

WMI Mapper: Accessing WMI data using CIM XML

Brian Harrison

Hewlett-Packard

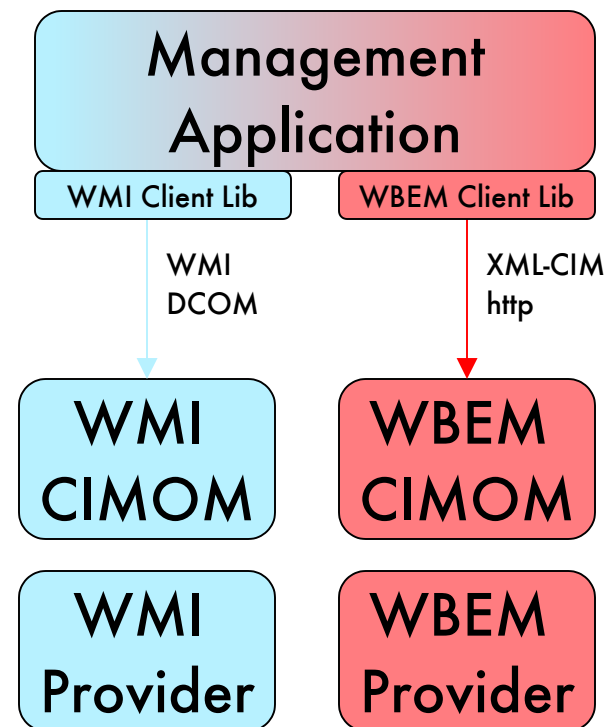
June 16-19 San Jose, California

Agenda

- The Problem
- Mapper Overview
- Remote connections & deployment
- Limitations & Issues
- Availability

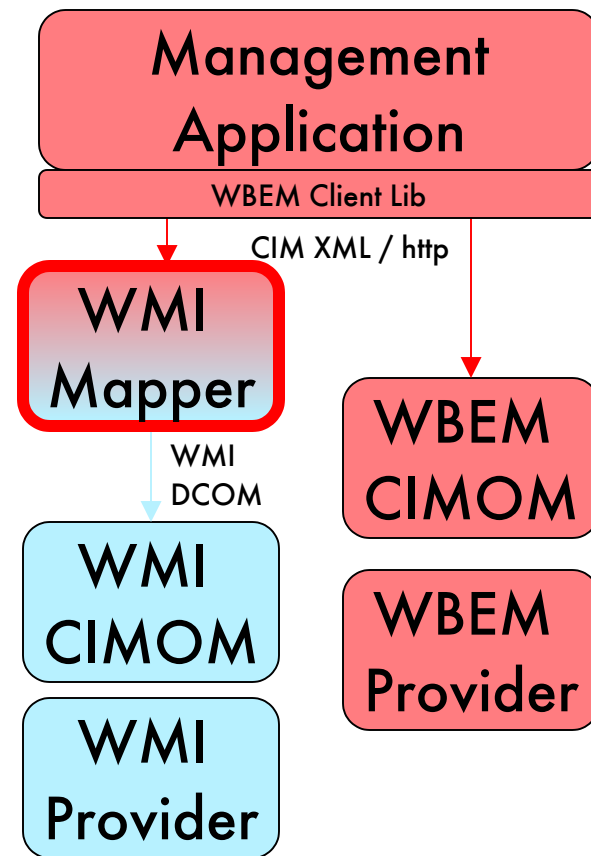
The Problem

- Microsoft Windows systems are well instrumented with CIM data through Windows Management Instrumentation (WMI) – *but*:
 - WMI uses a proprietary protocol
 - Windows OS needed to make WMI requests
 - Client applications are left with different models for Windows and other operating systems



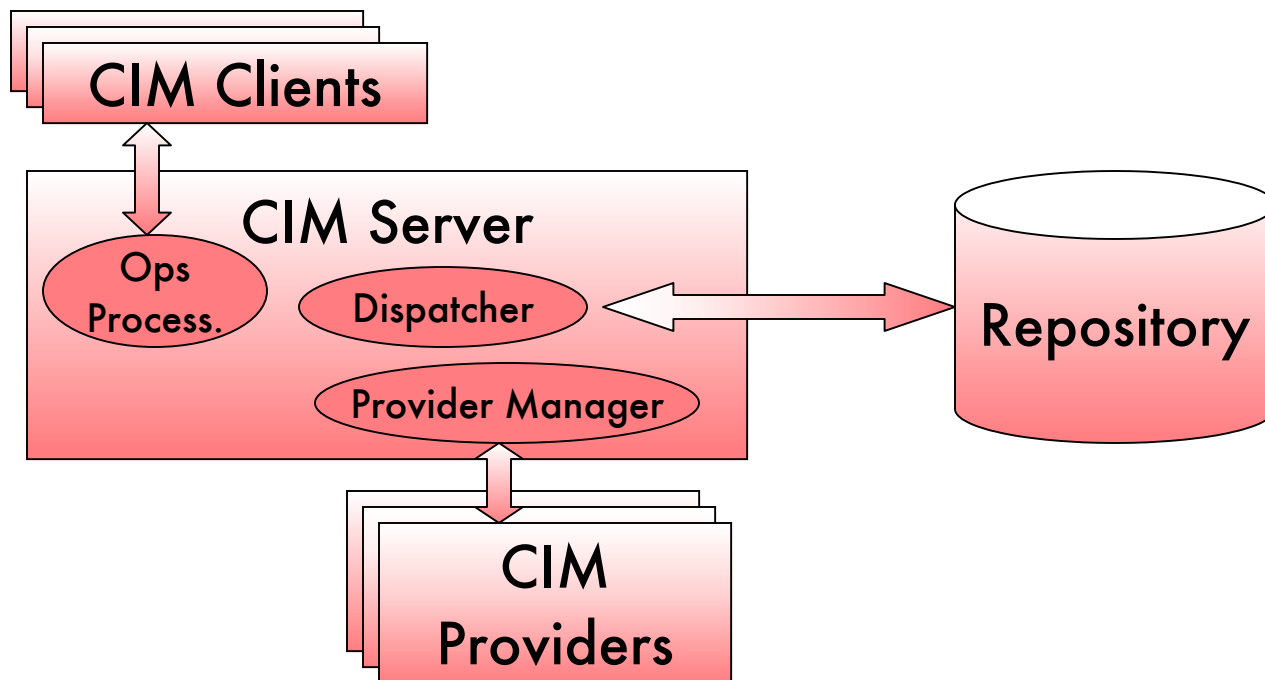
The Solution: WMI Mapper

- Write portable client applications using XML-CIM/http
- Requests to Windows systems are translated to WMI requests by the WMI mapper
- WMI responses returned as XML-CIM



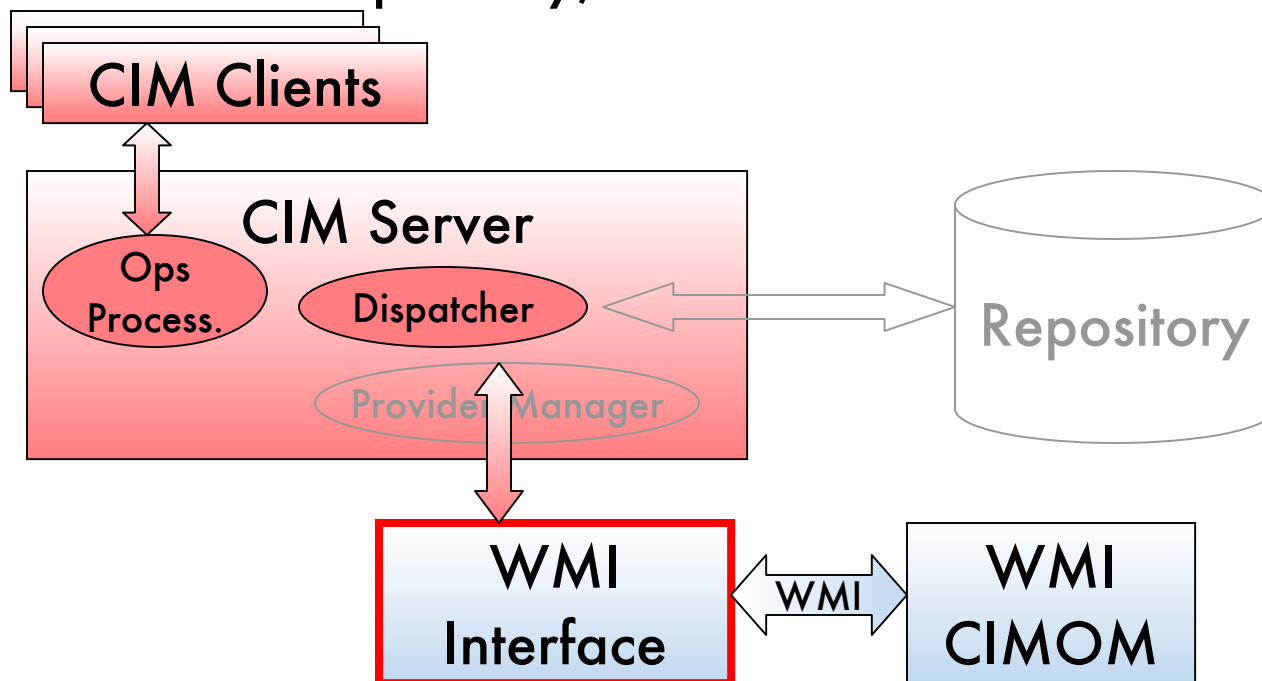
Pegasus

- Mapper based on the Pegasus CIMOM:



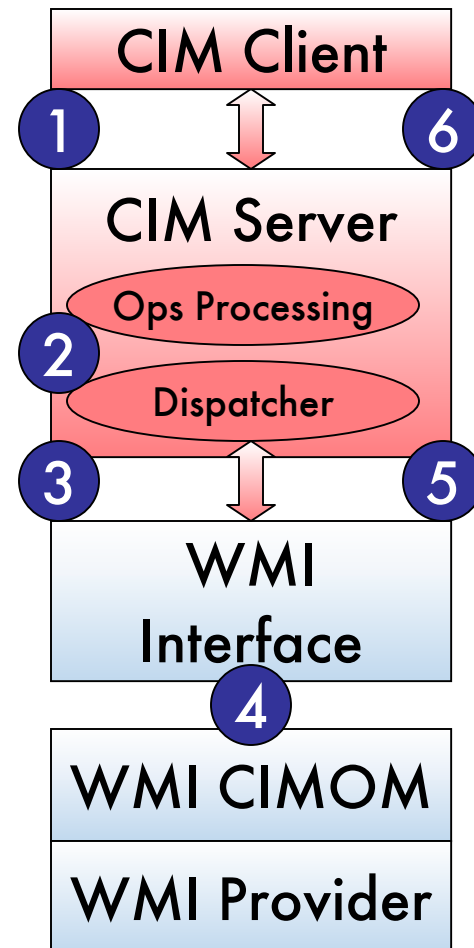
Pegasus: Changes for Mapper

- Goal: Minimize & localize changes to Pegasus
- Dispatcher sends all requests to WMI interface
- No need for repository, WMI has all information



Example Flow

1. XML request sent from client
2. Server parses XML, passes request to dispatcher
3. Dispatcher passes request to WMI provider
4. WMI interface calls WMI
5. WMI interface returns result to dispatcher
6. XML response built and returned to client



CIM Operations

- Class Operations
 - Create, Delete, Get, Modify, Enumerate, Enumerate Names
- Instance Operations
 - Create, Delete, Get, Modify, Enumerate, Enumerate Names
- Properties
 - Get, Set
- Invoke Method
- Exec Query
- Associations
 - Associators, Associator Names, References, Reference Names
- Qualifiers
 - **No methods supported**
 - Specify in class or instance operation

Qualifiers

- WBEM qualifiers are defined in a namespace; WMI qualifiers are more freely defined – any entity can have any qualifier
- The mapper cannot implement the WBEM qualifier methods to get, set, delete or enumerate qualifiers – no equivalent in WMI
- Client applications may specify qualifiers in class & instance operations (create, get, enumerate, modify)

Error Handling

- Some WMI errors can be mapped directly to CIM errors

```
<ERROR CODE="2"
```

```
DESCRIPTION="CIM_ERR_ACCESS_DENIED:
```

```
Access to a CIM resource was not available to the client:
```

```
&quot;LogonUser&quot;"/>
```

- Errors that cannot be mapped are returned as CIM_ERR_FAILED along with description from WMI

```
<ERROR CODE="1"
```

```
DESCRIPTION="CIM_ERR_FAILED:
```

```
A general error occurred that is not covered by a more specific error code:
```

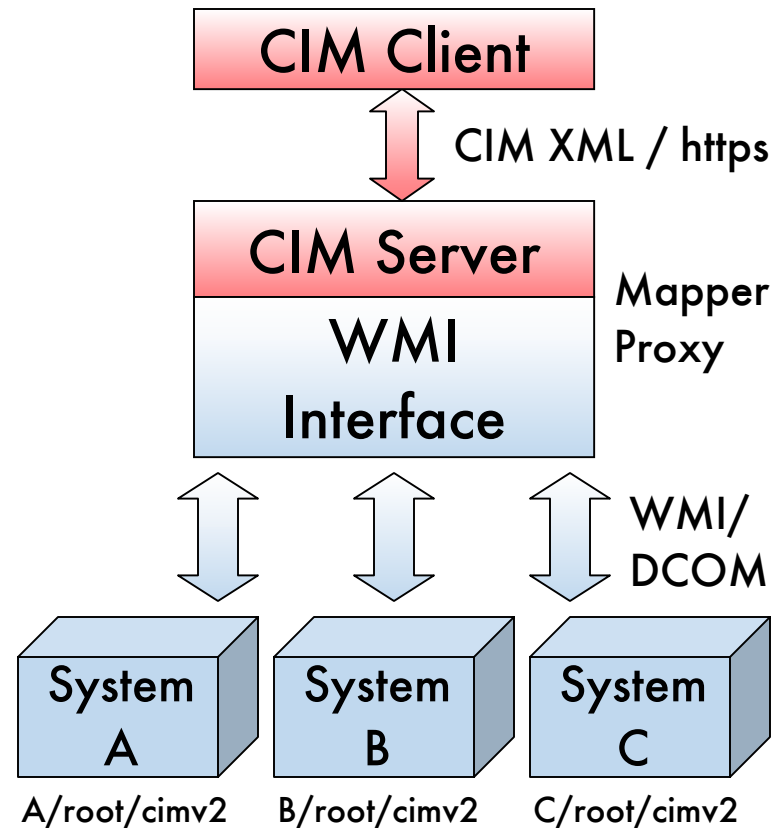
```
&quot;invalid CIM name: Win32_LocalTime=@&quot;"/>
```

Authentication & Authorization

- Mapper uses basic authentication over https to receive user name & password
- WMI interface then connects to WMI using these credentials
- User & password validated by WMI using Windows authentication. User name may include Windows domain.
- Authentication may take place locally on system with mapper, or remotely on the target system.
- All authorization performed by WMI – may return error to CIM client
- *Pegasus Enhancement Proposal to pass user credentials to dispatcher*

Remote Connections

- WMI Interface may connect to local system or to a remote Windows system (WMI over DCOM)
- Allows mapper to be installed in a 'proxy' mode: client requests funneled through mapper
- CIM XML request sent to proxy; use namespace to specify system name - e.g.:
 - Local: root/cimv2
 - Remote: system/root/cimv2
- Mapper requires privileges to access network
 - E.g. run as user account



Remote Namespace

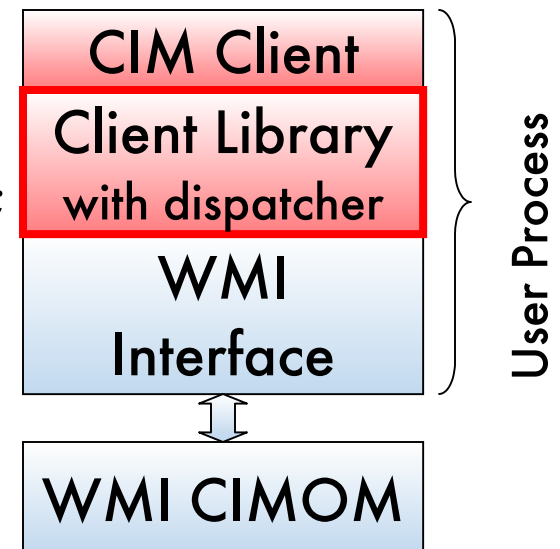
- No standard way to identify remote namespace
- Our approach is to use namespace to specify target:
 - Local namespace assumed to start with "root/"
 - E.g. "root/cimv2"
 - Assume characters before 'root' specifies system name
 - E.g. "mysystem/root/cimv2"
 - Domain name: Dots not valid in namespace; convert to '/'
 - E.g. *mysystem.hp.com*: "mysystem/hp/com/root/cimv2"
 - IP Address: Leading digits not valid; convert to '_'
 - E.g. *12.34.56.78*: "_12/_34/_56/_78/root/cimv2"

Deployment Models

- Mapper installed on target node
 - Supports CIM XML/https everywhere, providing consistent secure protocol on standard port
 - Transparent to clients - WMI system looks as if it supports WBEM natively
 - Additional installation required on each managed system
 - Adds some overhead to managed system in addition to WMI
 - One additional service process, <6MB memory footprint
- Proxy mode
 - Single mapper can support many WMI systems, given appropriate credentials
 - Requires WMI/DCOM protocol to target
 - No installation or overhead on target systems

Local Connect

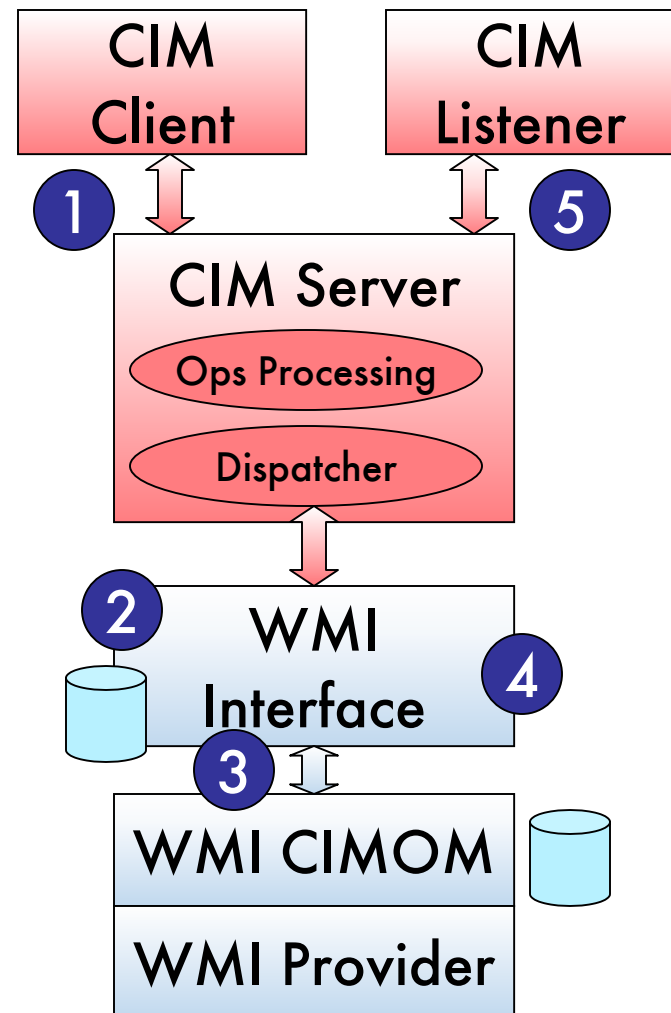
- Current Pegasus implementation supports 'local connect' without credentials – uses local user ID
 - Very useful for local management applications; avoids need to prompt user for password
- WMI also supports local connection without credentials; however if this is used by mapper the credentials of the mapper service would be used
- Solution: run mapper in client process
 - Special WMI version of CIM client library loads dispatcher directly
 - no XML encode/decode or HTTP
 - WMI provider makes local connection
- *Pegasus Enhancement Proposal to modify WBEM client library for local WMI connections*



Indications

Currently the mapper does not handle WMI indications

1. Indication subscription sent from client to CIM server
2. WMI Mapper needs to create subscription
3. Mapper then subscribes for WMI indication
4. When event occurs, WMI sends WMI indication to mapper (DCOM)
5. Mapper then forwards XML-CIM indication to listener (http)



Mapper Limitations

- Mapper must be installed on Windows system
- No support for Pegasus providers in mapper
 - All calls mapped to WMI, cannot mix with Pegasus
- No translation of CIM model differences between WMI & WBEM
- Non-standard workaround for remote namespace
- Proxy does not declare its capabilities
 - No discovery of mapper or its proxy capability
 - Can enumerate local namespaces (from root), or specific system
- Indications not currently supported
 - Model differences in indication subscription
 - No support for delivering indications

Client Issues

- Mapper allows client to transparently support heterogeneous systems – almost!
 - Client must supply credentials valid for WMI
 - Mapper-specific WBEM client library can remove this need
 - If proxy supported, client must send request to proxy and manipulate namespace to identify target
 - Client must avoid operations not supported by mapper – e.g. qualifier methods
 - Client must be aware of WMI CIM model
 - Some differences between WMI & Pegasus:
 - Singleton instances not supported in Pegasus
 - E.g. Win32_LocalTime=@
 - Some WMI instances do not have unique path

Availability

- Code developed by HP Brazil as modifications to the Pegasus open source CIMOM, from The Open Group
 - <http://www.openpegasus.org>
- Source code freely available under the Pegasus license
 - MIT open source license

Demo & Questions?

openCIMOM Browser

File Instances Class Namespaces Help

root/cimv2

- CIM_LogicalElement
 - Win32_PageFileUsage
 - Win32_Registry
 - + CIM_SoftwareFeature
 - + CIM_Process
 - + CIM_RedundancyGroup
 - Win32_QuickFixEngine
 - + CIM_SystemResource
 - Win32_NetworkClient
 - + CIM_SoftwareElement
 - + Win32_Session
 - Win32_ServerConnection
 - Win32_ServerSession
 - Win32_ComponentClass
 - CIM_JobDestination
 - Win32_NetworkProtocol
 - CIM_System
 - Win32_NTDomain
 - CIM_ApplicationServer
 - CIM_ComputerSystem
 - CIM_UnitaryClass
 - Win32_ComputerSystem
 - + CIM_ServiceAccessPoint
 - Win32_IP4PersistedFilter
 - Win32_LoadOrderGroup
 - + CIM_Service
 - Win32_Share
 - + Win32_COMApplication
 - Win32_COMClass

Properties | Methods | Qualifiers

Key	Name	Type
	AdminPasswordStatus	uint16
	AutomaticResetBootOption	boolean
	AutomaticResetCapability	boolean
	BootOptionOnLimit	uint16
	BootOptionOnWatchDog	uint16
	BootROMSupported	boolean
	BootupState	string
	Caption	string
	ChassisBootupState	uint16
	CreationClassName	string
	CurrentTimeZone	sint16
	DaylightInEffect	boolean
	Description	string
	Domain	string
	DomainRole	uint16
	EnableDaylightSavingsTime	boolean
	FrontPanelResetStatus	uint16
	InfraredSupported	boolean
	InitialLoadInfo	string
	InstallDate	datetime
	KeyboardPasswordStatus	uint16
	LastLoadInfo	string
	Manufacturer	string
	Model	string
	Name	string
	NameFormat	string
	NetworkServerModeEnabled	boolean
KEY		

Instances

Properties | Methods | Related objects | Associations

Key	Type	Name	Value
	uint16	AdminPasswordStatus	3
	boolean	AutomaticResetBootOption	true
	boolean	AutomaticResetCapability	true
	uint16	BootOptionOnLimit	
	uint16	BootOptionOnWatchDog	
	boolean	BootROMSupported	true
	string	BootupState	Normal boot
	string	Caption	HARRISON
	uint16	ChassisBootupState	3
	string	CreationClassName	Win32_ComputerSystem
	sint16	CurrentTimeZone	-420
	boolean	DaylightInEffect	true
	string	Description	AT/AT COMPATIBLE
	string	Domain	OPENVIEW.CUP.HP.COM
	uint16	DomainRole	1
	boolean	EnableDaylightSavingsTime	true
	uint16	FrontPanelResetStatus	3
	boolean	InfraredSupported	false
	string	InitialLoadInfo	
	datetime	InstallDate	
	uint16	KeyboardPasswordStatus	3
	string	LastLoadInfo	
	string	Manufacturer	Hewlett-Packard
	string	Model	hp workstation
KEY	string	Name	HARRISON