

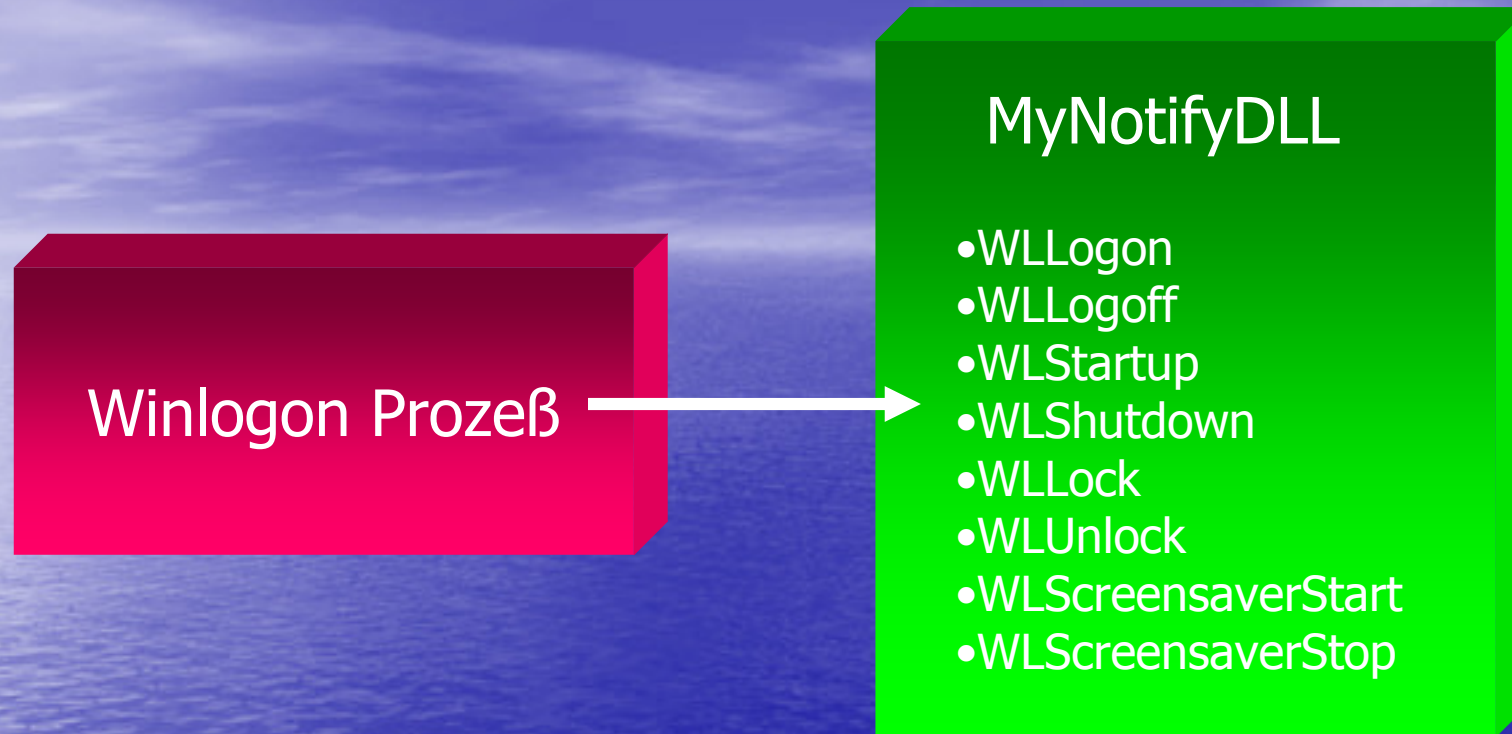
Windows Vista

TTT

Tips, Trick and Traps

Funktionalitäten die wegfallen

- MSGINA (Microsoft Graphical Identification and Authentication
Ersatz: ICredentialProvider
- Winlogon Notification API
Ersatz: Funktionalität in eigenen Service auslagern.
Neue SCM Messages
Achtung ! WL -> synchron oder asynchron
Service nur asynchron



Installation über Registry:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Notify

- OS Version Check !

keine harte Abfrage

```
if(OS.Major == 6)
```

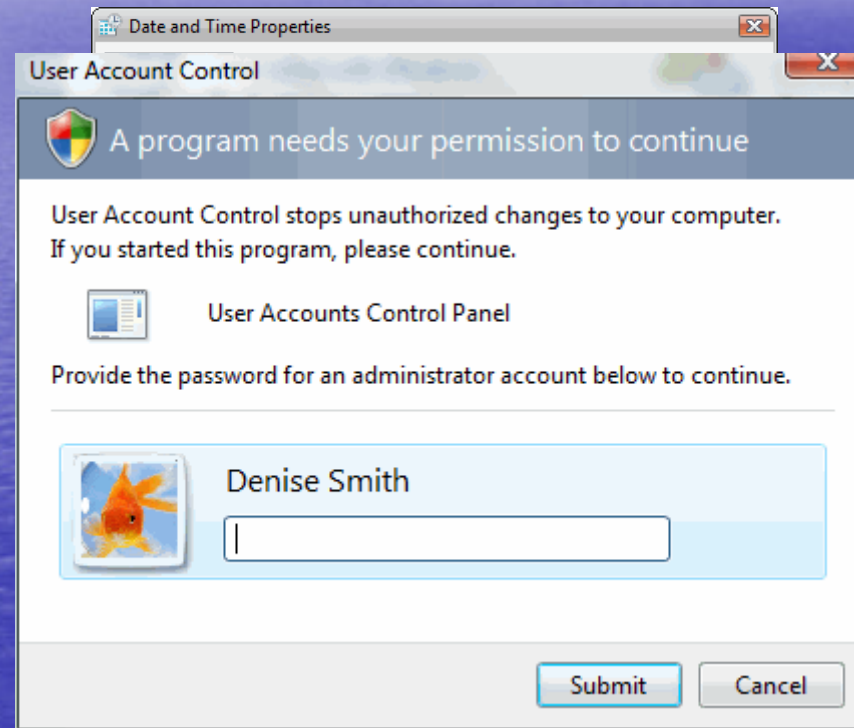
● UAC (User Account Control)

„ A fundamental step toward increasing the security of Windows is enabling interactive users to run with a standard user account, which gives them access to only a limited set of permissions and privileges. By default, Windows Vista will run every application as a standard user even if the users logs on as a member of the administrators' group. Conversely, when users attempt to launch an application that has been marked as requiring administrator permissions, the system will explicitly ask them to confirm their intention to do so. Only applications running with administrator privileges can modify system and global settings and behavior. This feature of Windows Vista is the User Account Control “

Administrator

- Token 1: normale Userrechte
- Token 2: Adminrechte

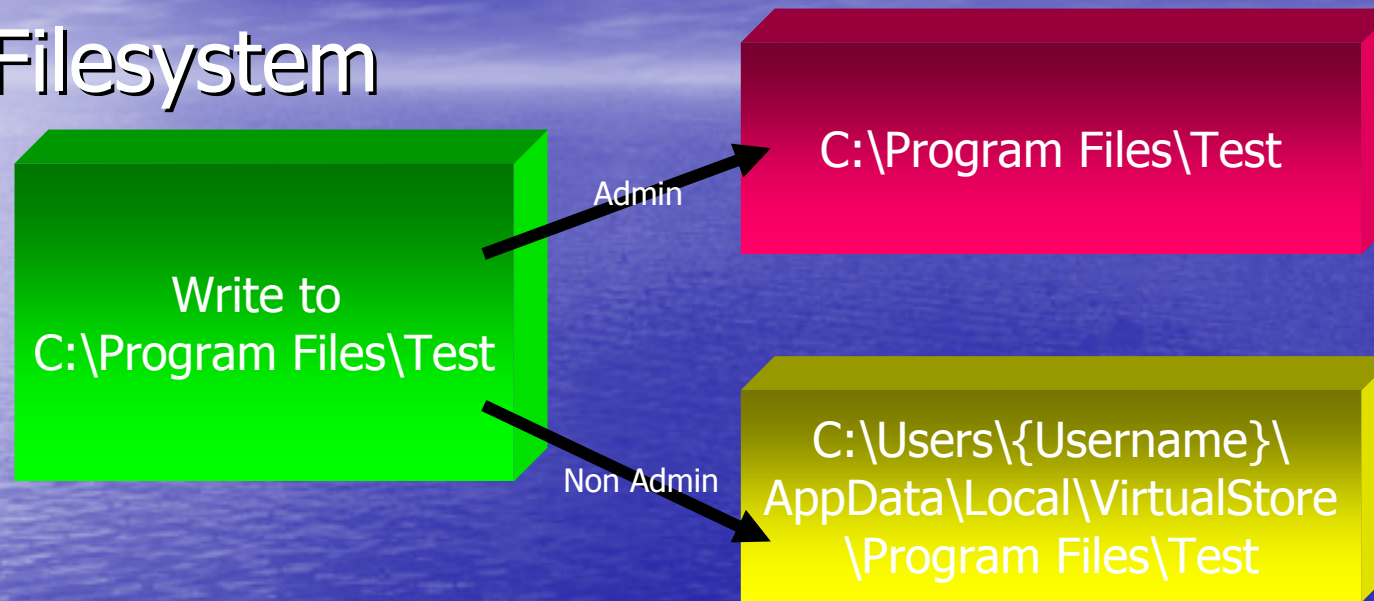
- Globale und Systemeinstellungen nur vom Admin zu ändern



•-> Demo

Virtualisierung abhängig vom User-Kontext

- Filesystem



- Registry

- -> Demo

Virtualisierung abhängig vom User-Kontext

Achtung !
Funktioniert nicht für Applikationen die für
Vista über eine Manifestdatei vorbereitet sind !!!

Vorbereitung UAC durch Manifestdatei

```
<?xml version="1.0" encoding="utf-8" ?>  
<assembly xmlns="urn:schemas-microsoft-com:asm.v1" manifestVersion="1.0">  
<assemblyIdentity version="1.0.0.0"  
  processorArchitecture="X86"  
  name="WindowsVistaReadiness"  
  type="win32" />
```

asInvoker	(inherit privilege of caller)
highestAvailable	(maximum potential privilege of the user)
requireAdministrator	(user must be administrator)

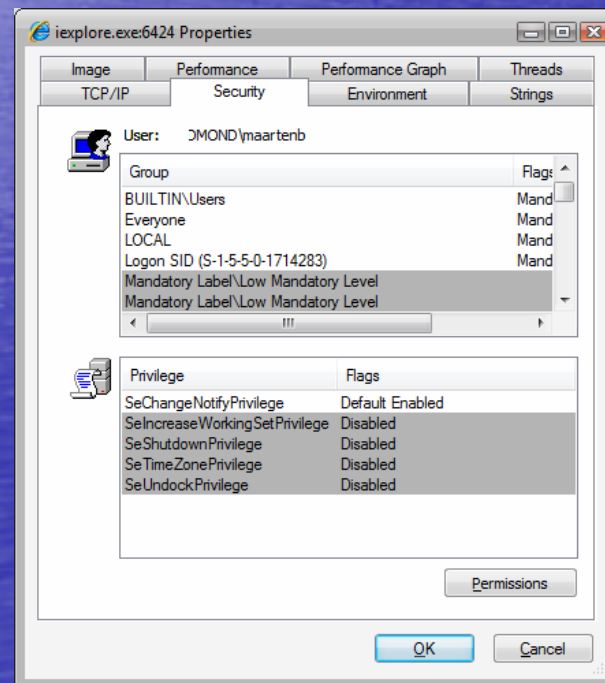
```
</requestedPrivileges>  
</security>  
</trustInfo>  
</assembly>
```

Manifestdatei:

- Programm.exe.manifest in Verzeichnis mit der ausführbaren Datei.
- Direkt zum Programm dazulinken
- -> Demo

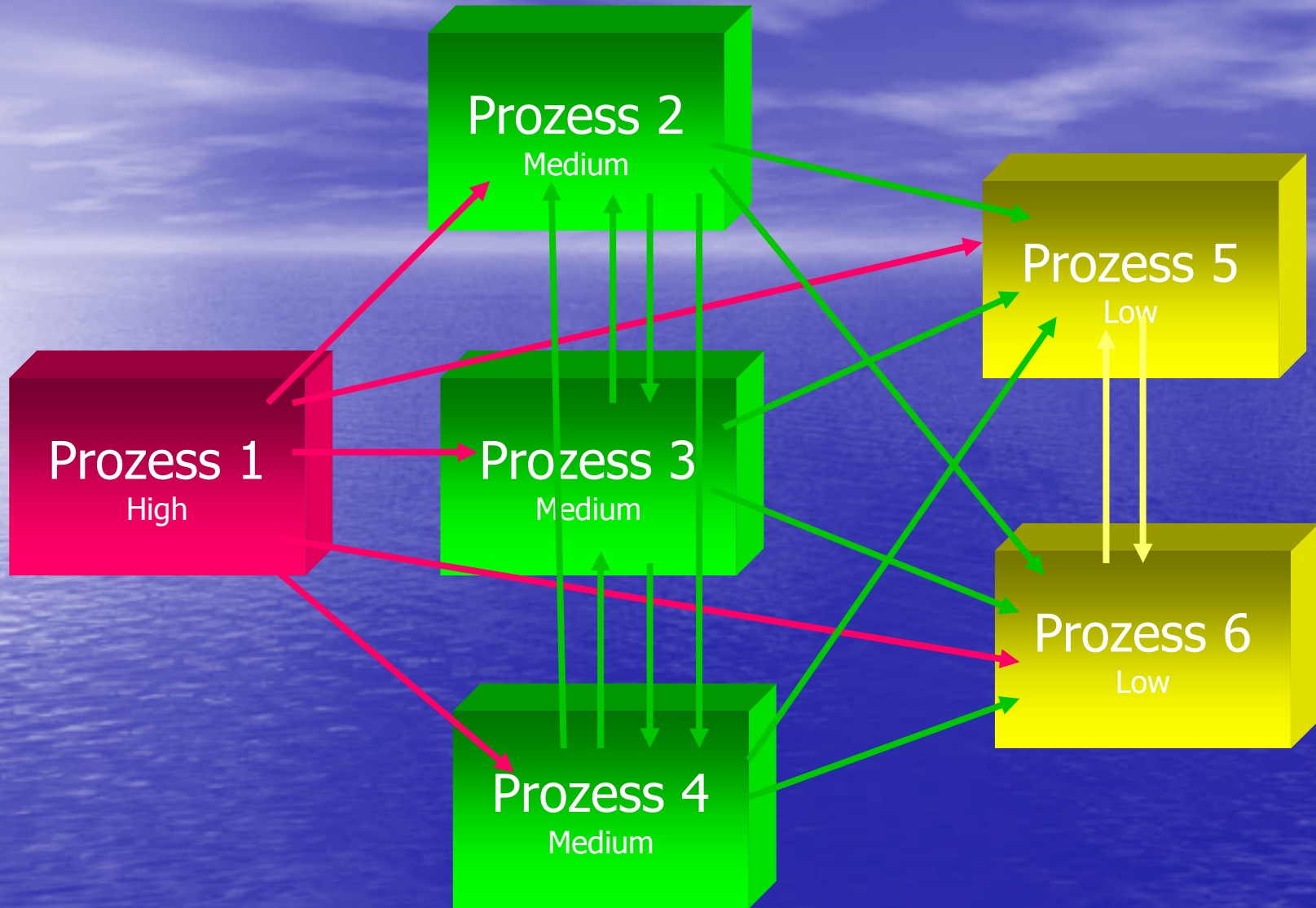
Mandatory Integrity Control (MIC) User Interface Privilege Isolation (UIPI)

- IPC via Windows Messages:
Send/Postmessage/FindWindow/EnumWindows...



3 Level:

- > Low
- > Medium
- > High



● -> Demo

Windows resource protection

- Windows Resource Protection (WRP) is designed to protect a Windows system in a read-only state in an attempt to increase system stability, predictability and reliability. This will affect specific files, folders, and registry keys. Updates to protected resources are restricted to the OS trusted installers, such as Windows Servicing. This enables components and applications that ship with the OS to be better protected from the impact of other applications and administrators.

- -> Demo

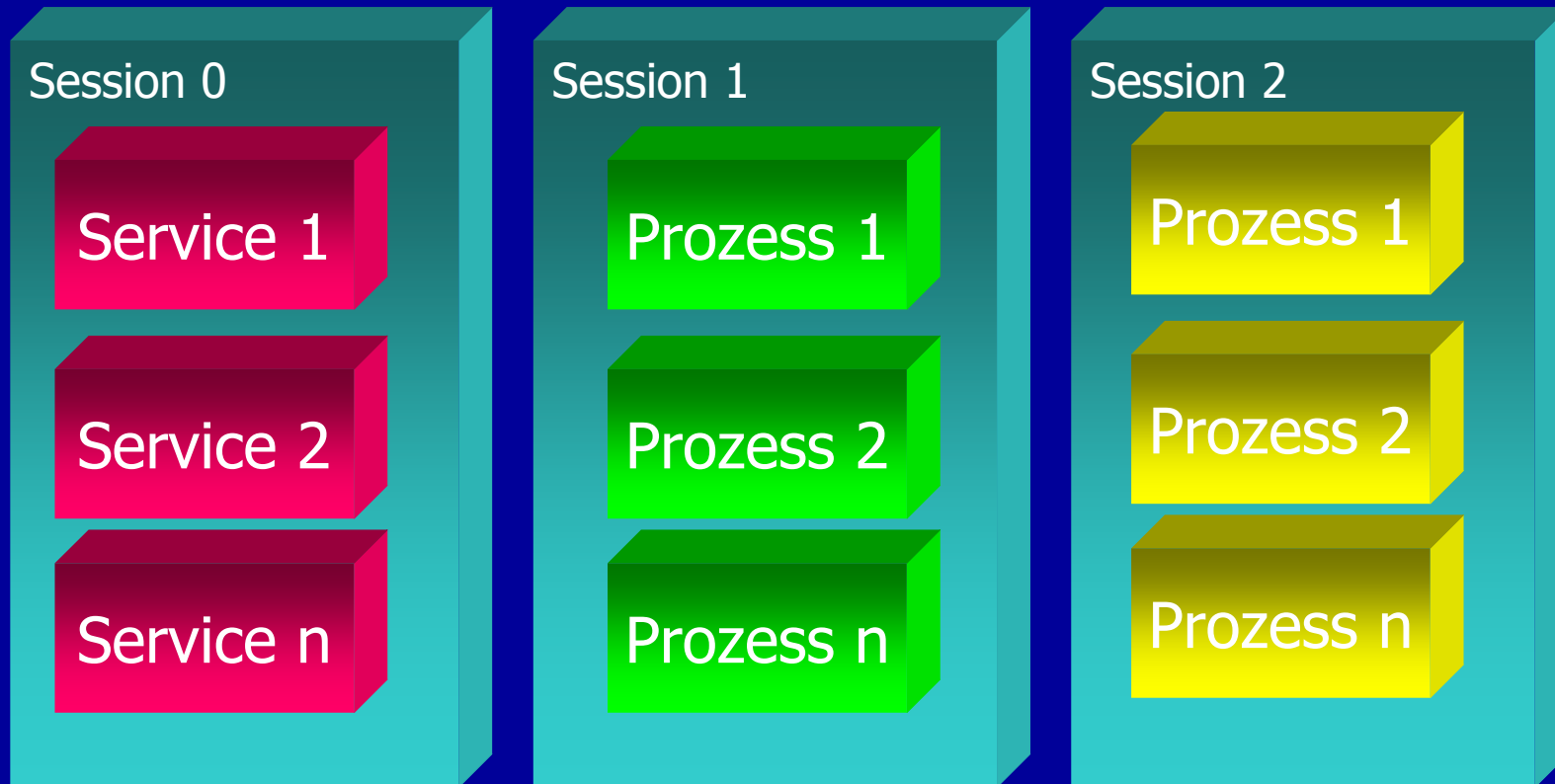
Session 0 isolation

W2K, XP, W2K3



Session 0 isolation

Vista



- Session 0 hat kein UI

- -> Demo

Dies und das ...

- Achtung beim Erzeugen von globalen Objekten (z.B. Mutex) Global\ oder Local\ berücksichtigen ! -> vgl. WTS
- Neue API Funktionen z.B. für Registry (RegGetValue), SH... etc. WTS Funktionen auch ohne WTS.
- Debugging von Services und (Webservices) unter VS2005 als NichtAdmin nicht möglich (attach Prozeß)

Enumerieren Windows Stations und Desktops

```
C:\WINNT\system32\cmd.exe
Station: WinSta0
  Desktop: ZZBoschRASPolicy
  Desktop: Default
  Desktop: Winlogon
Station: Service-0x0-3e7$
  Desktop: WinSta0
  Desktop: Service-0x0-3e7$
  Desktop: SAWinSta
  Desktop: Service-0x0-11510$
Station: SAWinSta
  Desktop: SADesktop
Station: Service-0x0-11510$
  Desktop: WinSta0
  Desktop: Service-0x0-3e7$
  Desktop: SAWinSta
  Desktop: Service-0x0-11510$
Drücken Sie eine beliebige Taste . . .
```

W2K / XP

```
Visual Studio 2005 Command Prompt
Station: WinSta0
  Desktop: Default
  Desktop: Disconnect
  Desktop: Winlogon
Station: Service-0x0-3e7$
  Desktop: WinSta0
  Desktop: Service-0x0-3e7$
  Desktop: Service-0x0-3e4$
  Desktop: Service-0x0-3e5$
Station: Service-0x0-3e4$
  Desktop: WinSta0
  Desktop: Service-0x0-3e7$
  Desktop: Service-0x0-3e4$
  Desktop: Service-0x0-3e5$
Station: Service-0x0-3e5$
  Desktop: WinSta0
  Desktop: Service-0x0-3e7$
  Desktop: Service-0x0-3e4$
  Desktop: Service-0x0-3e5$
```

Vista als Service

.Net unter Vista

- Viele Dinge sind nur über P/Invoke erreichbar
z.B. neue Service Notifications, WTS
Funktionen, Teile für Security (z.B. ACL
auf Prozesse ...)

Zusammenfassung

- UAC (noch nicht konsistent)
GUI-Tools von MS 95% ready
Commandline Tools - keine Manifestdateien

**Arbeiten ohne
Adminrechte !**

HKLM\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\POLICIES\SYSTEM\ENABLELUA -> 0/1

- Sicherheit größer ??
 - Arbeiten ohne Userrechten
 - Programme sollten ohne Admin laufen (VS2005 debuggen von Services / Webservices !)
 - wenn UAC zu oft nervt -> kontraproduktiv
 - weniger Vulnerabilities -> wird sich zeigen !
- -> Demo

Links

- www.devreadiness.org
- <http://msdn.microsoft.com/windowsvista/>
- <http://blogs.msdn.com/uac/>
- PDC 2005:
5+ways+to+ensure+your+app+will+not+run+on+Vista
+APPcompatGeneral.ppt
- <http://msdn.microsoft.com/windowsvista/default.aspx?pull=/library/en-us/dnlong/html/AccProtVista.asp>
- „Design a Windows NT Service to Exploit Special Operating System Facilities“, Jeffrey Richter – MSJ October 1998