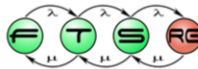


Konfigurációkezelés Windowson

Micskei Zoltán

<http://mit.bme.hu/~micskeiz>

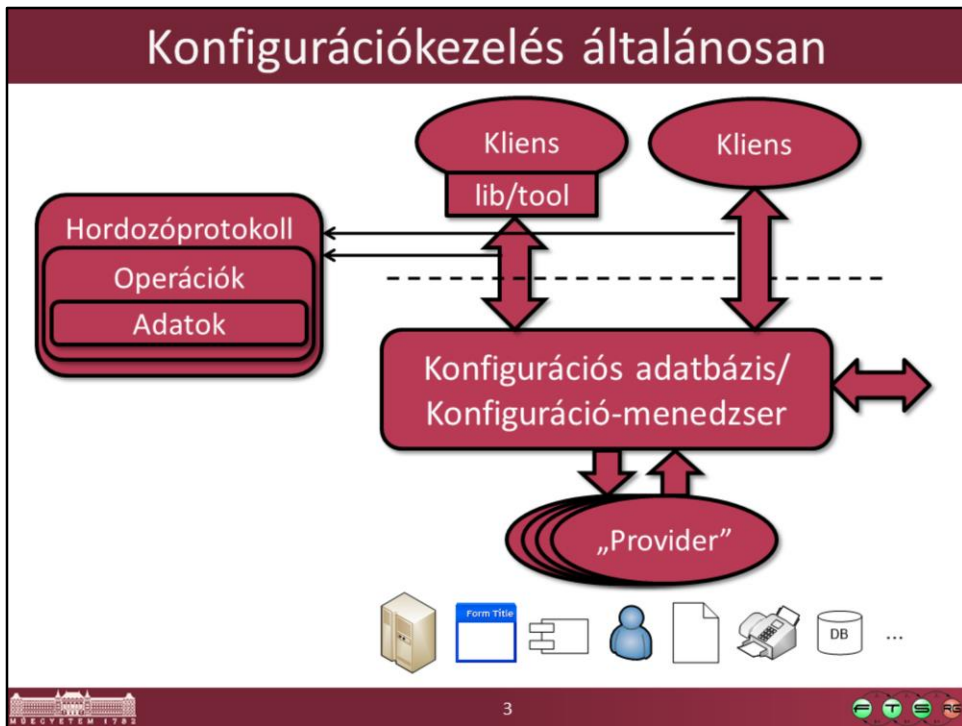


Utolsó módosítás: 2014. 03. 26.

A félév eddigi anyagai

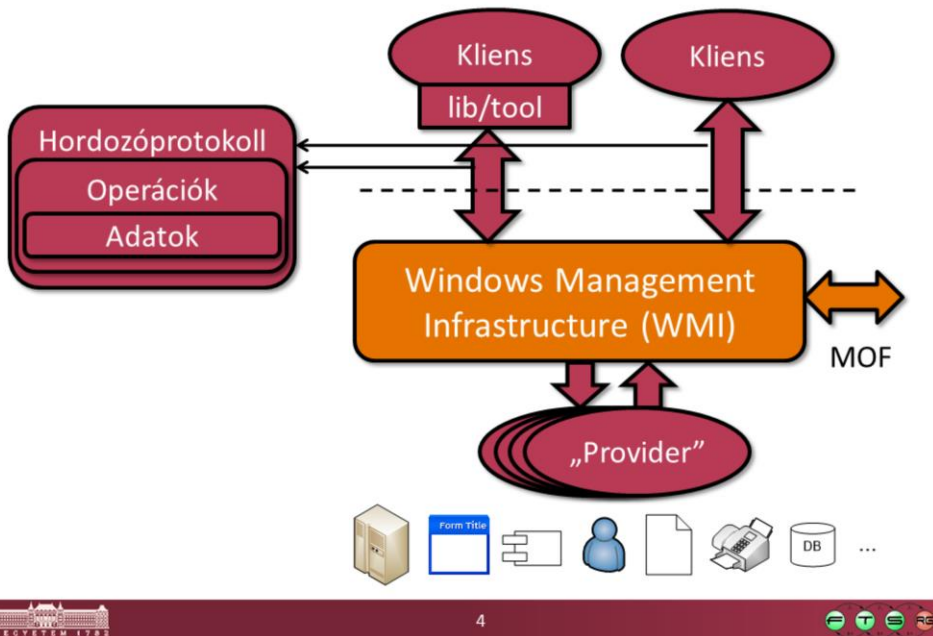
- Modellezés
- Szkriptelés
- Címtárak

- Konfigurációkezelés
 - Konfigurációkezelés alapok
 - **Konfigurációkezelés Windowson**



Ez volt az általános séma, amit előző előadáson felrajzoltunk.

Konfigurációkezelés Windowson



Windows Management Infrastructure (WMI)



WMI felhasználói szemszögből

- CIM Object Manager (CIMOM) Windowsra
 - „winmgmt” szolgáltatás
 - Kiterjesztett sémakészlet

- A modern Windowsokon elérhető
 - Felügyeleti megoldások alapja

- Interfész:
 - Belül alapvetően C++
 - De sokféle egyéb elérés: szkriptnyelvek, .NET...

Miért fontos a CIM / WMI?

Windows Management Framework:

- **Windows 8 vonal:** radikális változás
 - Újraírt API: Management Instrumentation (MI)
 - Igazodás a DMTF szabványokhoz
 - Nyílt forráskódú implementáció is: [OMI](#)
 - ~600 új CIM-et becsomagoló PS cmdlet
- **Miért?**
 - **Cloud OS** „Generalized cloud computing requires standards-based management.”
 - Összefoglaló: [Standards-based Management in Windows Server “8”](#)



A WMI korábbi neve ‚Windows Management Instrumentation’ volt, Windows 8 óta inkább ‚Windows Management Infrastructure’ néven hivatkoznak rá.

Windows Server Blog: **Standards-based Management in Windows Server “8”**

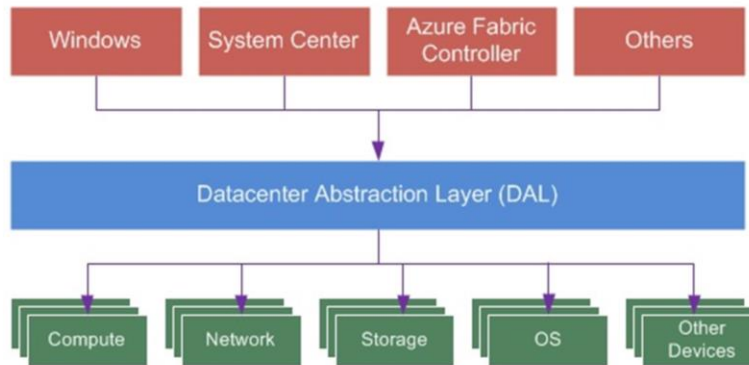
URL: <http://blogs.technet.com/b/windowsserver/archive/2012/03/30/standards-based-management-in-windows-server-8.aspx>

Az OMI implementációról (korábban NanoWBEM) bővebben:

Windows Server Blog: **Open Management Infrastructure**

URL: <http://blogs.technet.com/b/windowsserver/archive/2012/06/28/open-management-infrastructure.aspx>

Datacenter Abstraction Layer (DAL)



Forrás: <http://technet.microsoft.com/en-us/library/dn265975.aspx>

Microsoft: Datacenter Abstraction Layer (DAL) Overview, URL:
<http://technet.microsoft.com/en-us/library/dn265975.aspx>

DAL-nak hívják most azokat a technológiákat, PowerShell modulokat, CIM profilokat, amiknek a segítségével egy adatközpontban az alapvető hardverelemek (szerverek, hálózati elemek, tápegységek, tárhelyek...) jellegzetességeit el lehet fedni, és a legfontosabb feladatokhoz (pl. ki/bekapcsolás, firmware frissítés, hardverleltár...) egy általános, magas szintű interfészt biztosítanak. Ez főleg a saját Azure cloud controller fejlesztése kapcsán dolgozták ki.

CIM/WMI/Powershell jelentősége

~Az összes régi parancssori eszköz lecserélése

ipconfig.exe	Get-NetIPAddress
netsh.exe	Get-NetFirewallRule
diskpart.exe	Get-Disk
dnscmd.exe	Add-DnsServerPrimaryZone
wmic.exe	Get-CimInstance
...	...

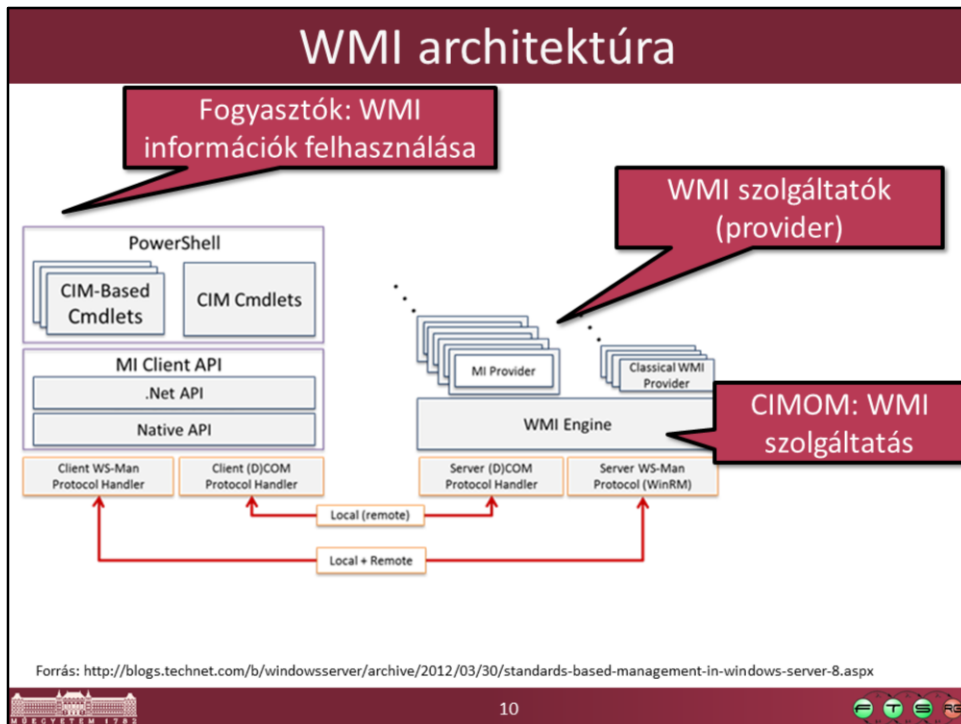
```
netsh>firewall
In future versions of Windows, Microsoft might remove the Netsh functionality
For Windows Firewall with Advanced Security.

Microsoft recommends that you transition to Windows PowerShell if you currently
use netsh to configure and manage Windows Firewall with Advanced Security.

Type Get-Command -Module NetSecurity at the Windows PowerShell prompt to view
a list of commands to manage Windows Firewall with Advanced Security.

Visit http://go.microsoft.com/fwlink/?LinkId=217627 for additional information
about PowerShell commands for Windows Firewall with Advanced Security.
netsh firewall>
```

- A legtöbb régi parancssori eszközt felváltja valamelyik PowerShell modul és a benne lévő cmdletek a következő Windowsokban.
- Ezeknek az eredménye szűrhető, kereshető, távoli hozzáférés egyszerűen, szabványosan megy.



Elemek:

- Managed objects: akiket a WMI kezel
- Providers: Managed objectról szolgáltató adatokat a WMI interfészének megfelelően. Egy DLL fájlból (megvalósítja az interfész műveleteit), és egy MOF fájlból áll (CIM osztályok, attribútumok és műveleteik leírása)
- WMI service: beolvassa és létrehozza a névtereket, végrehajtja a fogyasztók műveleteit
- Consumers: olyan alkalmazások, amik felhasználják a WMI-t (pl. felügyeleti szoftverek, szkriptek, stb.)

Példa: néhány WMI névtér, osztály

root\cimv2

- **Hardver**
 - Win32_BaseBoard
 - Win32_DiskDrive
- **OS**
 - Win32_PageFile
 - Win32_Registry
- **Telepített szoftver**
 - Win32_Patch
 - Win32_Product
- **Teljesítményszámlálók**
 - Win32_Perf
 - Win32_PerfRawData

directory\ldap

- ds_person
- ds_group

Microsoft\SqlServer

- SqlService
- ServerSettings

IntelNIC

- IANet_EthernetAdapter

DEMO WBEM könyvtár, MOF fájlok

- WMI részei:
 - C:\windows\system32\wbem

- Itt találhatóak:
 - *.exe
 - *.dll: provider-ek megvalósítása
 - *.mof



A MOF fájlokat általában át kell másolni valahova, hogy meg tudjuk nézni őket.

WMI Query Language (WQL)

- WMI: nagy mennyiségű adat
- WQL: keresés, szűrés ezekben
- SQL szerű szintaxis:
 - **SELECT** attrib **FROM** osztály **WHERE** feltétel

- Példa:

```
SELECT eventcode, message, sourcename  
FROM Win32_NTLogEvent  
WHERE type = 'error' AND logfile = 'system'
```



WMI felhasználása

- Ad-hoc lekérdezések, utasítások
- PowerShell szkriptek készítése
 - bonyolultabb műveletek, rendszeres feladatok
- Felhasználás .NET-es alkalmazásban
 - Microsoft.Management.Infrastructure névtér
- Windows Server 2012-től kezdve
 - Távoli menedzsment alapja a WMF
- Rendszermenedzsment alkalmazáson keresztül
 - Pl.: MS System Center, HP OpenView...

WMI felhasználása – PowerShell

- CIMCmdlets modul parancsaival:

```
PS C:\> Get-CimInstance Win32_Processor
```

DeviceID	Name	Caption	MaxClockSpeed	SocketDesignati on	Manufacturer
-----	----	-----	-----	-----	-----
CPU0	Intel(R) Core...	Intel64 Famil...	2001	Microprocessor	GenuineIntel

```
PS C:\> Get-CimInstance Win32_Processor | gm
```

TypeName: Microsoft.Management.Infrastructure.CimInstance#root/cimv2/Win32_Processor

- Van TAB kiegészítés a WMI osztályokra is!
- Visszaadott objektum:
 - Típus: lekérdezett CIM osztálynak megfelelően
 - CIM-ben definiált tulajdonságok és metódusok



Get-CimInstance Win32_Processor
Get-CimInstance Win32_Processor | Get-Member

PowerShell: WMI szűrés, keresés

- Beépített CIM osztályok:

```
PS C:\> (Get-CimClass -Namespace "root/cimv2").count  
1094
```

- Eredmények szűrése:

- -Query paraméter: WQL vagy CQL kifejezés
- -Filter paraméter: WQL where része csak
- vagy: PowerShell cmdletek használata

PowerShell: WMI szűrés, keresés

- Példa: keressük meg a memóriával kapcsolatos CIM objektumokat

```
PS C:\> Get-CimClass -ClassName *memory*

Namespace: ROOT/cimv2

CimClassName      CimClassMethods      CimClassProperties
-----
CIM_Memory         {SetPowerState, R... {Caption, Description, InstallDate, Nam...
CIM_CacheMemory   {SetPowerState, R... {Caption, Description, InstallDate, Nam...
Win32_CacheMemory {SetPowerState, R... {Caption, Description, InstallDate, Nam...
Win32_SMBIOSMemory {SetPowerState, R... {Caption, Description, InstallDate, Nam...
Win32_MemoryArray {SetPowerState, R... {Caption, Description, InstallDate, Nam...
Win32_MemoryDevice {SetPowerState, R... {Caption, Description, InstallDate, Nam...
```

- Nézzük meg, hogy fut-e a W32Time szolgáltatás:

```
PS C:\> Get-CimInstance -Class Win32_Service -Filter "Name = 'w32time'" | select DisplayName, Status, Started | fl

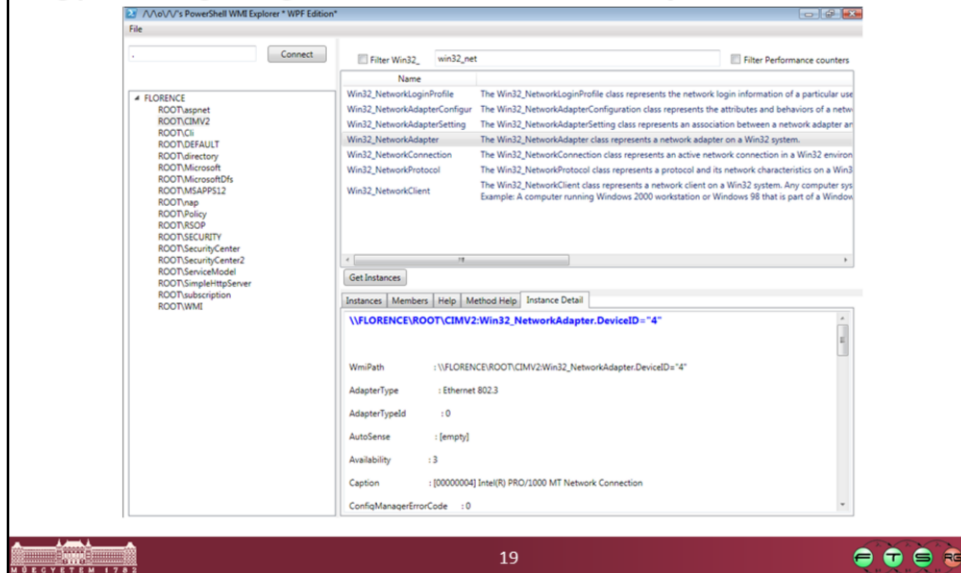
DisplayName : Windows Time
Status      : OK
Started     : False
```

Get-CimClass -ClassName *memory*

Get-CimInstance -Class Win32_Service -Filter "Name = 'w32time'" | select
DisplayName, Status, Started | fl

PowerShell WMI Explorer

Egy kis segítség: PowerShell WMI Explorer



Eredeti elérés: PowerShell WMI Explorer letöltése,
<http://thepowershellguy.com/blogs/posh/pages/powershell-wmi-explorer.aspx>
Ez a link már nem működik, de innen elérhető:
<http://mit.bme.hu/~micskeiz/files/WpfWmiExplorer.zip>

PowerShell: CIM kapcsolatok kezelése

- **Get-CimAssociatedInstance cmdlet:**

```
PS C:\> Get-CimAssociatedInstance -InputObject (Get-CimInstance -ClassName Win32_Processor) | select
Name, CimClass
-----
Name                                CimClass
----                                -
Cache Memory                        root/cimv2:Win32_CacheMemory
Cache Memory                        root/cimv2:Win32_CacheMemory
BELEG                               root/cimv2:Win32_ComputerSystem
```

- **Kapcsolóosztályok lekérdezése:**

- `Get-CimClass -ClassName "Win32_*" -QualifierName "Association"`

- **Kapcsolóosztály tartalma:**

```
PS C:\> Get-CimInstance Win32_DependentService | Select-Object Antecedent, Dependent
Antecedent                                Dependent
-----
Win32_SystemDriver (Name = "FltMgr")      Win32_SystemDriver (Name = "AppID")
Win32_SystemDriver (Name = "discache")    Win32_SystemDriver (Name = "AppID")
Win32_SystemDriver (Name = "rdbss")       Win32_SystemDriver (Name = "CSC")
Win32_SystemDriver (Name = "Mup")         Win32_SystemDriver (Name = "Dfsc")
Win32_SystemDriver (Name = "FltMgr")      Win32_SystemDriver (Name = "FileInfo")
```

`Get-CimAssociatedInstance -InputObject (Get-CimInstance -ClassName Win32_Processor) | Select-Object Name, CimClass`

`Get-CimInstance Win32_DependentService | Select-Object Antecedent, Dependent`

PowerShell: WMI kapcsolatok ábrázolása



Lásd: WMI kapcsolatok megjelenítése, <http://blog.inf.mit.bme.hu/?p=437>



Micskei Zoltán. **WMI kapcsolatok megjelenítése** URL:
<http://blog.inf.mit.bme.hu/?p=437>

Fontosabb CIM cmdletek összefoglalása

- **Kapcsolat kezelése:**
 - Get-CimSession, New-CimSession, Remove-CimSession, New-CimSessionOption
- **CIM osztályok:**
 - Get-CimClass
- **CIM példányok:**
 - Get-CimInstance, New-CimInstance, Remove-CimInstance, Set-CimInstance
- **Kapcsolódó példányok:**
 - Get-CimAssociatedInstance
- **Metódushívás:**
 - Invoke-CimMethod



Get-Command -Module CimCmdlets

PowerShell: CIM-based cmdlet

- Egy CIM/WMI osztályhoz generálható PS cmdlet
- Pl. Get-NetAdapter csak burkoló:
root/StandardCimv2/MSFT_NetAdapter
- Ehhez egy leképző XML-t kell megadni:
 - Cmdlet Definition XML (CDXML)
 - Akár ez mutathat távoli (nem Windows) gépre
 - Segítség: [CIM IDE](#)



23



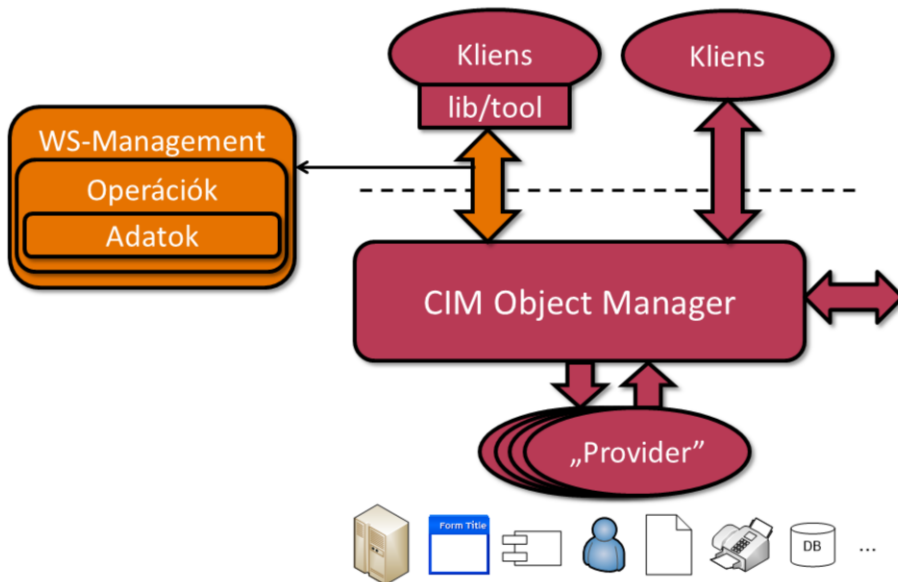
Lásd még: MSDN. **Getting started with CDXML** URL: [http://msdn.microsoft.com/en-us/library/windows/desktop/jj542525\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/jj542525(v=vs.85).aspx)

CIM IDE: <http://archive.msdn.microsoft.com/cimide>

Web Services for Management (WS-Management)

Kibocsátó:	Distributed Management Task Force (DMTF)
Megalkotók:	AMD, Dell, Intel, Microsoft, Sun...
Verzió:	első draft (2005. 09.), aktuális: 1.1.1 (2012. 10. 10.)
Cél:	Webszolgáltatás alapú protokoll rendszerek menedzseléséhez

Konfigurációkezelés általánosan



WS-Management

- WBEM egyik protokollja
 - 5985/5986 portot használja
- Web Service alapú
 - sok egyéb WS-* protokollra épül
- CIMOM elérése, lekérdezése, menedzselése
 - De használható nem csak CIMOM-hoz!

WS-Management példa üzenet

```
<s:Header>
  <wsa:To> Address </wsa:To>
  <wsa:Action> Action URI </wsa:Action>
  <wsman:ResourceURI>resURI</wsman:ResourceURI>
  <wsman:SelectorSet>
    <wsman:Selector Name="selector-name">
      selector-value
    </wsman:Selector>
  </wsman:SelectorSet>
  ...
</s:Header>
```

Melyik gép

Mit akarunk

Gép melyik erőforrásával

Ha az erőforrás egy csoport (pl. CPU-k), akkor a csoport melyik tagja

```
<s:Header>
  <wsa:To> Address </wsa:To>
  <wsa:Action> Action URI </wsa:Action>
  <wsman:ResourceURI>resURI</wsman:ResourceURI>
  <wsman:SelectorSet>
    <wsman:Selector Name="Selector-name">
      Selector-value
    </wsman:Selector>
  </wsman:SelectorSet>
  ...
</s:Header>
```

- A ResourceURI tipikusan egy osztályt azonosít, és a Selectorok segítségével azonosítjuk, hogy az osztály melyik példányát akarjuk elérni.
- Ezen kívül még minden üzenet szokott kapni egy MessageID egyedi azonosítót, amire egy másik üzenetben a RelatesTo fejléc elemmel tudunk hivatkozni.
- (Az üzenet a WS-Management úgynevezett default addressing model címzési módszerét használja, ezen kívül lehetne még másfajtaikat alkalmazni.)

Endpoint Reference (EPR)

- Erőforráspéldányok egyedi azonosítója
- Hálózati cím + ResourceURI (+ SelectorSet)
- Példa:

```
<wsa:EndpointReference>
  <wsa:Address>http://123.99.222.36/wsman</wsa:Address>
  <wsa:ReferenceParameters>
    <wsman:ResourceURI>http://example.org/hardware/physDisk</wsman:ResourceURI>
    <wsman:SelectorSet>
      <wsman:Selector Name="LUN">2</wsman:Selector>
    </wsman:SelectorSet>
  </wsa:ReferenceParameters>
</wsa:EndpointReference>
```



28



Példa EPR:

```
<wsa:EndpointReference>
  <wsa:Address>http://123.99.222.36/wsman</wsa:Address>
  <wsa:ReferenceParameters>
    <wsman:ResourceURI>http://example.org/hardware/physDisk</wsman:ResourceURI>
  >
    <wsman:SelectorSet>
      <wsman:Selector Name="LUN">2</wsman:Selector>
    </wsman:SelectorSet>
  </wsa:ReferenceParameters>
</wsa:EndpointReference>
```

WS-Management: tipikus műveletek

- **(IDENTIFY)**
 - távoli gép paramétereinek felderítése
- **GET, PUT, CREATE, DELETE**
 - erőforrások kezelése
- **ENUMERATE, PULL**
 - gyűjtemények elemeinek felsorolása
- **SUBSCRIBE, UNSUBSCRIBE...**
 - események kezelése
- **EXECUTE**
 - metódusok meghívása



29



- ENUMERATE esetén a legegyszerűbb esetben először egy úgynevezett enumeration contextet kap vissza a kliens, és ebből tudja az elemeket PULL segítségével lekérni.
- Események kezelésével nem fogunk most foglalkozni, az túlmutat a tárgy keretein.
- Az IDENTIFY igazából teljesen más címezést és XML elemeket használó művelet (pl. nem kell neki ResourceURI-t megadni), de a teljesség kedvéért itt is megemlítjük.

WS-Management és CIM

- Eddig semmi köze nem volt még a CIM-hez
- *CIM Binding Specification* adja meg a kapcsolatot
 - Milyen URI-kat és akciókat kell használni
- *WS-CIM mapping*:
 - CIM osztály leképezése egy XML névtérbe:
`http://schemas.dmtf.org/wbem/wscim/X/cim-schema/Y/ClassName`
(X: WS-CIM verziója, Y: CIM séma verziója)
 - Nem kerül bele a CIM névtér (pl. root/cimv2)!
 - Lehet gyártó-specifikus a névtér is
- Példa:
`http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_ComputerSystem`



30



- DMTF. „WS-Management CIM Binding Specification”, 1.1.0, DSP0227. URL:
http://www.dmtf.org/sites/default/files/standards/documents/DSP0227_1.1.0.pdf
- DMTF. „WS-CIM Mapping Specification”, 1.0.2, DSP0230. URL:
http://www.dmtf.org/sites/default/files/standards/documents/DSP0230_1.0.2.pdf
- A névteret egy speciális Selectorban kell megadni:
`<wsman:Selector Name="__cimnamespace">xs:anyURI</wsman:Selector>`
- Van egy speciális ResourceURI, az „All Classes” URI, amit pl. több osztályt érintő
kapcsolatok lekérdezésénél lehet használni:
`http://schemas.dmtf.org/wbem/wscim/1/*`
-



Implementációk

- **Windows Remote Management (WinRM)**
 - WMI lekérdezés és távoli parancssor
 - CLI, scripting API

- **Openwsman; F/OSS**
 - Kiszolgáló: CIM-XML-t „fordít” WS-Managementre
 - Adapter tervezési minta; SBLIM SFCC, mint kliens
 - CLI, mod_wsman, Ruby kötések

- **wiseman; F/OSS**
 - Java alapú kliens és kiszolgáló megvalósítás támogatása

Ki használja?

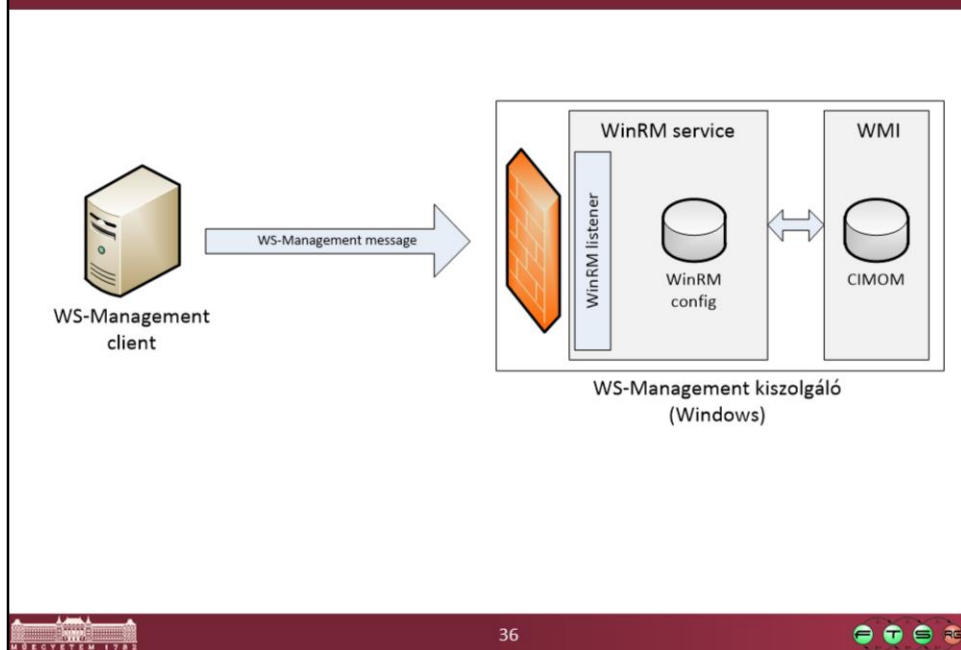
- Windows Server
 - Server Manager távoli gépre
 - PowerShell Remoting
- VMware ESXi
 - openwsman daemon futtatása
- Microsoft Operations Manager
 - Cross Platform: Linux gépek monitorozása
- HP System Insights Manager
- ...

Windows Remote Management (WinRM)

Windows Remote Management (WinRM)

- Új távoli menedzsment szolgáltatás a Windowsban
 - WS-Management implementáció
- Elérhetőség:
 - Windows 8, Server 2012: gyárilag (3.0)
 - Windows 7, Server 2008 R2: gyárilag (2.0), külön (3.0)
 - XP, Windows Server 2003 R2: külön telepíthető (2.0)
- WMI beágyazható WinRM-be
 - WMIv3-tól kezdve ezt használja a WMI alapból
 - De a WinRM nem csak WMI-hoz használható

WinRM architektúra



Egy gépen több listener is lehet, pl. az SSL-t használóhoz külön egy másik.

WinRM inicializálása

- WinRM részei:
 - WinRM szolgáltatás
 - WinRM listener: WS-Management üzenetek fogadására szolgáló végpont definíciója
- Inicializálás (kiszolgáló oldalán!):

```
PS C:\Windows\system32> Set-WSManQuickConfig

WinRM Quick Configuration
Running the Set-WSManQuickConfig command has significant security implications, as it enables
remote management through the WinRM service on this computer.
This command:
1. Checks whether the WinRM service is running. If the WinRM service is not running, the service
is started.
2. Sets the WinRM service startup type to automatic.
3. Creates a listener to accept requests on any IP address. By default, the transport is HTTP.
4. Enables a firewall exception for WS-Management traffic.
5. Enables Kerberos and Negotiate service authentication.
Do you want to enable remote management through the WinRM service on this computer?
[Y] Yes [N] No [S] Suspend [?] Help (default is "Y"):
```

Példa CIM lekérdezés WinRM-en keresztül

- Művelet: GET
- Erőforrás URI: CIM objektum
- Távoli gép, hitelesítési információ megadása

```
PS C:\> Get-WSManInstance -ResourceURI http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/Win32_LogicalDisk -SelectorSet
@{__cimnamespace="root/cimv2"; DeviceID="C:"} -ComputerName 10.6.17.44 -Credential $cred

xsi          : http://www.w3.org/2001/XMLSchema-instance
p            : http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/Win32_LogicalDisk
cim          : http://schemas.dmtf.org/wbem/wscim/1/common
type        : p:Win32_LogicalDisk_Type
lang        : hu-HU
Access      : 0
Availability : Availability
BlockSize   : BlockSize
Caption     : C:
Compressed  : false
ConfigManagerErrorCode : ConfigManagerErrorCode
ConfigManagerUserConfig : ConfigManagerUserConfig
CreationClassName : Win32_LogicalDisk
Description  : Local Fixed Disk
DeviceID    : C:
DriveType   : 3
ErrorCleared : ErrorCleared
ErrorDescription : ErrorDescription
ErrorMethodology : ErrorMethodology
FileSystem   : NTFS
FreeSpace    : 32499982336
InstallDate : InstallDate
LastErrorCode : LastErrorCode
```



38



```
Get-WSManInstance -ResourceURI http://schemas.dmtf.org/wbem/wscim/1/cim-
schema/2/Win32_LogicalDisk -SelectorSet @{__cimnamespace="root/cimv2";
DeviceID="C:"} -ComputerName 10.6.17.44 -Credential $cred
```

CIM objektum elérése (string)

- A Selector belerakható az URI-ba is:
 - prefix/osztály?tulajdonság=érték
- URI példa:
 - `http://schemas.microsoft.com/wbem/wsman/1/wmi/root/cimv2/Win32_Service?Name=Winmgmt`
- Ez nem ugyanaz, mint a CIM-XML ObjectPath!
- De mit használjunk prefix-nek?



39



Figyelem: a WSMAN ResourceURI más, mint a CIM object path-ban szereplő URL!

Ha több kulcs tulajdonsága lenne az adott osztálynak, akkor azokat + jellel lehet elválasztani. De ebben az esetben érdekesebb inkább már a -SelectorSet paramétert használni, és ott egy hash táblában lehet megadni.

URI alias-ok használata

winrm help alias

wmi = <http://schemas.microsoft.com/wbem/wsman/1/wmi>

wmicimv2 =
<http://schemas.microsoft.com/wbem/wsman/1/wmi/root/cimv2>

cimv2 = <http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2>



Figyeljünk arra, hogy más URL prefixet kell használni ha WMI specifikus osztályt, és mást, ha DMTF szabványos osztályt akarunk lekérdezni (különböző platformok közötti lekérdezéshez).

További műveletek

■ Enumerate:

- Több objektum lekérése
- `-Enumerate wmicimv2/Win32_LogicalDisk`

■ Invoke:

- Beavatkozás
- `Invoke-WSManAction -Action StopService`
`-ResourceURI wmicimv2/win32_service`
`-SelectorSet @{name="W32Time"}`

Biztonság

- **Titkosítás:**
 - Nincs / HTTPS csatorna / XML üzenet titkosítása
- **Hitelesítés:**
 - Basic / Negotiate / Kerberos ...
- **Távoli fél azonosítása:**
 - Tanúsítvány / tartományi környezetben Kerberos
- **Trusted hosts**

- **Leírások:**
 - `Get-Help about_Remote_Troubleshooting`

WinRM használata PowerShellből

- **Beépített alacsonyszintű cmdletek**
 - Lásd: `Get-Help about_wsman`
 - `Set-WSManQuickConfig`, `Test-WSMan`, `Get-WSManInstance`, `cd wsman:`
- **CIMCmdlets**: háttérben ezt használja
- **PowerShell Remoting**
 - parancsok/szkriptek végrehajtása távoli gépen
 - WinRM-et használ alul
 - Interaktív / batch mód (akár több gépen párhuzamosan)

DEMO ESXi lekérdezése WinRM-ből

- Kliens: Windows 8 (WinRM)
- Kiszolgáló: VMware ESXi (openwsman)

```
Get-WSManInstance cimv2/CIM_Processor  
-Enumerate  
-ComputerName 10.6.17.141  
-UseSSL -Port 443 -ApplicationName wsman  
-Authentication Basic -Credential $cred
```

- [VMware API leírás](#)

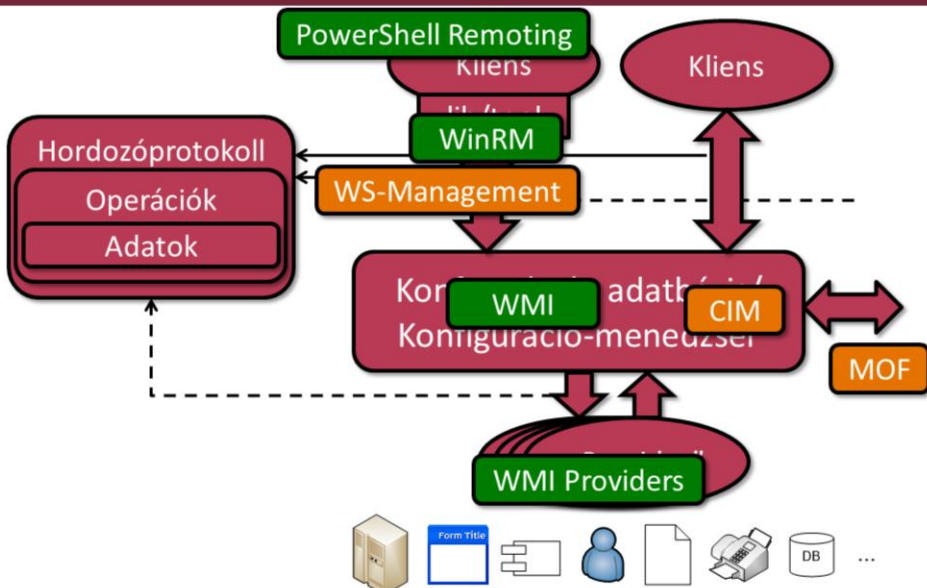
- VMware CIM API leírás: <http://www.vmware.com/support/developer/cim-sdk/>

```
$so = New-WSManSessionOption -SkipCACheck -SkipCNCheck -SkipRevocationCheck  
$cred = Get-Credential  
Get-WSManInstance cimv2/CIM_Processor -Enumerate -ComputerName  
10.6.17.141 -UseSSL -Port 443 -ApplicationName wsman -Authentication Basic -  
Credential $cred
```

DEMO PowerShell remoting

- Követelmény:
 - PowerShell 2.0, WinRM 2.0
 - Lásd: `Get-Help about_remoting_requirements`
- Remoting engedélyezése: `Enable-PSRemoting`
- Egy parancs távoli végrehajtása:
`Invoke-Command -ComputerName nev -ScriptBlock {...}`
- Távoli munkamenet:
 - `Enter-PSSession gepnev`
- Lásd: `Get-Help about_remoting`

Összefoglalás



További információ

- WMI
 - [MSDN: WMI](#)
 - [WMI Blog](#)
- WinRM
 - [MSDN: Windows Remote Management](#)
- Openwsman
 - <http://www.openwsman.org/>



- MSDN. Windows Management Instrumentation, [http://msdn.microsoft.com/en-us/library/aa394582\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/aa394582(VS.85).aspx)
- MSDN Blogs. Windows Management Infrastructure Blog, <http://blogs.msdn.com/b/wmi/>
- MSDN. Windows Remote Management, [http://msdn.microsoft.com/en-us/library/aa384426\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/aa384426(VS.85).aspx)

WinRM példa üzenet

■ Win32_LogicalDisk WMI objektum lekérdezése

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
  xmlns:a="http://schemas.xmlsoap.org/ws/2004/08/addressing"
  xmlns:w="http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd">
  <s:Header>
    <a:To>http://RemoteComputer:80/wsman</a:To>
    <w:ResourceURI s:mustUnderstand="true">
      http://schemas.microsoft.com/wbem/wsman/1/wmi/root/cimv2/win32_logicaldisk
    </w:ResourceURI>
    <a:ReplyTo>
    <a:Address s:mustUnderstand="true">
      http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
    </a:Address>
    <a:ReplyTo>
    <a:Action s:mustUnderstand="true">
      http://schemas.xmlsoap.org/ws/2004/09/transfer/Get
    </a:Action>
    <w:MaxEnvelopeSize s:mustUnderstand="true">153600</w:MaxEnvelopeSize>
    <a:MessageID>uuid:4ED2993C-4339-4E99-81FC-C2FD3812781A</a:MessageID>
    <w:Locale xml:lang="en-US" s:mustUnderstand="false"/>
    <w:SelectorSet>
    <w:Selector Name="DeviceId">c</w:Selector>
    </w:SelectorSet>
    <w:OperationTimeout>PT60.000S</w:OperationTimeout>
  </s:Header>
  <s:Body/>
</s:Envelope>
```

