

# Správa sítí pomocí OpenLMI

Radek Nováček

EurOpen – květen 2013

# Abstrakt

*Projekt OpenLMI si klade za cíl vytvořit jednotnou infrastrukturu pro správu linuxových systémů za použití otevřených standardů. Přednáška se bude zabývat představením tohoto projektu, úvodem do použitých standardů a dále návrhem rozhraní pro správu sítí a jeho implementací.*

# Obsah

- ▶ Co je OpenLMI
- ▶ WBEM a CIM
- ▶ Správa sítí s OpenLMI
- ▶ Závěr

# Co je OpenLMI

- ▶ Infrastruktura pro správu linuxových systémů za použití otevřených standardů
- ▶ Jednotné rozhraní pro konfiguraci systému
- ▶ Snadná integrace do existujících nástrojů

## Proč OpenLMI?

- ▶ Centrální bod pro správu
- ▶ Ostatní nástroje nepokrývají všechny požadavky (monitoring, správa, upozornění, atd.)
- ▶ Unifikace rozhraní

## Jak?

- ▶ Standardy: WBEM a CIM
- ▶ Open Source Software: LGPLv2.1+
- ▶ Pouze GNU/Linux
- ▶ Implementace providerů
- ▶ Skriptování (OpenLMI Shell)
- ▶ Dokumentace

## Současný stav

- ▶ Síťování
- ▶ Disková úložiště
- ▶ Software (RPM)
- ▶ Systémové služby
- ▶ Hardware
- ▶ Uživatelské účty
- ▶ Souborový systém
- ▶ atd.

# WBEM a CIM

- ▶ Organizace DMTF (Distributed Management Task Force)
- ▶ ~200 organizací (AMD, Broadcom, Cisco, Citrix, Fujitsu, HP, Huawei, IBM, Intel, Microsoft, Oracle, Red Hat, VMware, atd.)



## WBEM (Web Based Enterprise Management)

- ▶ Protokoly (CIM-XML, WS-Management)
- ▶ Standardní operace
- ▶ Model = CIM
- ▶ Dotazovací jazyk

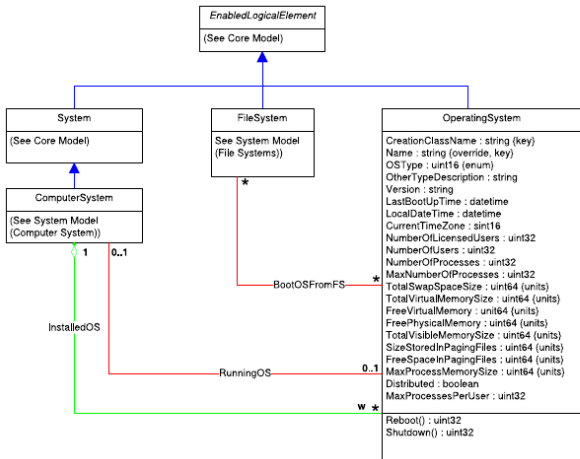
## CIM (Common Information Model)

- ▶ Specifikace infrastruktury
  - ▶ Jazyk pro CIM schéma (metamodel)
  - ▶ Mapování na ostatní modely (např. SNMP)
- ▶ Schéma
  - ▶ Popis objektového modelu

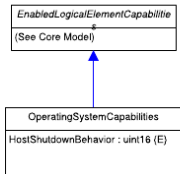
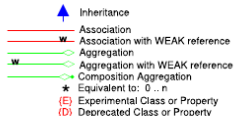
## Objektový model

- ▶ Třídy: vlastnosti, metody, kvalifikátory
- ▶ Metody - vnitřní a vnější
- ▶ Dědičnost
- ▶ Asociace - relace mezi třídami
- ▶ Indikace - výskyt události

# Příklad



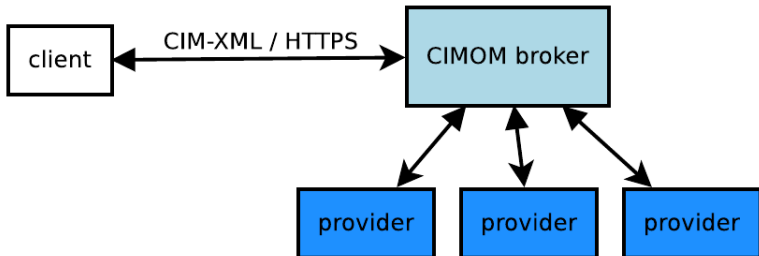
## Operating system



## Implementace

### ▶ CIMOM

- ▶ Síťový server – HTTP(s), CIM-XML
- ▶ Komunikace s providery
- ▶ Běží na spravovaném počítači



## Implementace

- ▶ Providery
  - ▶ Sdílené knihovny (DSO), načítány CIMOMem
  - ▶ CMPI = Rozhraní mezi CIMOMem a Providerem (pro jazyk C)
  - ▶ Bindings pro ostatní jazyky (Python, Java, Ruby, ...)

## CIM Operace

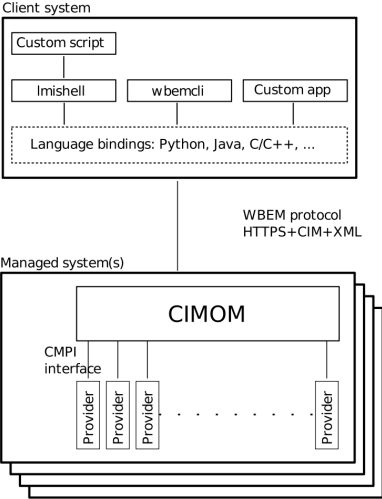
- ▶ Vnitřní metody (Intrinsic)
  - ▶ EnumerateClasses, EnumerateInstances
  - ▶ EnumerateClassNames, EnumerateInstanceNames
  - ▶ GetClass, GetInstance
  - ▶ CreateClass, CreateInstance
  - ▶ ModifyClass, ModifyInstance
  - ▶ DeleteClass, DeleteInstance
  - ▶ Associators, AssociatorNames
  - ▶ References, ReferenceNames
  - ▶ GetProperty, SetProperty
  - ▶ ExecQuery, ...
- ▶ Vnější metody (Extrinsic)
  - ▶ libovolné další, definovány ve schématu

## Profily

- ▶ Unifikovaný popis funkční oblasti
- ▶ Profil je popis modelu a jeho chování (třídy, vlastnosti, metody a hodnoty)
- ▶ Obsahuje případy užití (use cases)
- ▶ <http://dmtf.org/standards/profiles>



# OpenLMI – implementace

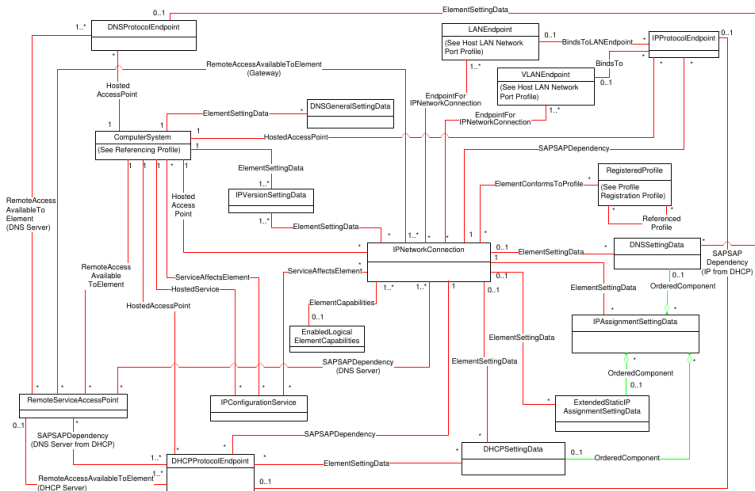


# Správa sítí s OpenLMI

- ▶ implementace profilu DSP1116 – IP Configuration Profile a částečně DSP1035 – Host LAN Network Port Profile
- ▶ používá NetworkManager jako backend
- ▶ “tenká” vrstva pro překlad mezi API NetworkManageru a CIMem

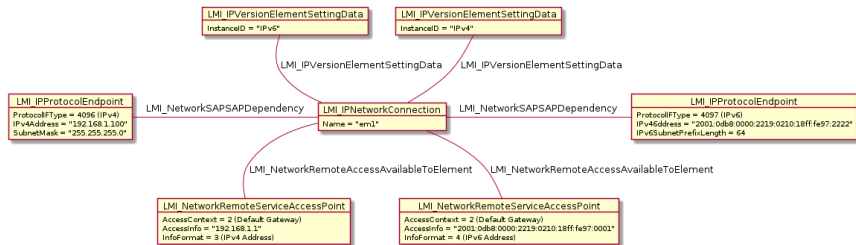
# Profil DSP1116

## ► Diagram tříd:



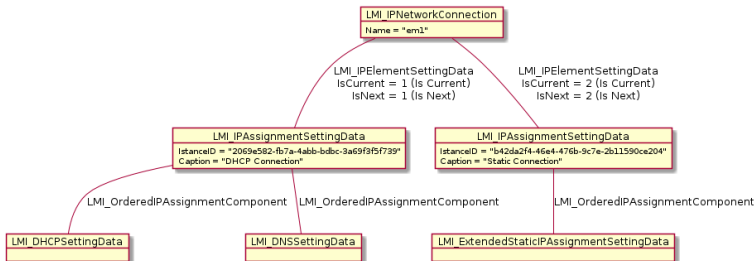
## Základní třídy

- ▶ CIM\_IPNetworkConnection – síťové rozhraní
- ▶ CIM\_IPProtocolEndpoint – koncový bod (= IP adresa)
- ▶ CIM\_RemoteServiceAccessPoint – funkce poskytnutá jiným systémem (Gateway, DNS server, DHCP server, ...)
- ▶ Diagram tříd:



# Síťová připojení

- ▶ odpovídají připojením v NetworkManageru
- ▶ CIM\_IPAssignmentSettingData – nadtřída pro jednotlivá nastavení
  - ▶ CIM\_DHCPSettingData
  - ▶ CIM\_ExtendedStaticIPAssignmentSettingData
  - ▶ CIM\_DNSSettingData



## Aktivace připojení

- ▶ Singleton `CIM_IPConfigurationService`
- ▶ Metoda `ApplySettingToIPNetworkConnection`
  - ▶ aplikuje `CIM_IPAssignmentSettingData` na `CIM_IPNetworkConnection`
  - ▶ parametr `Mode` – enumerace určující hodnoty `IsCurrent` a `IsNext`

## Vytváření připojení

- ▶ Není definováno profilem!
- ▶ Metoda `LMI_CreateIPSetting` třídy `CIM_IPNetworkConnectionCapabilities`
- ▶ Parametry:
  - ▶ název
  - ▶ metoda pro IPv4 (staticky, DHCP, vypnuto)
  - ▶ metoda pro IPv6 (staticky, DHCPv6, Stateless, vypnuto)
  - ▶ typ (Ethernet, Bridge, Bond, VPN, atd.)
  - ▶ vrací instanci `CIM_IPAssignmentSettingData`
- ▶ Nutné zavolat `ModifyInstance` na vytvořenou instanci

## Další možnosti

- ▶ Nastavení routování
- ▶ Nastavení DNS serverů
- ▶ Statistiky (množství přenesených dat, chyby)



## Příklady použití

- ▶ Zjištění MAC address pro síťové porty

```
for ethernetPort in c.root.cimv2.LMI_EthernetPort.instances():  
    print ethernetPort.DeviceID, ethernetPort.PermanentAddress
```

- ▶ Výpis nastavených IP adres

```
port = 'eth1'  
ipnetworkconnection = c.root.cimv2.LMI_IPNetworkConnection.\  
    first_instance(Key='Name', Value=port)  
for assoc in ipnetworkconnection.associators():  
    if assoc.classname == "LMI_IPProtocolEndpoint":  
        if assoc.ProtocolIFType in [4096, 4098]:  
            # IPv4  
            print "%s/%s" % (assoc.IPv4Address, assoc.SubnetMask)  
        if assoc.ProtocolIFType in [4097, 4098]:  
            # IPv6  
            print "%s/%d" % (assoc.IPv6Address, assoc.PrefixLength)
```

## Plány do budoucna

- ▶ Indikace
- ▶ Asynchronní aktivace připojení
- ▶ Enterprise nasazení
  - ▶ Bridging, Bonding, VLAN
  - ▶ Záleží na NetworkManageru
- ▶ Wifi a VPN není zatím v plánu

# Závěr

Děkuji za pozornost