Security Considerations

Denise Eckstein
Hewlett-Packard

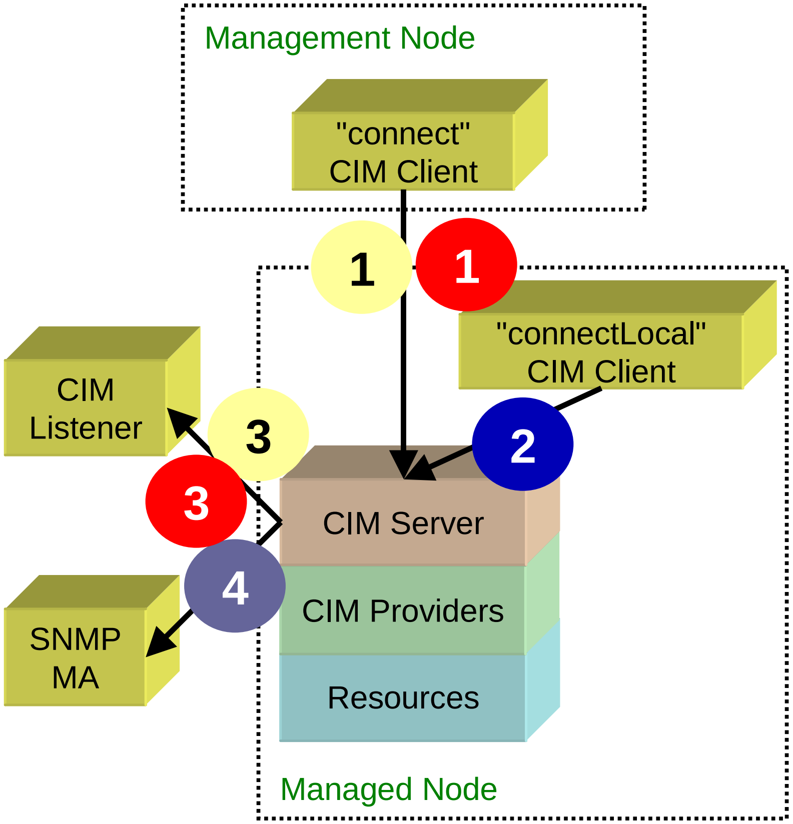
Module Content

HP WBEM Security

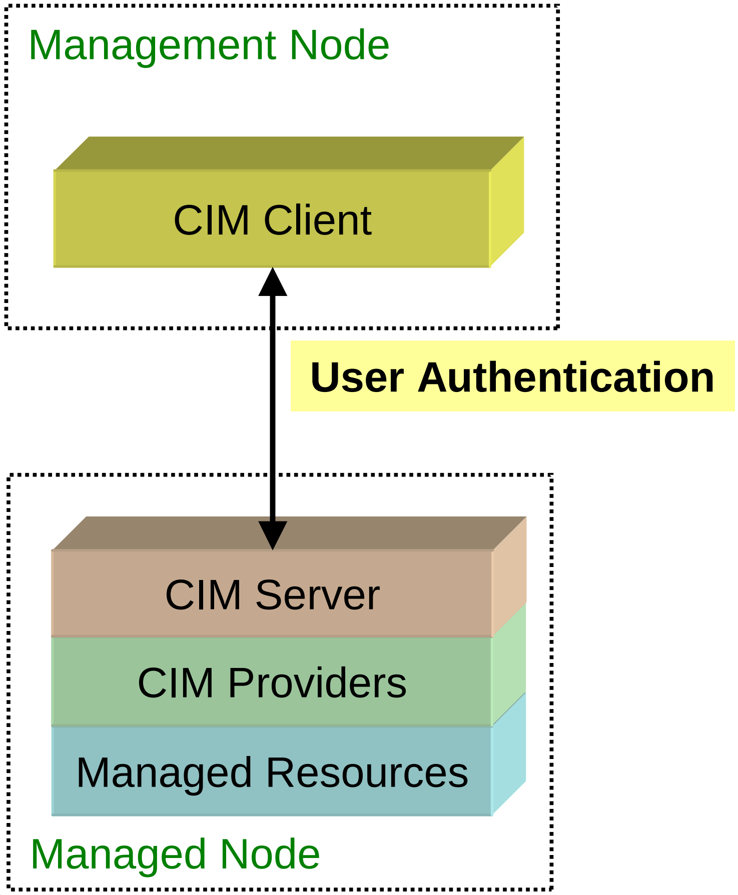
- **Overview**
- Secure Communication
- User Authentication
- User Authorization

Connection Points

ID	Requestor	Responder
1	"connect" CIM Client	CIM Server
2	"connectLocal" CIM Client	CIM Server
3	CIM Server	CIM Listener
4	CIM Server	SMNP MA

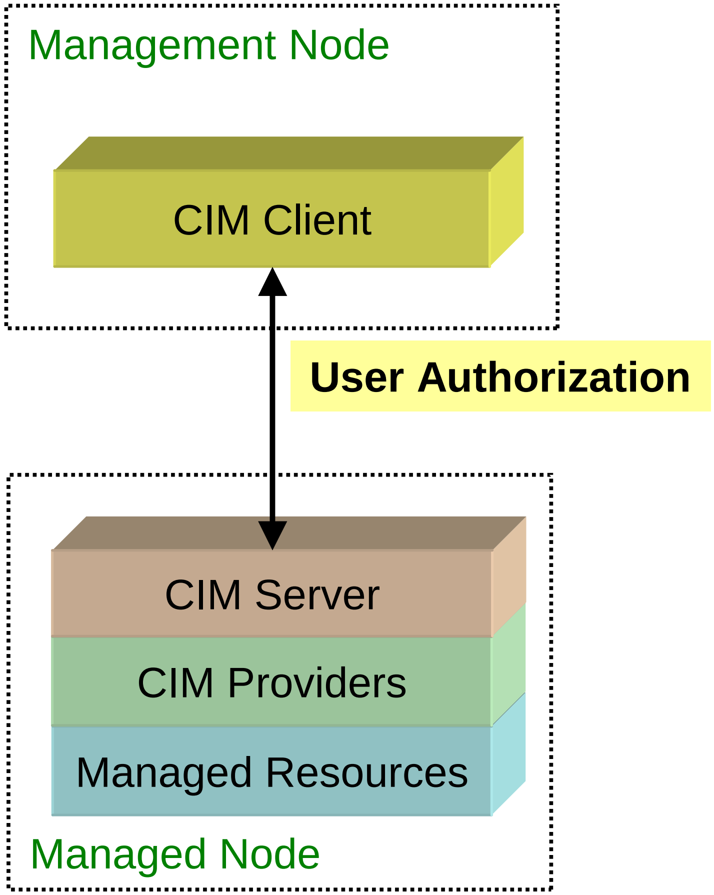


User Authentication



Authentication is the process of establishing the legitimacy of a user before allowing access to requested information.

User Authorization



Authorization is the process of granting permission to a user to perform an action that would be otherwise be prohibited by security policy.

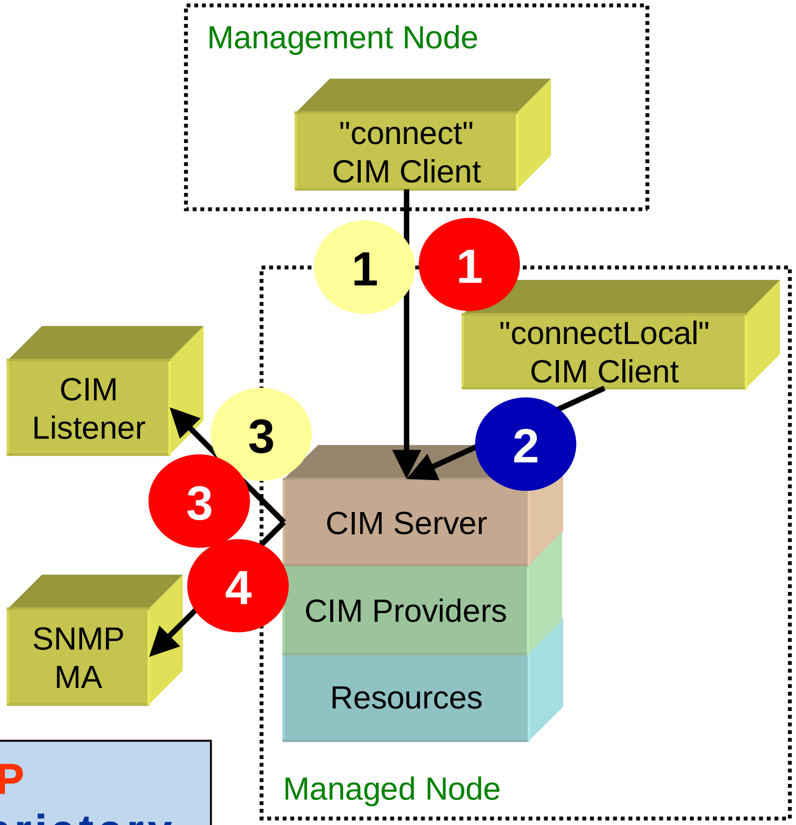
Module Content

HP WBEM Security

- Overview
- Secure Communication
  - **Connection Points**
  - SSL Overview
- User Authentication
- User Authorization

Connection Points

ID	Requestor	Responder
1	"connect" CIM Client	CIM Server
2	"connectLocal" CIM Client	CIM Server
3	CIM Server	CIM Listener
4	CIM Server	SMNP MA



HTTP
Proprietary
HTTPS

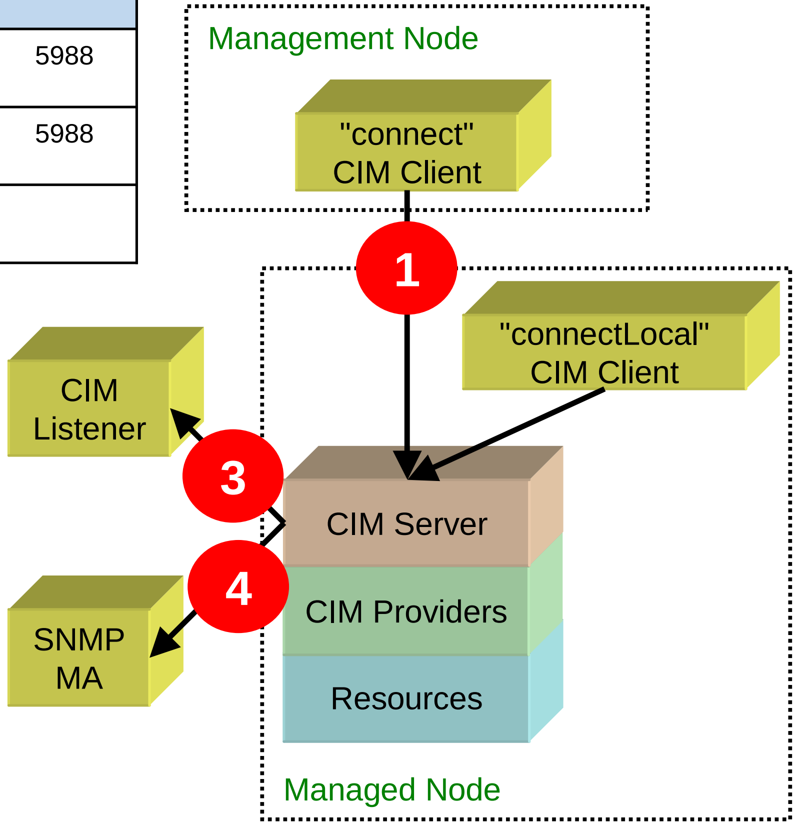
Non-Secure Connection Points

	Requester	Responder	Encoding	Protocol	Port
1	CIM Client	CIM Server	CIM-XML	HTTP over TCP/IP	5988
3	CIM Client	CIM Server	CIM-XML	HTTP over TCP/IP	5988
4	CIM Server	SNMP MA	MIB	SNMP Alert/Inform	

Warning: Use of these connection points is not recommended for confidential information in a high threat environment

These connection points will not be discussed in this module.

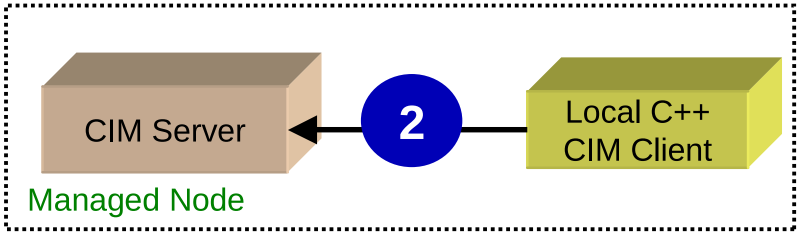
HTTP Connections



Proprietary Connection Points

Proprietary Connections

Requester	Responder	Encoding	Protocol	CIM Server Configuration Mechanisms
CIM Client	CIM Server	CIM-XML	Proprietary	Varies by platform. On HP-UX this option is not configurable. Always enabled.

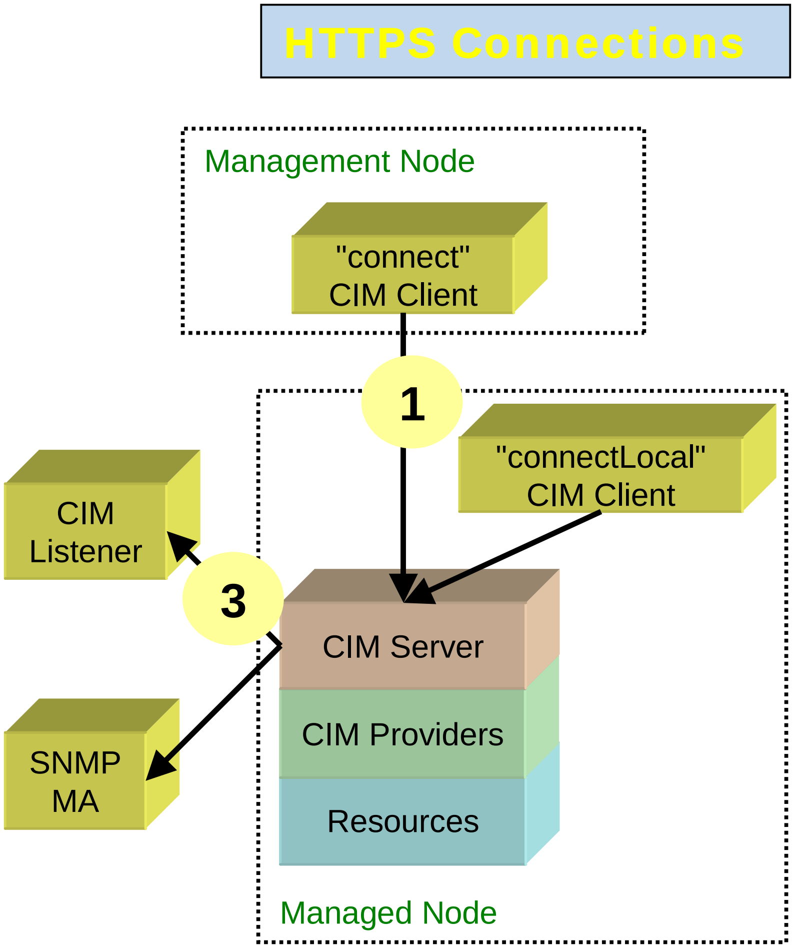


The **connectLocal()** Client API creates a connection to the server for local clients. The connection is automatically authenticated for the current user.

Note: The connectLocal interface is NOT STANDARD and only supported for use by C++ CIM Clients on certain platforms.

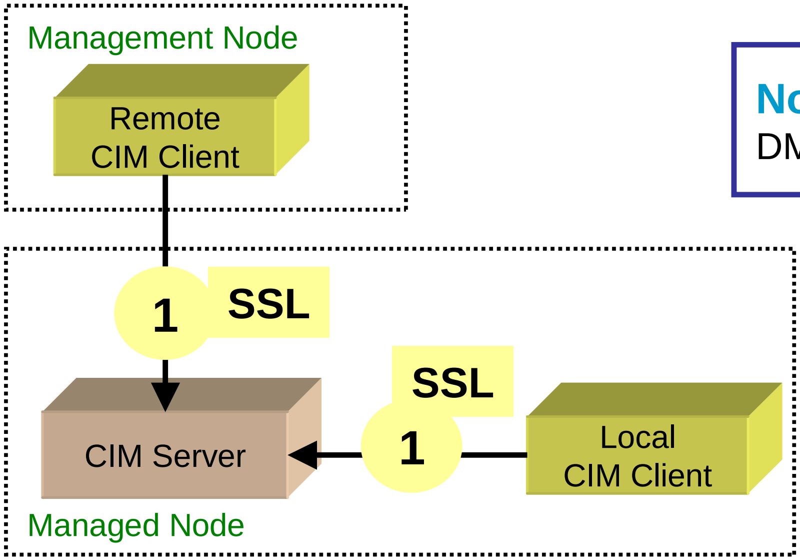
SSL Connection Points

ID	Requestor	Responder
1	"connect" CIM Client	CIM Server
2	"connectLocal" CIM Client	CIM Server
3	CIM Server	CIM Listener
4	CIM Server	SMNP MA



"connect" Connection

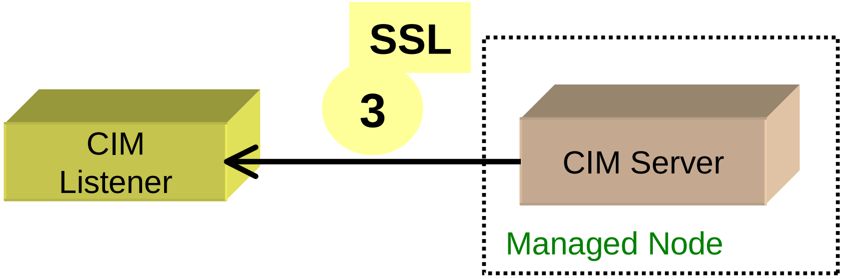
Requester	Responder	Encoding	Protocol	Port	CIM Server Configuration Parameter
CIM Client	CIM Server	CIM-XML	HTTPS over TCP/IP	5989	enableHttpsConnection Default = TRUE
CIM Client	CIM Server	CIM-XML	HTTP over TCP/IP	5988	enableHttpConnection Default = FALSE



Note: This interface implements the DMTF CIM-XML Standard.

CIM-XML Indication Delivery

Requester	Responder	Encoding	Protocol	Port	CIM Listener Configuration Mechanisms
CIM Server	CIM Listener	CIM-XML	HTTP over TCP/IP	Configurable	CIM_IndicationHandlerCIMXML
CIM Server	CIM Listener	CIM-XML	HTTPS over TCP/IP	Configurable	CIM_IndicationHandlerCIMXML



Module Content

HP WBEM Security

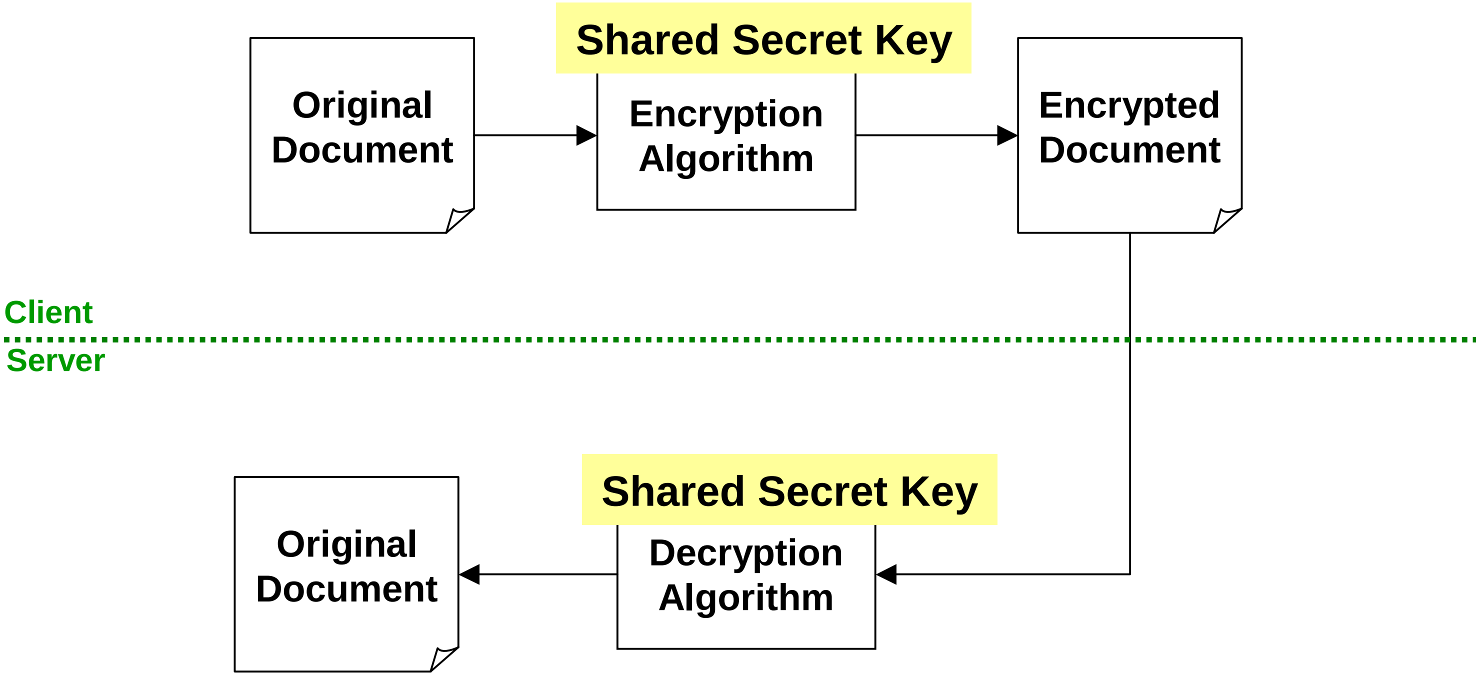
- Overview
- Secure Communication
  - Connection Points
  - **SSL Overview**
- User Authentication
- User Authorization

Cryptography

- Supports …
  - Authentication
  - Integrity
  - Confidentiality
  - Non-repudiation
- Mitigates …
  - Eavesdropping
  - Tampering
  - Spoofing
  - Connection Hijacking
  - Capture/Replay

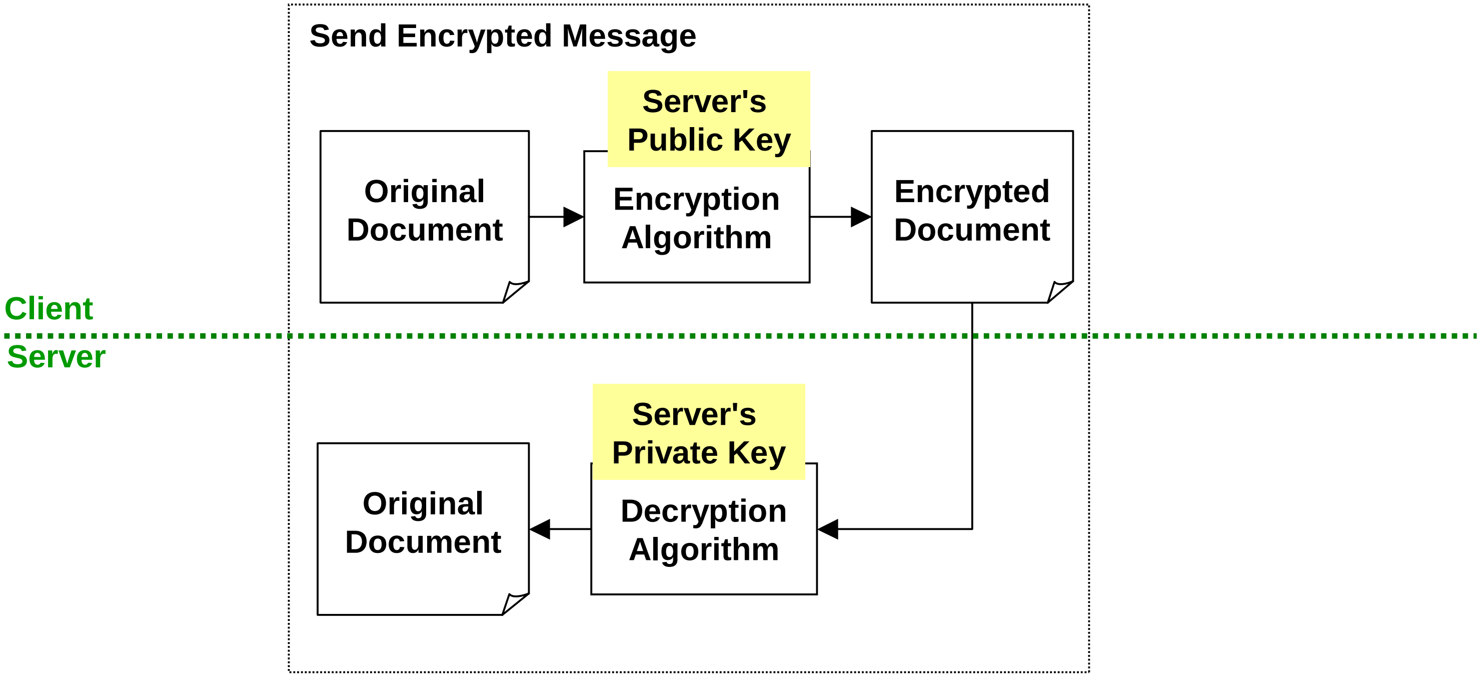
Symmetric Key Encryption

Benefit: Allows private data to be sent across an insure medium.
Issue: Key Distribution



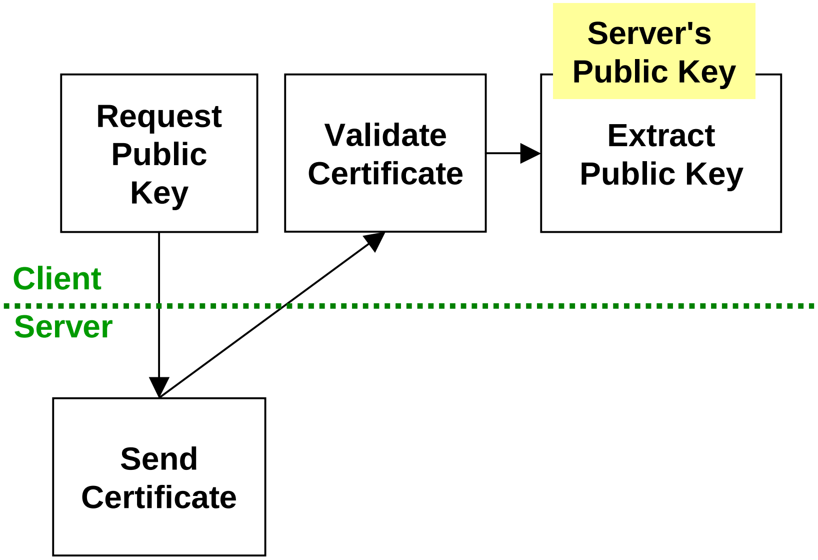
Public Key Encryption

Benefit: Solves key distribution problem.
Issue 1: Need to ensure validity of "Public Key"
Issue 2: Performance

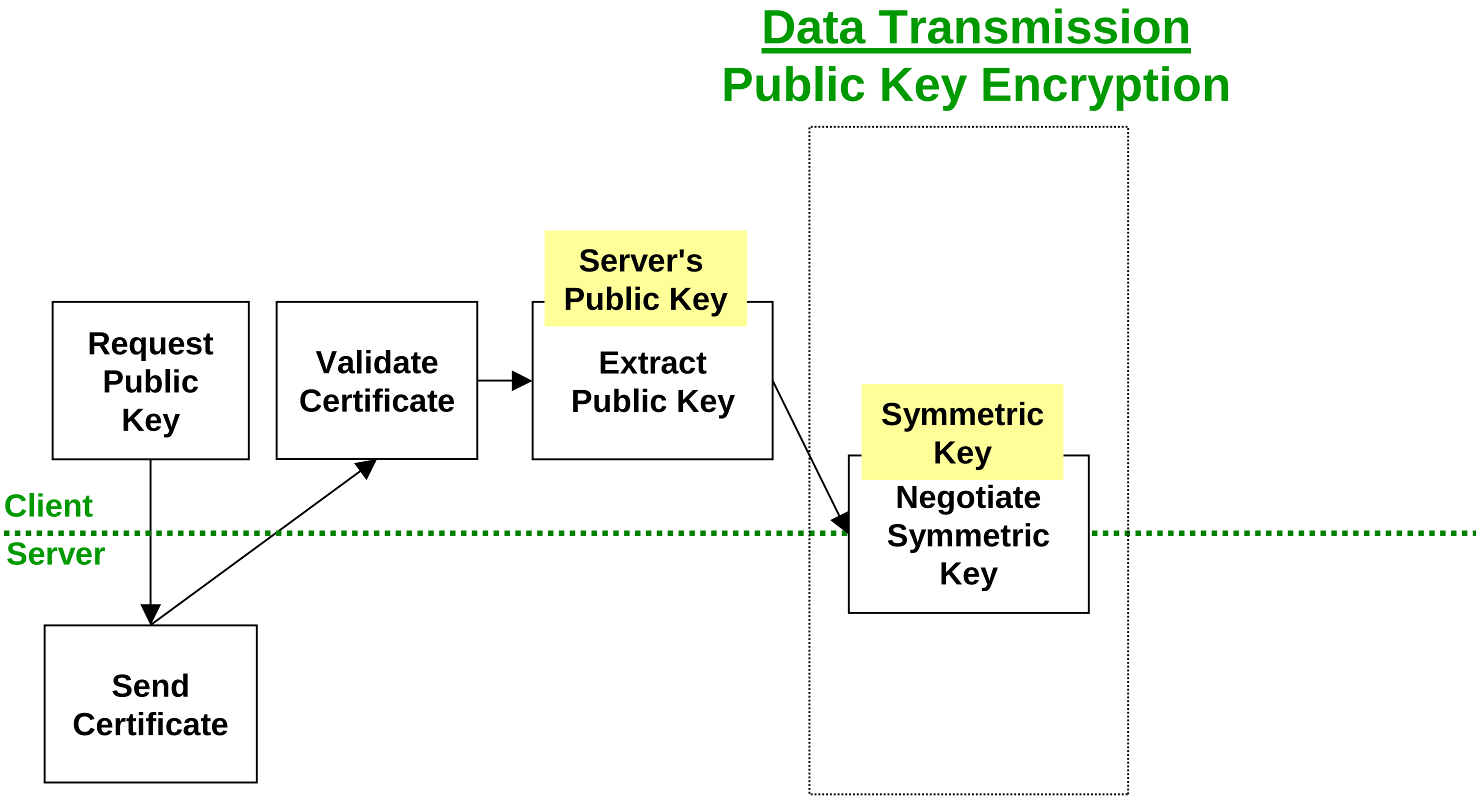


Certificates

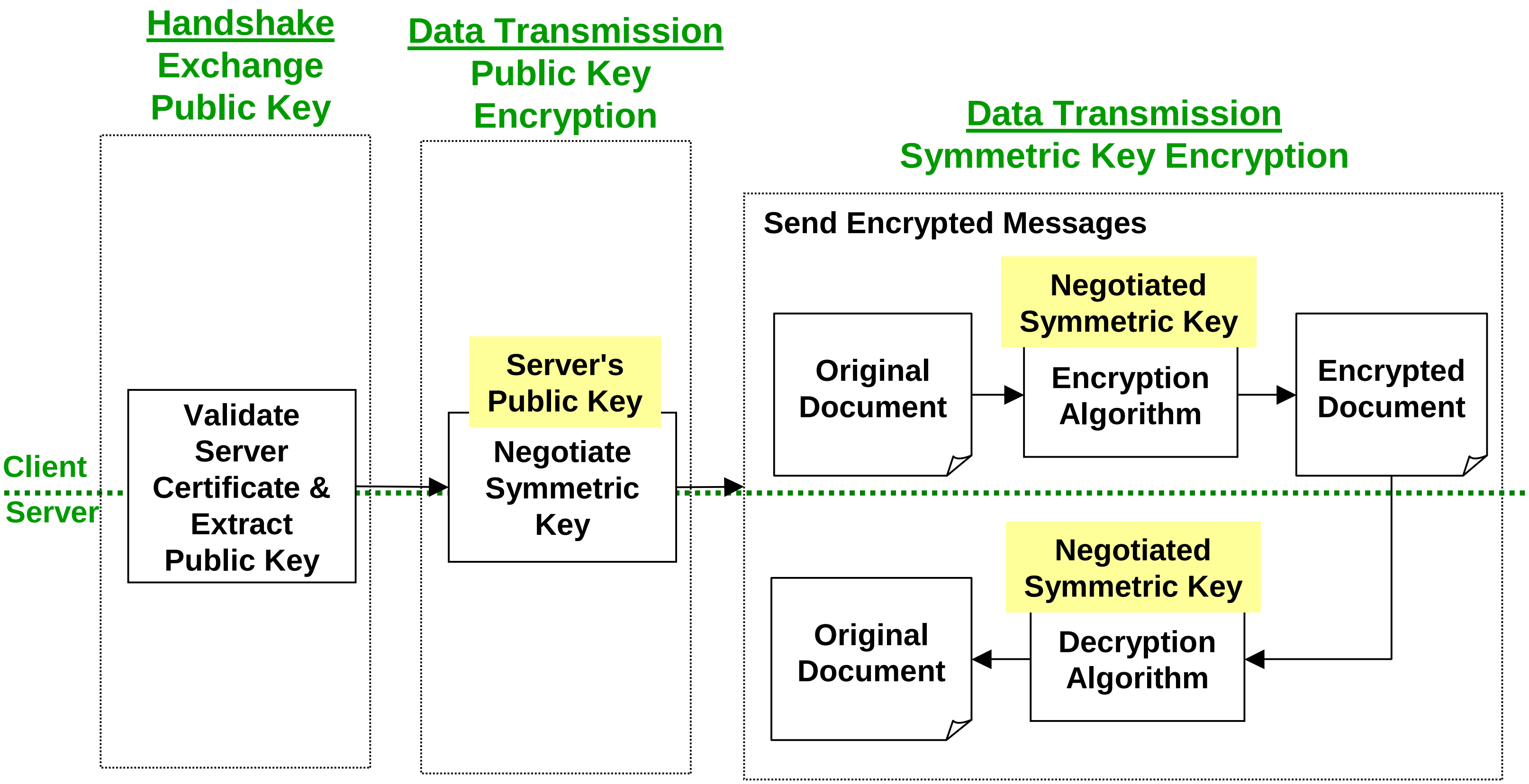
Certificate: Public Key + Identify Information
Benefit:Enables Client to verify validity of "Public Key" of Server



Negotiate Symmetric Key



SSL Protocol

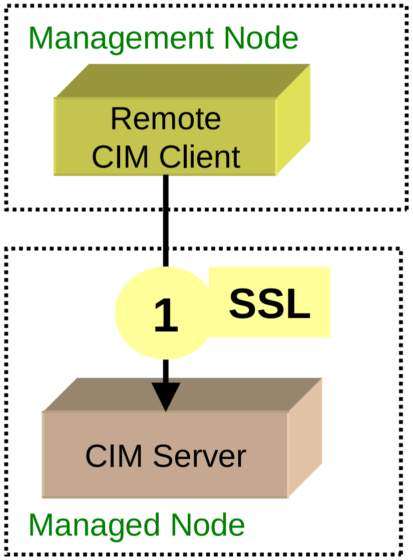


THE Open GROUP
OpenPegasus Developer Conference

invent

CIM Operation Client "connect"

Requester	Responder	Encoding	Protocol	Port	CIM Server Configuration Parameter
CIM Client	CIM Server	CIM-XML	HTTPS over TCP/IP	5989	enableHttpsConnection Default = TRUE



Management Node

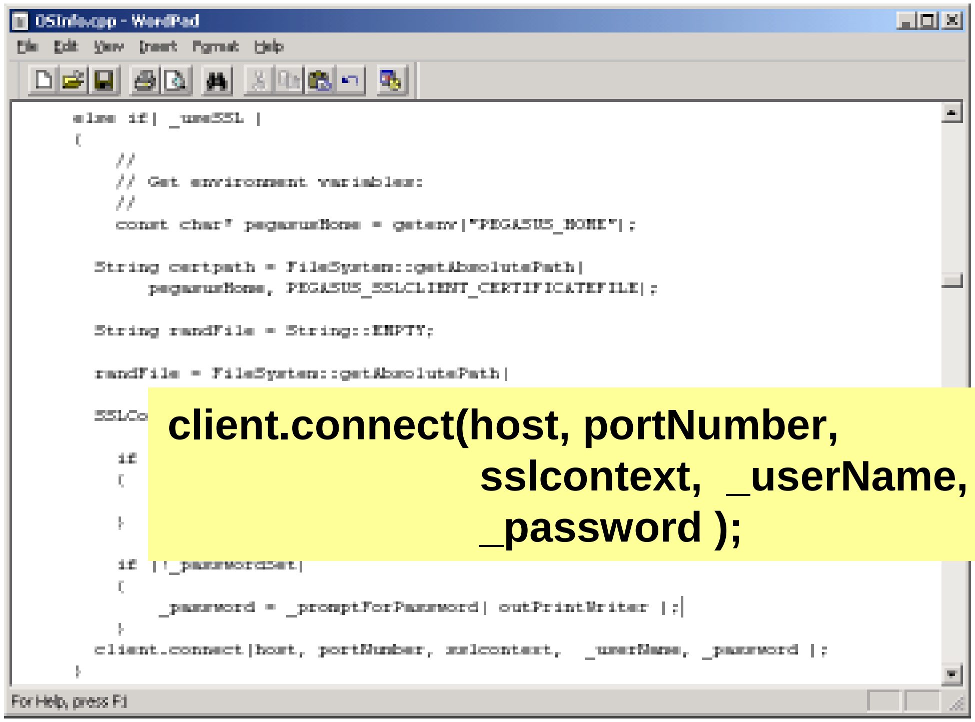
Remote CIM Client

1

SSL

CIM Server

Managed Node

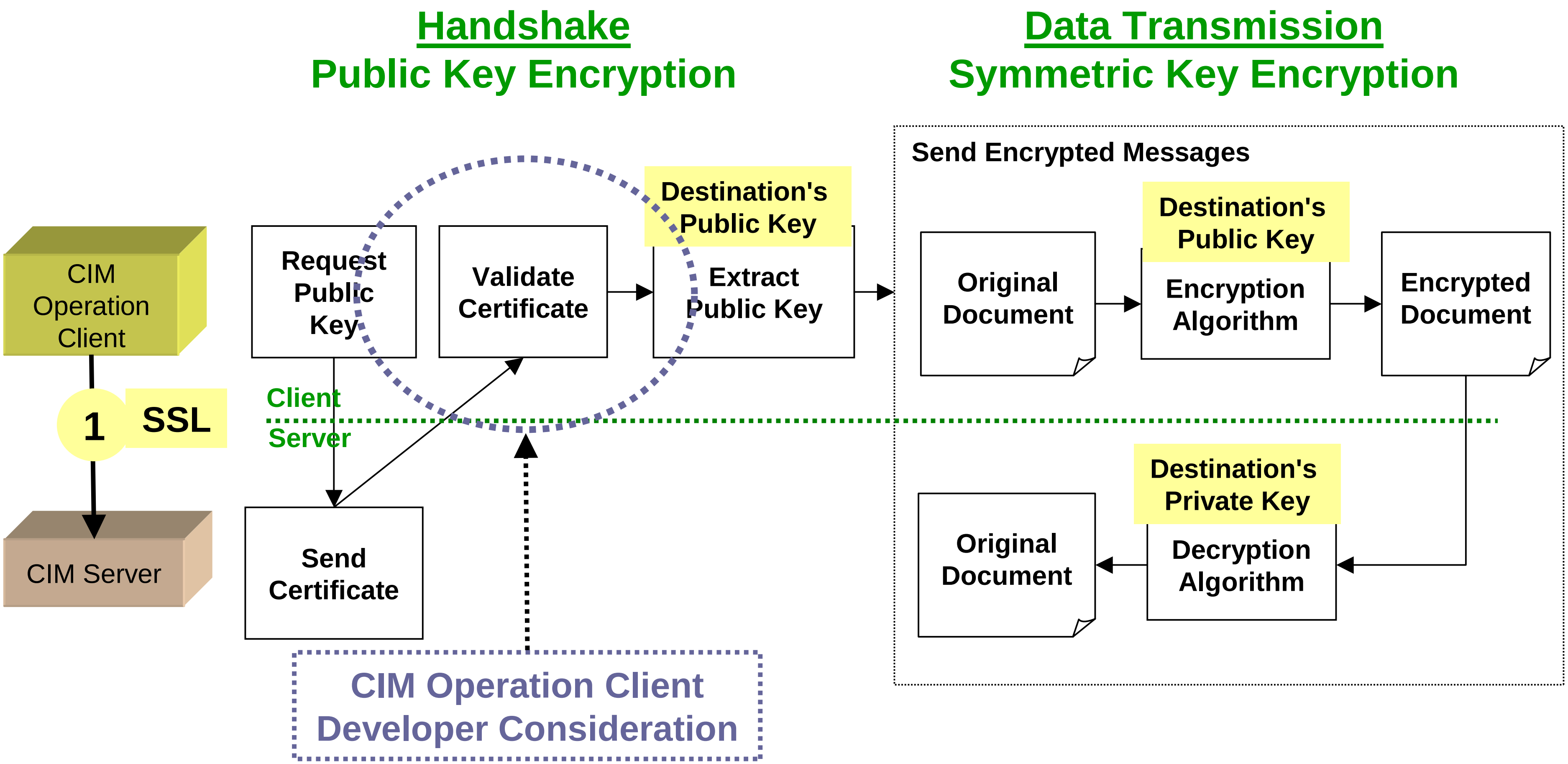


```
else if( _useSSL )
{
    //
    // Get environment variables:
    //
    const char* pegasusHome = getenv("PEGASUS_HOME");

  String certpath = FileSystem::getAbsolutePath(
       pegasusHome, PEGASUS_SSLCLIENT_CERTIFICATEFILE);

  String randFile = String::EMPTY;

  randFile = FileSystem::getAbsolutePath(

  SSLCo

    if
    {

    }

    if (!_passwordSet)
    {
        _password = _promptForPassword( outPrintWriter );
    }
  client.connect(host, portNumber, sslcontext,  _userName, _password );
}
```

**client.connect(host, portNumber,
sslcontext, _userName,
_password);**

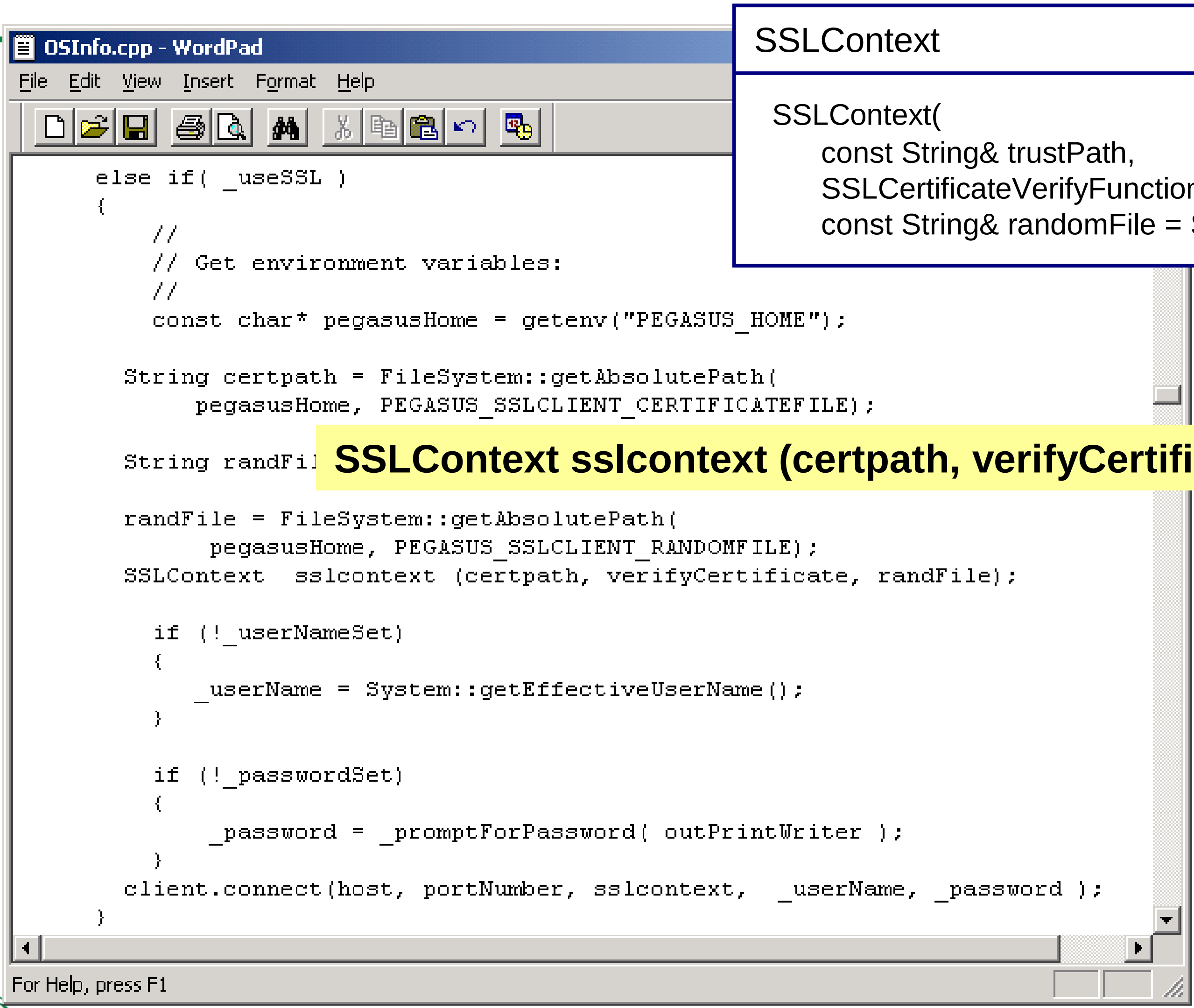
OpenPegasus Developer Conference

CIM Operation Client Considerations



THE Open GROUP
OpenPegasus Developer Conference

SSLContext Example



```
    else if( _useSSL )
    {
        //
        // Get environment variables:
        //
        const char* pegasusHome = getenv("PEGASUS_HOME");

      String certpath = FileSystem::getAbsolutePath(
          pegasusHome, PEGASUS_SSLCLIENT_CERTIFICATEFILE);

      String randFil

      randFile = FileSystem::getAbsolutePath(
            pegasusHome, PEGASUS_SSLCLIENT_RANDOMFILE);
      SSLContext  sslcontext (certpath, verifyCertificate, randFile);

        if (!_userNameSet)
        {
           _userName = System::getEffectiveUserName();
        }

        if (!_passwordSet)
        {
            _password = _promptForPassword( outPrintWriter );
        }
      client.connect(host, portNumber, sslcontext,  _userName, _password );
    }
```

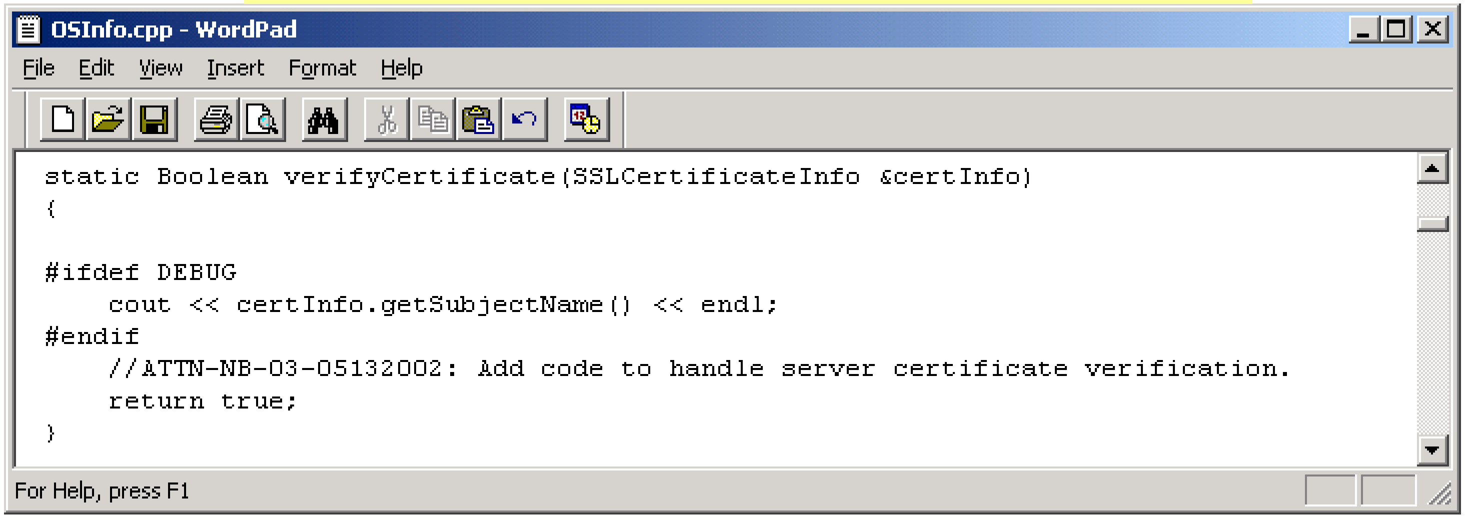
SSLContext

```
SSLContext(
    const String& trustPath,
    SSLCertificateVerifyFunction* verifyCert,
    const String& randomFile = String::EMPTY);
```

SSLContext sslcontext (certpath, verifyCertificate, randFile)

SSLContext Example

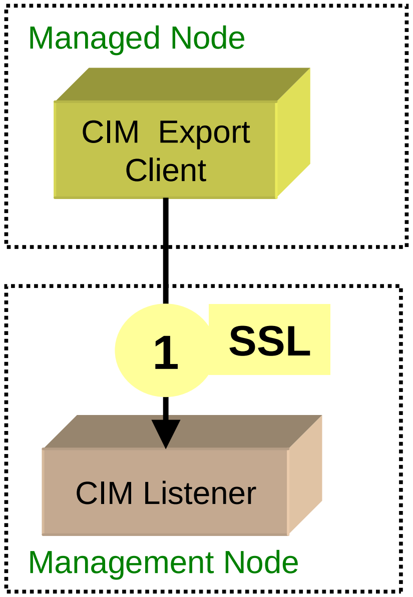
verifyCertificate(SSLICerticateInfo &certInfo



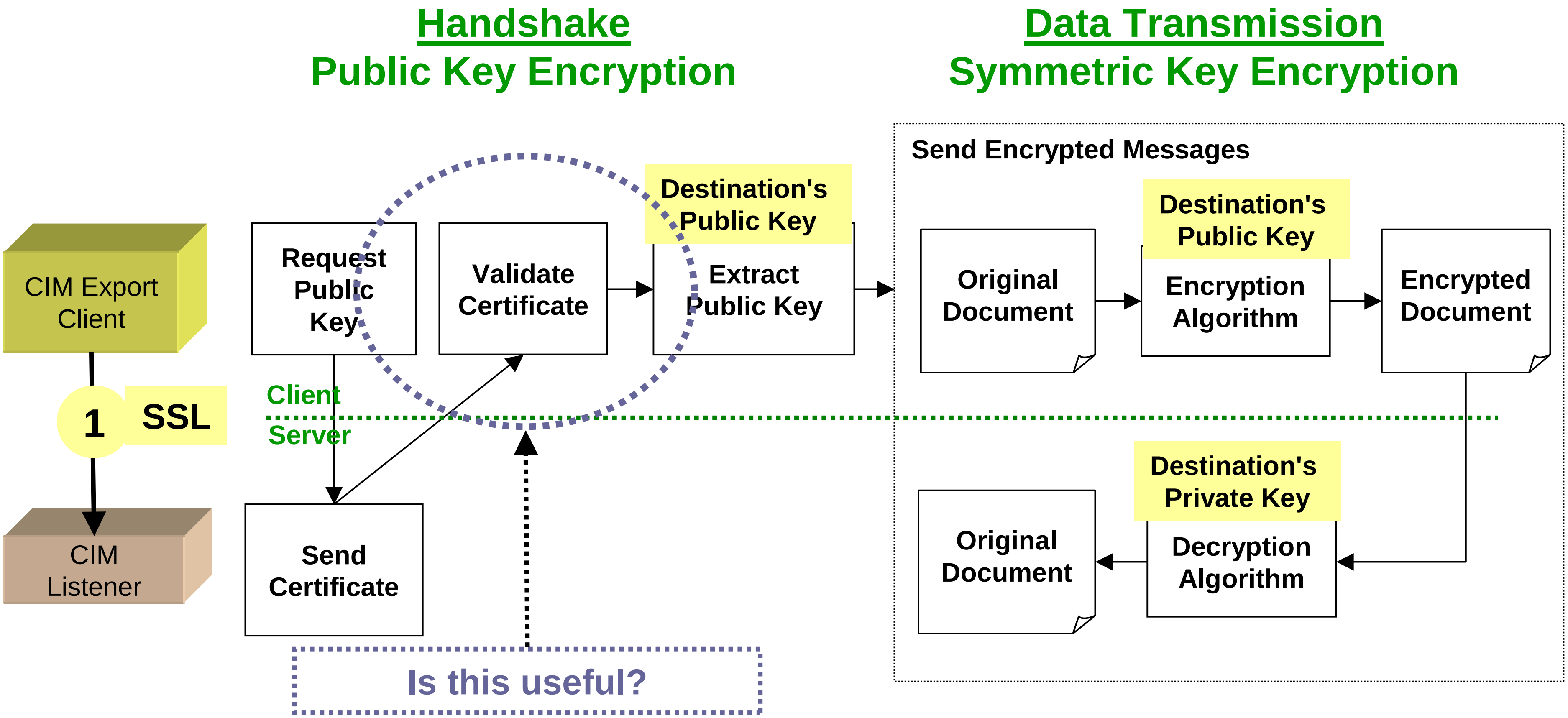
```
static Boolean verifyCertificate(SSLCertificateInfo &certInfo)
{

#ifdef DEBUG
    cout << certInfo.getSubjectName() << endl;
#endif
    //ATTN-NB-03-05132002: Add code to handle server certificate verification.
    return true;
}
```

CIM Export Client "connect"

Requester	Responder	Encoding	Protocol	Port	CIM Server Configuration Parameter
CIM Server	CIM Listener	CIM-XML	HTTPS over TCP/IP	Configurable	CIM_IndicationHandlerCIMXML

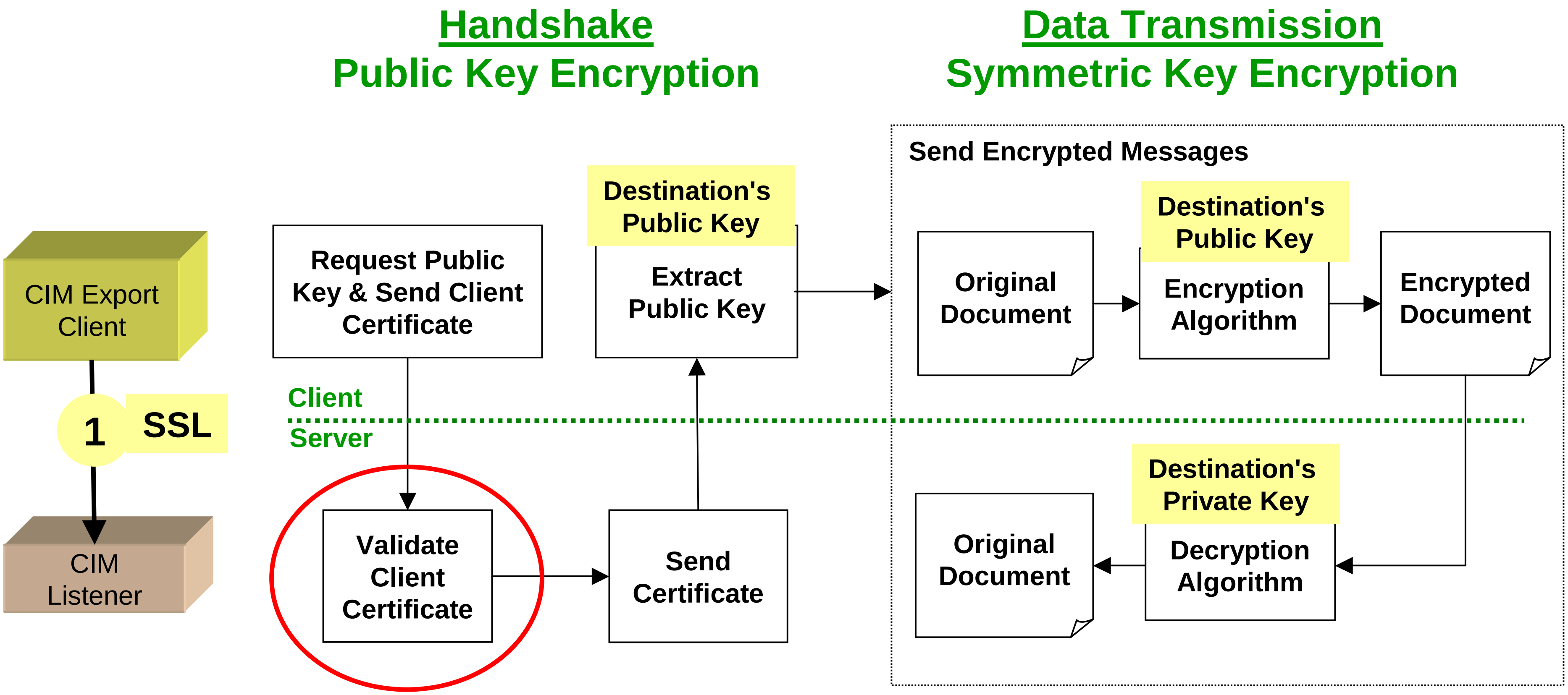


CIM Export Client Considerations



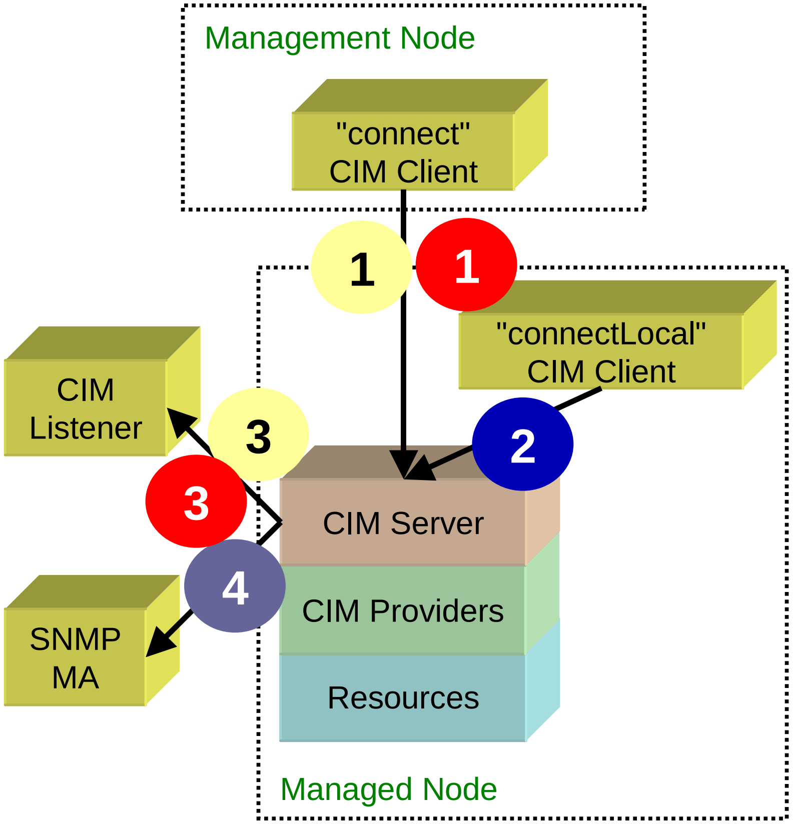
THE Open GROUP
OpenPegasus Developer Conference

CIM Export Client Considerations



Connection Point Summary

ID	Requestor	Responder
1	"connect" CIM Client	CIM Server
2	"connectLocal" CIM Client	CIM Server
3	CIM Server	CIM Listener
4	CIM Server	SMNP MA



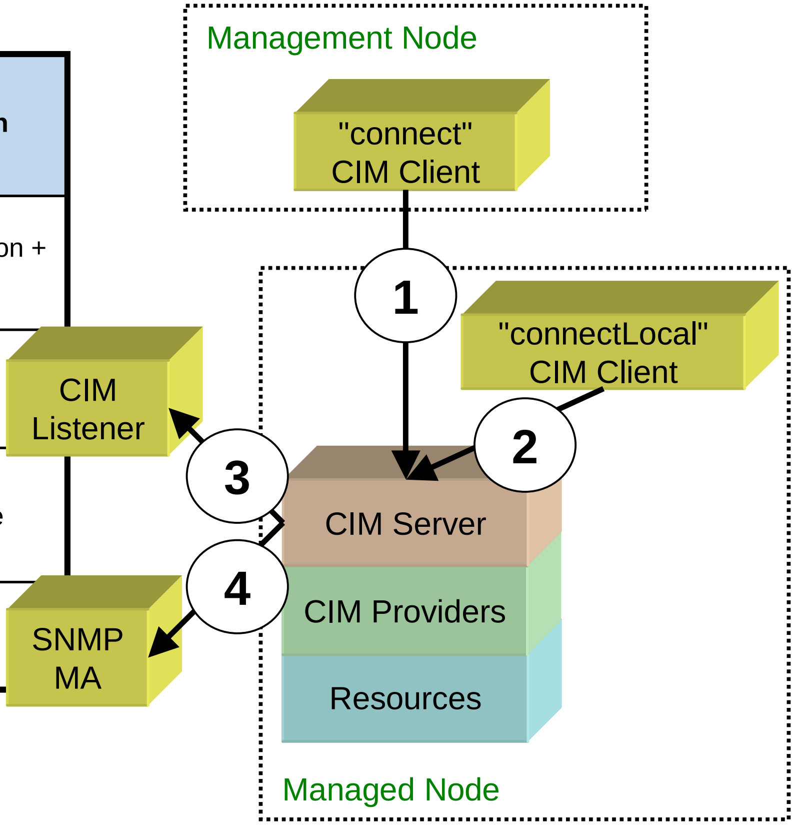
Module Content

HP WBEM Security

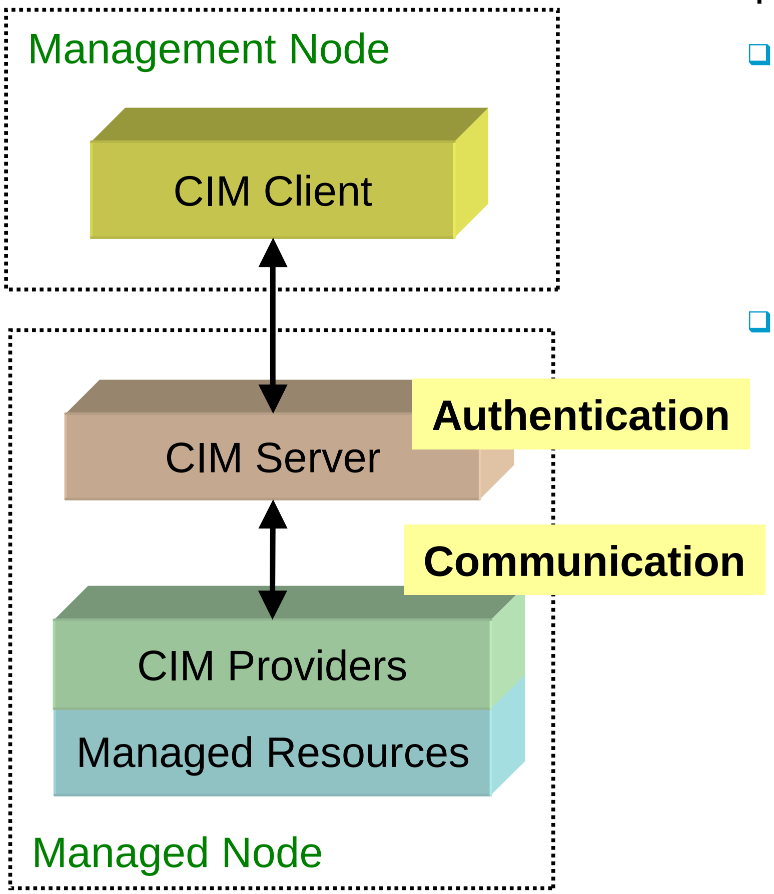
- Overview
- Secure Communication
- **User Authentication**
- User Authorization

Authentication Protocols

ID	Requestor	Responder	Requestor Authentication Protocol
1	"connect" CIM Client	CIM Server	Basic Authentication + PAM
2	"connectLocal" CIM Client	CIM Server	Proprietary
3	CIM Server	CIM Listener	SSL Certificate
4	CIM Server	SMNP MA	



CIM Server Role

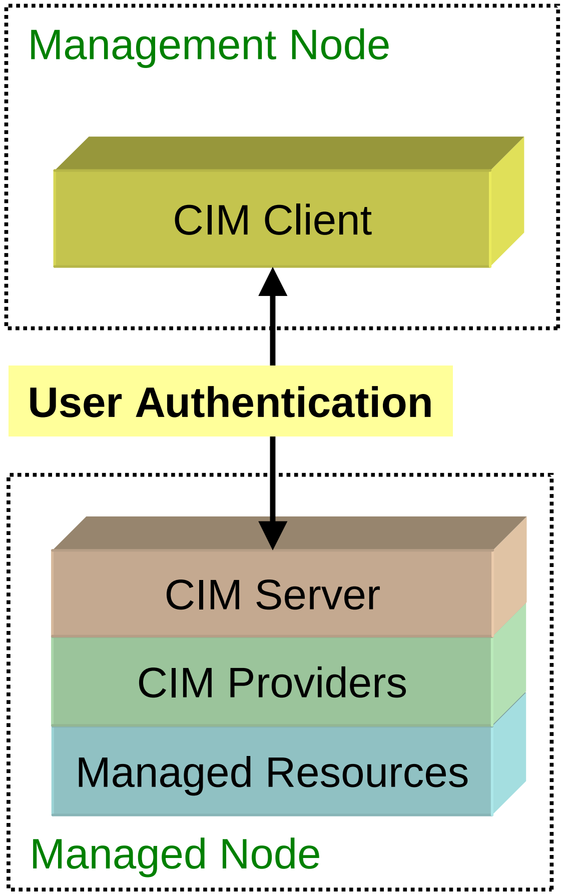


The **CIM Server** is responsible for

- Authenticating the user issuing the CIM Request. A CIM Request will be rejected if the user name is not valid on the system where CIM Server is running.
- Communicating the name of the authenticated user to the CIM Provider.

THE Open GROUP
OpenPegasus Developer Conference

Basic Authentication



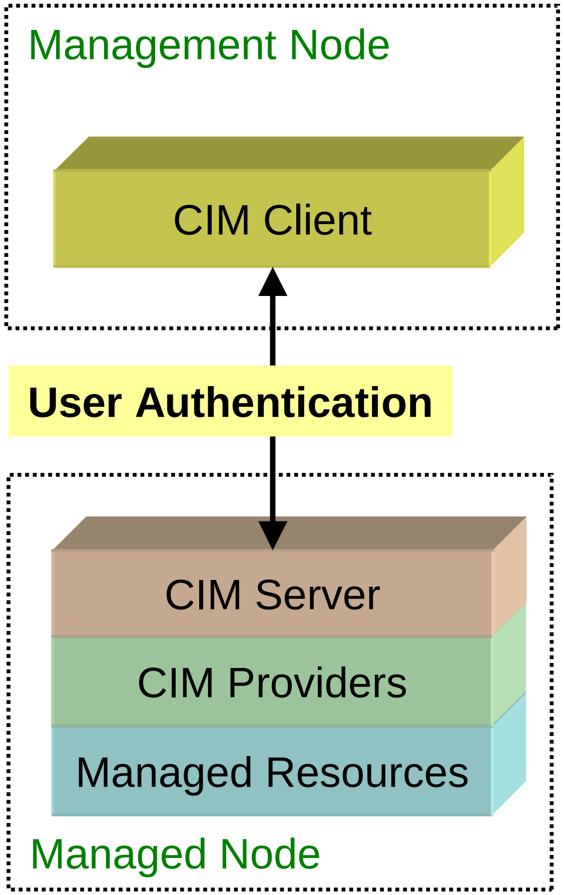
Key Fact: The CIM-XML Specification supports the use of Basic and Digest Authentication* as defined by the following HTTP Specifications:

1. Hypertext Transfer Protocol HTTP/1.0
   IETF RFC 1945, May 1996
   (http://www.ietf.org/rfc/rfc1945.txt)
2. Hypertext Transfer Protocol HTTP/1.1
   IETF RFC 2068, January 1997
   (http://www.ietf.org/rfc/rfc2068.txt)

HP WBEM Services Fact: By default, HP WBEM Services uses Basic Authentication, in conjunction with SSL, to challenge and validate remote CIM users.

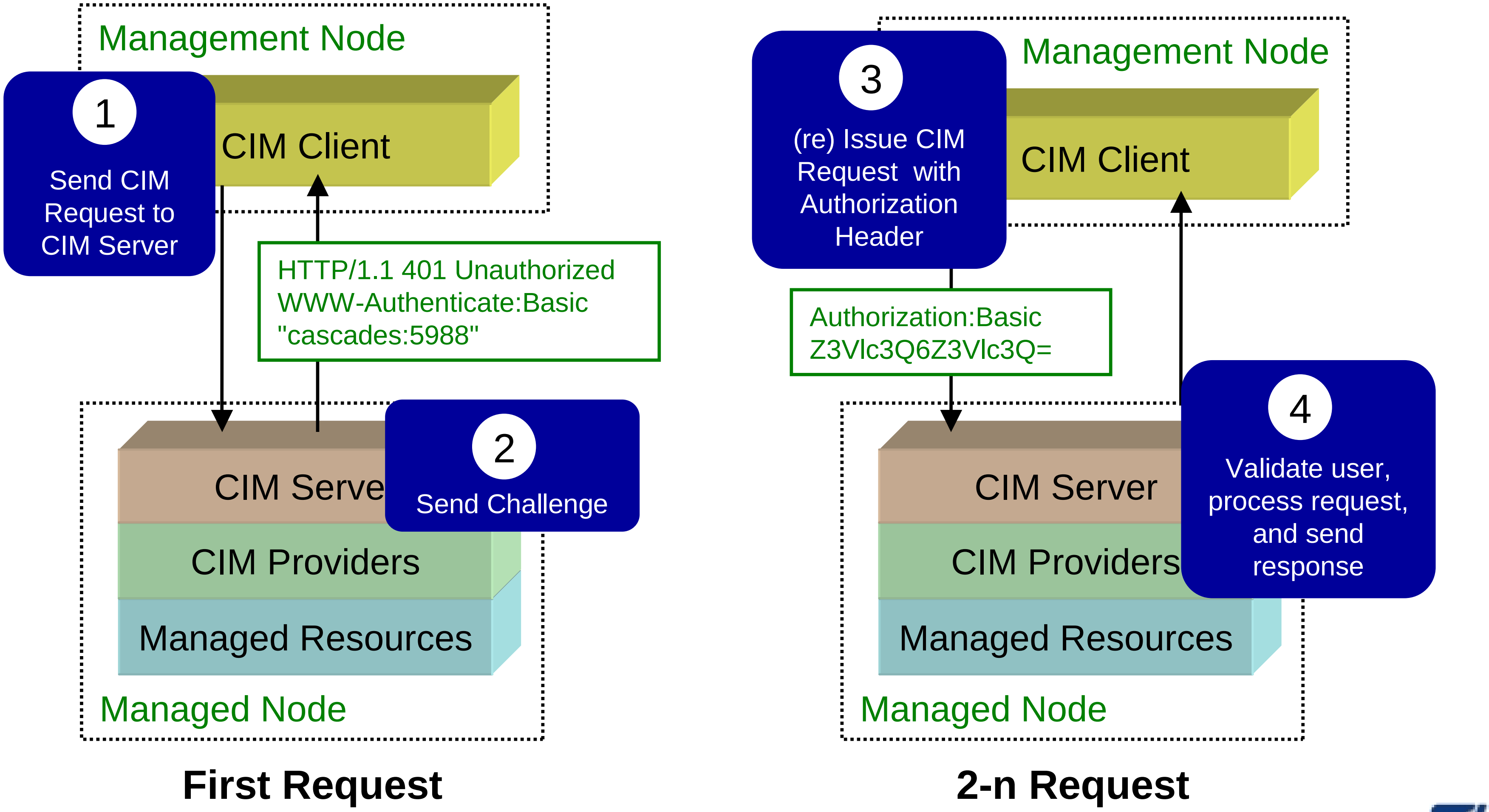
***Digest Authentication is NOT supported by the HP WBEM Services product. SSL is the recommended encryption mechanism.**

Basic Authentication



Warning: Basic Authentication requires the client to pass both the user name and password and uses Base64 encoding for the user name and password. This encoding is NOT secure. SSL should ONLY be disabled in environments where the transmission of clear text passwords is NOT an issue.

Basic Authentication Protocol

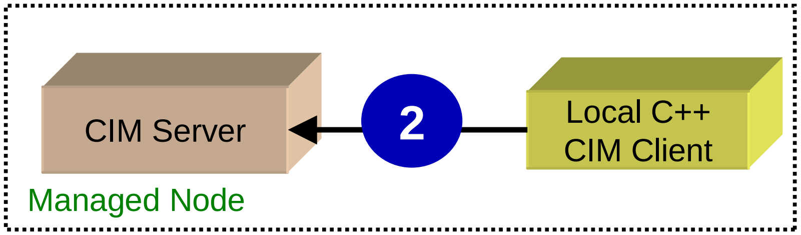


THE Open GROUP
OpenPegasus Developer Conference

invent

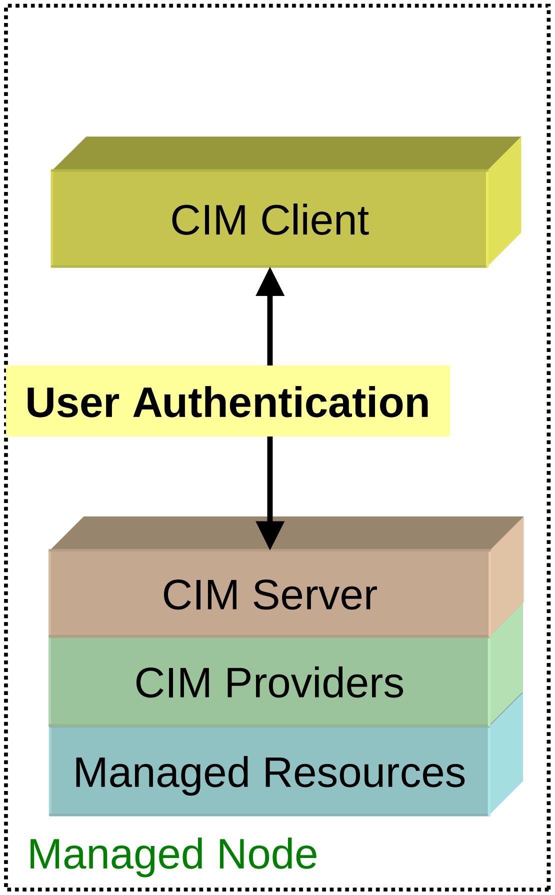
"connectLocal" Connection

Requester	Responder	Encoding	Protocol	Authentication Protocol
CIM Client	CIM Server	CIM-XML	Proprietary	Proprietary Variations on Basic Authentication



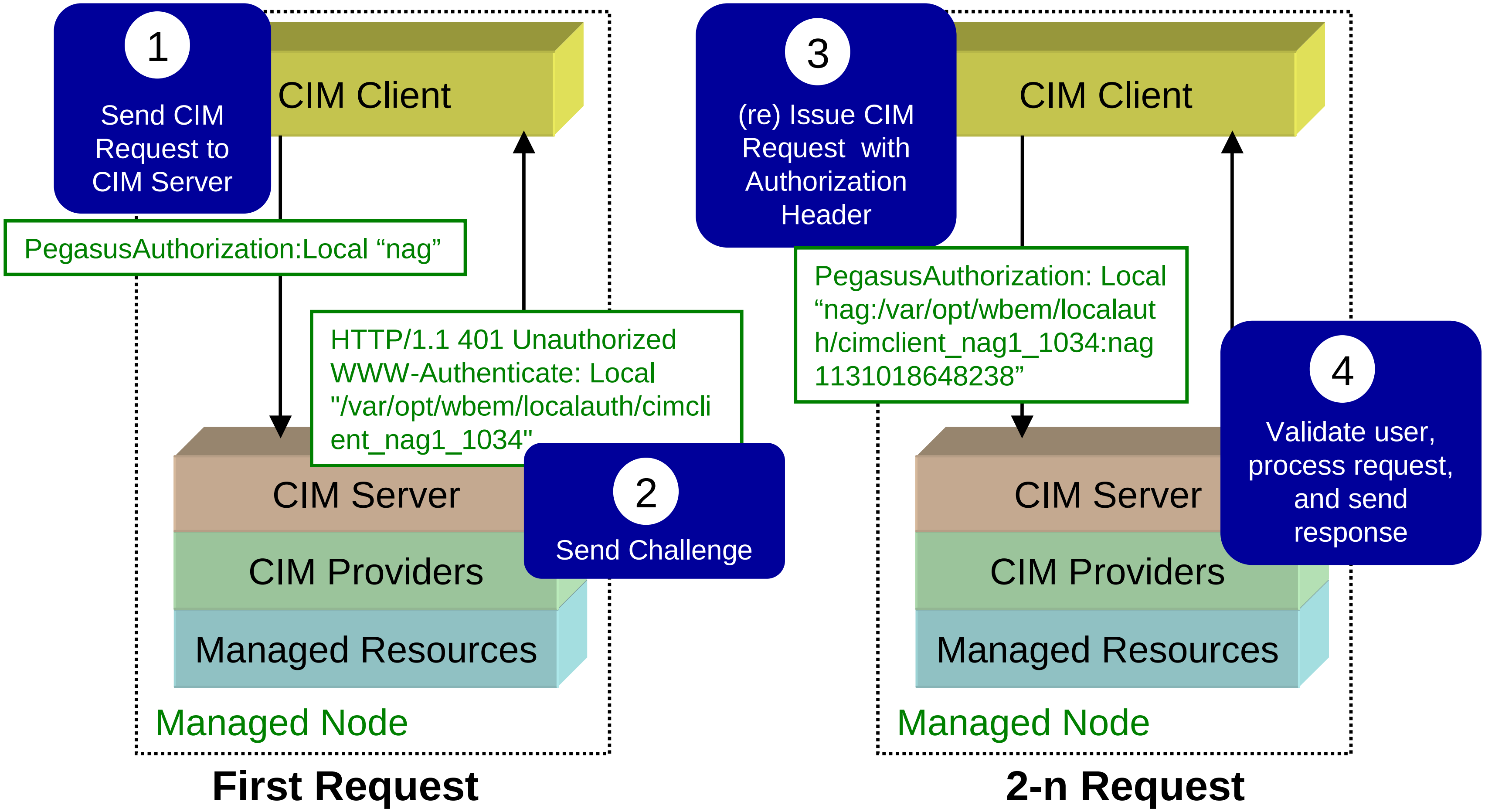
Note: The connectLocal interface is NOT STANDARD and only supported for CIM Clients built with the HP WBEM Services SDK.

Local Authentication



Fact: Local Authentication does not require the Client to send a password. This eliminates the need for user to specify a user name or password when issuing management commands on the local system (e.g., as a command line argument in a batch job). Instead the CIM Server uses the system "user name" associated with the process running the CIM Client application.

Local Authentication


1
Send CIM Request to CIM Server
CIM Client
PegasusAuthorization:Local “nag”
HTTP/1.1 401 Unauthorized
WWW-Authenticate: Local "/var/opt/wbem/localauth/cimclient_nag1_1034"
2
Send Challenge
CIM Server
CIM Providers
Managed Resources
Managed Node
First Request
3
(re) Issue CIM Request with Authorization Header
CIM Client
PegasusAuthorization: Local “nag:/var/opt/wbem/localauth/cimclient_nag1_1034:nag 1131018648238”
4
Validate user, process request, and send response
CIM Server
CIM Providers
Managed Resources
Managed Node
2-n Request

Local Authentication Algorithm

CIM Server

1. Create a file with a unique file name.
2. Set the permissions so that only the owner can read this file.
3. Generate a random token (pseudorandom number) and write to the file.
4. Change the file owner to the user issuing the request.
5. Send a challenge to the requesting application.

CIM Client

6. The Client reads the random number from the file and sends it with request.

CIM Server

7. CIM Server authenticates the requesting user, processes the request and sends the response.

2

Send Challenge

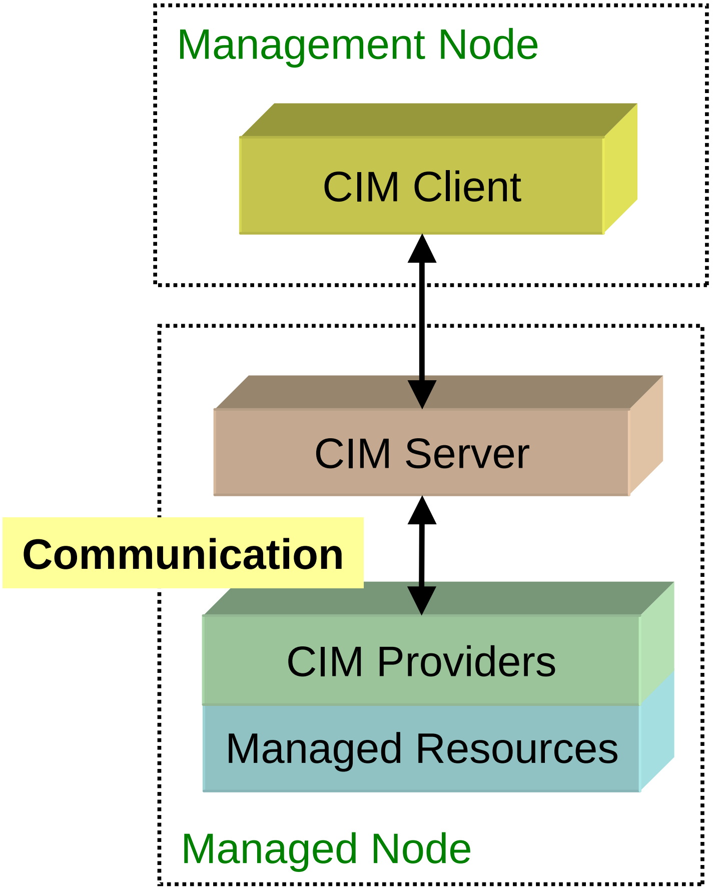
HTTP/1.1 401 Unauthorized
WWW-Authenticate: Local
"/var/opt/wbem/localauth/cimclient_nag1_1034"

3

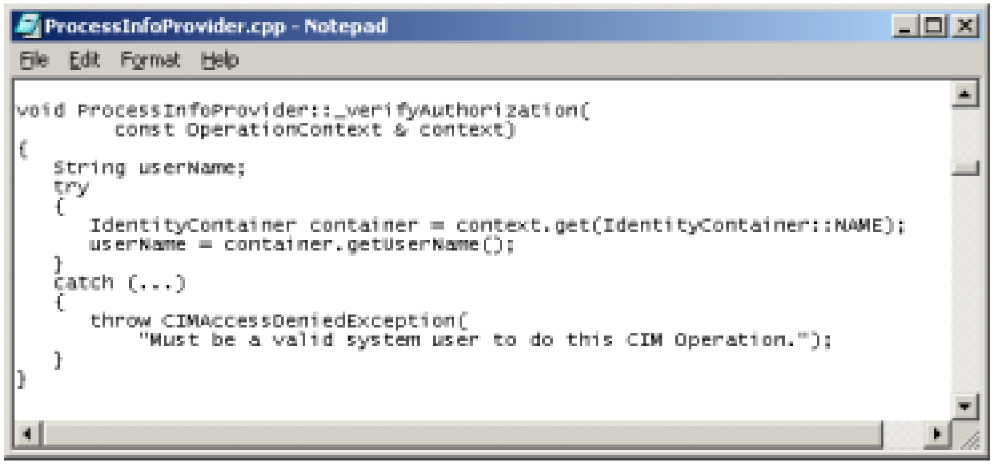
Re-issue CIM Request with Authorization Header

PegasusAuthorization: Local
“nag:/var/opt/wbem/localauth/cimclient_nag1_1034:nag1131018648238”

CIM Server Role



The **OperationContext** parameter is used to communicate the name of the authenticated user to the CIM Provider.



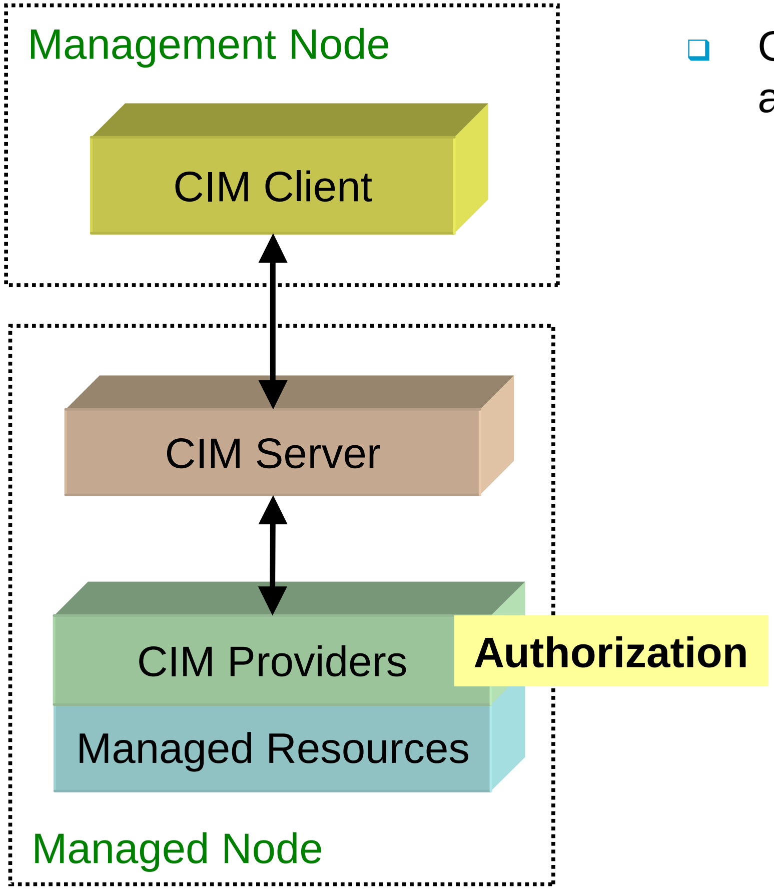
```
void ProcessInfoProvider::_verifyAuthorization(
        const OperationContext & context)
{
    String userName;
    try
    {
        IdentityContainer container = context.get(IdentityContainer::NAME);
        userName = container.getUserName();
    }
    catch (...)
    {
        throw CIMAccessDeniedException(
            "Must be a valid system user to do this CIM Operation.");
    }
}
```

Module Content

HP WBEM Security

- Overview
- Secure Communication
- User Authentication
- **User Authorization**

CIM Provider Role

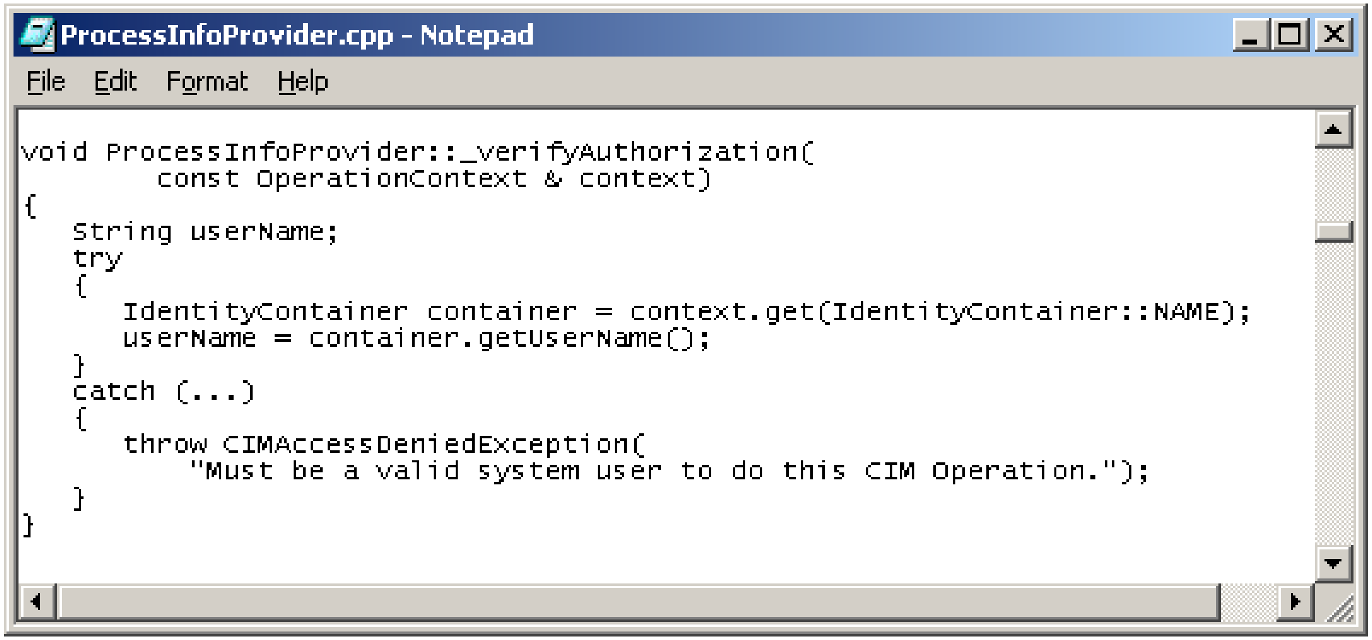


The **CIM Provider** is responsible for

- Granting the requesting user authorization to perform the operation.

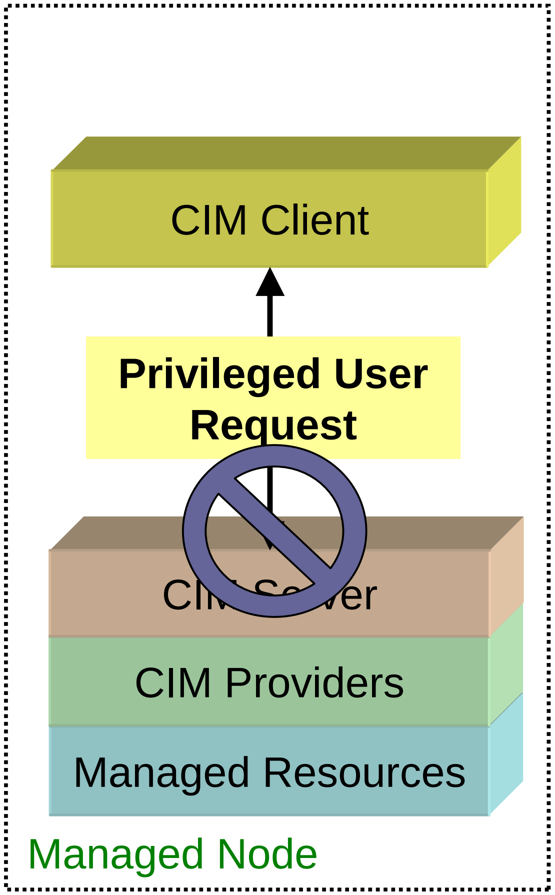
enumerateInstanceNames

Verify Authorization



```
void ProcessInfoProvider::_verifyAuthorization(
        const OperationContext & context)
{
   String userName;
   try
   {
      IdentityContainer container = context.get(IdentityContainer::NAME);
      userName = container.getUserName();
   }
   catch (...)
   {
      throw CIMAccessDeniedException(
          "Must be a valid system user to do this CIM Operation.");
   }
}
```

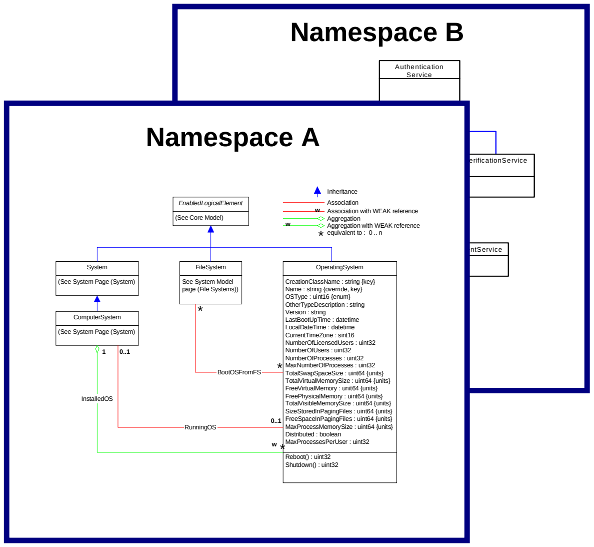
CIM Server "Hardening"



Fact: As an additional security feature the CIM Server is configured, by default, to REJECT all remote requests by a Privileged user (i.e., a user with UID 0). This option is configurable.

Namespace Authorization

Fact: As an additional security feature, the CIM Server can be configured to support Namespace Authorization. If Namespace Authorization is enabled, a user must be granted the appropriate permission (i.e., read or write) on the target Namespace before the CIM Server will pass the CIM Operation Request to the Provider(s). Namespace Authorization can be used to restrict access to a resource that would otherwise be granted by the Provider.


Namespace B
Authentication Service
erificationService
ntService
Namespace A
EnabledLogicalElement
(See Core Model)
Inheritance
Association
Association with WEAK reference
Aggregation
Aggregation with WEAK reference
equivalent to : 0 .. n
System
(See System Page (System))
ComputerSystem
(See System Page (System))
FileSystem
See System Model page (File Systems))
OperatingSystem
CreationClassName : string {key}
Name : string {override, key}
OSType : uint16 {enum}
OtherTypeDescription : string
Version : string
LastBootUpTime : datetime
LocalDateTime : datetime
CurrentTimeZone : sint16
NumberOfLicensedUsers : uint32
NumberOfUsers : uint32
NumberOfProcesses : uint32
MaxNumberOfProcesses : uint32
TotalSwapSpaceSize : uint64 {units}
TotalVirtualMemorySize : uint64 {units}
FreeVirtualMemory : unit64 {units}
FreePhysicalMemory : uint64 {units}
TotalVisibleMemorySize : uint64 {units}
SizeStoredInPagingFiles : uint64 {units}
FreeSpaceInPagingFiles : uint64 {units}
MaxProcessMemorySize : uint64 {units}
Distributed : boolean
MaxProcessesPerUser : uint32
Reboot() : uint32
Shutdown() : uint32
1
0..1
InstalledOS
RunningOS
BootOSFromFS
w

Note: The CIM Provider is still responsible for authorizing access to the resource.

Fact: By default, Namespace Authorization is NOT enabled.

Namespace Authorization

Read Operations	Write Operations
GetClass	SetProperty
GetInstance	SetQualifier
GetProperty	CreateClass
GetQualifier	CreateInstance
References	ModifyInstance
ReferenceNames	ModifyClass
Associators	DeleteClass
AssociatorNames	DeleteInstance
EnumerateClassNames	DeleteQualifier
EnumerateInstanceNames	InvokeMethod
EnumerateQualifiers	
EnumerateClasses	
EnumerateInstances	
ExecQuery	