



Overview of the Pegasus CIM/WBEM Implementation

July 24th 2002

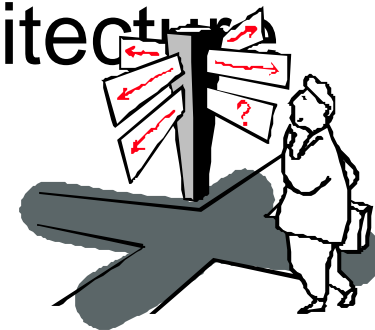
Karl Schopmeyer

Chair Open Group Enterprise Mgt. Forum

k.schopmeyer@opengroup.org

Agenda

- ❑ CIM/WBEM Overview
- ❑ Overview -What (and why)is Pegasus?
- ❑ The Pegasus Environment
- ❑ The Pegasus Software Architecture
- ❑ Pegasus Status Today
- ❑ The Pegasus Project



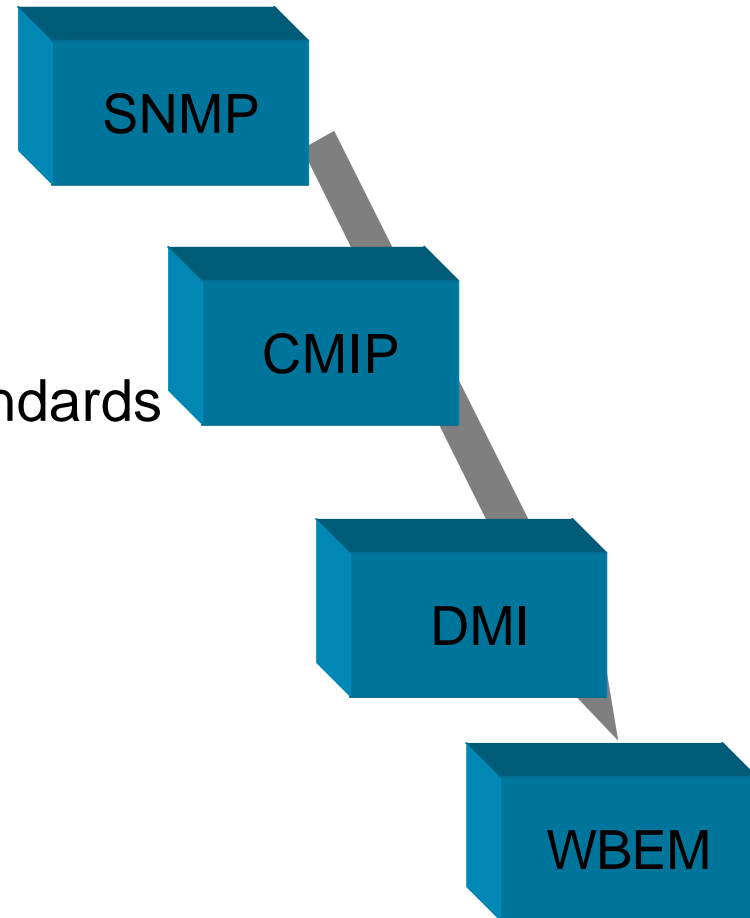


1. CIM and WBEM

The Management Standards Groups

THE *Open* GROUP

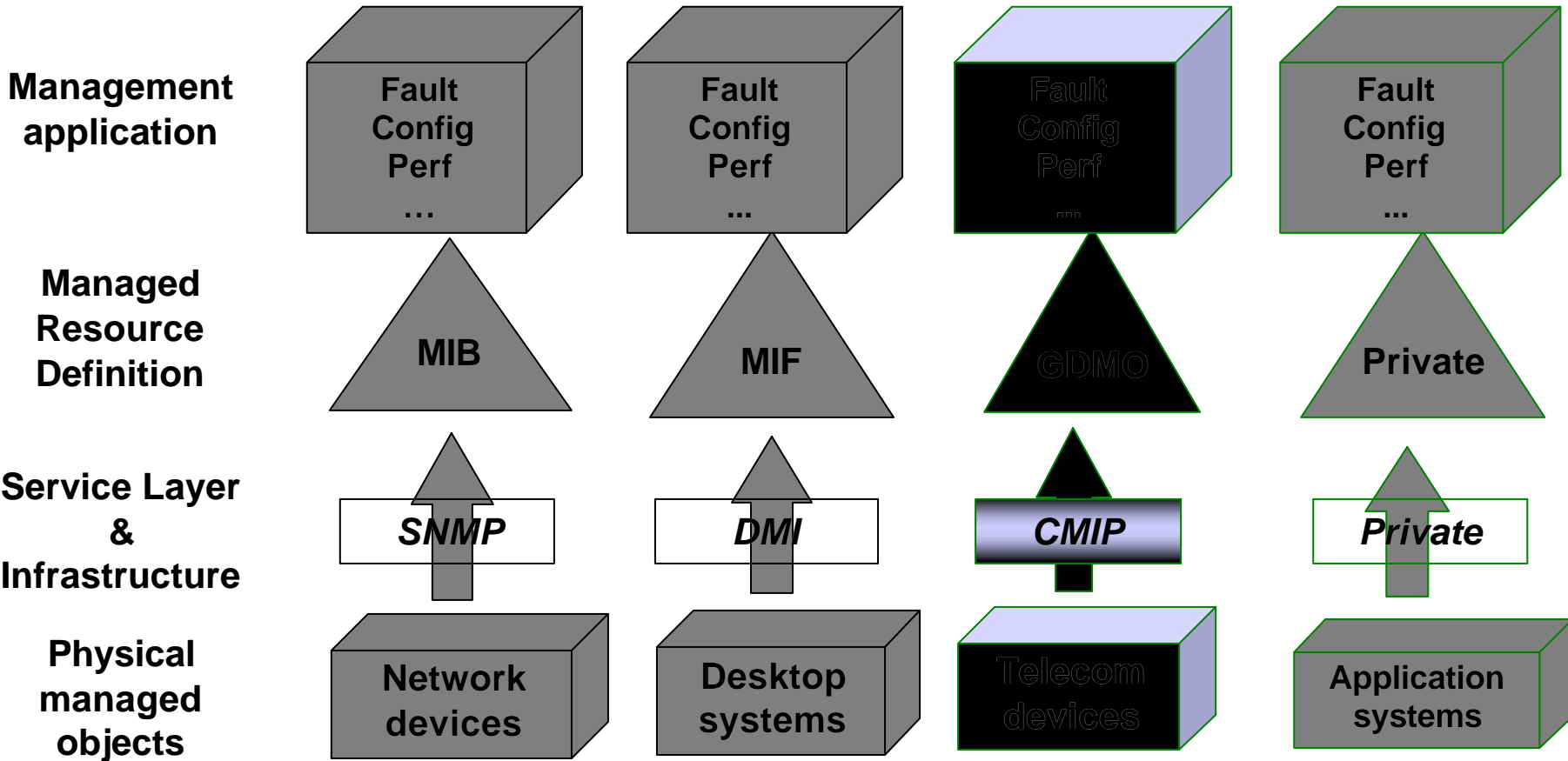
- DMTF
 - DMI
 - CIM / WBEM
- The Open Group
 - Unix Management Standards
 - Application Management Standards
 - CIM / WBEM
- IETF
 - SNMP
- TMF
 - CMIP
 - OSS/NG





Management Environment Today

THE *Open* GROUP





A “Management” Problem

THE *Open* GROUP

- ❑ Example: Managing a Fibre Channel SAN
 - Computer system and its devices
 - Storage enclosures
 - Hubs and switches
- ❑ Manageability difficult, not unique to Fibre Channel
 - Different data standards (SNMP, DMI, SES, ...)
 - Different terminology
 - Proprietary MIBs
 - Missing data (topologies and dependencies)

WBEM Architectures

THE *Open* GROUP

WBEM

Data Description

CIM

• **DMI**

Discovery

</xml/CIM>

Transport Encoding

HTTP

Access

Why CIM?

- ❑ Enterprise-wide management
 - Wide breadth of objects + name scoping
 - Repository independent
 - Unifies and extends existing standards (SNMP, X.500, ...)
- ❑ OO design
 - Abstraction, inheritance, ability to “classify”
 - Reduces complexity, provides predictability
 - Well-defined “locations” and usage semantics for classes and associations
 - Extensibility via subclassing
- ❑ Associations depict relationships
 - Dependencies, topologies, aggregations, scoping, ...
- ❑ “Standard”, inheritable methods

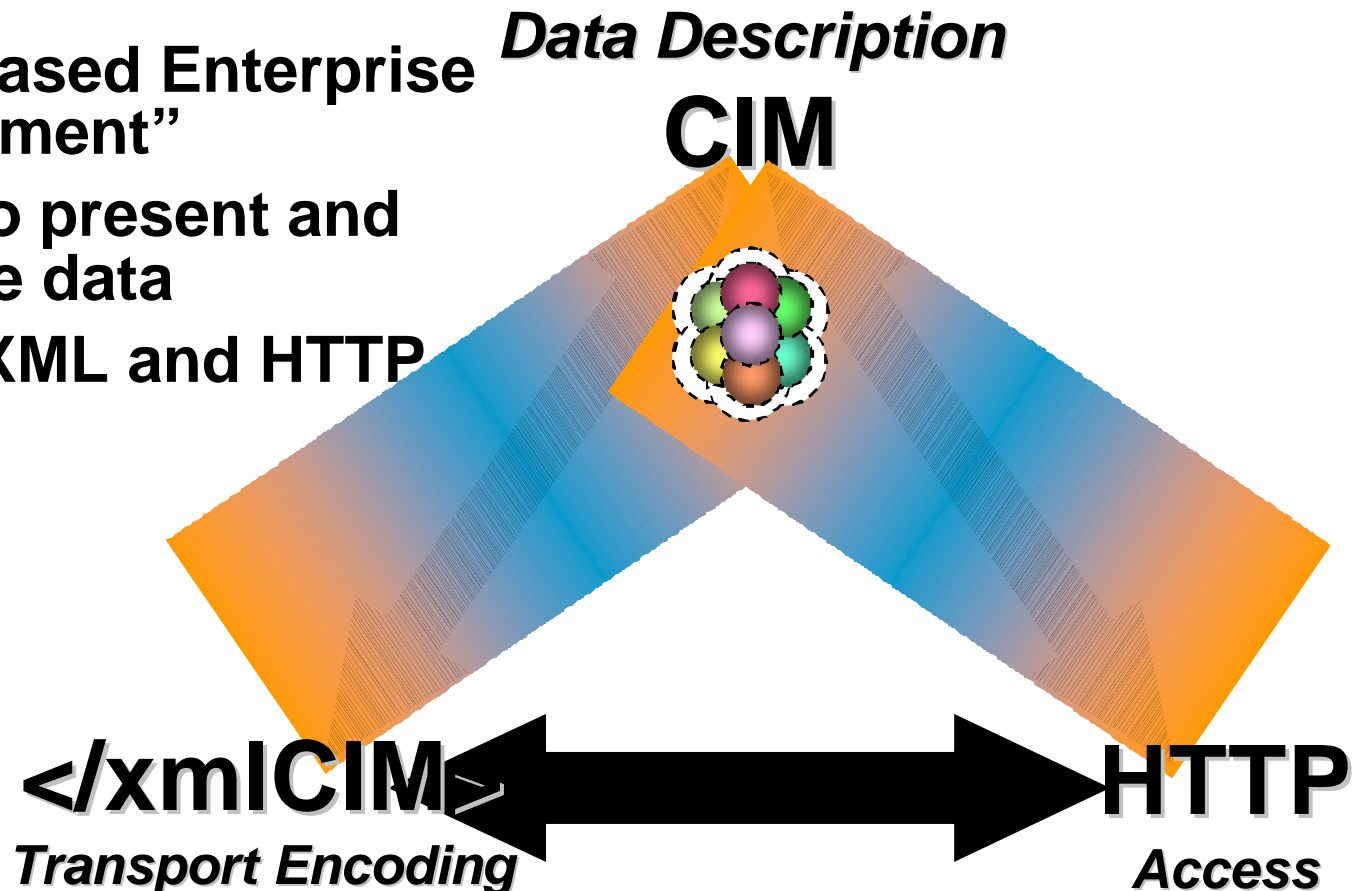
What Is CIM?

- ❑ Object-oriented data model
- ❑ Core Specification
 - “Meta”-model, high level concepts and language definitions
- ❑ “Core” and “Common” Models
 - Core Model contains info applicable to all management domains
 - Common Models address specific domains - Systems, Devices, Applications, Networks, Users, ...
 - Subclass from the Core Model
 - Models overlap and cross-reference
 - Vendor extensions allowed

What Is WBEM?

THE *Open* GROUP

- ◆ “Web-Based Enterprise Management”
- ◆ Model to present and organize data
- ◆ Use of XML and HTTP



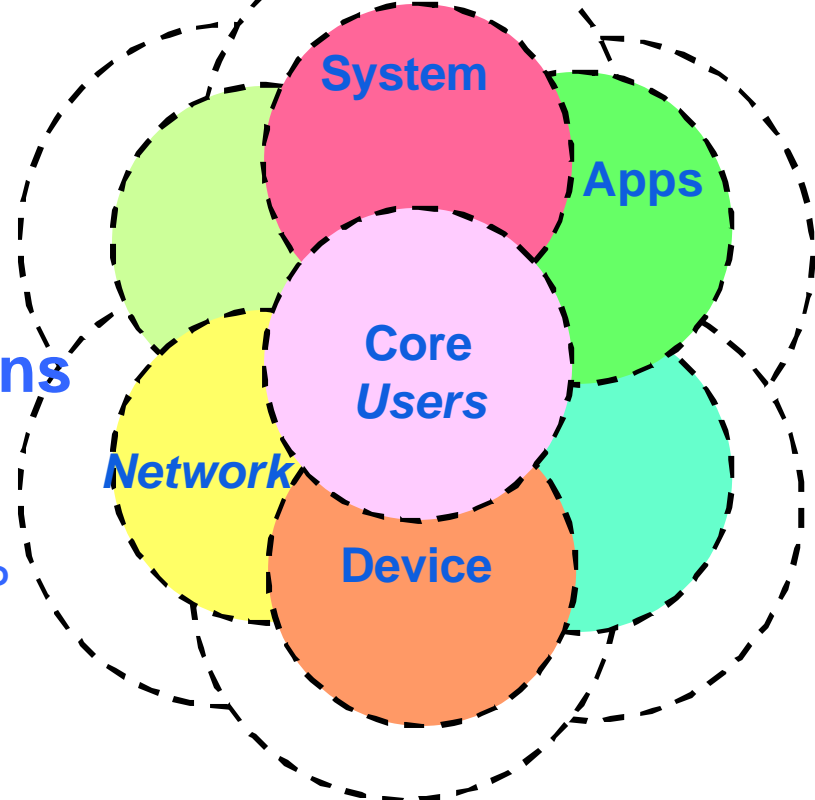
DMTF WBEM Components

THE *Open* GROUP

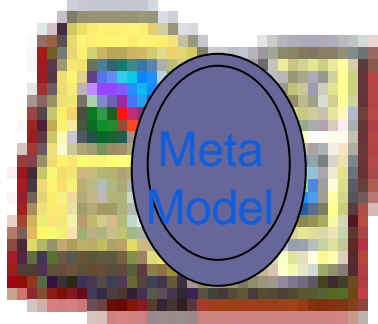
CIM Schema v2.7

(900+ classes)

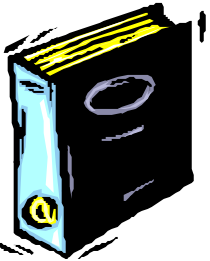
Extension
Schema



CIM Specification V2



Interoperability Specifications



- CIM Operations over HTTP
- CIM XML



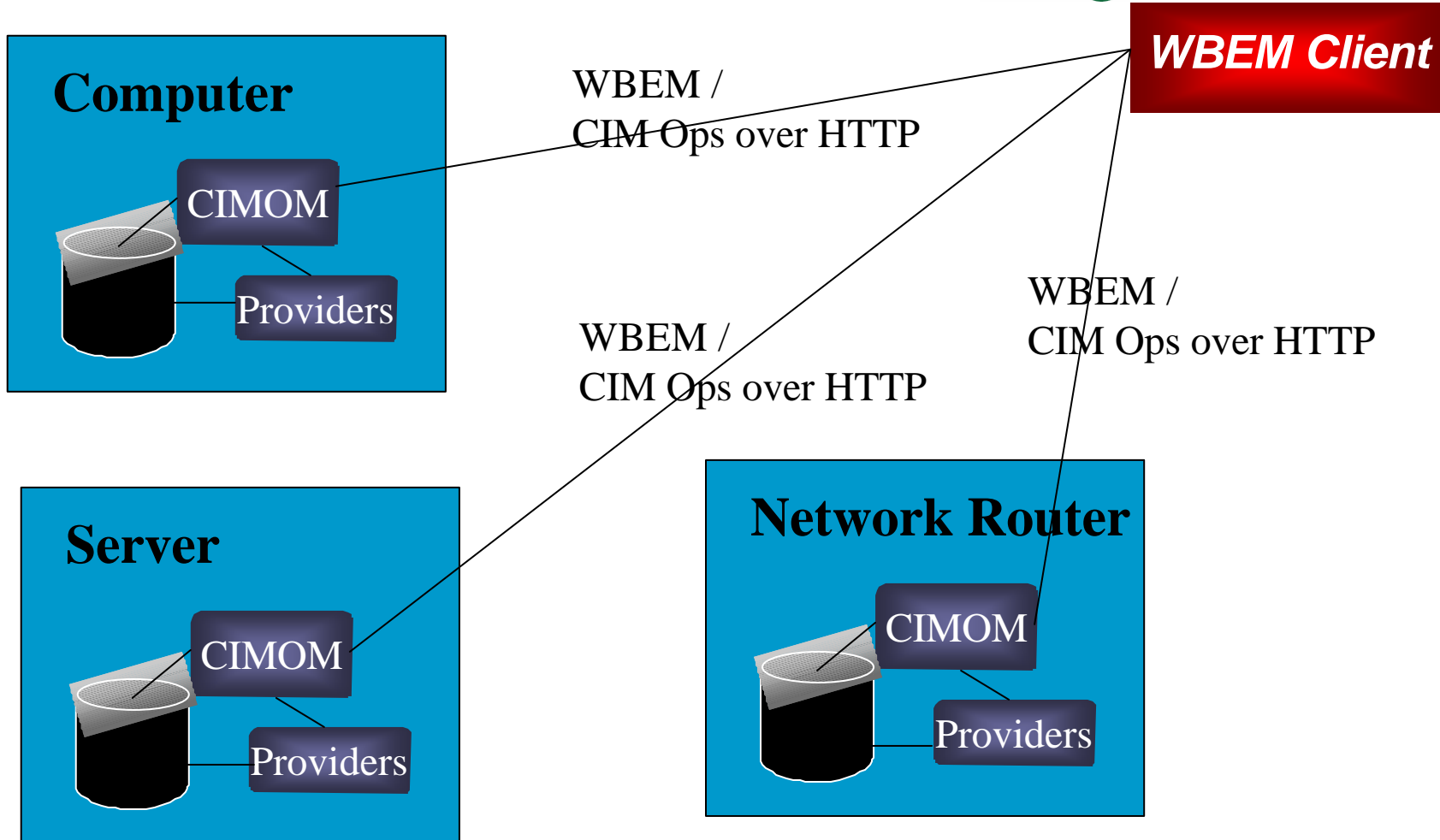
The Major Schemas Today

THE *Open* GROUP

- ❑ Core
- ❑ System
- ❑ Devices
- ❑ Networks
- ❑ Applications (today lifecycle)
- ❑ Metrics (originally for ARM support)
- ❑ User
- ❑ Policy (was originated as service)
- ❑ Databases – Dormant
- ❑ Unix System Extensions
- ❑ Support

WBEM Architecture (Client-Server)

THE *Open* GROUP





What does CIM/WBEM Provide?

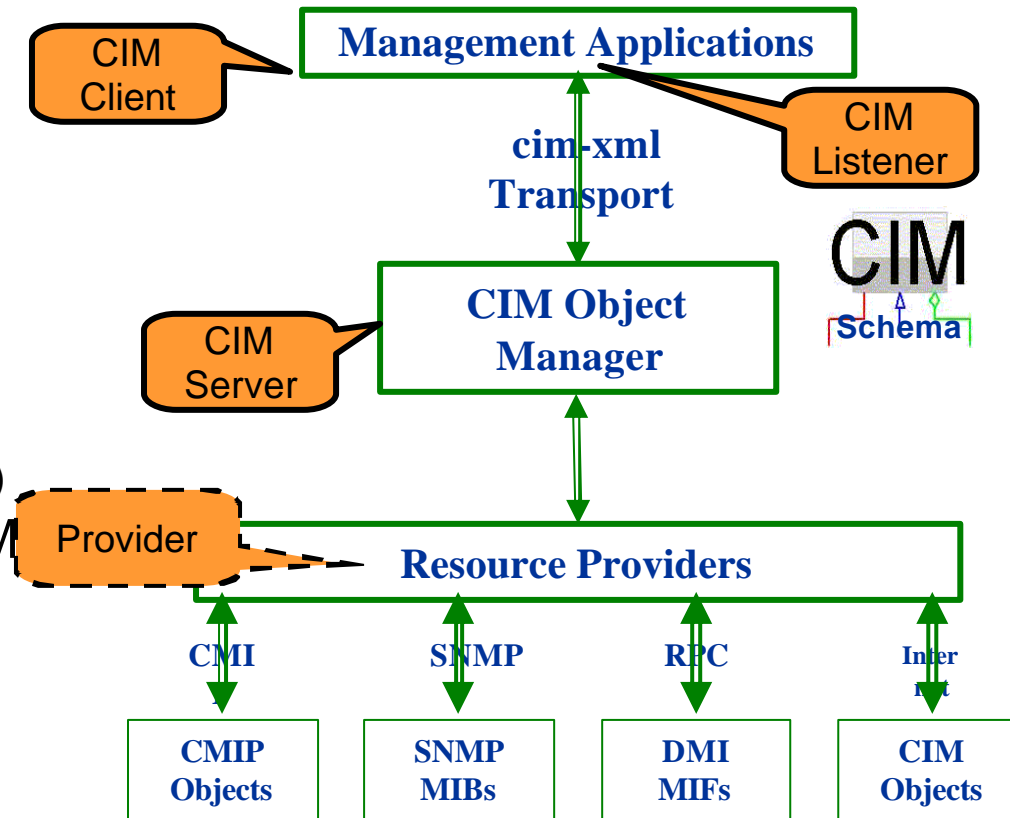
THE *Open* GROUP

- ❑ Abstractions of Managed objects
- ❑ A common semantic for objects
- ❑ Rich set of defined objects
- ❑ Relationships between objects
- ❑ Interoperability between management components.

WBEM Architecture Components

THE *Open* GROUP

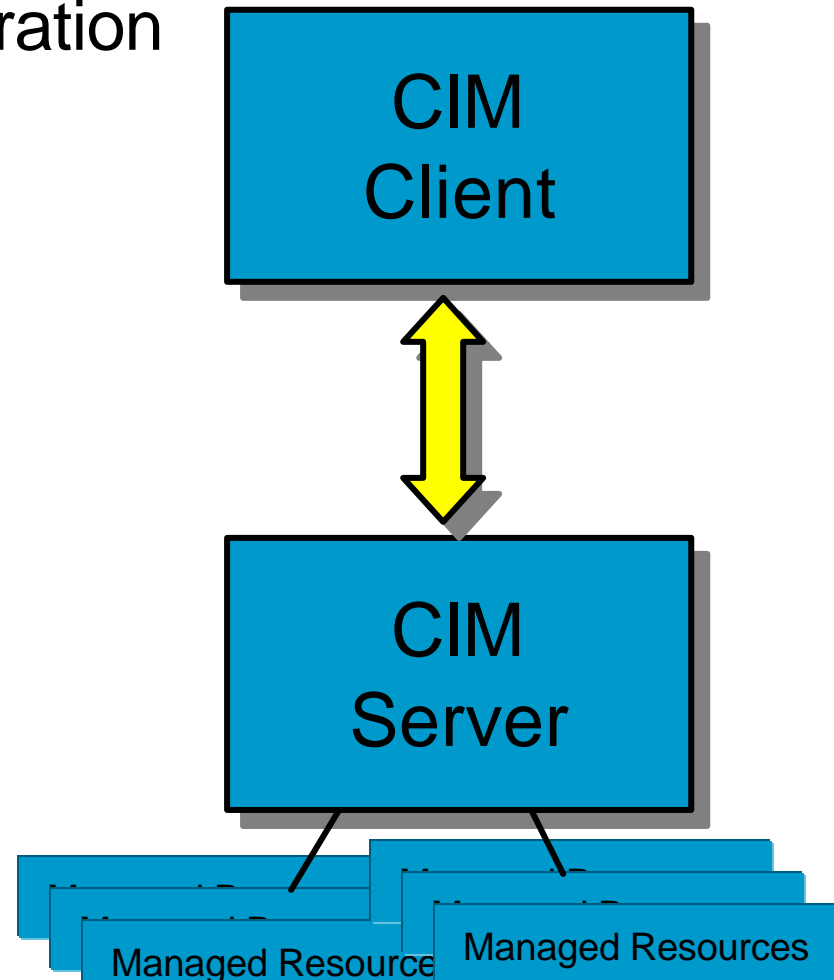
- Information Model
 - CIM Schema (Core, System,...)
- Communication Model
 - CIM Operations over HTTP
- Transport Encoding
 - Cim-xml – CIM/XML mapping
- Event Model
 - CIM indications (new in CIM 2.5)
- CIM Object Manager (CIMOM)
 - Operation Routing
 - Result Aggregation
- Repository
 - Class and Instance Persistence
- Resource Providers
 - Instrumentation subagents



Interoperability Characteristics

THE *Open* GROUP

- ❑ XML encoding
 - Definition for each operation
- ❑ HTTP Transport
 - HTTP 1.0 and 1.1
- ❑ Common Operations Semantics
 - Data
 - Meta data
 - Queries?
 - Methods



How Is CIM Schema Defined?

THE *Open* GROUP

- ❑ MOF - Managed Object Format (ASCII or Unicode)
- ❑ VISIO for UML (Unified Modeling Language)
- ❑ Whitepapers
- ❑ XML - eXtensible Markup Language
 - XML grammar can be used to describe CIM metaschema, Detailed in DTD (Document Type Defn)
 - DTD defines tags such as CLASS, INSTANCE and QUALIFIER
 - Associations are described via an ASSOCIATION.CLASS tag (Distinguished because they include references as properties)

Managed Object Format (MOF)

Qualifiers (Meta data)

```
[Abstract, Description (
  "An abstraction or emulation of a hardware entity, that may "
  "or may not be Realized in physical hardware. ... ") ]
class CIM_LogicalDevice : CIM_LogicalElement
{
  . . .
  [Key, MaxLen (64), Description (
    "An address or other identifying information to uniquely "
    "name the LogicalDevice.") ]
  string DeviceID;
  [Description (
    "Boolean indicating that the Device can be power "
    "managed. ...") ]
  boolean PowerManagementSupported;
  [Description (
    "Requests that the LogicalDevice be enabled (\\"Enabled\\" "
    "input parameter = TRUE) or disabled (= FALSE). ...") ]
  uint32 EnableDevice([IN] boolean Enabled);
  . . .
};
```

Class Name and Inheritance

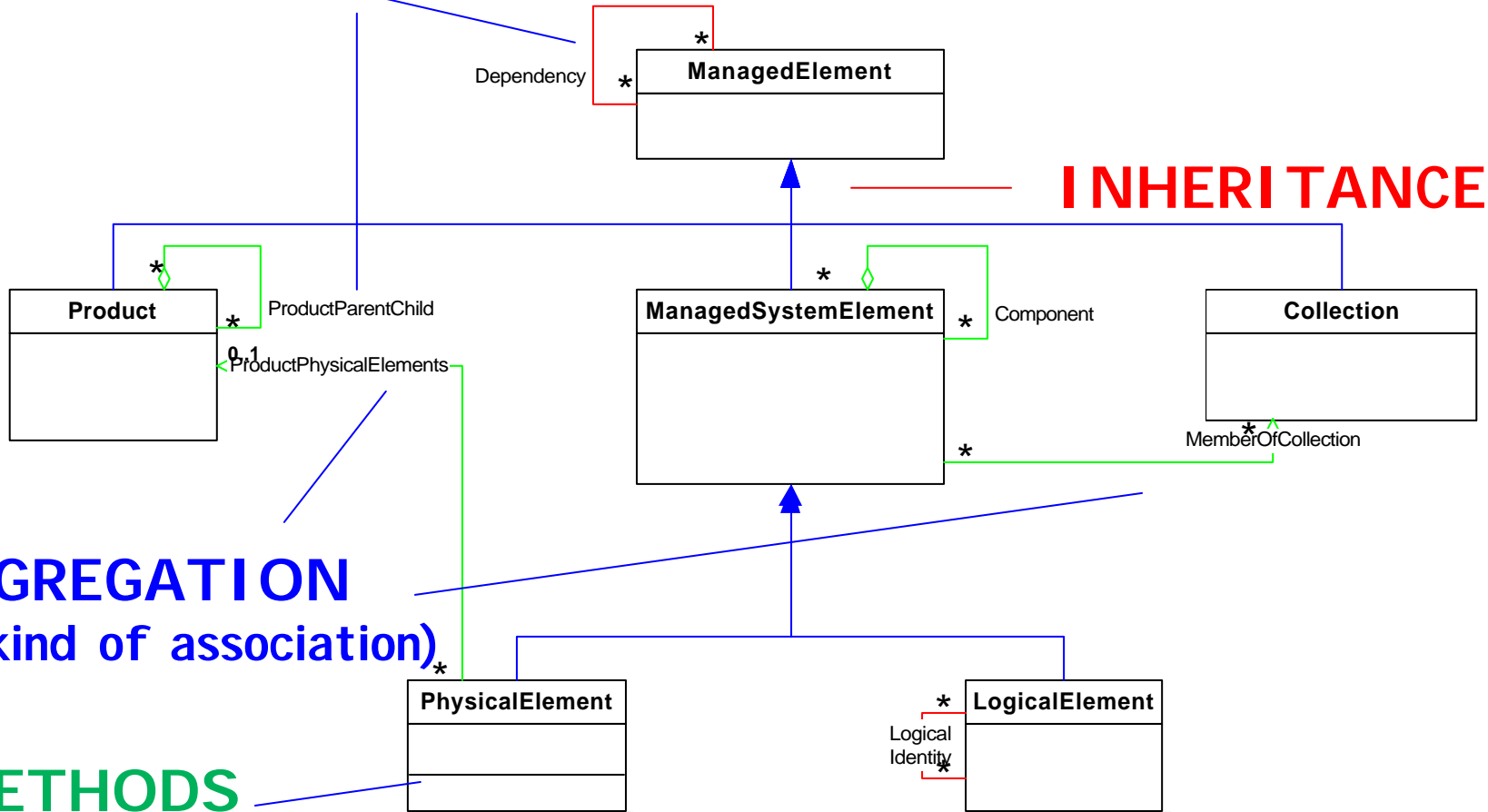
Properties

Methods

UML/Visio Example

THE *Open* GROUP

ASSOCIATIONS



XML Example

THE *Open* GROUP

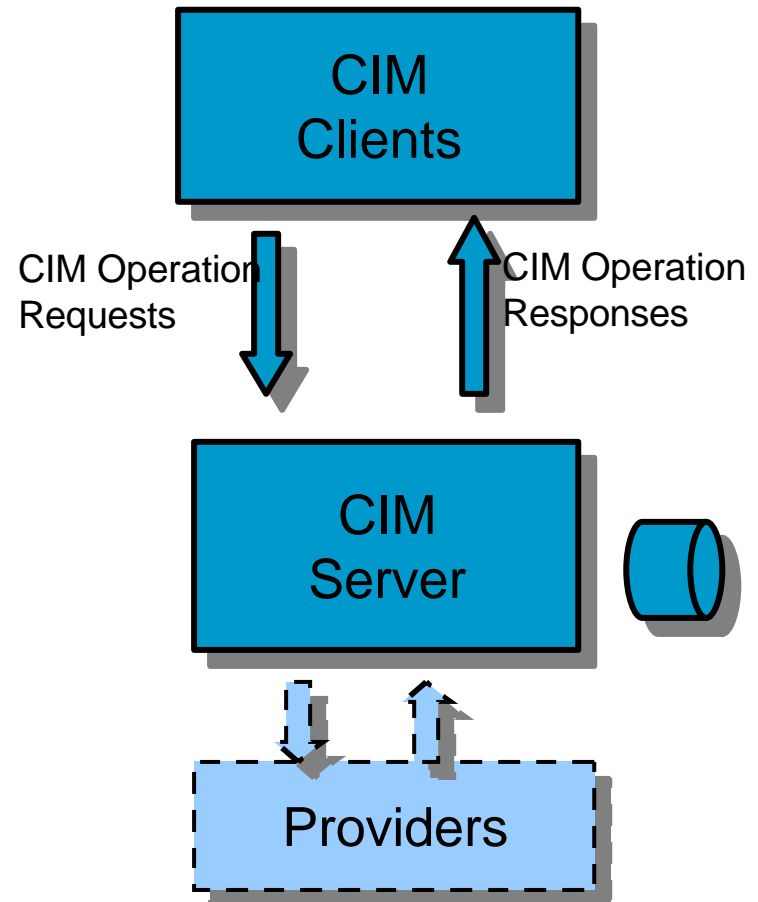
```
<?XML Version="1.0"?>
<!DOCTYPE CIM SYSTEM
"http://www.dmtf.org/cim-
v2.dtd/">
<CIM VERSION="2.0">
<CLASS
NAME="ManagedSystemElement">
  <QUALIFIER NAME="abstract">
  </QUALIFIER>
  <PROPERTY NAME="Caption"
    TYPE="string">
  <QUALIFIER NAME="MaxLen"
    TYPE="sint32">
  <VALUE>64</VALUE>
  </QUALIFIER>
  </PROPERTY>
  <PROPERTY NAME="Description"
    TYPE="string">
  </PROPERTY>
```

```
<PROPERTY NAME="InstallDate"
  TYPE="datetime">
<QUALIFIER NAME="MappingStrings"
  TYPE="string">
<VALUE>MIF.DMTF|ComponentID|001.5
</VALUE>
</QUALIFIER>
</PROPERTY>
<PROPERTY NAME="Status"
  TYPE="string">
<QUALIFIER NAME="Values"
  TYPE="string" ARRAY="TRUE">
<VALUE>OK</VALUE>
<VALUE>Error</VALUE>
<VALUE>Degraded</VALUE>
<VALUE>Unknown</VALUE>
</QUALIFIER>
</PROPERTY>
</CLASS>
</CIM >
```

CIM Operations

THE *Open* GROUP

- Intrinsic Operations
 - Create, Modify, Delete
 - Qualifiers
 - Classes
 - Instances
 - Properties
 - Invoke Query
- Extrinsic Operations
 - Execute Methods



```
<instance> GetInstance (  
  [IN] <instanceName> InstanceName,  
  [IN,OPTIONAL] boolean LocalOnly = true,  
  [IN,OPTIONAL] boolean IncludeQualifiers = false,  
  [IN,OPTIONAL] boolean IncludeClassOrigin = false,  
  [IN,OPTIONAL,NULL] string PropertyList [] = NULL
```

CIM Indications (Events)

THE *Open* GROUP

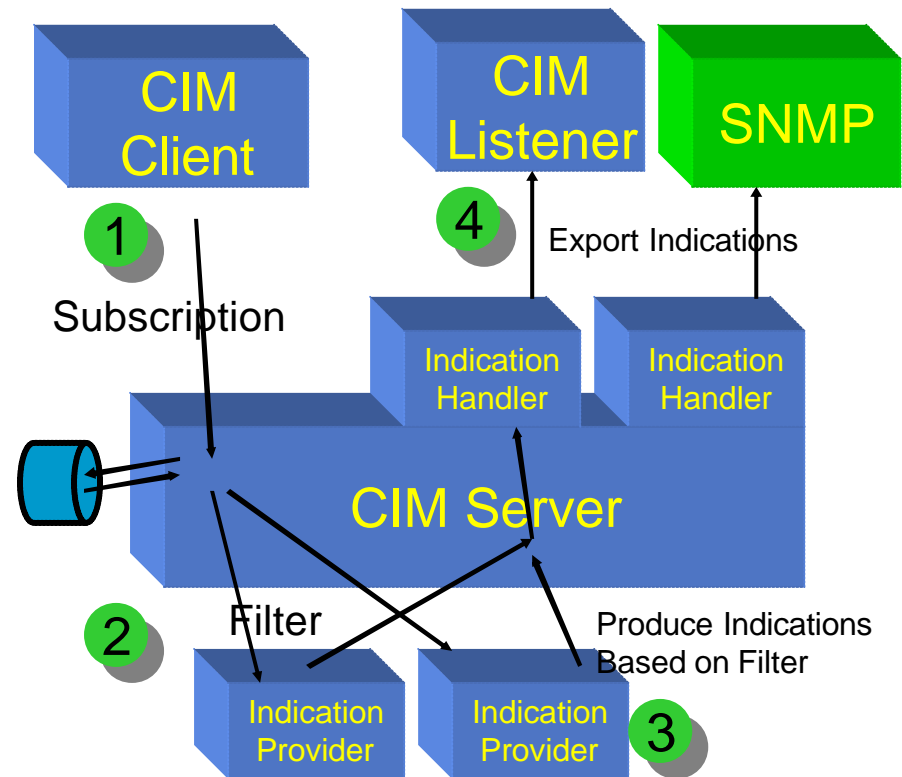
Indication Types

- Process Indications
 - External Events
- Lifecycle indications
 - Changes to CIM Objects

Indication
Subscription
Object

CIM Indication
Handler Object

CIM Filter
Object



Query Language - WQL

THE *Open* GROUP

- ❑ SQL based Query Language
- ❑ Used in:
 - CIM Indication Filters
 - CIM Queries (CIM Operation)
- ❑ 4 levels of Language
 - 1 – Simple queries from single tables
 - 2 – Adds DELETE INSERT UPDATE
 - 3 – Add complex expressions and specific data constructs
 - 4 Add Joins, etc.

Subset of SQL supports queries in the form
SELECT <row set>
FROM <table list>
WHERE <selection expression>



Industry Work

THE *Open* GROUP

- ❑ Microsoft's WMI (Windows Mgmt Instrumentation)
- ❑ Sun's WBEM Services
- ❑ Cisco's CiscoWorks2000 and LDAP mapping
- ❑ Coordinated development with IETF
 - Policy Framework
 - IPsec Policy
- ❑ The Open Group
 - Unix and software modeling
 - Pegasus (Manageability Services Broker) open source



Industry Work

THE *Open* GROUP

- ❑ SNIA (Storage Networking Industry Association)
 - Fibre Channel and Tape / Storage Library modeling and media management
 - Disk Resource Management - CIM prototype
 - CIM Object Manager open source (Now Open Group)
- ❑ WBEMSource (open source) initiative
- ❑ Applications and instrumentation for the various “object managers”



2. Pegasus Overview



What is Pegasus?

THE *Open* GROUP

- ❑ Open-source implementation of the DMTF WBEM specifications
- ❑ Work project of the Open Group Enterprise Management Forum
 - Pegasus is primarily volunteer labor but through company commitments, not just individuals
- ❑ Platform for building application management
- ❑ Pegasus is a function-rich, production-quality open-source implementation designed to be used in high volume server implementations.



Why Produce Pegasus?

THE *Open* GROUP

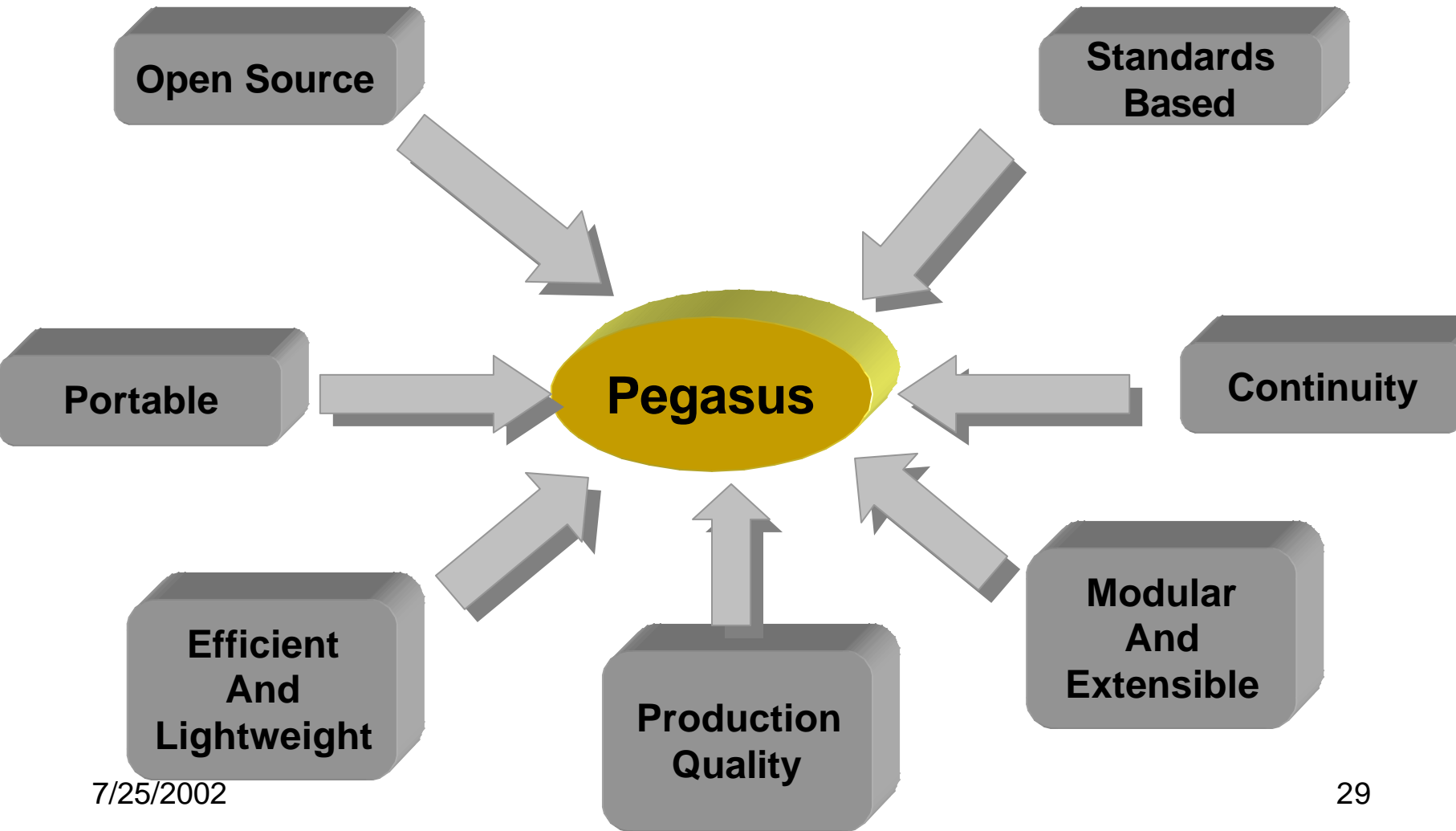
- ❑ Demonstrate manageability concepts.
- ❑ Provide additional standards for WBEM
- ❑ Provide a working implementation of WBEM technologies
- ❑ Provide an effective modular implementation
- ❑ Support other Open Group manageability standards
- ❑ Base Platform for Open Group Application management Projects
- ❑ **Help make WBEM real**

Pegasus was Initiated in 2000 by the Open Group in collaboration with:

- BMC Software
- IBM
- Tivoli Systems

Key Pegasus Objectives

THE *Open* GROUP





Pegasus Working Group Philosophy

THE *Open* GROUP

- Manageability not management
 - The Pegasus working group's objective is not to manage systems but to make them **manageable** by promoting a standard instrumentation environment
 - The actual **management** of systems is left to systems management vendors

- No standards without implementation
 - The process of implementation provides a rigorous process for testing the validity of standards
 - Therefore all standards must be validated by implementation

Open Source

THE *Open* GROUP

- ❑ Code and documentation freely available
 - Open Group and Source Forge
- ❑ MIT source license
 - Minimal Demands on licensee
 - No GPL demands
 - License used for X-Windows
- ❑ Open to contributions
 - But process controlled to keep strategy
- ❑ No commitment to Open Group to
 - Use code
 - Contribute code
- ❑ Royalty Free

But in a controlled environment



Portable

THE *Open* GROUP

- ❑ Designed for multi-platform, multi-OS, multi-compiler implementation
- ❑ Platforms ported today
 - UNIX (AIX, HPUX, Solaris, Tru-Unix)
 - Linux
 - Windows Platforms (NT, 2000, 9x)
 - Compaq Himalaya (Tandem)
 - OS 400*
 - Z/OS*
- ❑ MultiPlatform Build
 - Only Make and compiler required to build
- ❑ Multiple Compilers on most platforms
 - GNU compilers preferred



Efficient and Lightweight

THE *Open* GROUP

- ❑ Core written in C++
- ❑ Designed for execution efficiency
 - Custom xml parser
 - No STL
 - Minimal Templates, etc.
- ❑ Designed to be production-quality solution
- ❑ Message/Queue based multithreaded Architecture



Standards Based

THE *Open* GROUP

- ❑ Based on DMTF CIM and CIM-XML specifications
- ❑ Open Group is active partner in DMTF
- ❑ Growth through participation in specification growth
- ❑ Commitment to continue DMTF compliance



Modular, Extensible, Flexible

THE *Open* GROUP

- ❑ Minimize core object broker.
- ❑ Maximize extensibility through plug-in components
- ❑ Component types
 - Providers
 - Standard Provider interfaces
 - Clients – Connection local and cim-xml
 - Repositories (additional repository handlers)
 - Manageability service extensions
 - Protocol Adapters
 - Modules (extend and modify core functions)

Modularity is key to doing parallel development and to extensibility



Project for Continued Development

THE *Open* GROUP

- ❑ WBEM will continue to develop functionality and standards
- ❑ Open Group will develop application management partly around Pegasus
- ❑ Pegasus is a project, not just a technology
- ❑ Pegasus Development will continue beyond current versions
 - Integrate contributions
 - Add basic new functionality
 - Grow with DMTF/Open Group standards
- ❑ Light Weight Versions are part of our vision



The Open Group

THE *Open* GROUP

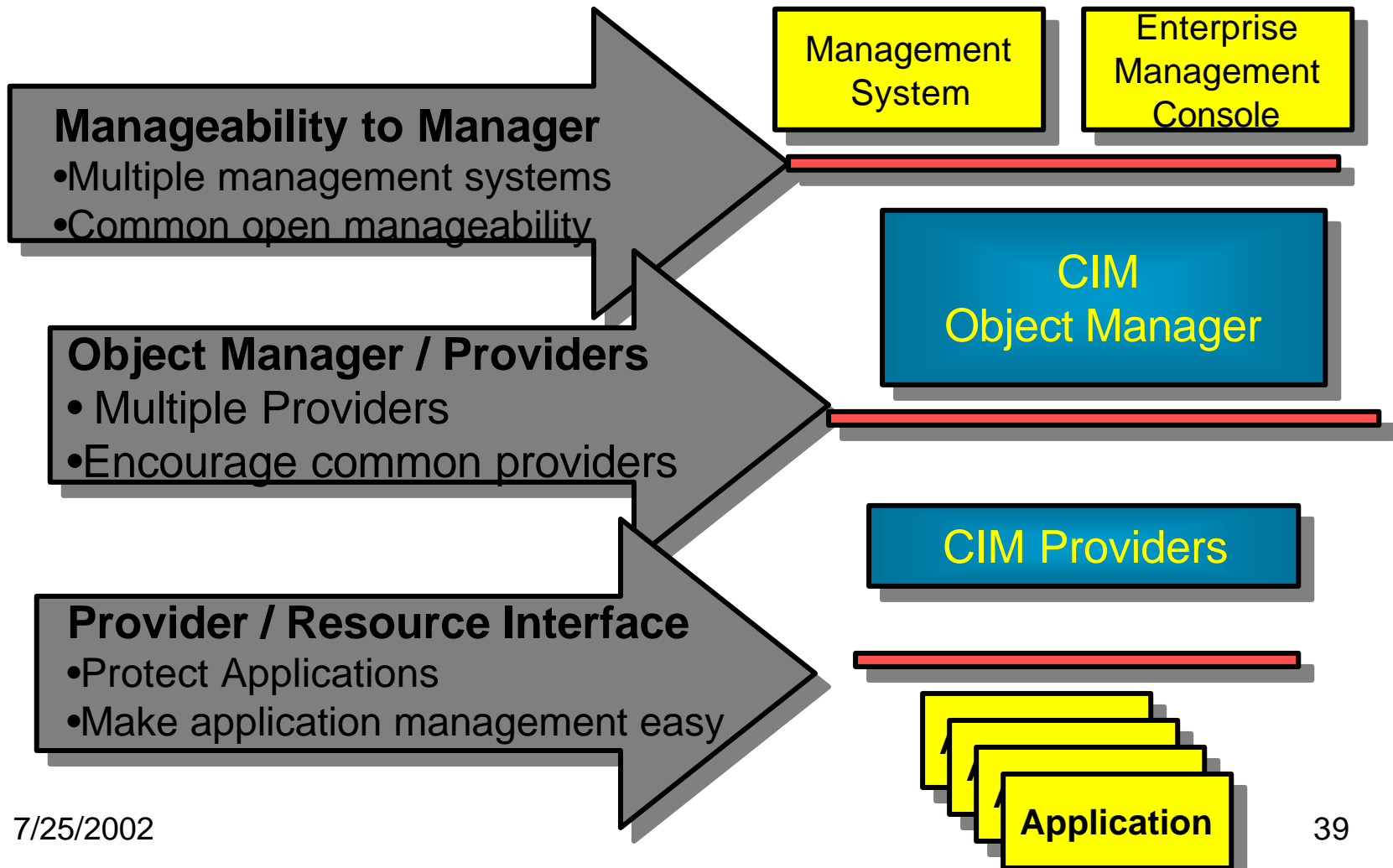
- Enterprise Management Forum
 - Projects include:
 - Pegasus
 - Ex-SNIA Java Based open-source Implementation
 - Application Management APIs
 - ARM
 - AIC
 - Software Licensing Standards
- QoS Task Force
 - User/Supplier interaction forum to advance QoS and Service Level Standards
 - Working with DMTF, IETF, TMN



3. The Pegasus Environment

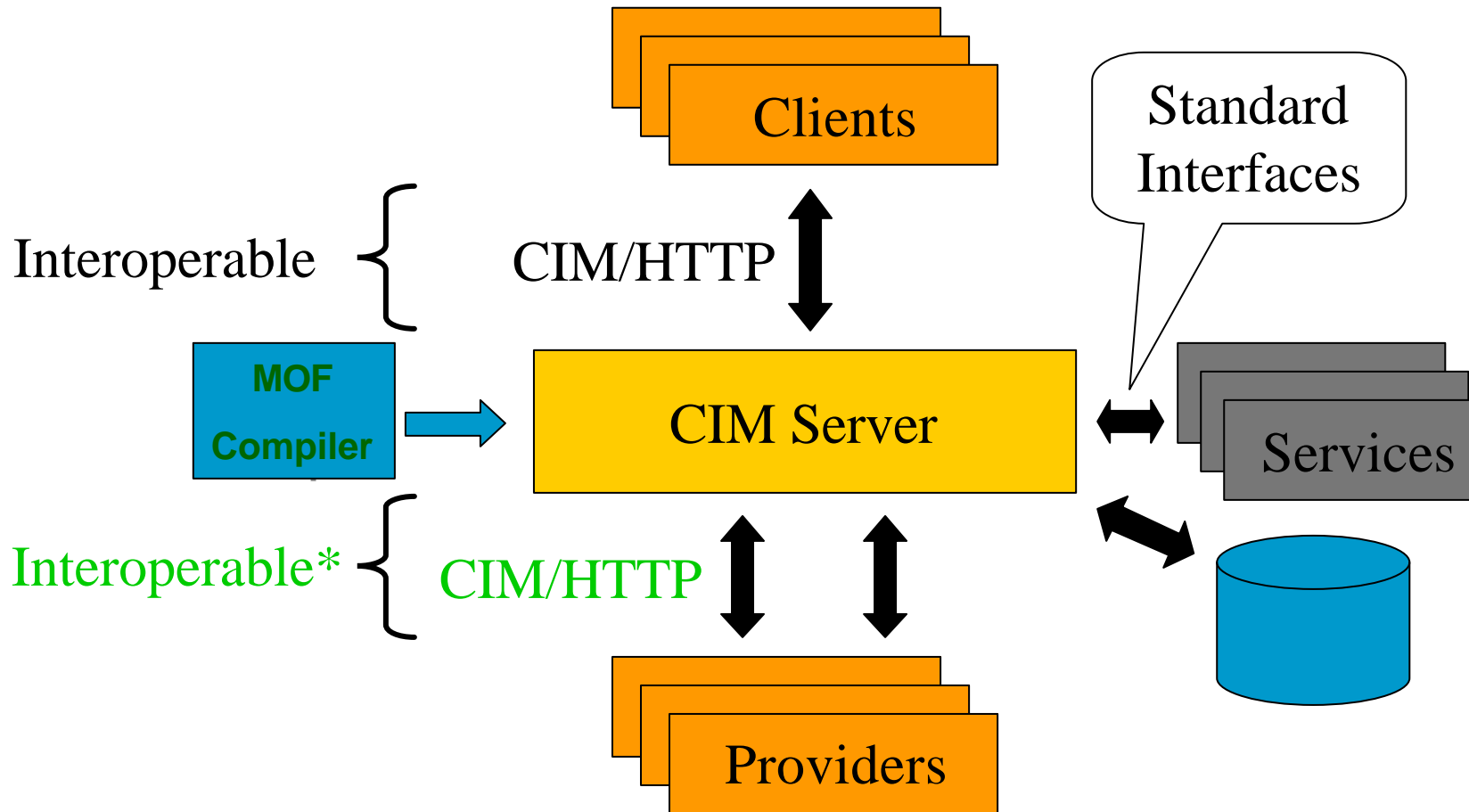
Key Interoperability Interfaces

THE *Open* GROUP



Pegasus Architecture

THE *Open* GROUP



CIMOM Capabilities

THE *Open* GROUP

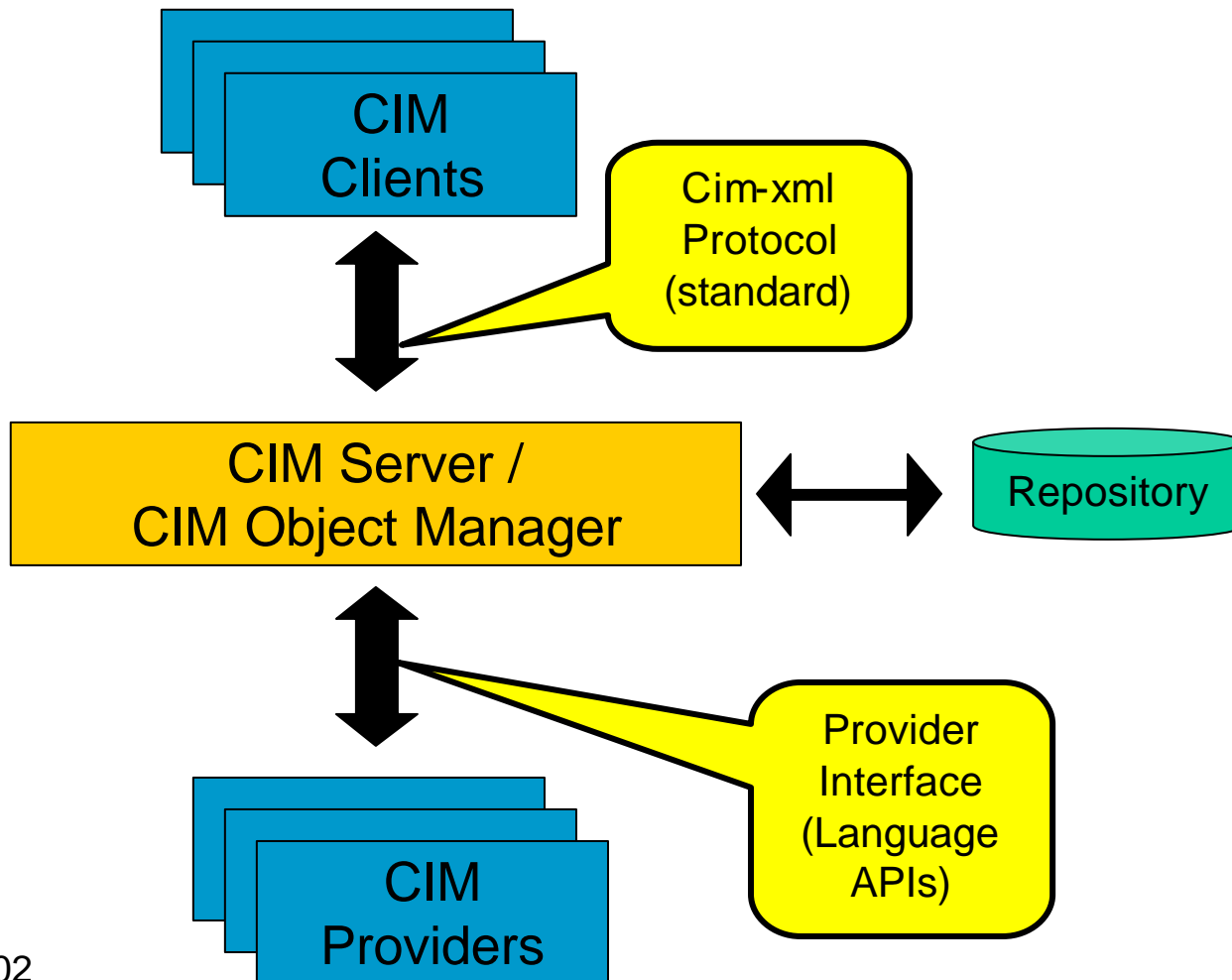
- ❑ Respond to Operations defined in “CIM Operations” spec.
 - Create, Modify, Delete operations on
 - **Class, Instance, Property, Qualifier**
- ❑ Manage Providers
 - Install, register, load
 - Interface for Operations and Indications
- ❑ Forward Requests to Providers, repositories, etc.
- ❑ Read/Write access to Management Information
 - Maintain Class/Instance Information
- ❑ Traversal of Associations
- ❑ Use of WBEM Query Language
- ❑ Syntax/Semantic checking (with Qualifiers)



4. The Pegasus Software Architecture

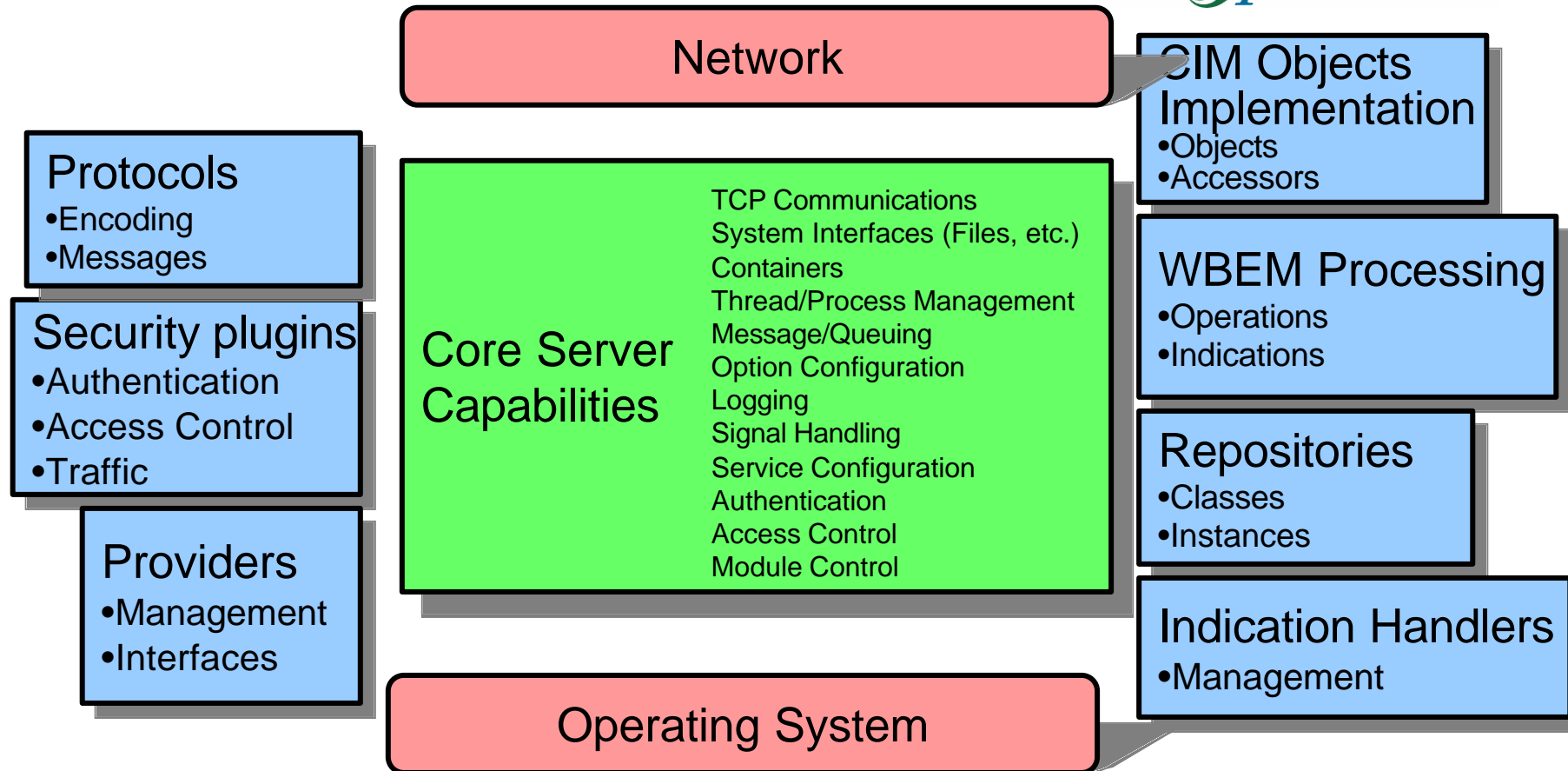
Major CIM/WBEM Components

THE *Open* GROUP



CIM Object Manager Capabilities

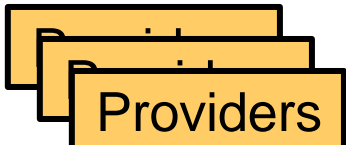
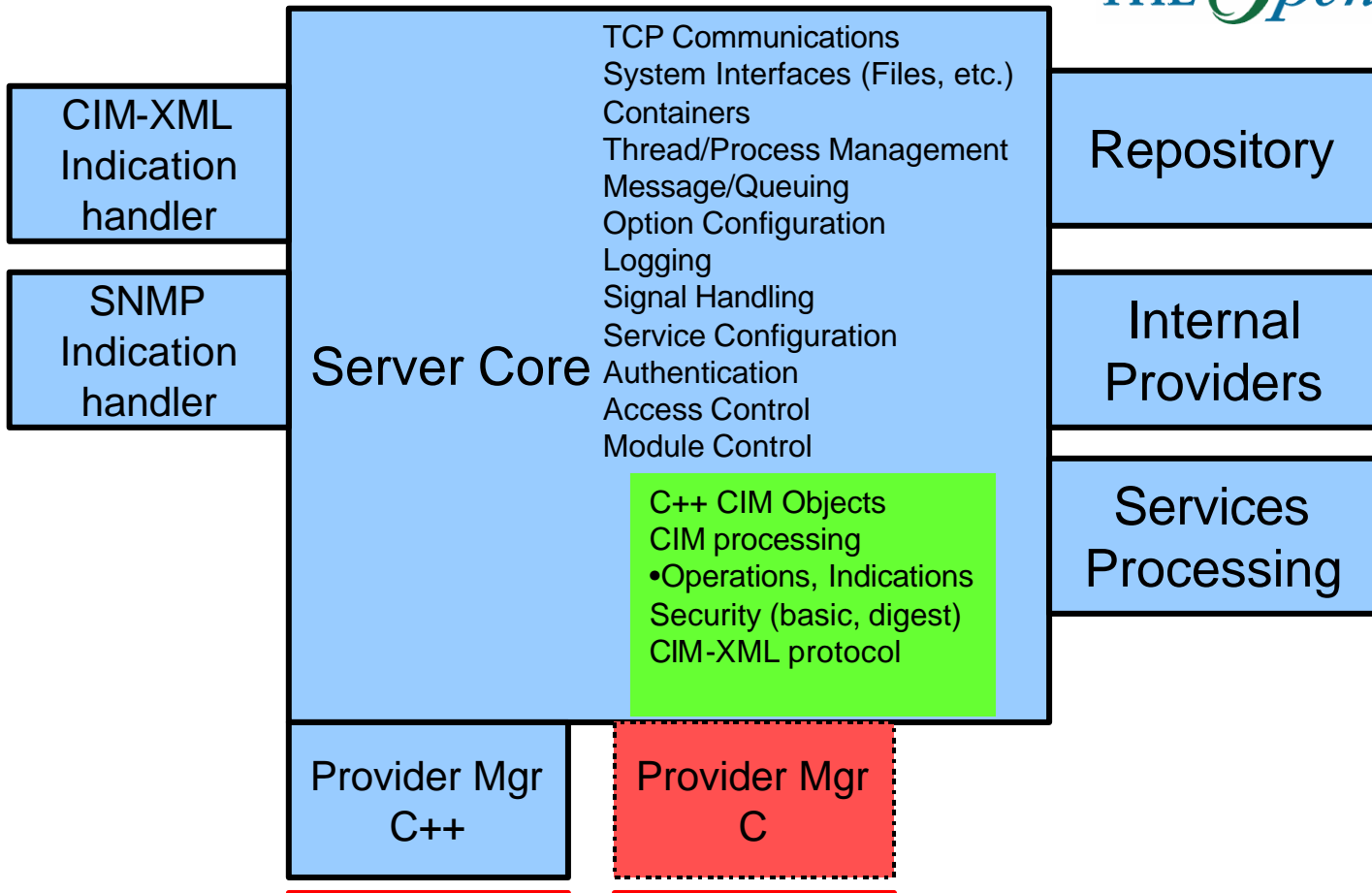
THE *Open* GROUP





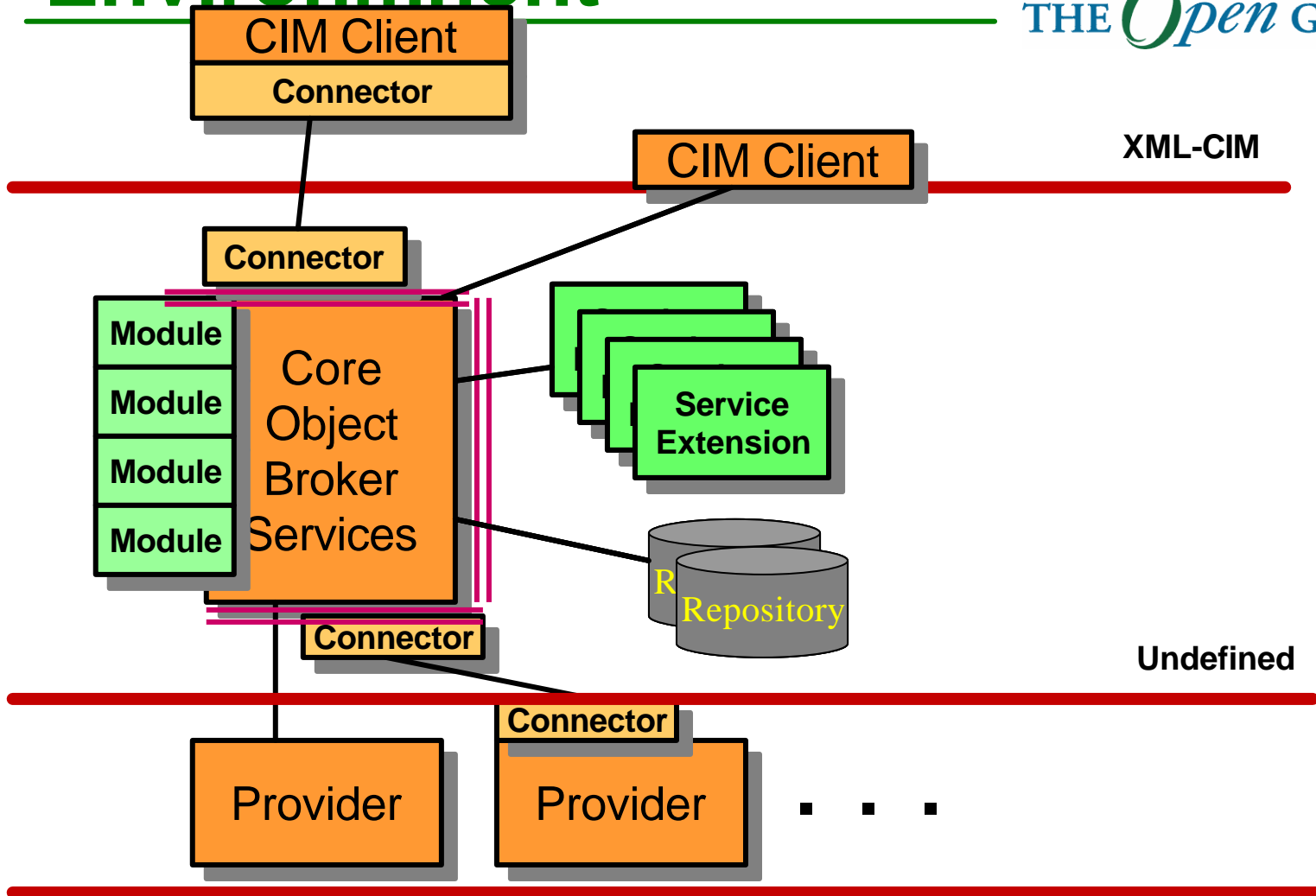
Pegasus Server Today Components

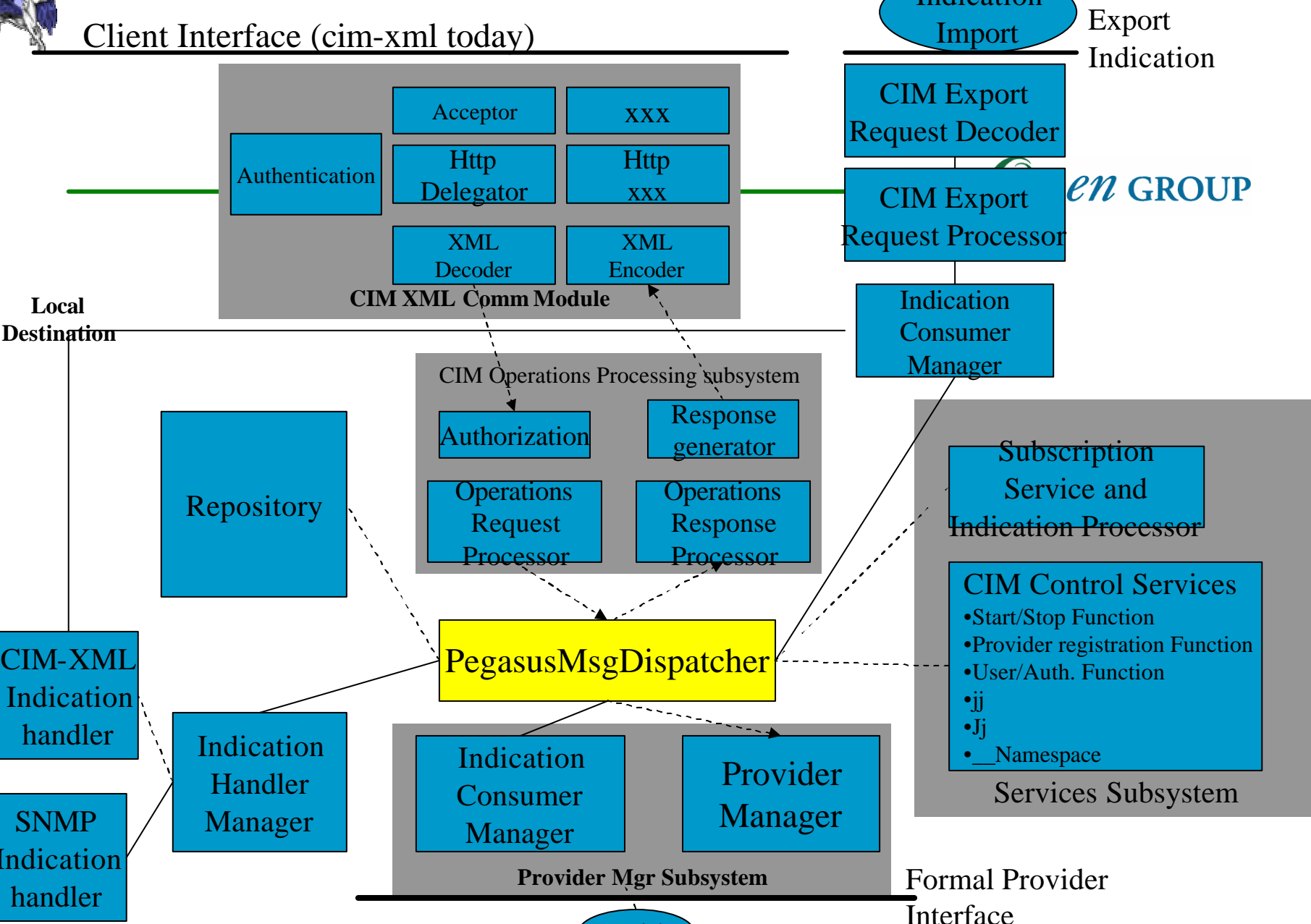
THE *Open* GROUP



Building A Modular Manageability Environment

THE *Open* GROUP





5. The Pegasus Project Today And Tomorrow

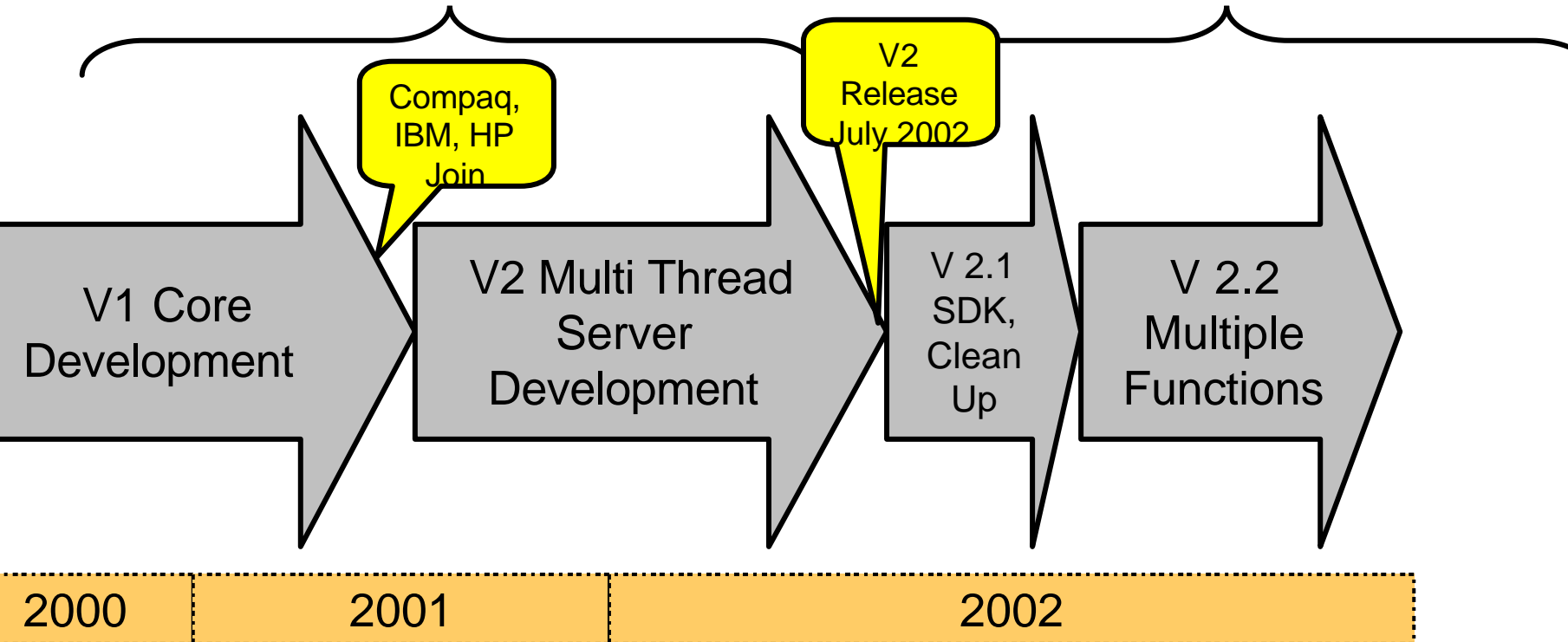


Pegasus Project In Context

THE *Open* GROUP

Core System Development

Incremental Function growth



The Pegasus Functions Today

THE *Open* GROUP

- CIMOM Functions
 - Class and Instance
 - Security
 - Threading
 - Queued execution model
 - Async CIM Operations APIs
 - Modular Services
 - C++ Provider Manager
 - Provider Control
- WBEM Functions
 - Provider Registration
 - CIM Operations
 - Indications (Process)
- Clients
 - Test Clients
- Providers
 - In-Process
 - Test
 - Sample Implementations
 - OS
 - ComputerSystem
- Platforms
 - Unix (AIX, HP-UX, Tru-UNIX)
 - Compaq Himyla
 - Linux
 - Windows



Planned Extensions

- CIMOM Functions
 - Lifecycle Indications
 - Increased Modularity
 - Enhance Service Configuration
 - Additional Provider Managers
 - C Provider Interface
 - Java Provider Interface
 - Out-of-Process Providers
 - External Providers
- WBEM Functions
 - Discovery
 - CIMOM Object Manager Mgt. (Interop Schema)
 - Chunking
- Specifications
 - C Provider Interface Standard (Sept 02)
 - C++ Provider Interface standard (Oct 02)
- Security (pluggable and extra security)
- WMI Mapping
- Clients
 - Object Browser
 - Test Clients
 - Real SDK
- Protocol Adapters
 - Add protocol Adapters
- Providers
 - Test Providers
 - Extend Generic Providers
 - Create real SDK
- Platforms
 - Easier portability
 - More platforms
- Testing
 - Extend Unit and System tests



A Pegasus Lite Project

THE *Open* GROUP

- ❑ Two Solutions
 - Lighten existing Pegasus (6 mm ext) – 1 – 2 MB
 - Lightweight core/model (15 mm est) – 300-500 KB
- ❑ Today requirements vary widely
 - Footprint
 - System resources
 - Functionality
- ❑ Either or both doable but requires commitment
 - Membership in Pegasus community
 - Better definition of requirements
 - Possibly specific extension projects
- ❑ This will be incremental
 - Both solutions will probably exist with later merge.



Portability of Pegasus

THE *Open* GROUP

- ❑ Any GNU based environment is easy
- ❑ Minimal tools required
 - C++ compiler
 - GNU Make
 - Project Provides one make portability tool
- ❑ Compiler issues
 - Minor compiler differences
 - Exceptions, templates
- ❑ Environment issues
 - Threads support
- ❑ Most ports – one week or less



Planned Standards

- C++ Provider API
 - Pegasus Internal Review of all APIs in process
 - Release SDK with Pegasus 2.1 Sept 2002
 - Review as Standard Q4 2002
- C Provider API
 - Draft available today for review
 - Formal review cycle (~September 2002)
- C++ Client API
 - In parallel with C++ Provider API
 - Need not clear today (Java, etc. for Clients)
 - Except for provider/clients.



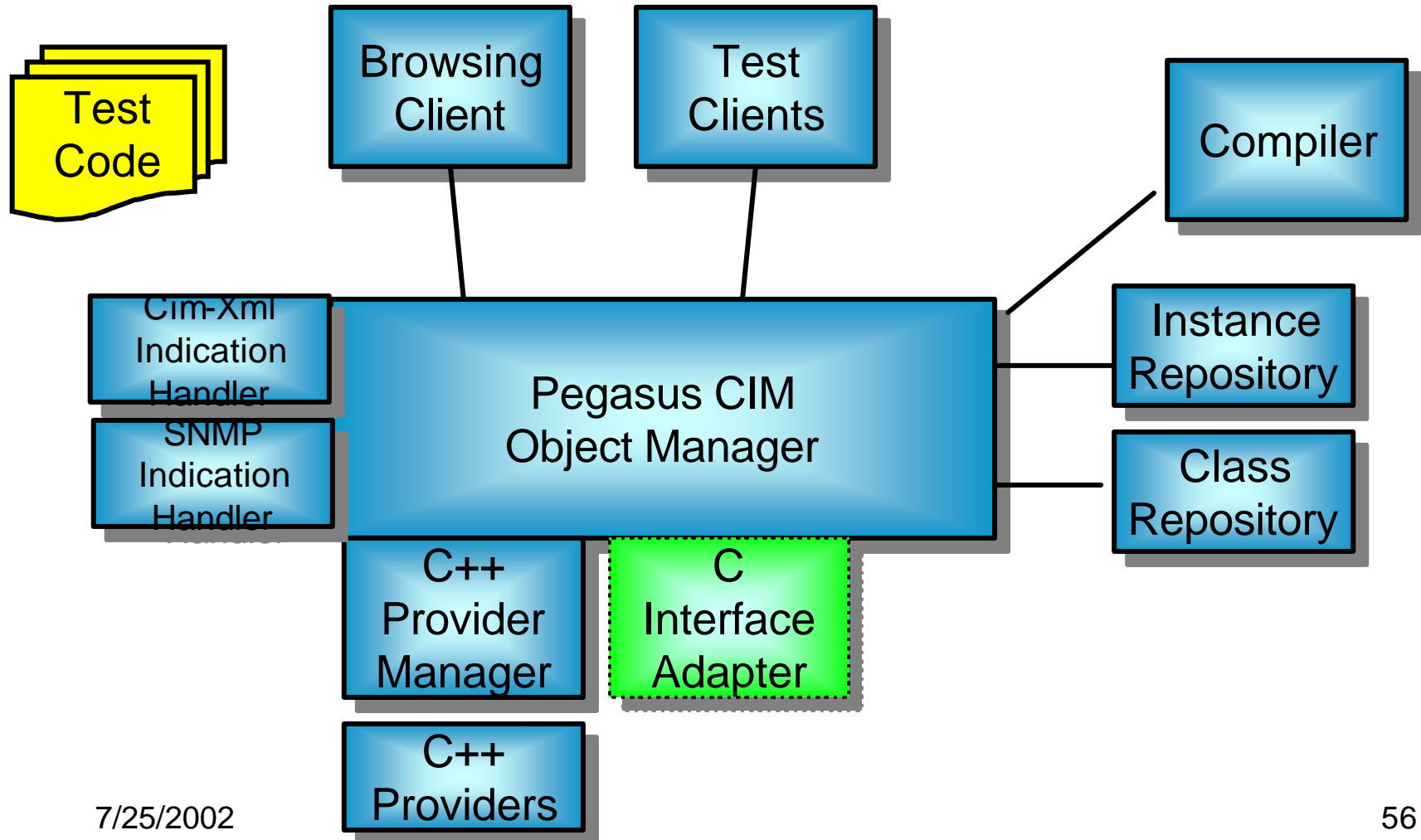
Pegasus State Today

THE *Open* GROUP

- ❑ 120,000 Lines of executable code
- ❑ Ported to at least 5 platforms
- ❑ 15 – 20 man years effort to date
- ❑ Components already used in multiple applications
- ❑ Major usage commitment from HP and IBM
- ❑ First “integrated” use this year

The Components Today

THE *Open* GROUP





Pegasus Who's Who

THE *Open* GROUP

□ The Major Pegasus Contributors

- IBM
- Compaq
- HP
- BMC

□ The Pegasus Users Today

- IBM
- Compaq
- HP
- BMC
- ...

IBM Commitment

IBM has chosen CIM/Wbem as the strategic technology for hardware platform instrumentation and manageability of all server platforms.

Pegasus is the CIMOM that IBM has selected to include with eServer platforms in 2002 and beyond.

During 2002 IBM is planning on maintaining and contributing to Pegasus for the Linux, Windows, AIX, OS/400 and System 390 operating environments.

Managing The Pegasus Project

THE *Open* GROUP

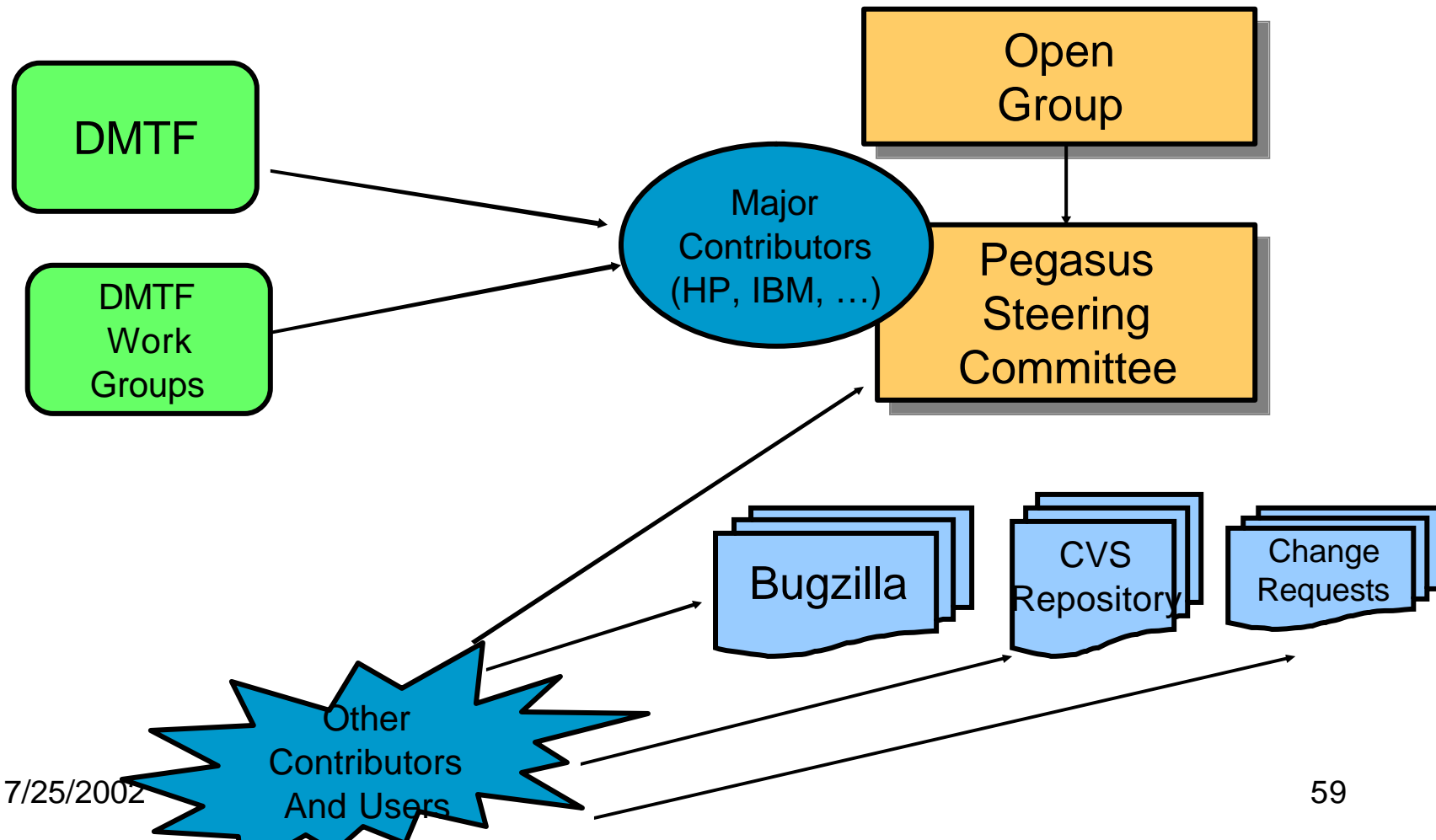
- ❑ Open Group
 - Sponsors the Project, Web sites, CVS, etc.
 - Creates standards for APIs
 - Environment for user understanding
- ❑ Steering Committee
 - Sets strategy, priorities, release commitments
- ❑ Contributors
 - Contribute via change requests, code , doc.
- ❑ Users
 - Take documentation, source, binary
- ❑ Project Manager
- ❑ DMTF / DMTF Work Groups
 - Standards, Interoperability

Management
Tools

- Web Site
- Teleconferences
- Group Mail
- Change Requests
- CVS

Managing The Pegasus Project

THE *Open* GROUP





Working With the Pegasus Project

THE *Open* GROUP

- ❑ Using the Code
 - Free for use
- ❑ Contributing to the Project
 - Outside contributors
 - In Company
 - Specific financed projects
 - Contribute via patches or authorized developers
 - Join the Steering Committee
 - Influences priorities, commitments, releases.

We want and NEED participation from users and Potential users



Open Group

THE *Open* GROUP

- ❑ The Enterprise Management Forum
 - <http://www.opengroup.org/management>
- ❑ The Pegasus Web Site and code
 - <http://www.opengroup.org/pegasus>
- ❑ The ex-SNIA web site and code
 - <http://www.opengroup.org/snias>



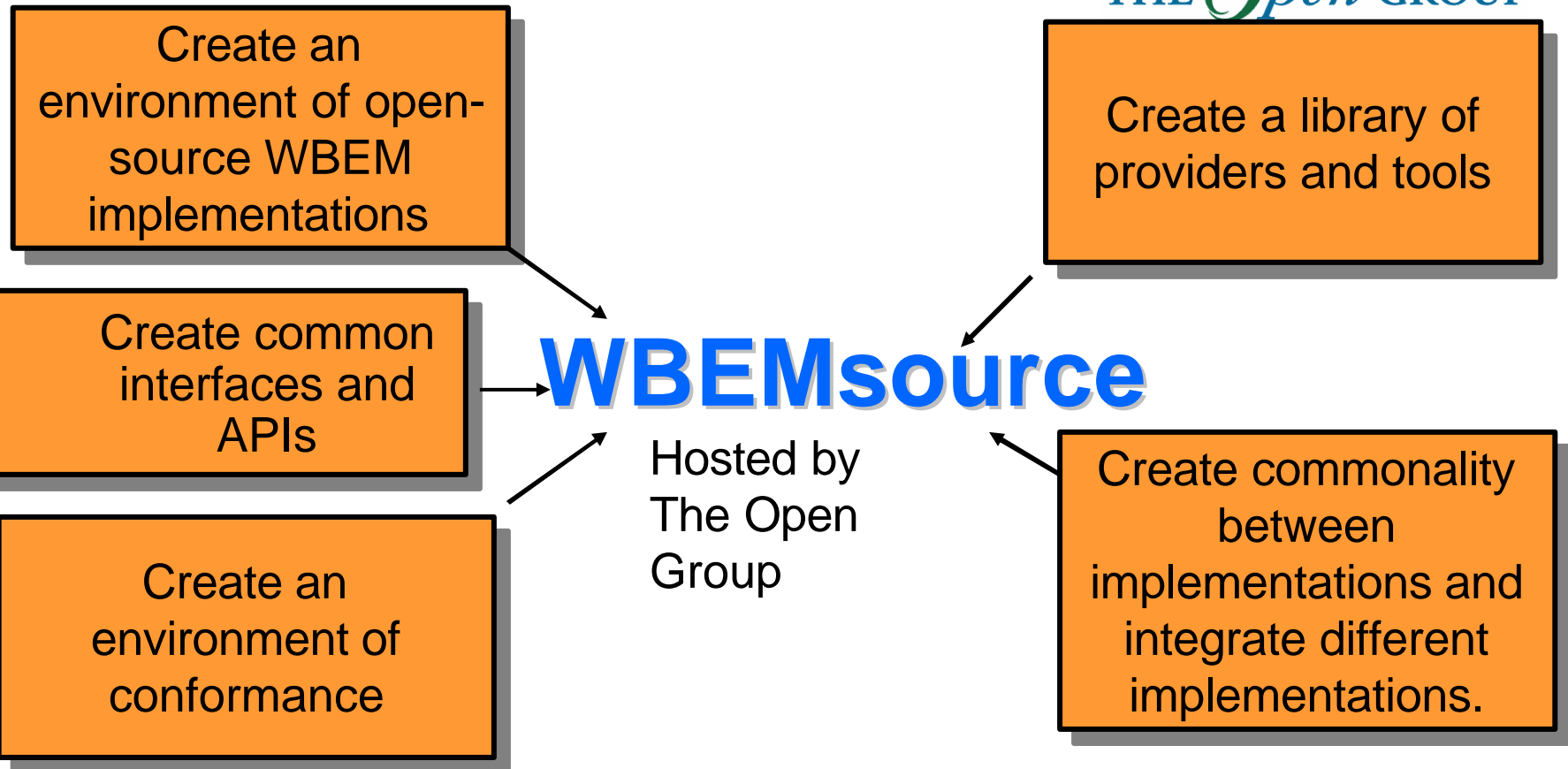
Other Manageability Projects

THE *Open* GROUP

- ❑ AIC – Application and Control
 - AIC as a Pegasus Provider
- ❑ ARM – Applications Response Measurement
 - ARM and DMTF DAP Information as Pegasus Provider
- ❑ Other possible Providers
 - JMX (Java)
 - SNMP
 - DMI
 - WMI

WBEMSource Consortium

THE *Open* GROUP



Members Include:

Sun, Compaq, HP, IBM, Tivoli, Open Group, SNIA, Caldera, Novell, Nokia, Intel



6. QUESTIONS?