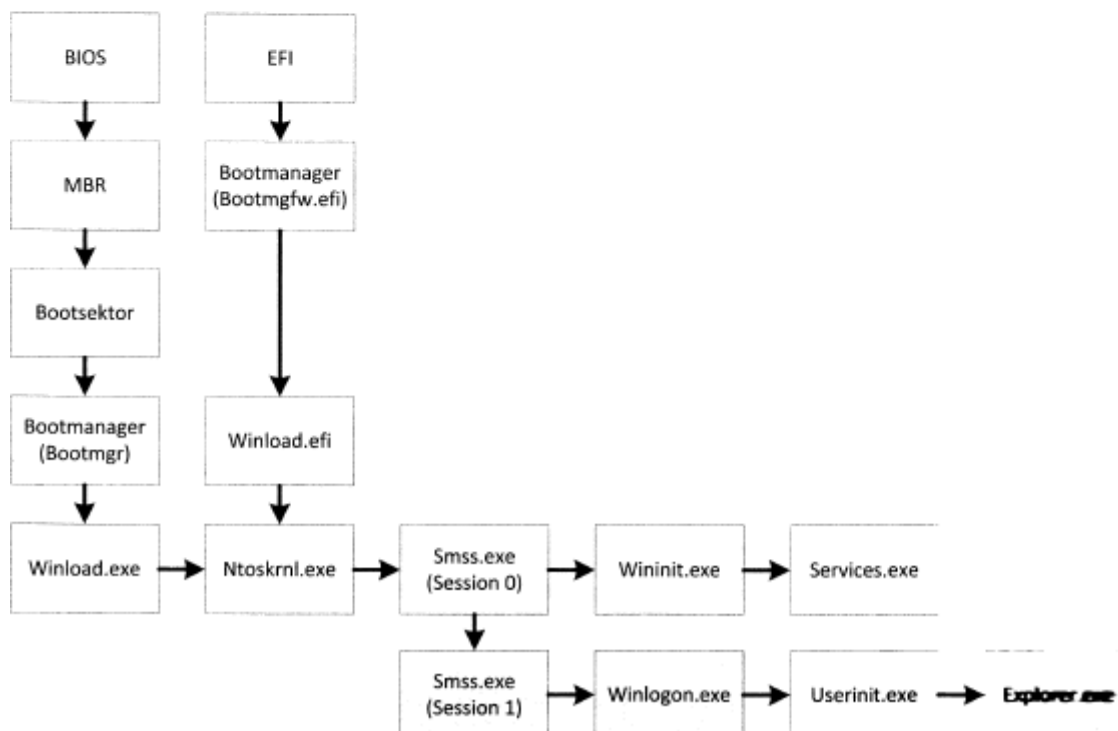


1. **MBR (Master Boot Record)** – Der MBR verweist auf die aktive Partition des Laufwerks, die den Bootsektor enthält.
2. **Bootsektor** – Der Bootsektor unter Windows 7 hat einen festen Verweis auf den Boot-Manager (bootmgr). Das BIOS startet diese Datei.
3. **Boot-Manager (bootmgr)** – (Wechsel von 16-Bit- in 32-Bit-/64-Bit-Mode) Liest BCD (Boot Configuration Database) und zeigt Bootmenü, wenn mehrere Einträge im BCD vorhanden sind. Startet den Bootloader des OS.
4. **Winload.exe (Bootloader)** – Lädt Ntoskrnl.exe, Hal.dll, Boot-Start-Device-Treiber und Dateisystemtreiber der Partition (z. B. NTFS.sys).
5. **Ntoskrnl.exe** – Führt u. a. Funktionen der Hal.dll, Bootvid.dll und Ntdll.dll aus. Startet Systemstarttreiber. Startet Smss.exe.
6. **Smss.exe (Sitzungs-Manager)** – Startet jeweils eine Instanz von sich selbst pro Session. In Session 0 wird u. a. Win32k.sys, Csrss.exe, Wininit.exe gestartet. In Session 1-n wird u. a. Csrss.exe und Winlogon.exe gestartet.

Session 0:

1. **Wininit.exe** – Startet SCM (Service Control Manager), LSASS (Local Security Authority Subsystem) und LSM (Local Session Manager).
2. **SCM (Services.exe)** – Startet Autostarttreiber und -dienste.
 - **SCM (Delay)** – Nach einer Verzögerung von 120 Sekunden werden die Delayed-Autostartdienste (verzögerter Start) gestartet.



Einfache Darstellung des Bootprozesses unter Windows 7.

Session 1-n:

1. **Winlogon.exe** – Lädt *LogonUI.exe* (Anmeldefenster). Nach bestätigter Authentifizierung wird das Benutzerprofil geladen und *Userinit.exe* gestartet.
2. **Userinit.exe** – Startet Skripte und die Shell (*Explorer.exe*).
3. **Explorer.exe** – Die erste Instanz von *Explorer.exe* in einer Session stellt den Desktop dar.

Die einzelnen Schritte im Detail

Der Boot-Manager liest die Systemstartinformationen aus der BCD (Boot Configuration Database). Dort steht in der Regel als *default* der Bezeichner *{current}*. Als Pfad ist für den Eintrag *{current}* der Bootloader von Windows 7 konfiguriert (*\Windows\system32\winload.exe*). Bootmgr startet *Winload.exe* in diesem Fall automatisch.



Sollte das System aus dem Ruhezustand (Hibernation) aufwachen, startet der Boot-Manager *winresume.exe*, das den gespeicherten RAM-Inhalt aus der Datei *hiberfil.sys* zurück in den Hauptspeicher lädt.

Winload.exe ist das erste Programm, das vollständig als 32-Bit- bzw. 64-Bit-Programm läuft. Es holt sich vom ACPI-BIOS Informationen über das System (Datum/Zeit, Festplatteninformationen und Legacy-Hardware wie PCI-Bus oder Parallel-Port). Außerdem lädt *Winload.exe* den Windows-Kernel *Ntoskrnl.exe* und die von *Ntoskrnl.exe* benötigten Dateien (*Ci.dll*, *Cifs.sys*, *Hal.dll*, *Kdcom.dll* und *Pshed.dll*) in den Speicher.

Anschließend ermittelt und startet es die Boot-Start-Device-Treiber und den Dateisystemtreiber für die Partition (NTFS.sys). Als letzte Aktion ruft *Winload.exe* die vorbereitete *Ntoskrnl.exe* auf.

Ntoskrnl initialisiert das System (weitere Prozessoren, I/O-Manager, Memory-Manager, Kernel-Debugger, Plug & Play-Manager etc.)

Da *Winload* bereits alle kritischen Treiber geladen hat, kann *Ntoskrnl* nun bereits über diese Treiber auf die Geräte wie die Festplatte zugreifen und die Systemstarttreiber und weitere Dateien laden. Als Letztes wird der Session Manager (Sitzungs-Manager – *Smss.exe*) gestartet.

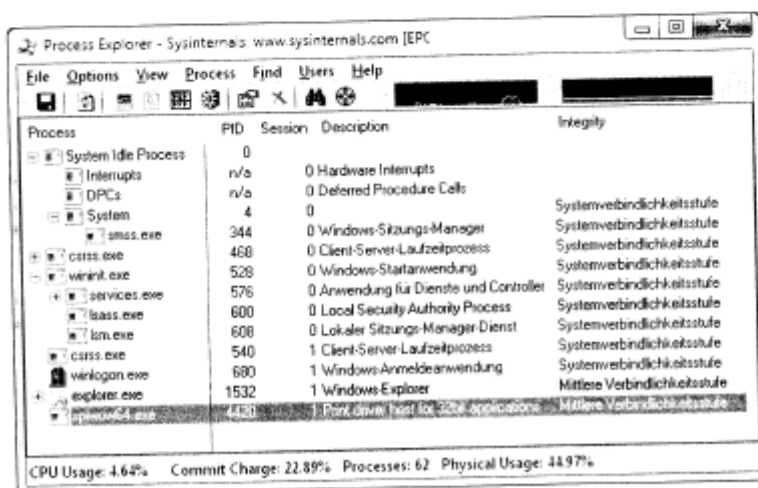
Der **Sitzungs-Manager (Smss)** erzeugt und verwaltet die einzelnen Sessions. Smss läuft selbst in der Session 0 und startet jeweils eine Kopie von sich selbst für jede weitere Session. Die Kernel-Treiber und Systemdienste laufen in Session 0, der Desktop des Benutzers läuft in Session 1.

Die Hauptversion von Smss in Session 0 startet den Kernel-Teil des Windows-Subsystems (Win32k.sys), eine Instanz der Client-Server-Laufzeitumgebung (Csrss.exe) und die Windows-Startanwendung (Wininit.exe). Außerdem führt der Sitzungs-Manager die Pending File Rename Operations durch, bei denen Dateien gelöscht werden, die während eines Updates oder einer Installation noch im Zugriff waren.

In Session 1 und jeder weiteren Session startet Smss eine zusätzliche Instanz der Client-Server-Laufzeitumgebung (Csrss.exe) und die Winlogon.exe.

Die Schritte in Session 0

Wininit.exe (Session 0) startet den **Service Control Manager (SCM – services.exe)**, das **Local Security Authority Subsystem (LSASS)** und den **Local Session Manager (LSM)**. Wininit startet all die Prozesse, die Winlogon benötigt. Wenn Wininit mit einem Fehler abbricht, werden Sie in den anderen Sessions keinen Anmeldebildschirm sehen. Außerdem stellt Wininit einen Desktop für die Session 0 bereit, obwohl kein interaktives Arbeiten mit Session 0 vorgesehen ist. Der Desktop in Session 0 sorgt aber für Kompatibilität, wenn alte oder fehlerhafte Dienste versuchen, ein Fenster anzuzeigen. Der Anwender (Session 1) sieht diese Fenster zwar nicht, aber die Dienste laufen wenigstens nicht in eine Exception.



The screenshot shows the Process Explorer window with the following data table:

Process	PID	Session	Description	Integrity
System Idle Process	0			
System	4	0	0 Hardware Interrupts	
smss.exe	344	0	0 Deferred Procedure Calls	Systemverbundlichkeitsstufe
csrss.exe	468	0	0 Windows-Sitzungs-Manager	Systemverbundlichkeitsstufe
wininit.exe	528	0	0 Client-Server-Laufzeitprozess	Systemverbundlichkeitsstufe
services.exe	576	0	0 Windows-Startanwendung	Systemverbundlichkeitsstufe
lsass.exe	600	0	0 Anwendung für Dienste und Controller	Systemverbundlichkeitsstufe
lsm.exe	608	0	0 Local Security Authority Process	Systemverbundlichkeitsstufe
csrss.exe	540	1	0 Lokaler Sitzungs-Manager-Dienst	Systemverbundlichkeitsstufe
winlogon.exe	680	1	1 Client-Server-Laufzeitprozess	Systemverbundlichkeitsstufe
explorer.exe	1532	1	1 Windows-Anmeldeanwendung	Mittlere Verbundlichkeitsstufe
explorer.exe			1 Windows-Explorer	Mittlere Verbundlichkeitsstufe

At the bottom of the window, system statistics are displayed: CPU Usage: 4.64%, Commit Charge: 22.89%, Processes: 62, Physical Usage: 44.97%.

Process Explorer: Ansicht der Prozess-Struktur mit Session-Nr. und Verbindlichkeitsstufe.

Der SCM (services.exe – Anwendung für Dienste und Controller) ist für die Verwaltung der Dienste zuständig und startet alle Autostartdienste und -treiber. Diese Dienste waren für den Bootvorgang nicht notwendig, stellen aber z. B. wichtige Funktionen bereit, die für eine Anmeldung und ein interaktives Arbeiten notwen-

dig sind. Sobald alle Dienste gestartet wurden, ruft SCM eine Unterroutine auf, die standardmäßig 120 Sekunden wartet und dann alle Delayed-Auto-Start-Dienste (verzögerter Start) startet.

Außerdem überwacht der SCM Ereignisse im System und verwaltet die Trigger, die dafür sorgen, dass Ondemand-Dienste beim Eintreten eines bestimmten Ereignisses gestartet oder gestoppt werden.

Die Schritte in Session 1

Die Winlogon.exe (Session 1-n) kümmert sich um die Anmeldung des Benutzers. Dazu zeigt sie das Anmeldefenster an (LogonUI.exe) und prüft die dort eingegebenen Anmeldedaten. Nach erfolgreicher Authentifizierung lädt Winlogon.exe das Profil des Benutzers (NTUSER.DAT) in die Registry. Anschließend wird Userinit.exe gestartet.

Bevor Winlogon.exe die Anmeldung durchführen kann, muss Wininit.exe das Sicherheitssystem (LSASS) gestartet haben.

Winlogon Notification

Die unter Windows XP häufig verwendeten Winlogon Notification wurden abgeschafft. Unter XP hat Winlogon bestimmte DLLs bereits vor der Anmeldung geladen und diese über Ereignisse wie Logon/Logoff informiert. Die DLLs unter *HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Notify* werden seit Windows Vista nicht mehr gestartet. Lesen Sie dazu auch <http://technet.microsoft.com/de-de/library/cc721961.aspx>.

Userinit.exe führt Skripte aus, die in den Gruppenrichtlinien definiert sind, startet die proquota.exe, falls eine Quota für das Profil definiert wurde, und startet die Shell, die in der Regel die Explorer.exe ist (*HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Shell=explorer.exe*).

In jeder Session stellt die erste Instanz der Explorer.exe den Desktop dar. Kann die Explorer.exe nicht gestartet werden oder der Prozess wird beendet, verschwinden auch das Startmenü und die Desktop-Icons.

Im Process Explorer ist schön zu sehen, wie die Prozess-Struktur aussieht. Man sieht z. B., dass lsm.exe, lsass.exe und services.exe Folgeprozesse von Wininit.exe sind. In der Spalte *Sessions* ist auch zu erkennen, dass Winlogon.exe und der Explorer in Session 1 laufen, alle Dienste (Services.exe) aber in Session 0.