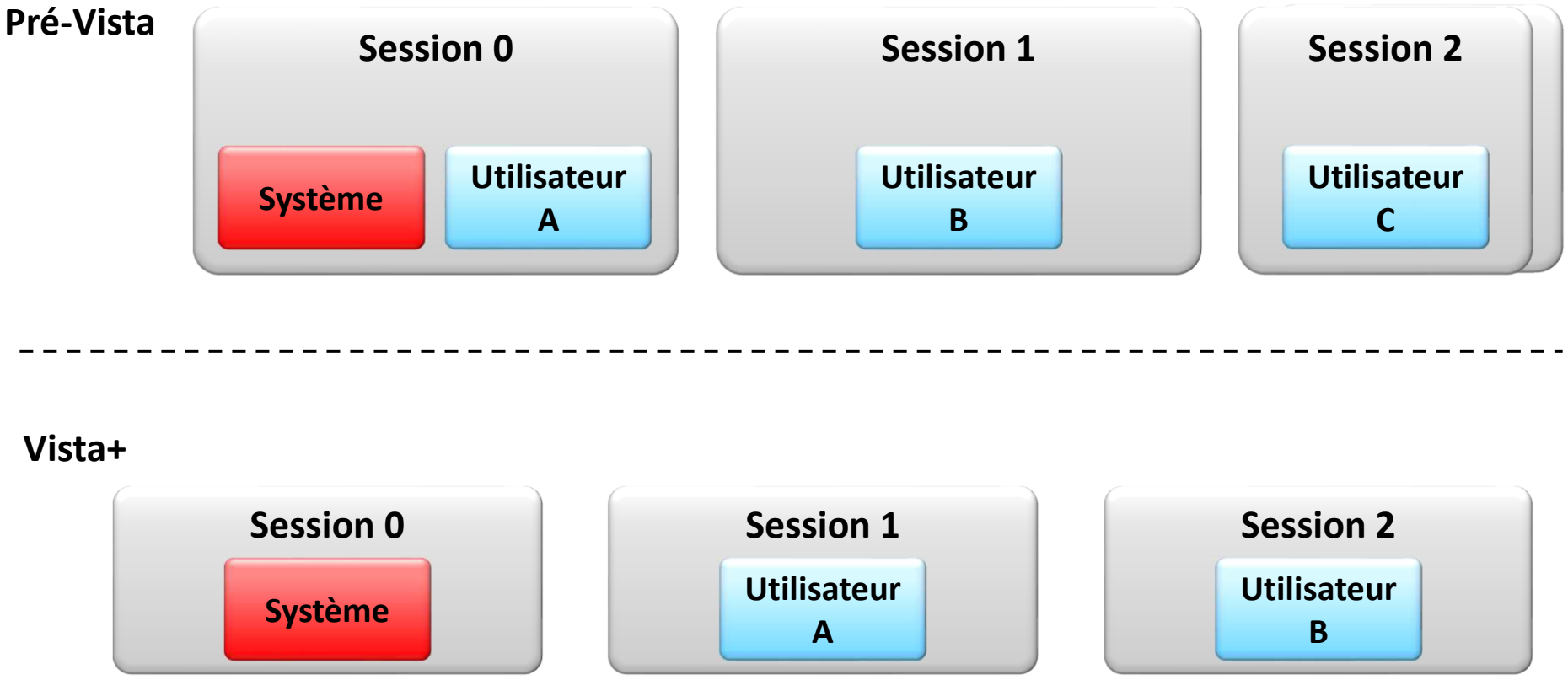
- Les sessions sont apparues avec Windows NT 4 TSE (*Terminal Server Edition*) afin de permettre la connexion simultanée de plusieurs utilisateurs sur un même système
- Les sessions sont identifiées par un `SessionID` et chaque processus est associé à une session
- Des processus dans des sessions différentes sont « isolés », en particulier :
  - Le système graphique
  - Certains objets du noyau (mutex, event, job, timer, etc.)
- Tous les processus du système et des services sont associés à la session 0

Sessions Windows

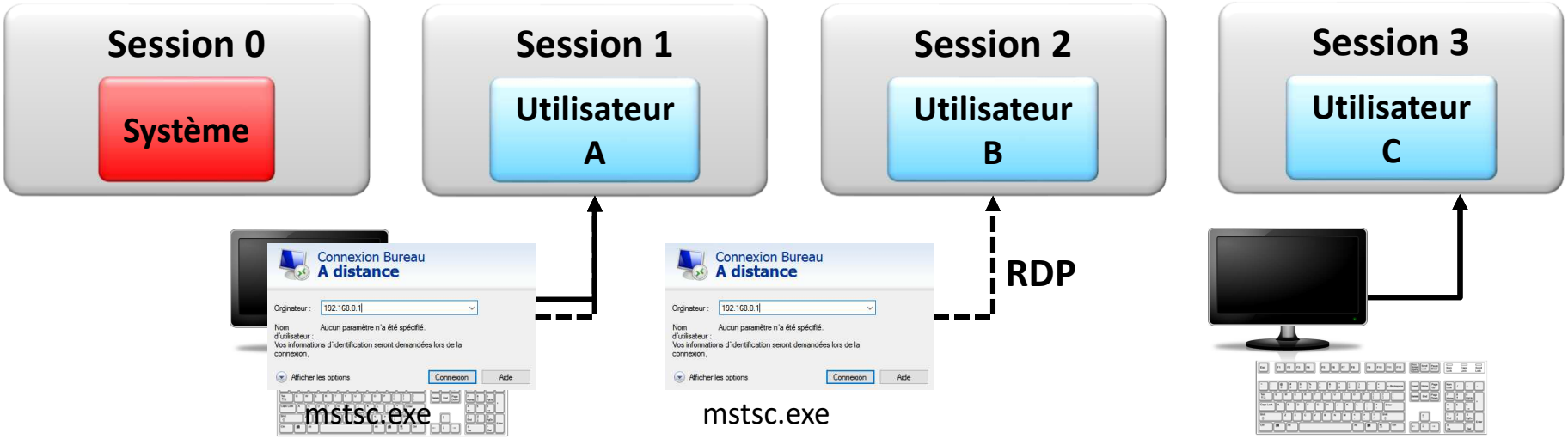
- Les sessions sont mises en œuvre par le mécanisme *Terminal Services* et utilisées par :
  - Le *Fast User Switching*
  - Le bureau à distance (accédé par le protocole RDP)
  - Les sessions « étendues » (*Enhanced Session Mode*) d'Hyper-V
    - Apparues avec Windows 8.1 / Windows Server 2012 R2

- Windows Vista a introduit « l'isolation de la session des services » : la session 0 est réservée exclusivement aux processus du système et aux services

Services isolation in Session 0



Fonctionnement des sessions



Différencier :

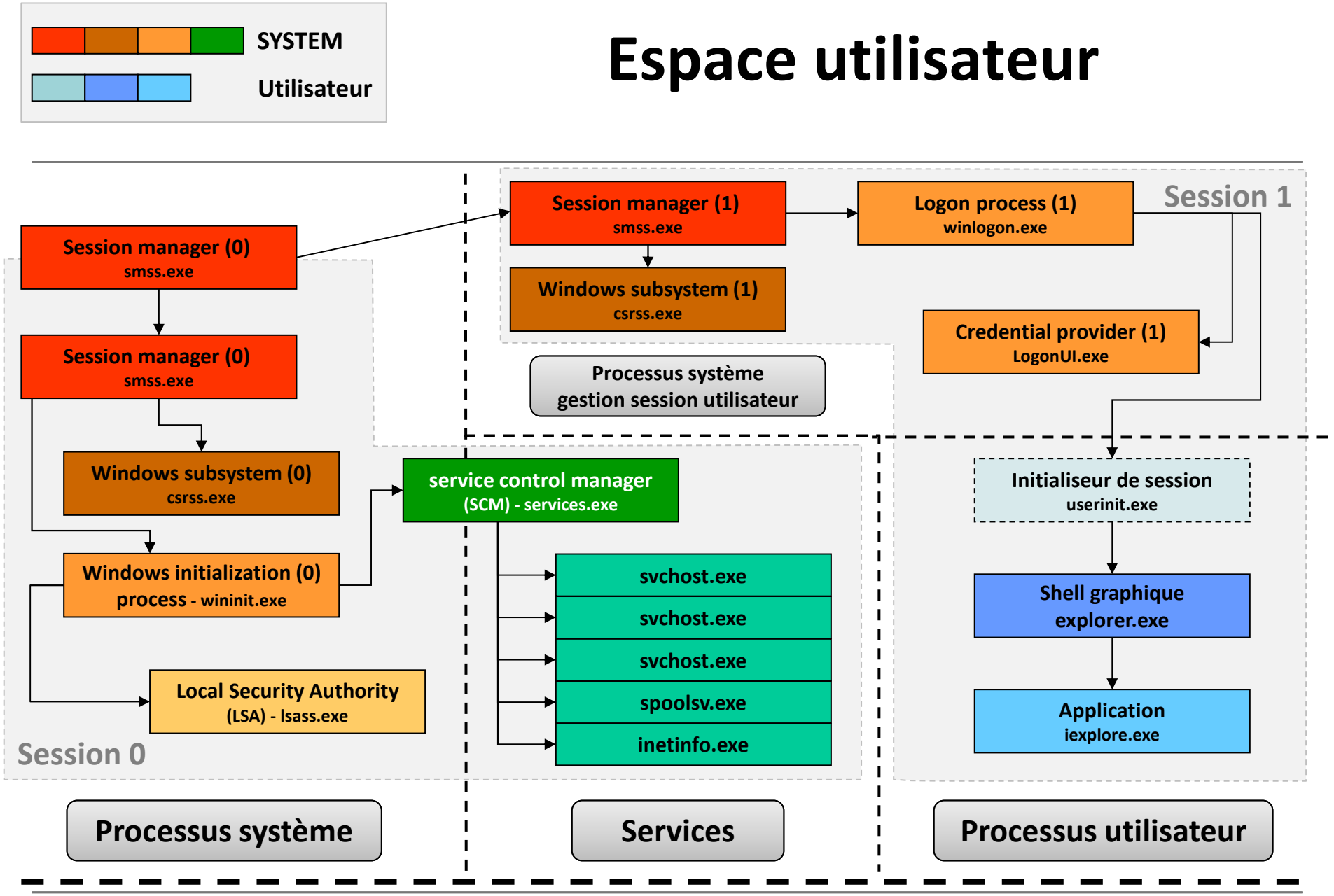
- Verrouiller la session (*Lock*)
- Se déconnecter (*Disconnect*)
- Fermer la session (*Sign out*, *Logoff*)

Divers

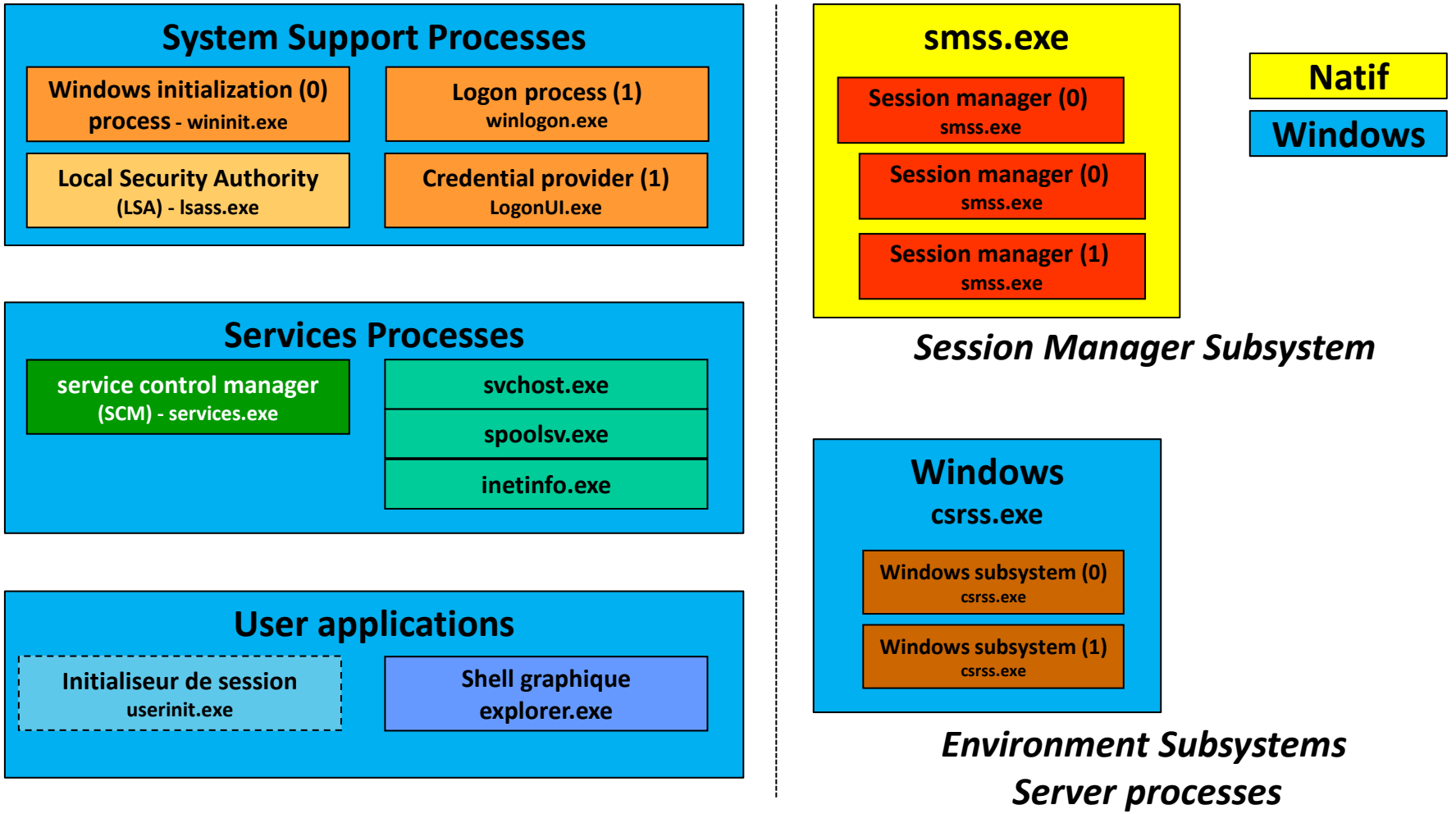
- Outils divers :
  - `query.exe { PROCESS | SESSION | TERMSERVER | USER }`
  - `qwinsta.exe / qprocess.exe / quser.exe`
  - `tscon.exe / tsdiscon.exe / tskill.exe`
- Possibilité de « dupliquer » une session :
  - shadow.exe (XP/2003 -> 7/2008 R2)
  - mstsc.exe /shadow:<sessionID> (8.1 / 2012 R2 +)
- Par défaut, une seule session interactive (locale ou distance) est autorisée par utilisateur : si un utilisateur a déjà une session ouverte, celle-ci est reconnectée en cas de nouvelle authentification (comportement modifiable par GPO)

Processus de démarrage

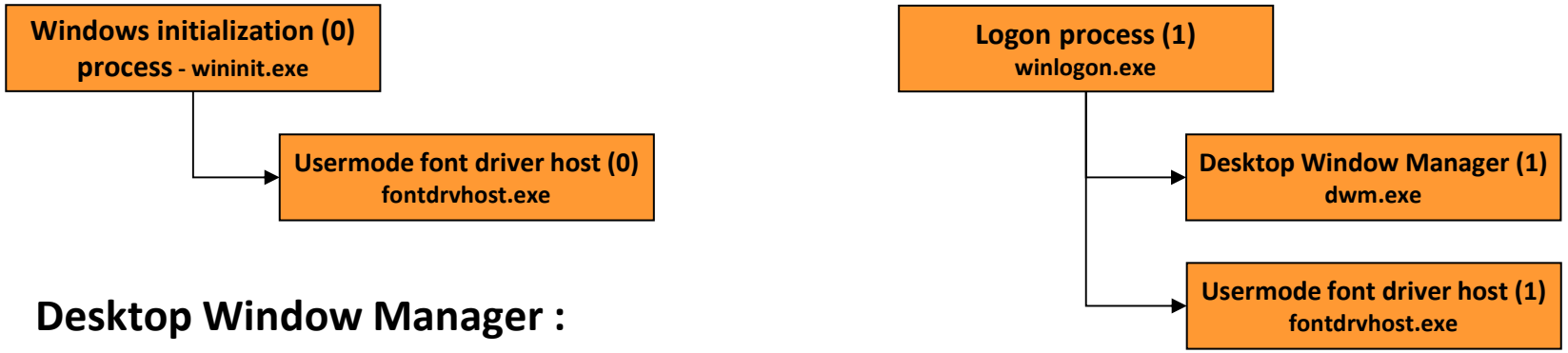
Espace utilisateur



Architecture générale du système

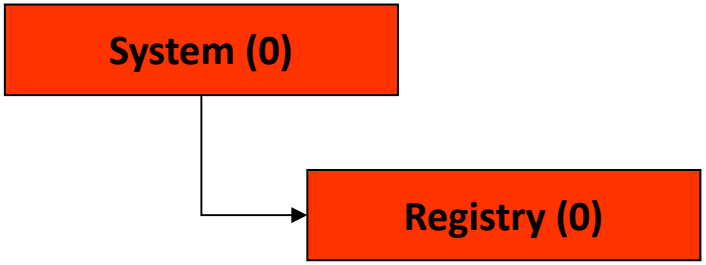


Processus supplémentaires (Windows 10)



- **Desktop Window Manager :**
  - Niveau d'intégrité « System »
  - Window Manager\DWM-X (S-1-5-90-0-X)
  - S-1-5-90-0 (Window Manager\Window Manager Group)
- **Usermode Font Driver Host :**
  - Niveau d'intégrité « Low »
  - Font Driver Host\UMFD-X (S-1-5-96-0-X)
  - S-1-5-96-0
  - Application AppContainer

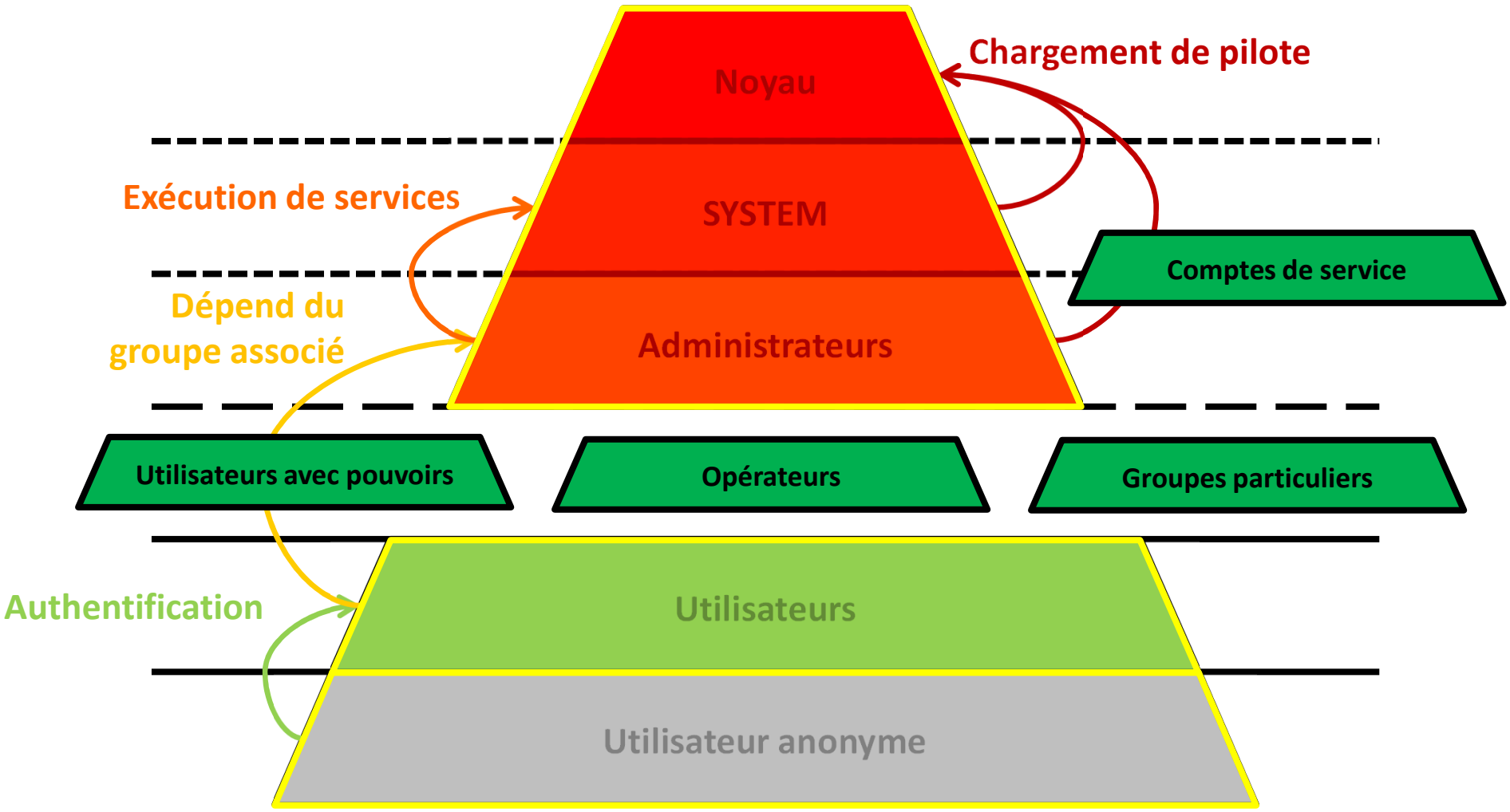
Processus supplémentaires (Windows 10)



- Apparu avec Windows 10 1803
- Processus de type « *minimal process* » hébergeant les pages mémoire liées au Registre

Pyramide des droits

Pyramide des droits



Principaux composants

Composants liés à la sécurité

- **Stratégie de sécurité locale** : base de configuration contenant divers paramètres relatifs à la sécurité locale du système (comptes, journalisation, pare-feu, restrictions logicielles, *etc.*)
  - `secpol.msc`
- **Stratégie de groupe locale (LGPO)** : base de configuration contenant la configuration locale de l'ordinateur et des utilisateurs
  - `gpedit.msc`
  - La stratégie de sécurité locale est incluse dans la stratégie de groupe locale
  - Des stratégies de groupes peuvent être appliquées depuis un domaine Active Directory via les GPO (*Group Policy Object*)
    - Les GPO du domaine sont prédominantes par rapport aux LGPO

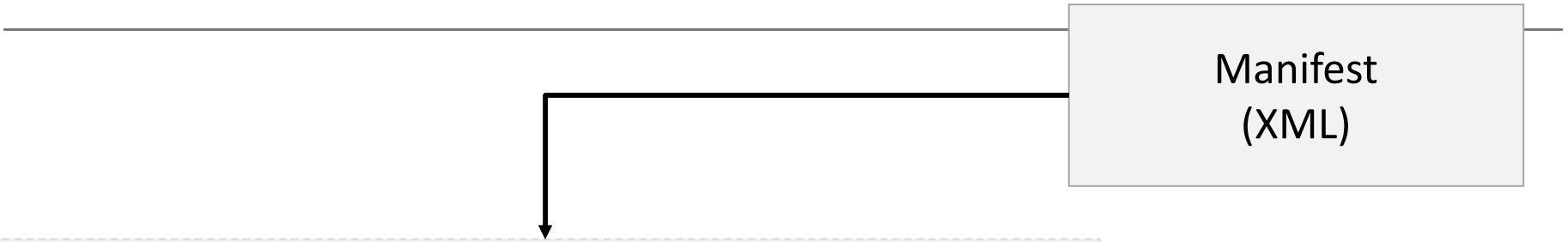
Composants liés à la sécurité

- **Base SAM (*Security Account Manager*)** : base contenant les comptes et groupes utilisateurs, ainsi que les secrets d'authentification (gérée par le service SamSs)
  - `net user / net localgroup`
  - `lusrmgr.msc`
  - `Get-LocalUser / Get-LocalGroup`
- **Observateur d'évènements** : permet d'accès aux journaux Windows (Application, Sécurité et Système), ainsi qu'aux journaux des applications et des services
  - `wevtutil qe System` (qe | query-events)
  - `eventvwr.exe`
  - `Get-EventLog System`

Journalisation

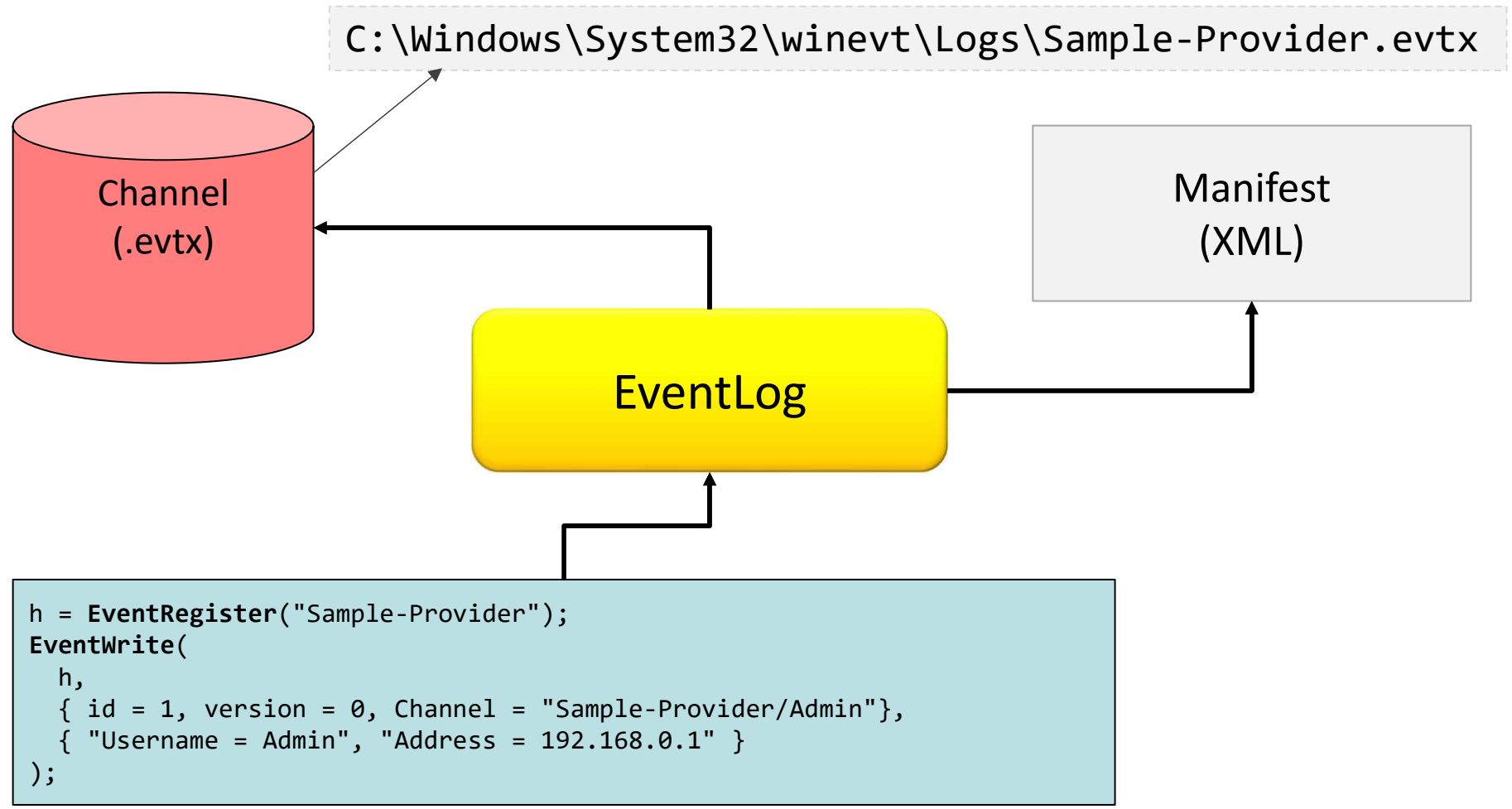
(*Windows Event Log*)

Définition du *Manifest*

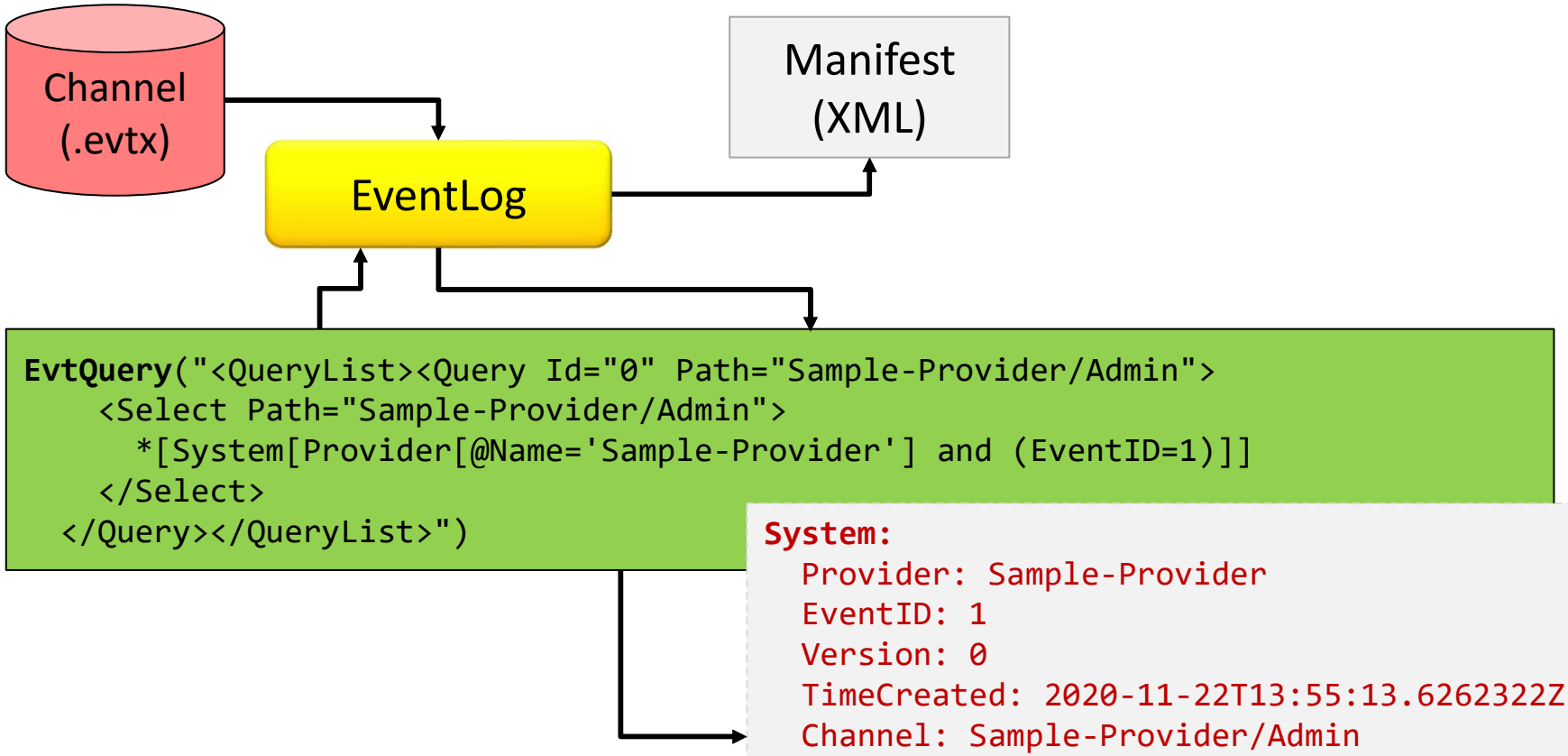



```
<instrumentationManifest><instrumentation><events>
  <provider name="Sample-Provider" />
    <channels>
      <channel name="Sample-Provider/Admin" />
    </channels>
    <templates>
      <template tid="t1">
        <data name="Username" inType="win:UnicodeString" />
        <data name="Address" inType="win:IPv4" />
      </template>
    </templates>
    <events>
      <event value="1" version="0"
        template="t1"
        message="User %1 connected from %2." />
    </events>
  </provider>
</instrumentation></instrumentationManifest>
```

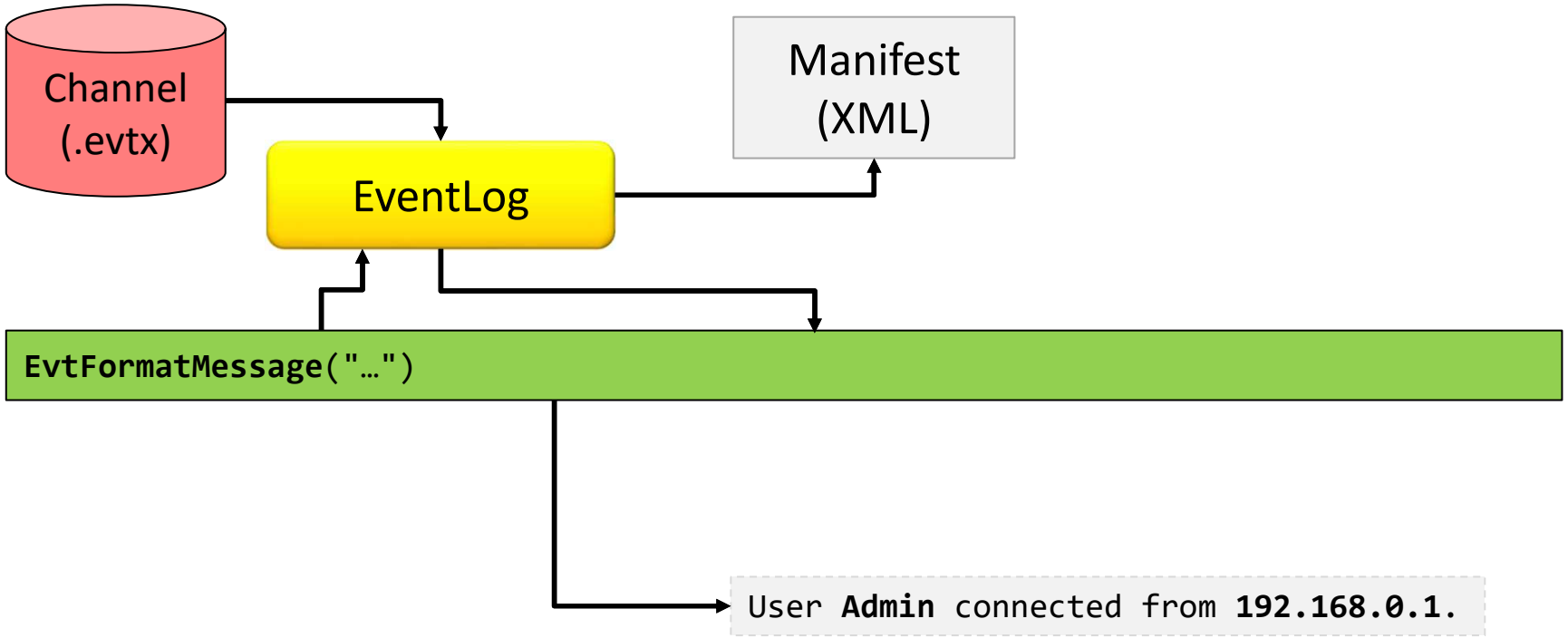
Écriture d'un événement



Requête d'un événement

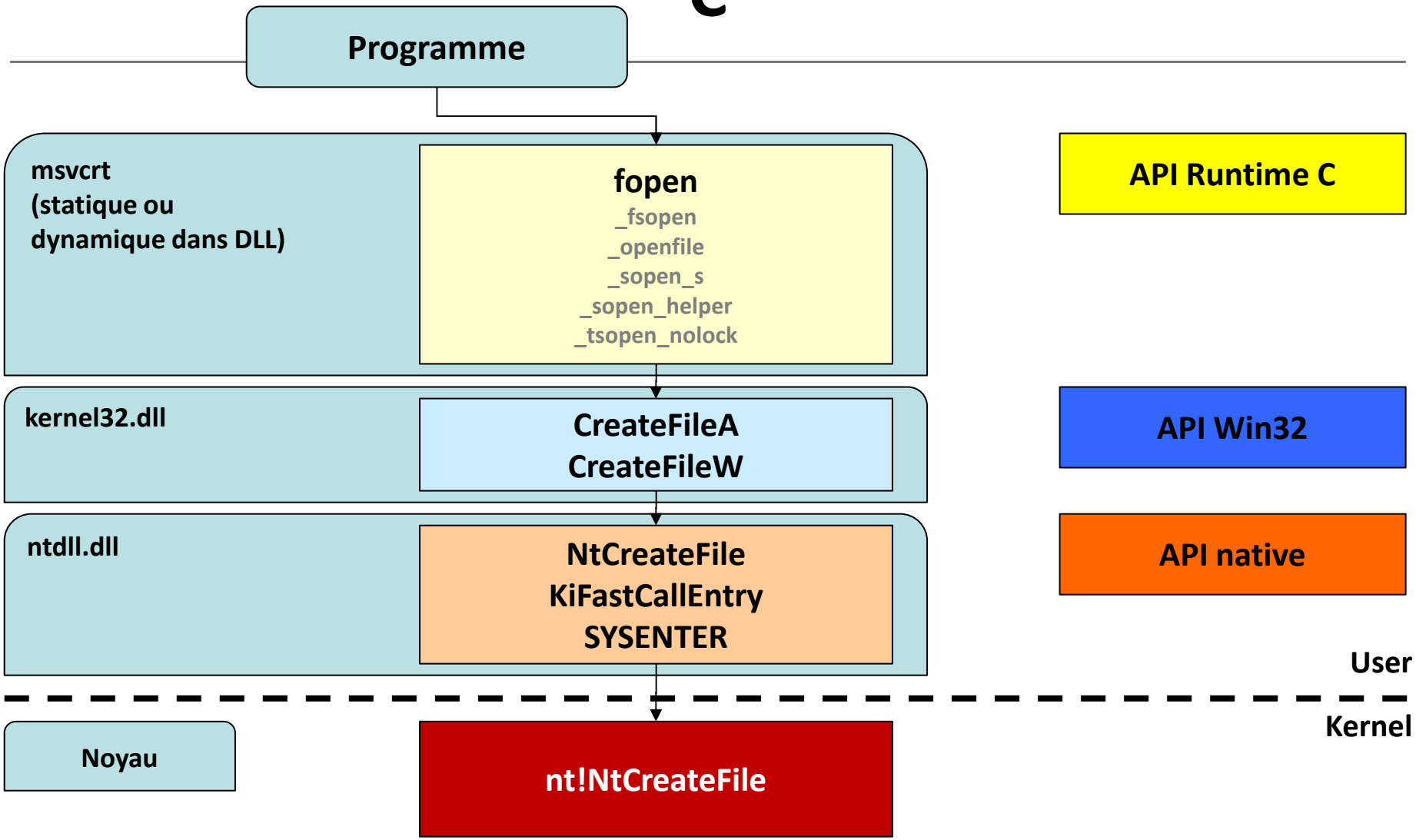


Rendu d'un événement



Appels en mode utilisateur

Appels bibliothèque / système C



Appels bibliothèque / système Win32