



IBM Corporation

# Integration of Nagios monitoring tools with IBM's solutions

Wojciech Kocjan  
IBM Corporation  
[wojciech.kocjan@pl.ibm.com](mailto:wojciech.kocjan@pl.ibm.com)

# Agenda

- Introduction
  - Integration bottlenecks
  - Why open standards?
  
- CIM and WBEM
  - Open standards for systems management
  - Using CIM and WBEM from Nagios
  
- CMDBf and COSMOS
  - What is federated systems management?
  - Nagios in the federated approach



IBM Corporation

# Introduction

# Monitoring in complex environments

- Various hardware and software
  - Different operating systems
  - Multiple virtualization types (LPAR, VMware, KVM, Xen, domains)
  - Managing devices (routers, printers, mobile devices)
  
- Multiple monitoring systems
  - Keeping list of assets up to date
  - Integrating event flow across systems
  - Managing downtimes, maintenance windows and SLAs
  
- Networking
  - Setting up dependencies between assets
  - Firewall traversal

# Typical issues without standards and integration

- Multiple platform types and devices
  - Dedicated handling for each type of platform/device
  - Platform specific configuration of systems and/or devices
  
- Monitoring remote systems
  - Deploying dedicated agents for each product
  - Issues with firewalls and private subnets
  - Different methods of authenticating connections and applications
  
- Dynamic IT environments
  - Frequent reinstalls and/or migrations of machines
  - Virtualization makes it easy to set up machines for a few days or weeks
  - Keeping multiple configuration systems up to date is a challenge

## Integration with IBM products – example

- Infrastructure consists of IBM and customer owned systems
  - Customer using 100 servers owned and managed by IBM (IBM AIX®, Microsoft Windows®, Linux®)
  - Nagios used for monitoring customer's part of the infrastructure
  - IBM Tivoli Enterprise Console ® used for monitoring IBM managed servers
  - Multiple services running: IBM DB2®, Oracle, IBM WebSphere®, file server, print server etc
  
- Integration of Tivoli Enterprise Console® with Nagios
  - Events from Tivoli Monitoring passed to Nagios
  - Configuration of hosts and services set up for Nagios
  - Active checks disabled on Nagios for IBM owned systems

# Standards to aid integration

- **CIM and WBEM**
  - Industry standard for retrieving information from local and remote systems
  - Standard protocols and methods to access information
  - Generic model along with vendor specific extensions
  
- **CMDBf**
  - Integration point for IT management
  - Integration point for other system monitoring and IT management applications
  
- **COSMOS**
  - Open source projects for system management tools
  - Uses CMDBf for federating information
  - Integration with various system monitoring software, including Nagios

## Plans for the future

- Move away from dedicated, product specific agents
  - Reuse interfaces provided by platform
  - Move to agentless deployments if possible
  - Integrated agent approach when possible
  - Decreasing number of agents reduces maintenance effort
  
- Integration across products and technologies
  - Focus on point of integration – CMDBf
  - Federation of models for interchange between management applications
  - Allow integration with third party software



IBM Corporation

# CIM and WBEM

# The need for common standards

- Unify information – CIM
  - Provides mechanisms for describing data model
  - Defines how information is described and stored
  - CIM Specification – defines syntax and rules
  - CIM Schema – defines base classes, properties and methods
  
- Unify access – WBEM
  - Standardization of requesting and receiving information
  - Methods for retrieving CIM-based information
  - Support for query languages (CQL, WQL); query languages not part of standard
  - Support for multiple protocols (http/https, RPC) and discovery (DNS, NetBIOS)

# Benefits of using CIM and WBEM

- **Unification of data management**
  - Same or similar CIM queries for different platforms
  - Same mechanisms for managing majority of endpoints
  
- **No additional software deployed to target machine to monitor it**
  - Using information already provided by operating system and/or software on system
  - Large number of software and agents already support CIM models
  
- **Multiple platform support**
  - Offers platform-specific extensions to generic CIM schema
  - Allows querying any machine supporting specified protocol; usually over HTTP

# What is Common Information Model?

- CIM is a hierarchical, object-oriented model for defining class hierarchy and relations
- CIM is only an information model; it does not require particular instrumentation or persistent information repository format
- CIM defines language for defining meta schema, Managed Object Format (MOF)
- CIM does not define any APIs, methods for accessing or interfaces across providers; CIM is only the model for representing information

## CIM – how data is defined

### **CIM Specification**

- Allows describing CIM schemas
- Object-oriented meta model based on UML
- Allows definitions of classes, properties, methods and associations between objects

### **CIM Schema**

- Defines standard set of classes and their relations
- Combination of Core and Common Models
- Extension schema allows specific vendors to extend common models

## CIM and standardizing data

- CIM Schema defines standard classes that describe various types of information
- Defines relations and associations between classes to describe how parts of the system are related
- Covers majority of aspects regarding hardware and software
- Specific CIM implementations offer a subset of CIM standards

# Scope of CIM Schema

- Network – systems, services, protocols, technologies
- Physical assets – enclosures, physical components, association, cabling
- Devices – computer systems, storage, memory, power supply etc
- Applications – software products, features, elements
- Databases – database system, database services, additions
- Systems – collections, combine other parts of CIM data model
- Policies – service-level policies, security policies, conditions

# What is Web-Based Enterprise Management?

- WBEM provides an interoperable and extensible means of managing CIM information
- Set of technologies to unify management of distributed computing environments
- Provides ability to exchange CIM information in an interoperable and efficient manner
- Includes protocols, discovery mechanisms, mappings and everything needed to exchange CIM information
- Powerful end-to-end management solution when combined with CIM

# CIM and WBEM in system monitoring

- Local and remote retrieval of information
  - Allows querying any machine supporting specified protocol
  - Standardized, operating system-independent way of querying data
  
- Does not require dedicated agent
  - Many vendors support CIM and WBEM standards
  - Use existing software that exports its data as CIM
  
- Platform independence
  - Adding support for new platforms much easier
  - Small set of protocols that need to be supported by the monitoring system

## Monitoring Microsoft Windows®

- Windows Management Infrastructure uses CIM+WBEM model
- Allows local and remote querying of attributes
- Uses COM for local and/or DCOM for remote access to repository
- Can be queried from UNIX systems (using WMI client) and Java (using J-Integra)

# WBEM over HTTP

- Supports authentication and encryption
  - Allows authenticating connections using username and password
  - Connections can use SSL, which can also be used for authentication
  
- Supported by major hardware and/or software vendors
  - IBM
  - Sun
  - HP
  
- Multiple open source projects offer client libraries and/or CIM servers
  - SBLIM – server and client
  - OpenWBEM
  - OpenPegasus

# Using wbemcli for getting CIM information

- Part of SBLIM project
  - Command line client for CIM
  - Native binary, no dependencies on additional software
- Using wbemcli to retrieve information over http:
  - Enumerating all available classes, passing username and password  
`wbemcli ecn http://username:password@machine:5988/root/cimv2`
  - Enumerating instances of a class, without authentication  
`wbemcli ei http://machine:5988/root/cimv2:CIM_Process`

# Using wmi-client for getting information from WMI

- Using wmi-client
  - Command line utility to retrieve WMI data over RPC/DCOM
  - Code base extracted from Samba4 code; additions to handle WMI
  - Support for systems from Windows® 2000 up to Vista/Windows® Server 2008
  
- Using wmic command to retrieve information over DCOM:
  - Querying operating system information

```
wmic -U username%password //win2003 \  
"SELECT * FROM win32_ComputerSystem"
```

# Modules for Python

- Pywbem – communication over HTTP protocol
  - Querying CIM information over WBEM on top of HTTP
  - Supports non-SSL and SSL enabled connections
  - Allows enumerating instances and executing queries
  
- Python-wmi
  - Communication with WMI over RPC/DCOM
  - Uses DNS and/or NetBIOS locator services for finding hosts
  - Allows enumerating instances and executing queries

## Sample code – Nagios plug-in using pywbem

- Checking if a field or number of rows returned from a specified query matches predefined range

```
import pywbem
[hostname, query, field, vfrom, vto] = sys.argv[1:]
vfrom = int(vfrom)
vto = int(vto)
url = 'http://%s:5988' % hostname
conn = pywbem.WBEMConnection(url, ('', ''), default_namespace='root/cimv2')
items = conn.ExecQuery('CQL', query)
if field == 'COUNT':
    item = len(items)
else:
    if len(items) == 0:
        print "CQL: No rows returned"
        sys.exit(3)
    item = int(items[0][field])
if item < vfrom:
    print "CQL: %d is lower than minimum value" % item
    sys.exit(2)
if item > vto:
    print "CQL: %d is higher than maximum value" % item
    sys.exit(2)
print "CQL: Current value is %d" % item
sys.exit(0)
```

## Sample code – Nagios plug-in using python-wmi

- Checking if a field or number of rows returned from a specified query matches predefined range

```
import win32com.client
[hostname, username, password, query, field, vfrom, vto] = sys.argv[1:]
vfrom = int(vfrom)
vto = int(vto)
locator = win32com.client.SWbemLocator();
conn = locator.ConnectServer(name=hostname, namespace="root/cimv2",
                             user=username, passwd=password)
items = conn.ExecQuery(query)
if field == 'COUNT':
    item = len(items)
else:
    if len(items) == 0:
        print "WMI: No rows returned"
        sys.exit(3)
    item = int(items[0][field])
if item < vfrom:
    print "WMI: %d is lower than minimum value" % item
    sys.exit(2)
if item > vto:
    print "WMI: %d is higher than maximum value" % item
    sys.exit(2)
print "WMI: Current value is %d" % item
sys.exit(0)
```



IBM Corporation

# CMDBf and COSMOS

# CMDB Federation

- Federated approach towards Configuration Management Database
  - Standard for querying, registration and providing information about infrastructure
  - Multiple sources of information – products can provide partial or full information
  - Long-term approach of keeping and integrating information about system
  - Avoiding replicating data between management repositories
  
- Standard defined by key vendors in the IT industry
  - IBM
  - CA
  - BMC
  - Fujitsu
  - HP
  - Microsoft

## Goals of CMDBf

- Point of integration between IT management processes
  - Keep information about configuration items, relations and other types of records
  - Handle configuration, change, availability, incident and capacity management
  - Allow IT management software to provide information to CMDBf
  - Keep track of which products keep track certain information
  - Unified, vendor independant interface for integrating multiple products
  
- Provide aggregated information to products
  - Handle queries from IT management software and/or other software
  - Pass requests to applications that provide specified type of information
  - Aggregate results and pass them back to applications
  
- Products providing information can also consume them

# Future of CMDBf

- CMDBf as core part of management products
  - IT management software transforms towards federated system based on standards
  - Multiple, domain specific applications can provide information about infrastructure
  - Applications using CMDBf use data provided by other applications
  
- Long term solution for integration of applications
  - IT management applications will integrate with CMDBf
  - Remove redundancy in configuration and managing of asset information

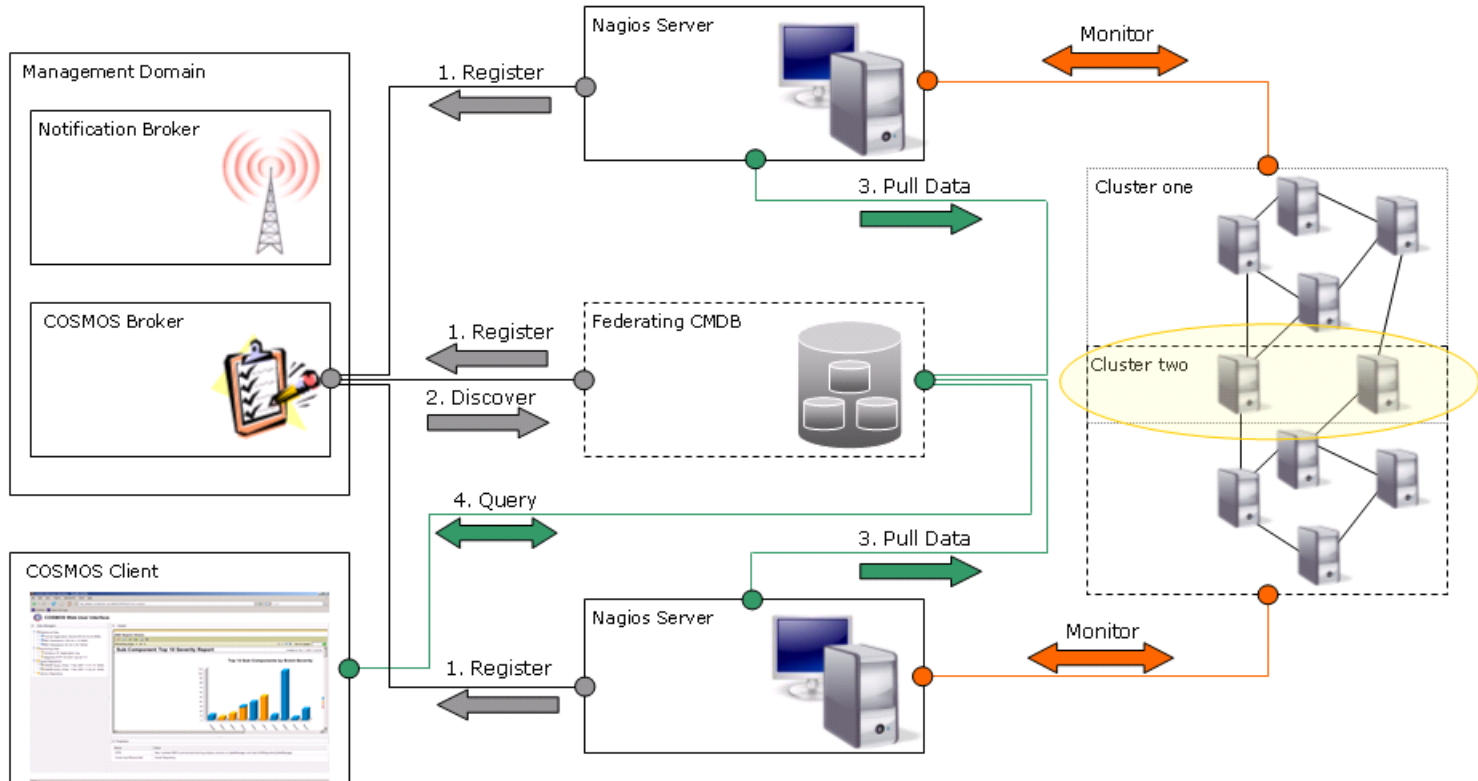
# What is COSMOS?

- Extensible, standards-based projects for developing system management applications
  - Ease development of system management applications
  - Set of management capabilities to improve integration
  - Integration of existing system management applications with CMDBf
  - Point of integration for other system monitoring software – i.e. Nagios
  
- CMDBf in COSMOS work
  - Retrieve information gathered by other software from CMDBf
  - COSMOS as one of providers of information in CMDBf
  - Integration of asset information with CMDBf

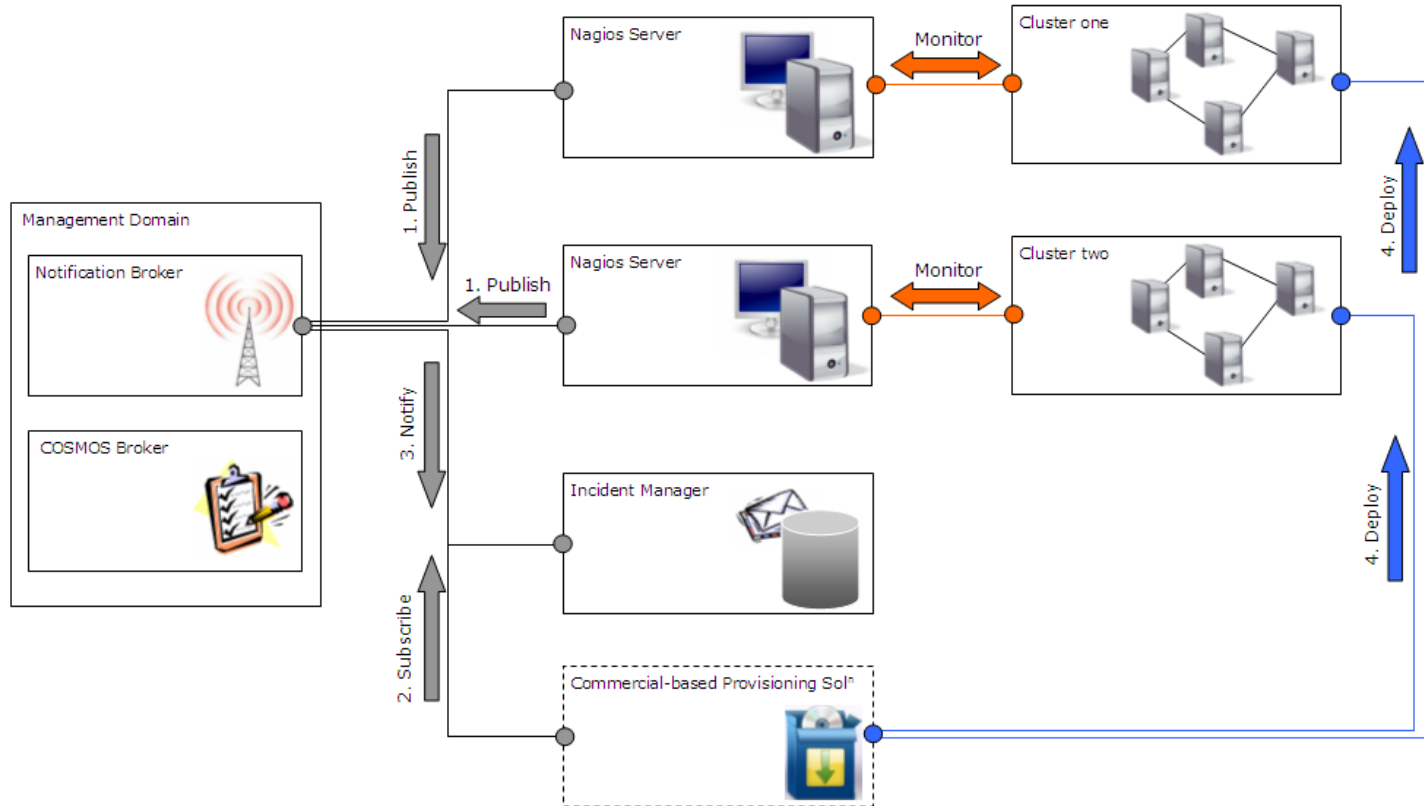
# Integration with Nagios

- Nagios servers register in COSMOS and in CMDBf as providers
  - COSMOS is aware of Nagios instances
  - CMDBf registers information about which hosts and services Nagios offers
- CMDBf pulls information from Nagios servers
  - CMDBf retrieves information from Nagios servers when a query is received
  - CMDBf offers an aggregated view from all Nagios instances
  - Machines monitored by multiple instances are correctly handled by CMDBf
- Nagios sends events directly to Notification broker
  - IT management software subscribes to specific events
  - Nagios events are published to notification broker
  - Broker notifies subscribed applications

# Integration with Nagios – information flow



# Integration with Nagios – notification flow





IBM Corporation

Questions?

# References

- IBM Tivoli Enterprise Console  
<http://www-01.ibm.com/software/tivoli/products/enterprise-console/>
- Distributed Management Task Force  
<http://www.dmtf.org/>
- CIM and WBEM standards  
<http://www.dmtf.org/standards/>
- Eclipse COSMOS  
<http://www.eclipse.org/cosmos/>

## References contd.

- WMI Samba based native client (wmic and winexe)  
<http://eol.ovh.org/winexe/>
- Java COM/WMI client – J-Integra  
<http://j-integra.intrinsyc.com/products/com/>
- Pywbem  
<http://pywbem.wiki.sourceforge.net/>
- Python-WMI  
<http://dev.zenoss.com/trac/browser/trunk/wmi>

## References contd.

- SBLIM  
<http://sblim.wiki.sourceforge.net/>
- OpenPegasus  
<http://www.openpegasus.org/>
- OpenWBEM  
<http://www.openwbem.org/>

# Trademarks

IBM, the IBM logo, ibm.com, Tivoli and Tivoli Enterprise Console are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml)

## Trademarks – contd.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Nagios is a trademark or registered trademark of Nagios Enterprises in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.



IBM Corporation

Thank you