November 14-17, 2011

MANAGEMENT DEVELOPERS CONFERENCE

San Mateo Marriott
San Mateo, CA

Open Pegasus
Part 1 – Overview and Update

The OpenPegasus Architecture Team
pegasus-architecture@opengroup.org

Karl Schopmeyer
Project Coordinator, Pegasus Open Source Project
k.schopmeyer@opengroup.org

This presentation will be available On the MDC and OpenPegasus websites

V 1.6 11/17/11 – Final Version

THE Open GROUP
Making standards work®

Agenda

- **Part 1**
  - 1. What is OpenPegasus?
  - 2. What's New?
  - 3. Pegasus Features Overview
  - 4. Technical Subjects
  - 5. How to use and work with Pegasus
  - 6. Issues
  - 7. Discussion and Feedback
- **Part 2 –Advanced Topics**
  - The Pull Operations
  - CIM_Error
  - Registering Pegasus Providers
  - Debugging in the Pegasus Environment

Rumors

30 Second Summary

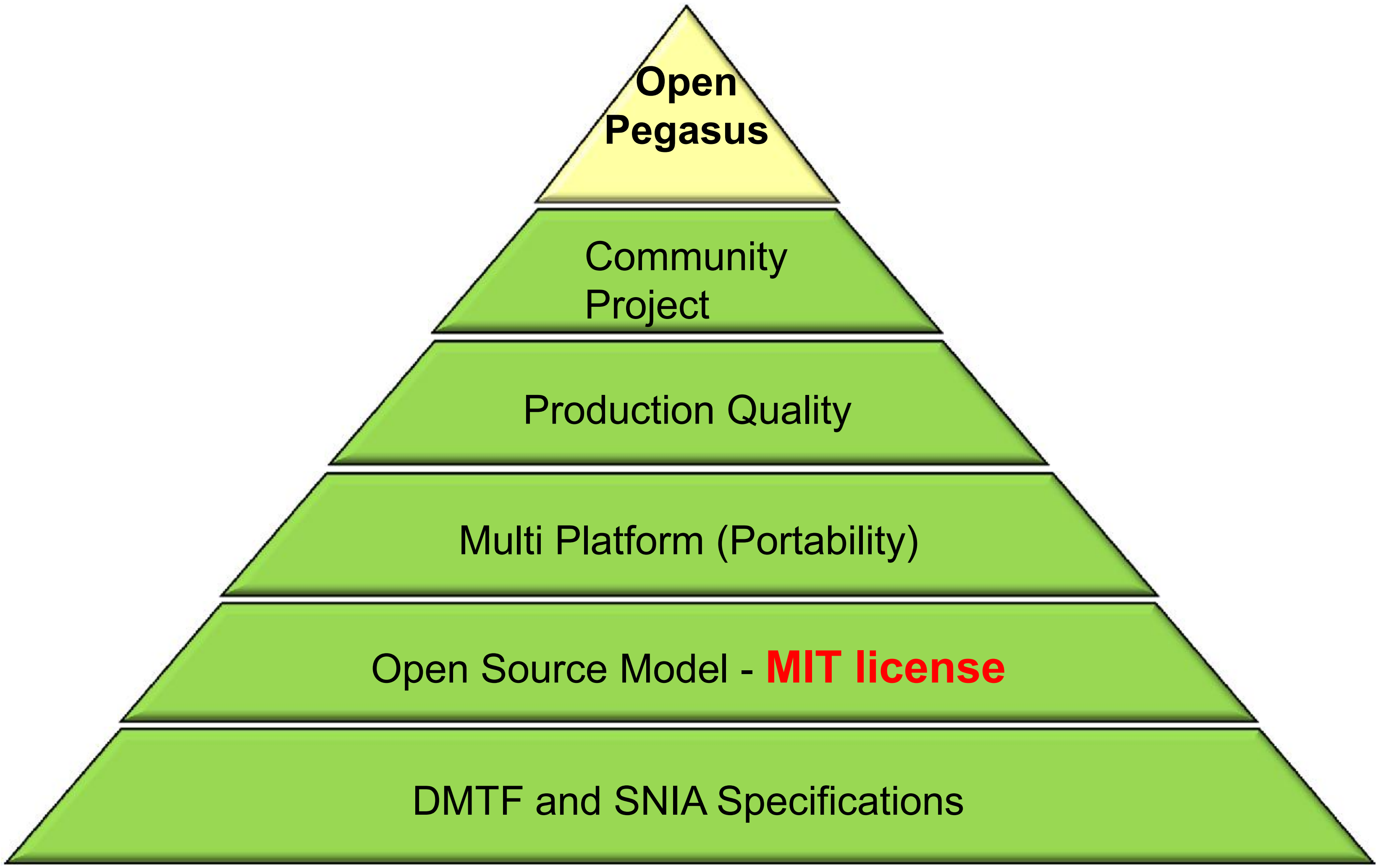
- OpenPegasus community active
- One Release and point release this year
- Probably Release and Major release next year
- Continuing to match DMTF specs with minimal exceptions and actively working with DMTF and SNIA
- Multiple implementations for both large scale and small scale systems.
- Community becoming less formal but more productive and with better quality output

Section 1.1

QUICK OPENPEGASUS OVERVIEW

- Goals
- The Project
- Architectural Overview

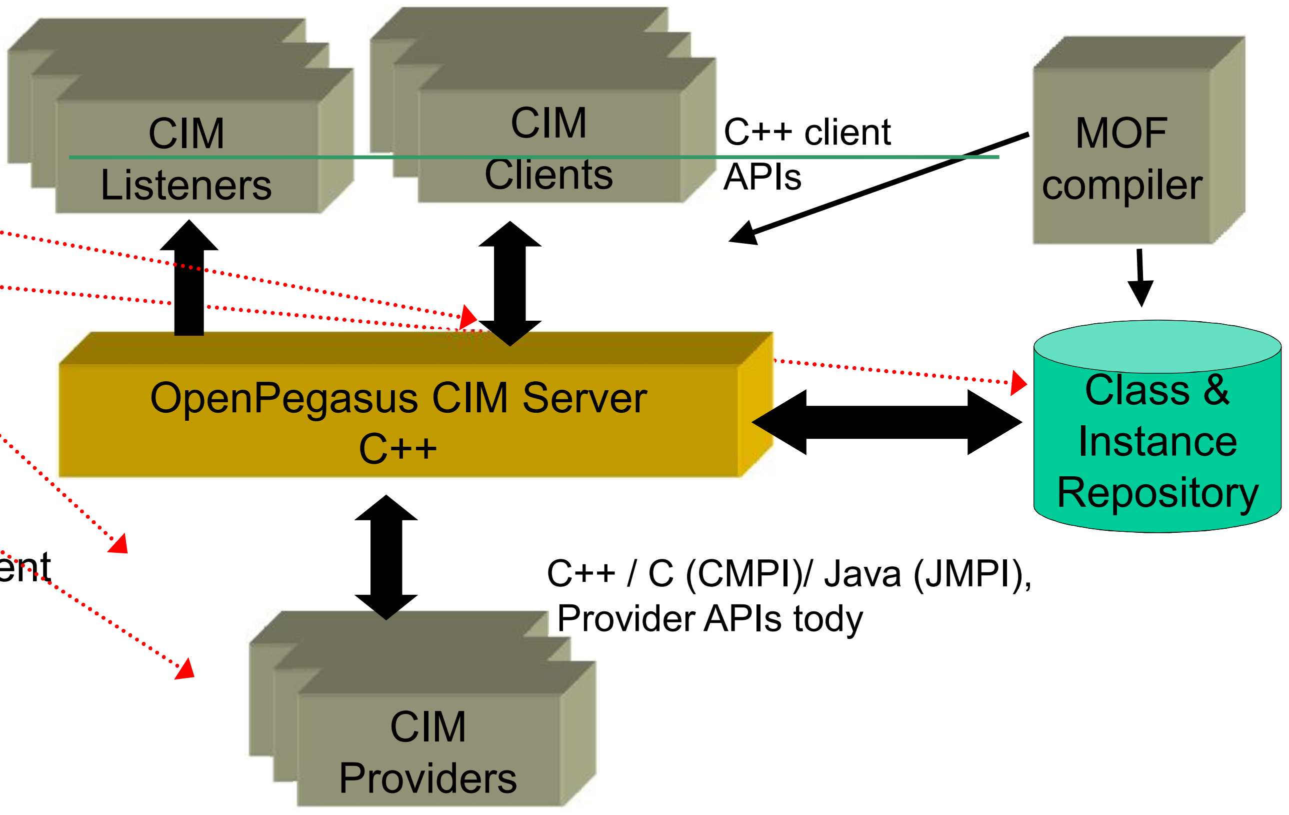
Our Objectives


Open Pegasus
Community Project
Production Quality
Multi Platform (Portability)
Open Source Model - **MIT license**
DMTF and SNIA Specifications

MANAGEMENT DEVELOPERS CONFERENCE

OpenPegasus Architecture

- **OpenPegasus Components**
  - CIM Client and Listener infrastructure
  - CIMServer
  - Server Repository
  - CIM Provider Interfaces
  - CIM Providers
  - MOF Compiler
  - Build and Test Environment



CIM Listeners
CIM Clients
C++ client APIs
MOF compiler
OpenPegasus CIM Server C++
Class & Instance Repository
C++ / C (CMPI)/ Java (JMPI), Provider APIs tody
CIM Providers

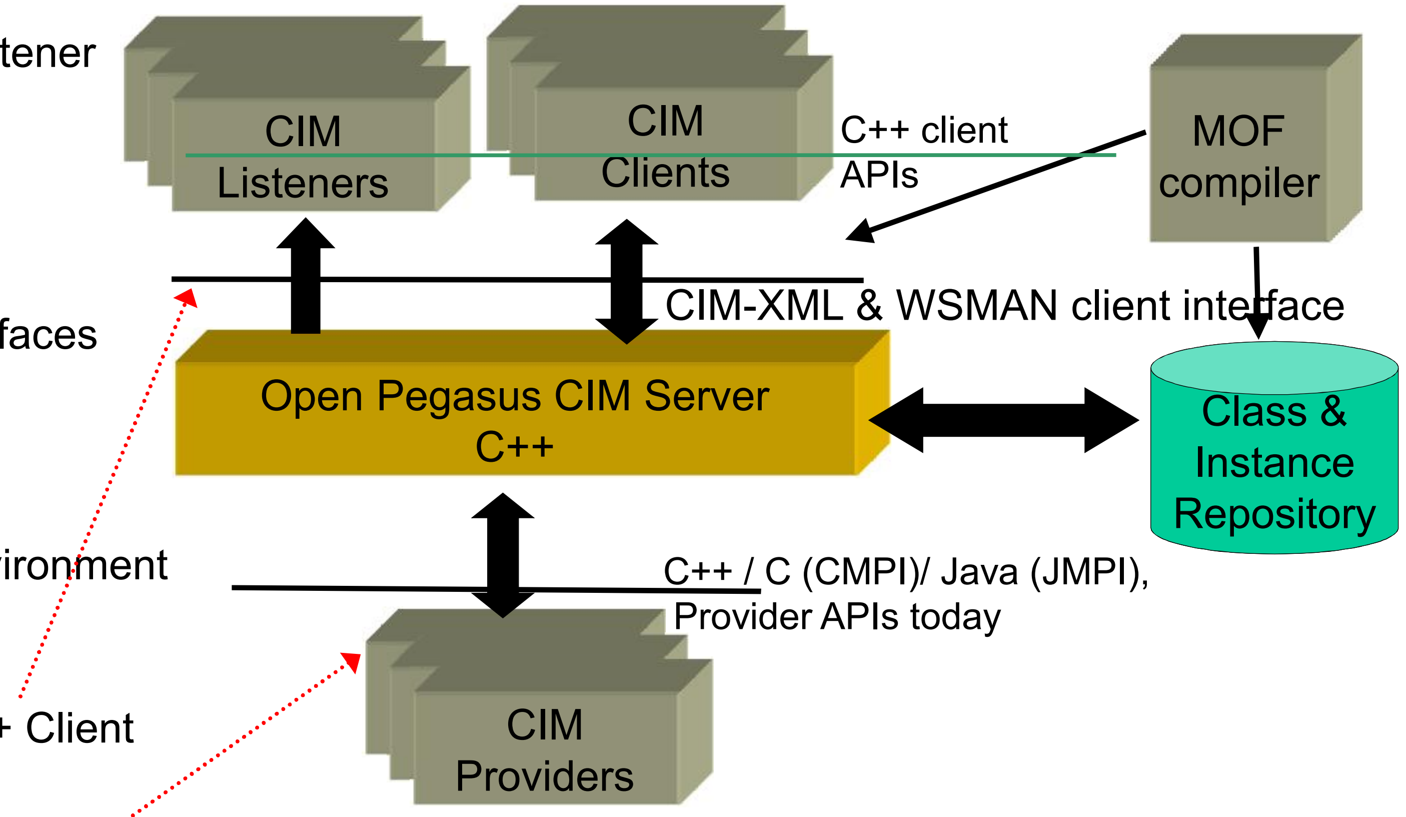
OpenPegasus Architecture

•OpenPegasus Components

- •CIM Client and Listener infrastructure
- •CIMServer
- •Server Repository
- •CIM Provider Interfaces
- •CIM Providers
- •MOF Compiler
- •Build and Test Environment

•Public Interfaces

- •OpenPegasus C++ Client Interface
- •CMPI & C++ Provider Interface
- •SNIA Java Client Interface


CIM Listeners
CIM Clients
C++ client APIs
MOF compiler
CIM-XML & WSMAN client interface
Open Pegasus CIM Server C++
Class & Instance Repository
C++ / C (CMPI)/ Java (JMPI), Provider APIs today
CIM Providers

Specifications and OpenPegasus

- Goals
  - Conform to DMTF and SNIA specifications
  - Limit noncompliant functionality
  - Work with DMTF and SNIA to grow specifications

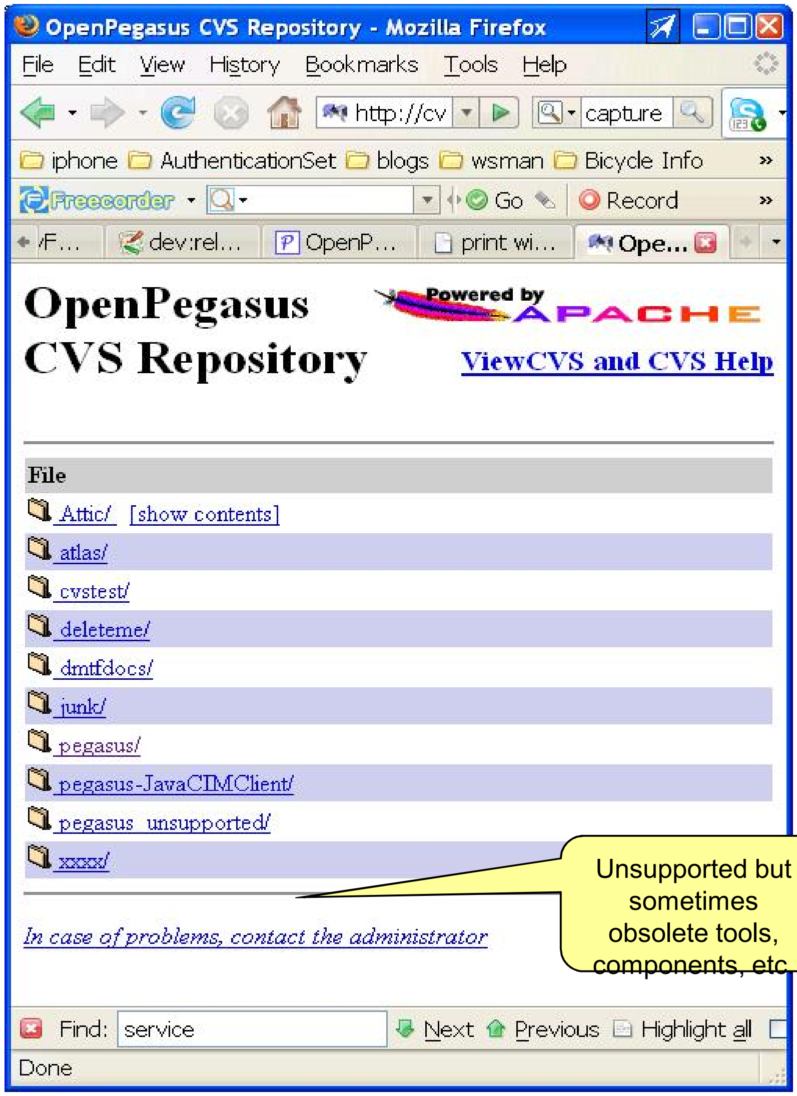
- Client Protocol Specs.
  - WSMAN – DMTF DSP0206, 0207, 0230
  - CIM/XML – DMTF DSP0200 & DSP0201
- Provider Interfaces
  - OpenGroup CMPI C interface specification V2
- Object Model
  - DMTF DSP0004
- Query Languages
  - CQL - DMTF DSP0202
  - WQL – Informal DMTF specs
- Profiles
  - Selected Server Profiles

Profile Implementation

- OpenPegasus Implements selected profiles
  - Server Control profiles
    - WBEM Server profile
  - Basic top level profiles
    - Profile Registration Profile
  - Major Services implemented by the Server
    - DMTF Indication Profile
- Will look at extending to new profiles in the future.

Additional Components

- SNIA Java Client and browser
  - Pegasus-JavaCIMClient module in OpenPegasus cvs repository
- WMI mapper
  - pegasus cvs module (pegasus/src/WMIMapper)
- configure
  - Build configure script in pegasus-unsupported module


OpenPegasus CVS Repository - Mozilla Firefox

OpenPegasus CVS Repository

ViewCVS and CVS Help

File

- Attic/ [show contents]
- atlas/
- cvstest/
- deleteme/
- dmtfdocs/
- junk/
- pegasus/
- pegasus-JavaCIMClient/
- pegasus_unsupported/
- xxxx/

In case of problems, contact the administrator

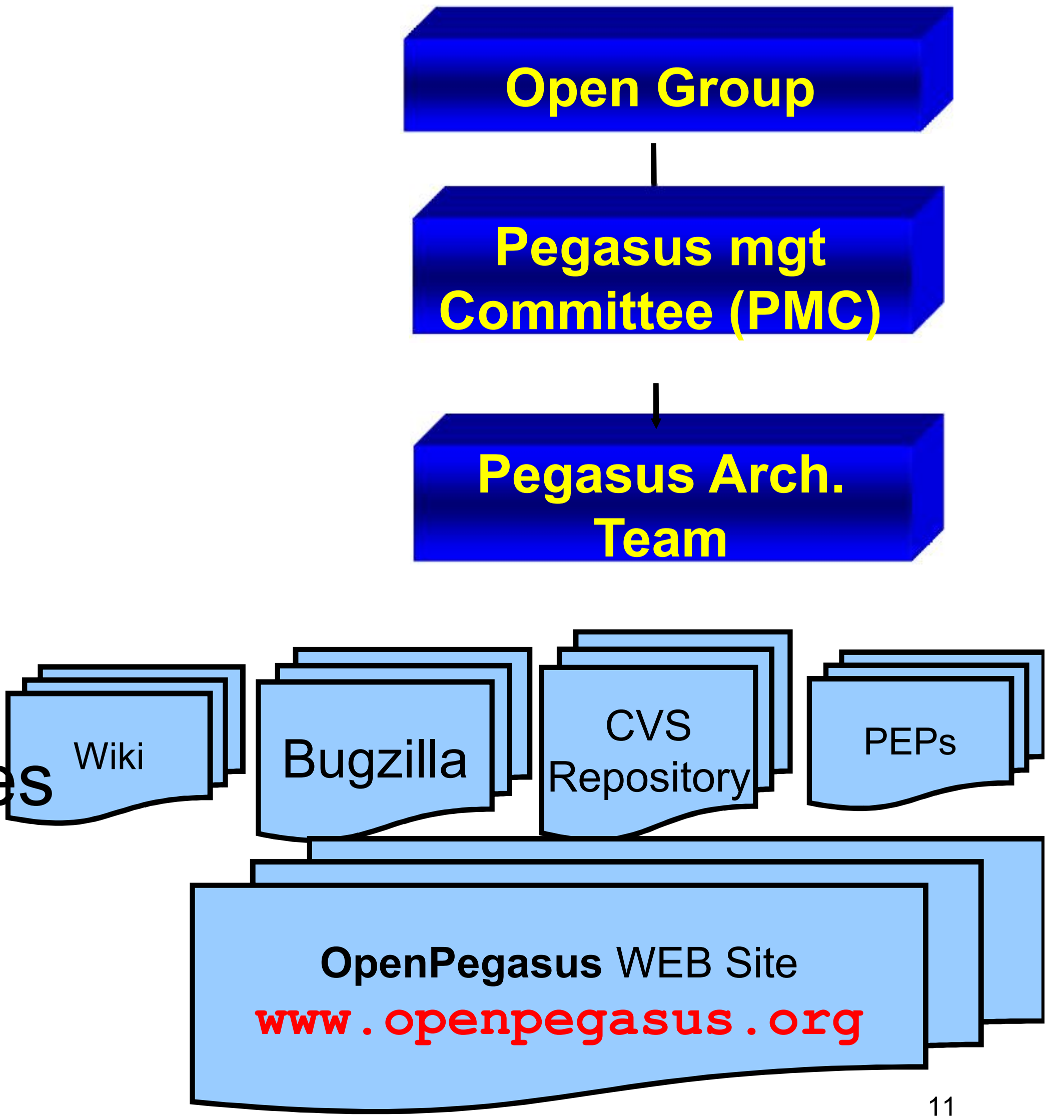
Done

Unsupported but sometimes obsolete tools, components, etc

MANAGEMENT DEVELOPERS CONFERENCE

OpenPegasus Project

- Community Project
  - Multiple supporters
  - Multiple developers
  - Multiple users
- Open source-code, open project
- Meritocracy based project model
- Documented processes
  - Support tools (bugzilla, websites, WIKI, etc.)
  - Defined releases, commit procedures, etc.



OpenPegasus Wiki

- OpenPegasus Release Planning now resides in the Wiki
  - https://wiki.opengroup.org/pegasus-wiki/doku.php?id=dev:openpegasusreleasestatus

- Architecture Team Telecon Minutes
  - https://wiki.opengroup.org/pegasus-wiki/doku.php?id=dev:architecture_team_minutes

- Open Pegasus Strategy and Planning
  - a list of possible items to attack in the future
  - https://wiki.opengroup.org/pegasus-wiki/doku.php?id=dev:planning:planning_top_page

- FAQ
  - https://wiki.opengroup.org/pegasus-wiki/doku.php?id=faq:frequently_asked_questions
  - Indications, Building and Installing OpenPegasus, Pegasus Server Administration

OpenPegasus Releases

- OpenPegasus Formal Version Releases
  - Version (ex. 2.11)
    - New functionality
    - Backward compatibility
    - Maintain binary interface compatibility
    - Extensive testing
    - Release Documentation
    - ~ each 9 – 12 months
  - Point Releases (ex. 2.11.1) – Largely bug fixes
    - New functionality only in special cases
    - ~ each 6 months depending on bugs
  - Major Version
    - Ex Version 2.x – Will change only when we have incompatible changes
- Project maintains
  - Current release (ex. 2.11)
  - Two back version releases (2.9, 2.10)

OpenPegasus Availability

- OpenPegasus source freely available
  - Releases are on OpenPegasus web site
    - Source tarball
    - Source rpms
  - No binary releases
- Available as part of some OS releases
  - VMS, ZOS, HPUX etc.
- Available on Several Linux distributions as binary RPM

Platform Support

- Platforms Supported
  - Unix / Linux
  - Windows
  - VMS
  - ZOs
- Fully Supported Platforms
  - Tested nightly and for release
- Supported Platforms
  - Include configuration, some testing but no maintainer currently for regular testing

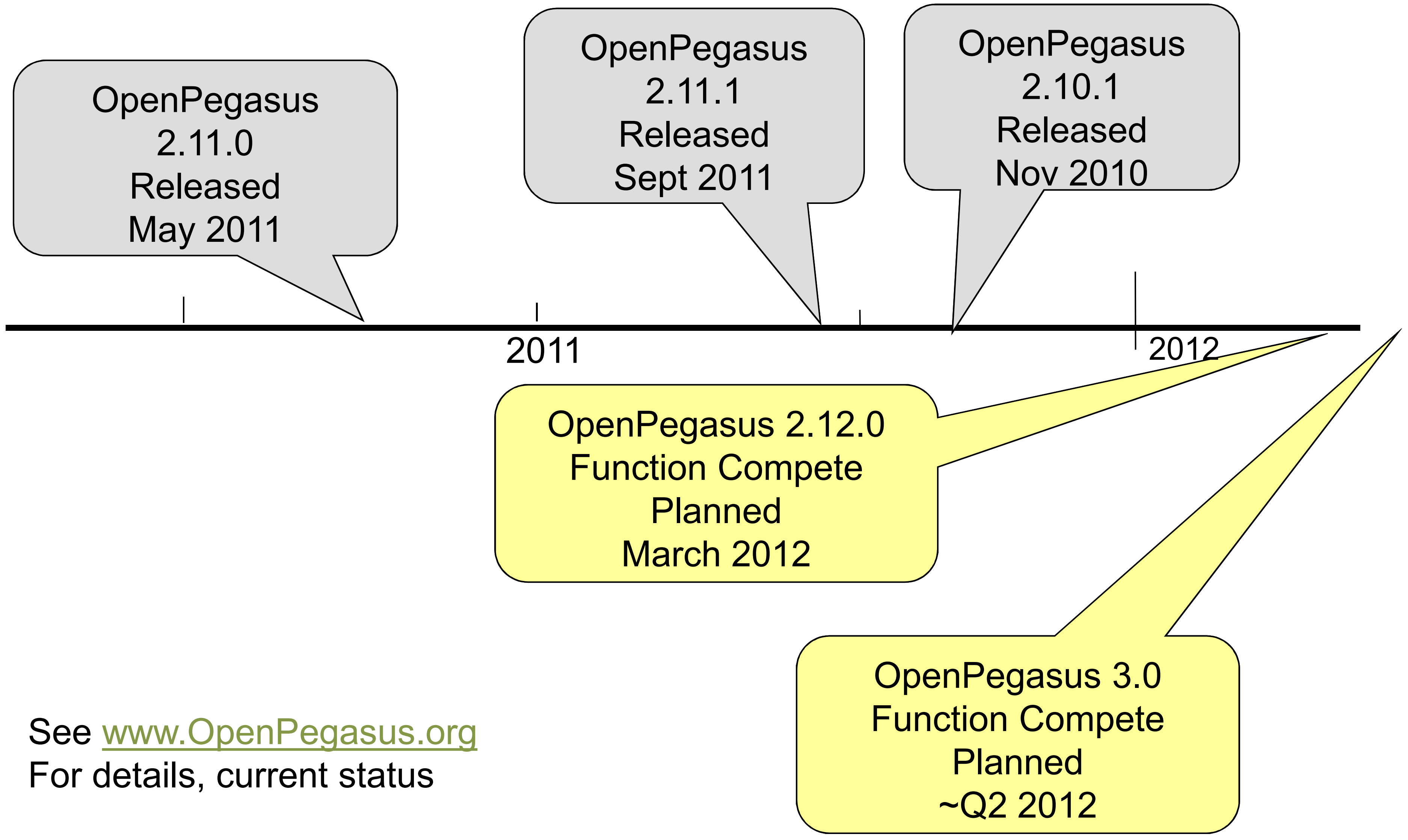
Supporting OpenPegasus

- The project keeps important bug fixes through 2 previous versions.
- **All** fixes to OpenPegasus are documented in bugs
- All bugs/fixes are in the Bugzilla database
  - Find bugs through version search
    - All versions are tagged (ex. RELEASE_2_9_0)
- Support through
  - OpenPegasus email
  - Support in specific organizations (ex SNIA plugfests provide specific support)

Section 1.2

OPENPEGASUS
VERSION OVERVIEW

What’s New in 2011/ 2012


OpenPegasus 2.11.0 Released May 2011
OpenPegasus 2.11.1 Released Sept 2011
OpenPegasus 2.10.1 Released Nov 2010
2011
2012
OpenPegasus 2.12.0 Function Compete Planned March 2012
OpenPegasus 3.0 Function Compete Planned ~Q2 2012

See www.OpenPegasus.org
For details, current status

2.12.0 New Functionality

- CIM/XML Pull Operations
- WSMAN Eventing support
- SSL cipher suite support
- Restful Services (experimental and doubtful)
- Update ICU services
- LifeCycle Indications to support provider management
- interop namespace
- Eliminate/reduce SNIA/SMIS differences
- Expand cimcli for embedded instance support
- Clean up bug list

A Wish List is not a commitment.

Commitments only come when someone agrees to do the work, not just need the result.

Details in OpenPegasus PEPs and Bugs:
https://wiki.opengroup.org/pegasus-wiki/doku.php?id=dev:release:2.12_x

CIM/XML Pull Operations

- Implemented per DMTF Specification DSP 0200 and DSO0201
- Implemented internally so that they can be used with CIM/XML and WSMAN operations
- Implemented for compatibility with existing providers.
- Further information in Part 2 (Advanced topics)

WSMAN Eventing support

Features planned for 2.12

- Subscribe (wsme:Subscribe)
- Unsubscribe (wsme:Unsubscribe)
- Subscription response (wsme:SubscribeResponse)
- Delivery mode Push (http://schemas.xmlsoap.org/ws/2004/08/eventing/DeliveryModes/Push)
- Filters with WQL language(wsme:Filter)
- Connection retries(wsman:ConnectionRetry)

Cipher Suite support in cimserver

- Cipher Suite can be specified for the cimserver using the option sslCipherSuite.
- This directive uses a colon-separated *cipher-spec* string consisting of OpenSSL cipher specifications to configure the Cipher Suite the client is permitted to negotiate in the SSL handshake phase.

Example : cimconfig sslCipherSuite=RSA:!EXP:!NULL:+HIGH:-LOW

(all ciphers using RSA key exchange and Triple-DES(HIGH) but not export ciphers(EXP), ciphers using no encryption(NULL) and all low strength ciphers(LOW))

ICU Update

- ICU is OpenPegasus internationalization library
- OpenPegasus currently supports old version of ICU(v 3.2)
  - Obsolete
  - Not easily available
  - Not current version on distributions
- ICU 4.0 represents incompatible changes
- Update OpenPegasus to support ICU 4.0

LifeCycle Indication Support

- Issue today with admin lack of knowledge of status of failed OOP providers
- Adds lifecycle indication support for the OpenPegasus provider module class
- Pegasus PEP 360 for details

Expand CIMCLI to support embedded instances

- CIMCLI is OpenPegasus command line client tool for testing and production use
- Allows all instance operations and all class operations except create/modify
- Added cleaner cli input for create/modify
  - Cimcli ci myClass id=3128 name=fred
- Apply this new definition to all objects
  - Cimcli getInstance myClass id=2138
- Add creation/modification/display of instances containing embedded instance properties
- Expand display capabilities

Interop Namespace

- OpenPegasus uses root/PG_Interop as interop namespace name
- Embedded systems can change at build
- Issue with systems that upgrade without removing repository
- Solution: alias namespace mechanism so root/interop is alias for root/PG_Interop

Sigh – we finally beat one issue down; or it is as bad to get to far ahead as to far behind.

Reduce SNIA/SMIS differences

- Currently several compile time flags that specialize Pegasus for SNIA SMIS compatibility
- Goal
  - Remove compile time options
  - Remove special SNIA code
  - If there are differences they should be driven by profiles, not compile time flags

Supporting Restful services

- Create a new client adapter (parallel to WSMAN) for the restful protocols.
- With state of specifications very slim change that this will get into 2.12
- Will create a service parallel to WSMServer to map Pegasus internal operations to Restful operation request/responses
- PEP in review and early code done

2.11 New Functionality

- 32 Bit Providers in 64 Bit system
- Provider Module Grouping
- WSMan Adapter
  - Association Filters per DSP0227, Sect 8.2
- DMTF Indication Profile
- SNMP v3 trap generation
- Improve OOP provider failure recovery

Details in Pegasus PEPs and Bugs
https://wiki.opengroup.org/pegasus-wiki/doku.php?id=dev:release:2_11_x

Pegasus 2.11 Enhancements

- Improve Release Builds
  - External SLP support
- Multiple Directories for Providers
- Improve quality checking on provider responses (ie. Handle PropertyList)
  - Server filters properties not on property list
  - For CMPI improves performance
  - Recommendation is that providers use property list only for properties that affect performance

OpenPegasus 3.0

- Goal – Remove behavior issues that are not consistent with spec.
- Major version change because it changes behavior, not because major change to the platform.
- See Bugzilla Keyword TARGET_3_0 for details
- Schedule: Unknown but Post 2.12.0

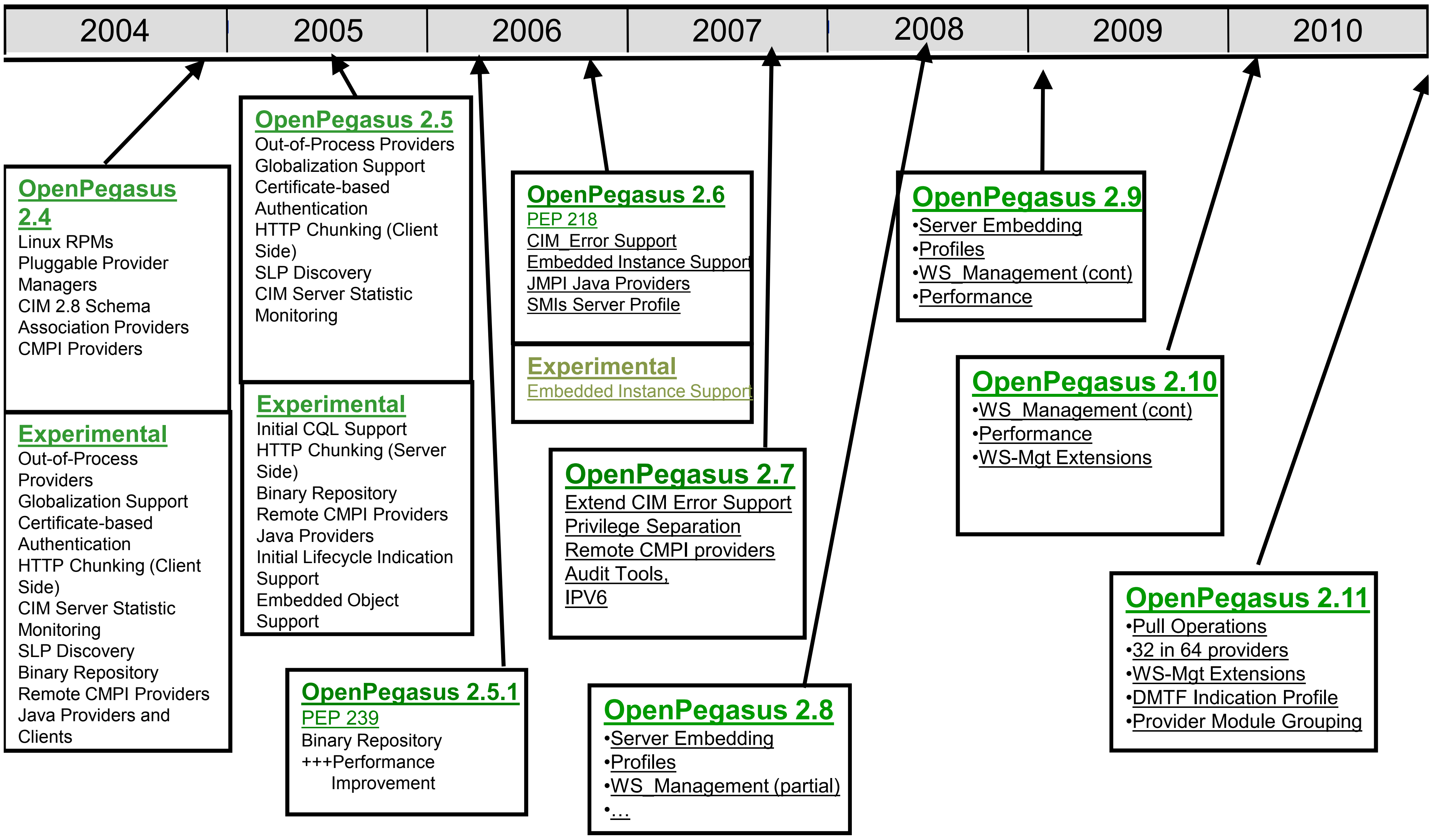
Current issues list for OpenPegasus V3

- Whitespace in CIM/XML issue
- Repository modify instance behavior
- Some incorrect return status codes
- Indications subscriptions succeed sometimes when shouldn’t
- CIMValue Null vs. value
- Rebase String class (utf-16 to utf-8)

OpenPegasus and CIM 3.0

- OpenPegasus team participating in V3 planning
- Schedule and form of solution undefined today because
  - Early in planning stage
  - OpenPegasus could not really start planning until DMTF Work In Progress specs available.
- OpenPegasus 3.0 is NOT CIM 3.0

OpenPegasus Release History



OpenPegasus 2.4
Linux RPMs
Pluggable Provider Managers
CIM 2.8 Schema
Association Providers
CMPI Providers

Experimental
Out-of-Process Providers
Globalization Support
Certificate-based Authentication
HTTP Chunking (Client Side)
CIM Server Statistic Monitoring
SLP Discovery
Binary Repository
Remote CMPI Providers
Java Providers and Clients

OpenPegasus 2.5
Out-of-Process Providers
Globalization Support
Certificate-based Authentication
HTTP Chunking (Client Side)
SLP Discovery
CIM Server Statistic Monitoring

Experimental
Initial CQL Support
HTTP Chunking (Server Side)
Binary Repository
Remote CMPI Providers
Java Providers
Initial Lifecycle Indication Support
Embedded Object Support

OpenPegasus 2.5.1
PEP 239
Binary Repository
+++Performance
Improvement

OpenPegasus 2.6
PEP 218
CIM_Error Support
Embedded Instance Support
JMPI Java Providers
SMIs Server Profile

Experimental
Embedded Instance Support

OpenPegasus 2.7
Extend CIM Error Support
Privilege Separation
Remote CMPI providers
Audit Tools,
IPV6

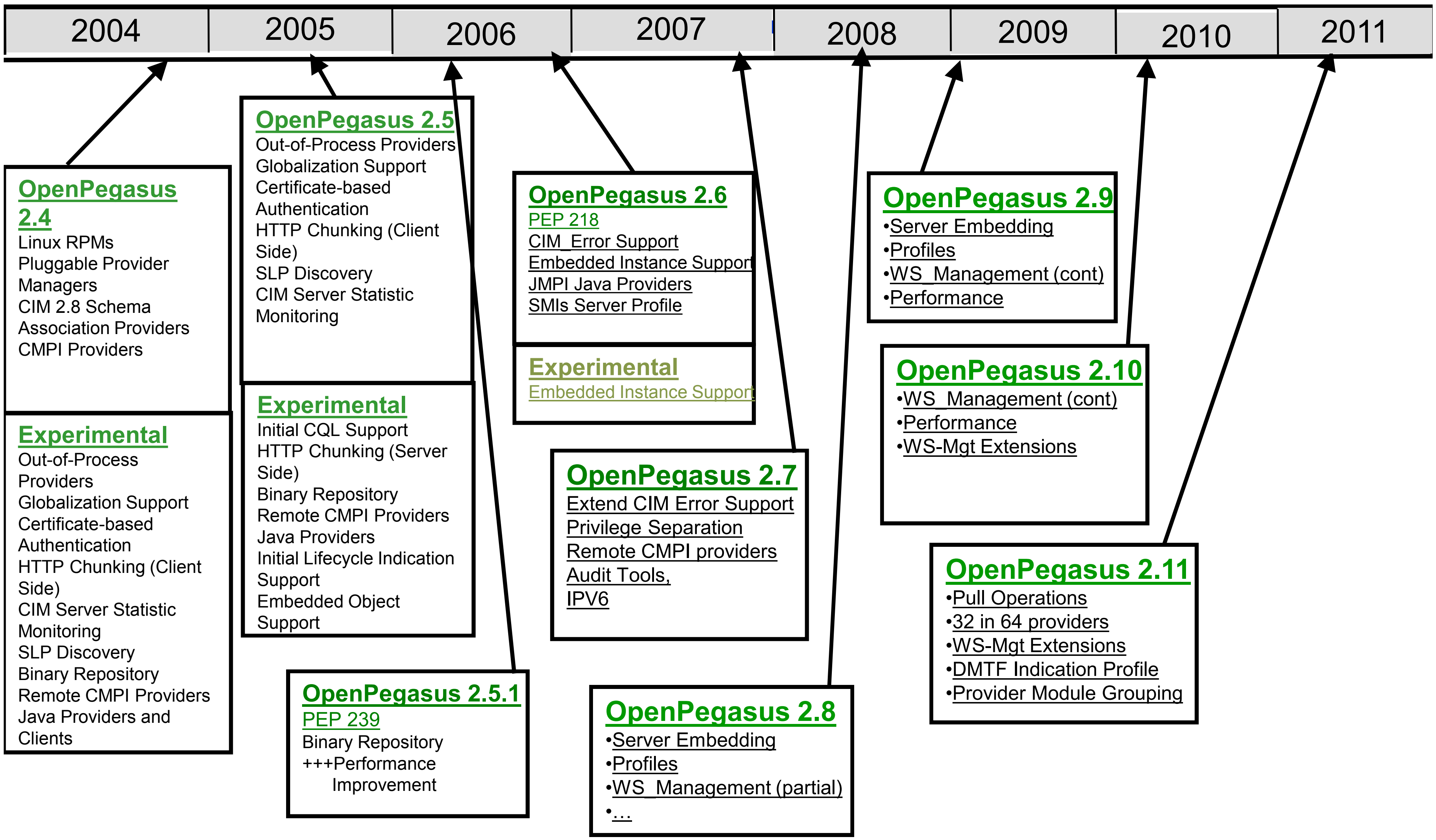
OpenPegasus 2.8
•Server Embedding
•Profiles
•WS_Management (partial)
•…

OpenPegasus 2.9
•Server Embedding
•Profiles
•WS_Management (cont)
•Performance

OpenPegasus 2.10
•WS_Management (cont)
•Performance
•WS-Mgt Extensions

OpenPegasus 2.11
•Pull Operations
•32 in 64 providers
•WS-Mgt Extensions
•DMTF Indication Profile
•Provider Module Grouping

OpenPegasus Release History



OpenPegasus 2.4
Linux RPMs
Pluggable Provider Managers
CIM 2.8 Schema
Association Providers
CMPI Providers

Experimental
Out-of-Process Providers
Globalization Support
Certificate-based Authentication
HTTP Chunking (Client Side)
CIM Server Statistic Monitoring
SLP Discovery
Binary Repository
Remote CMPI Providers
Java Providers and Clients

OpenPegasus 2.5
Out-of-Process Providers
Globalization Support
Certificate-based Authentication
HTTP Chunking (Client Side)
SLP Discovery
CIM Server Statistic Monitoring

Experimental
Initial CQL Support
HTTP Chunking (Server Side)
Binary Repository
Remote CMPI Providers
Java Providers
Initial Lifecycle Indication Support
Embedded Object Support

OpenPegasus 2.5.1
PEP 239
Binary Repository
+++Performance
Improvement

OpenPegasus 2.6
PEP 218
CIM_Error Support
Embedded Instance Support
JMPI Java Providers
SMIs Server Profile

Experimental
Embedded Instance Support

OpenPegasus 2.7
Extend CIM Error Support
Privilege Separation
Remote CMPI providers
Audit Tools,
IPV6

OpenPegasus 2.8
- Server Embedding
- Profiles
- WS_Management (partial)
- …

OpenPegasus 2.9
- Server Embedding
- Profiles
- WS_Management (cont)
- Performance

OpenPegasus 2.10
- WS_Management (cont)
- Performance
- WS-Mgt Extensions

OpenPegasus 2.11
- Pull Operations
- 32 in 64 providers
- WS-Mgt Extensions
- DMTF Indication Profile
- Provider Module Grouping

2.10 New Functionality

- Speed Improvements
  - Single-Object Memory model for at least some SCMO functionality (cmpi responses)
- Expanded Indication support
  - Indication Profile
  - Algorithms to improve indication delivery reliability
- Support for multiple OpenPegasus servers in a single system
- Expanded WS-Man support
  - wsmid:Identify, WS_Enumeration filter support (WQL) and Custom Actions (i.e. CIM extrinsic Methods)

OpenPegasus 2.9 New Functionality

- Function Changes
  - SQLite based alternate class and instance repository
  - Expand WS-Management integrated support
    - ws-enumerate)
  - Server performance enhancements (out-of-process providers)
  - Solaris port enhanced
  - Binary internal and Client protocol

OpenPegasus 2.8 New functionality

- Version 2.8
  - Embedded Server Extensions (Memory Resident Repository)
  - Initial WS-Management integrated support (Infrastructure, get, put)
  - DMTF Indication Profile partial support
  - DMTF Profile Registration Profile
  - Pluggable Provider Manager support
  - Support Indication statistics
  - Internal Server support (improved tracing, etc.)
  - Python provider manager (available from Novell)
  - Incremental performance improvements
  - Build and configuration options

Major Functionality By Version

- Version 2.6 (PEP 218)
  - Initial CIM_Error support
  - Integrate SMIs server profile
  - Embedded instance support
  - Server footprint reduction (~40%)
  - Repository archive utility
  - CMPI provider interface current to V2 specification
  - Add server audit log
  - Add indications to remote CMPI
  - SSL trust store utilities
  - SLP enhancements
  - Indication Subscription management utility
- Version 2.6.1
  - IPV6 Support, experimental normally disabled
- Version 2.7 (PEP 296)
  - Support for IBM i5/PASE platform
  - Create privilege separation executor
  - Support for IPV6
  - Enable Remote CMPI for Windows
  - Enhanced log file support
  - Refactoring Queuing and OS primitives for performance (~ +30%)
  - Refactoring Class objects
  - Audit Logging (special log for operations that modify information)
- Version 2.7 Feature status changes
  - See the feature page

Major Functionality By Version

- Version 2.4 (PEP 97)
  - Linux RPMs
  - Pluggable Provider Managers
  - CIM 2.8 Schema
  - Association Providers
  - CMPI Providers
  - Out-of-Process Providers
  - Globalization Support
  - Certificate-based Authentication
  - HTTP Chunking (Client Side)
  - CIM Server Statistic Monitoring
  - SLP Discovery
  - Binary Repository
  - Remote CMPI Providers
  - Java Providers and Clients
- Version 2.5 (PEP180)
  - HTTP Chunking and Internal Response Segmentation
  - Remote CMPI Providers
  - Shared Namespaces
  - Java Providers (JMPI)
  - Initial Lifecycle Indication Support
  - CQL – Stage 1
  - Dynamic CIM Listener
  - Compressed Repository
  - Static Memory Size Reduction
- Version 2.5.1
  - Performance enhancement for Operations (approx 10 – 1)
- Version 2.5.2
  - Size reduction and static build options

Section 1.3

OPENPEGASUS FEATURES

OpenPegasus Features

- **CIMServer**
  - Core Infrastructure
  - CIM Operations
  - Indication Processing
  - Query Languages
  - Server Configuration
  - Provider Management
  - Indication Subscription Management
  - Local Domain Sockets
  - Chunked Transfer
  - Localization
  - Object Normalizer
  - OutOfProcess Providers
  - Statistics
- **Repository**
  - Core
  - MOF Compiler
  - Encodings
  - Shared Schema
  - Upgrade Utility
  - Archive
- **Provider Interfaces**
  - C++
  - CMPI
  - JMPI
  - Remote CMPI
  - Python
  - User Context

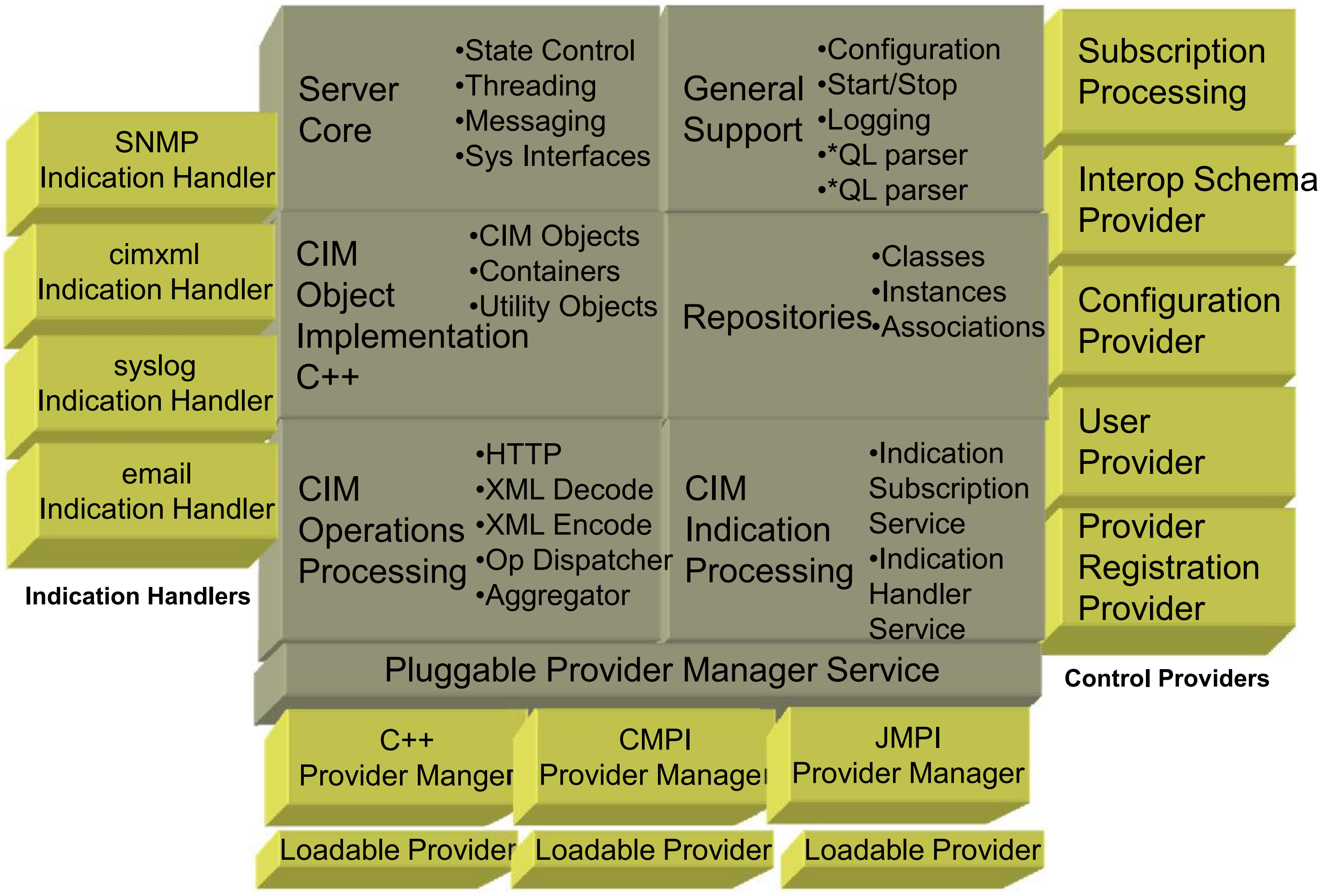
Features (cont)

- **Providers**
  - Interop Provider
  - Mgd. Server Providers
  - Profile Providers
    - DMTF Provider registration profile
    - DMTF Indication Profile
    - SNIA Server Profile
- **Client Interfaces**
  - CIM-XML
  - WS-Managmenent
- **Indication Listener**
- **Indication Handlers**
  - CIM-XML
  - SNMP
  - SysLog
  - Email
- **Security**
  - SSL
  - Local Authentication
  - PAM Authentication
  - Authentication Mgt
  - Cert based Auth
  - SSL Cert Management
  - Privilege Separation
  - Audit Logging
- **SLP**
  - Internal
  - OpenSLP interface
- **WMI Mapper**

MANAGEMENT DEVELOPERS CONFERENCE

OpenPegasus Functionality - Server

- CIMServer
  - Core Infrastructure
  - CIM Operations
  - Indication Processing
  - Query Languages
  - Server Configuration
  - Provider Management
  - Indication Subscription Management
  - Local Domain Sockets
  - Chunked Transfer
  - Localization
  - Object Normalizer
  - Out-Of-Process Providers
  - Statistics

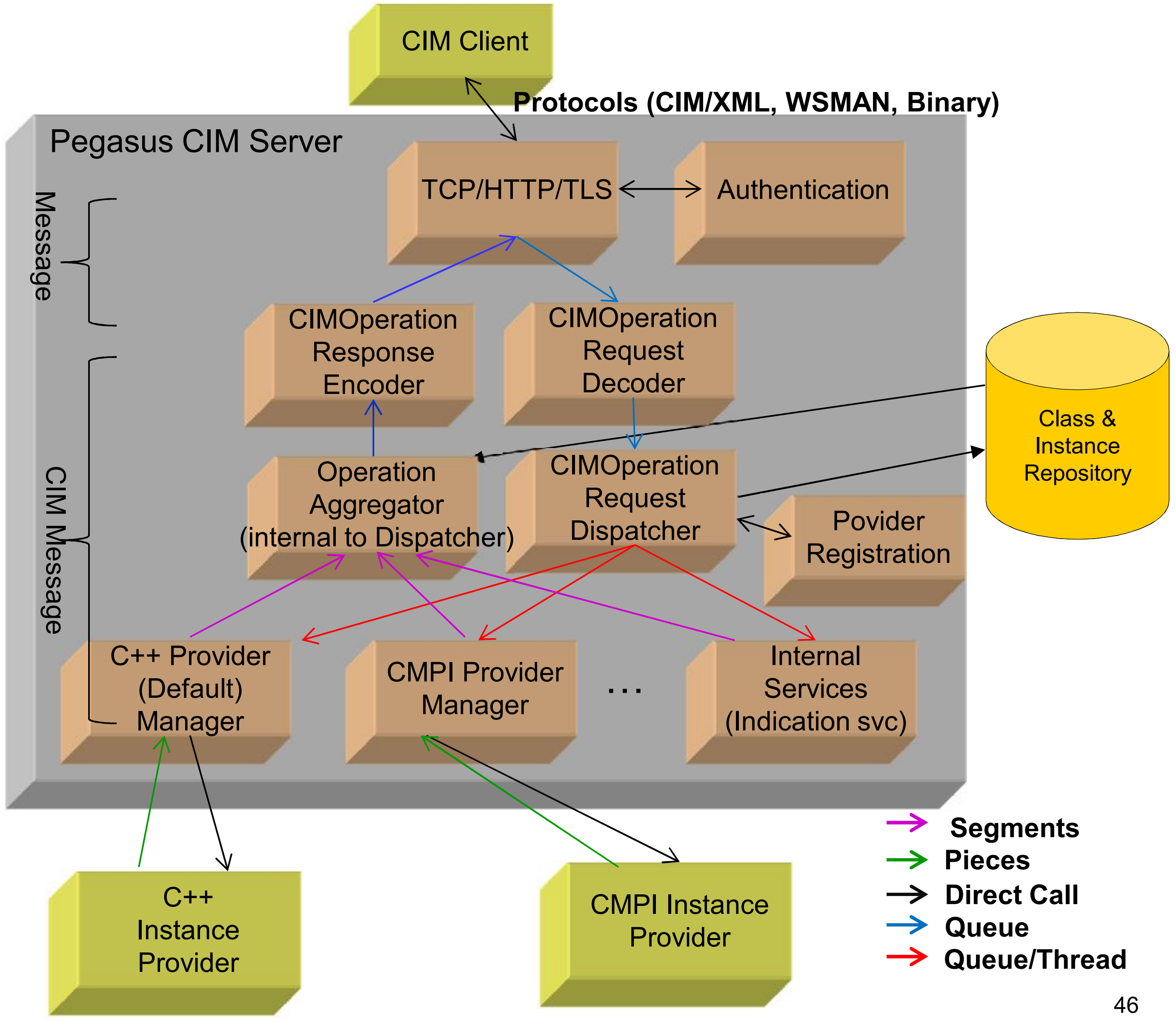


OpenPegasus Server/Client Protocols

- CIM-XML (DMTF DSP 0200)
  - Implements all operations in v 1.2 spec
  - Extend to Pull operations v 1.3 spec for next release
- WS-Management(DMTF DSP 0226, 0227, 0230)
  - Implements required operations except assoc filters (add assoc with 2.11).
  - Assoc filters planned for next release

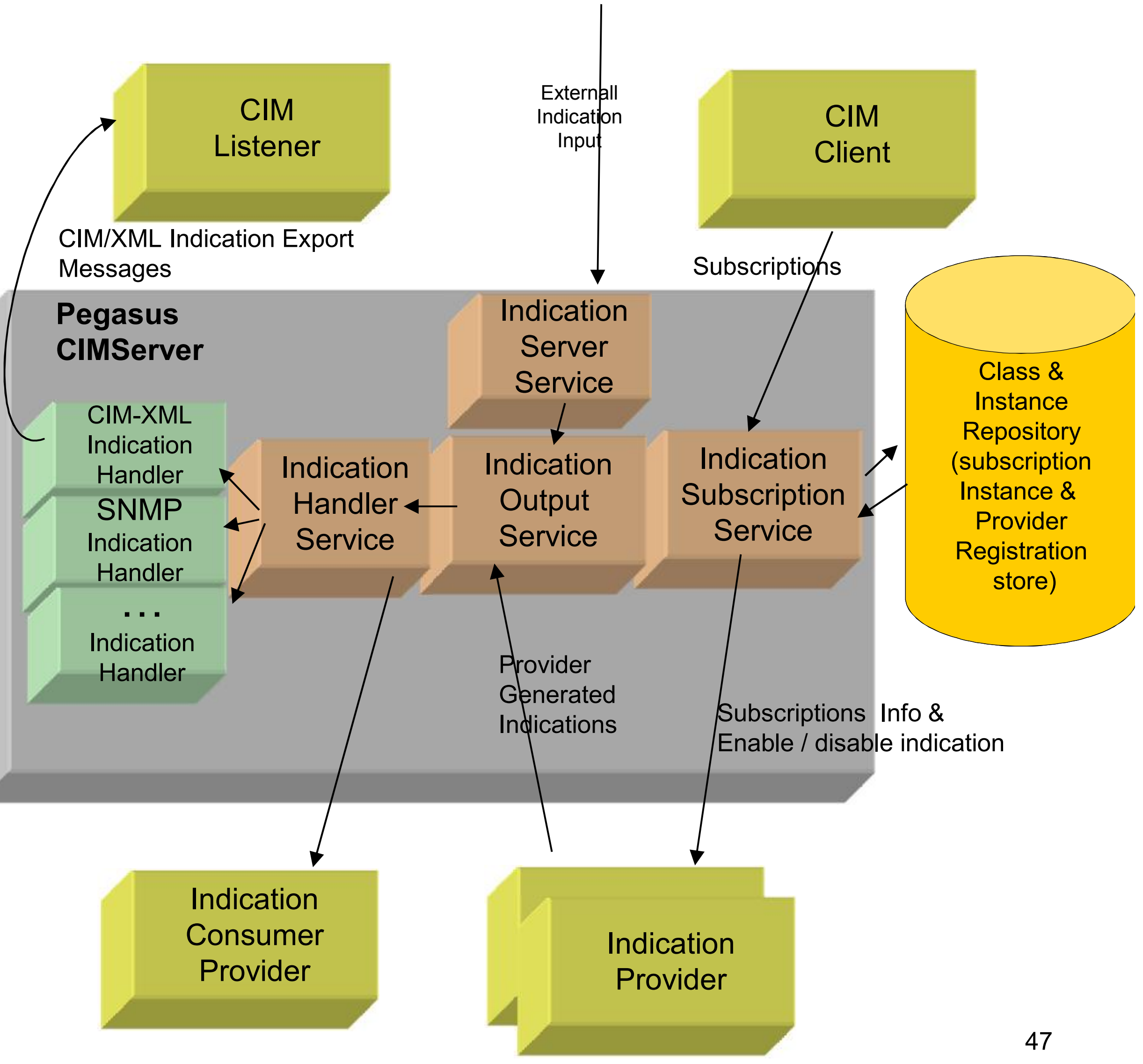
OpenPegasus Operation Execution EnumerateInstances

- Internal communication is message passing through queues
- Messages are based on Message and CIMMessage classes
- Some interfaces execute on separate threads


CIM Client
Protocols (CIM/XML, WSMAN, Binary)
Pegasus CIM Server
TCP/HTTP/TLS
Authentication
Message
CIMOperation Response Encoder
CIMOperation Request Decoder
Class & Instance Repository
CIM Message
Operation Aggregator (internal to Dispatcher)
CIMOperation Request Dispatcher
Povider Registration
C++ Provider (Default) Manager
CMPI Provider Manager
. . .
Internal Services (Indication svc)
C++ Instance Provider
CMPI Instance Provider
Segments
Pieces
Direct Call
Queue
Queue/Thread

OpenPegasus Indication Support

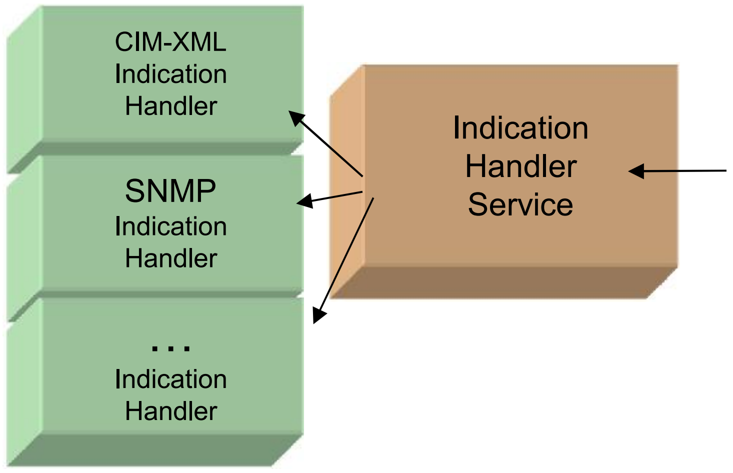
- Lifecycle and process indications
- Only indications supported by providers
- Support both CQL and WQL queries
  - Dynamic parsing and evaluation
- Multiple indication handlers
- Persistent indication subscriptions
- Indication Consumer Provider Type
- Accept External Indications



CIM Listener
Externall Indication Input
CIM Client
CIM/XML Indication Export Messages
Subscriptions
Pegasus CIMServer
Indication Server Service
CIM-XML Indication Handler
SNMP Indication Handler
. . . Indication Handler
Indication Handler Service
Indication Output Service
Indication Subscription Service
Class & Instance Repository (subscription Instance & Provider Registration store)
Provider Generated Indications
Subscriptions Info & Enable / disable indication
Indication Consumer Provider
Indication Provider

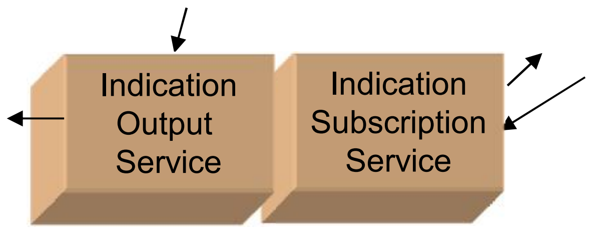
Indication Handlers

- Indication Handlers represent Indication delivery protocols
- Service extensions to core server
- Separate services for each handler type
- Support today for:
  - CIM-XML handler
  - SNMP trap handler
  - Syslog handler
  - Email handler



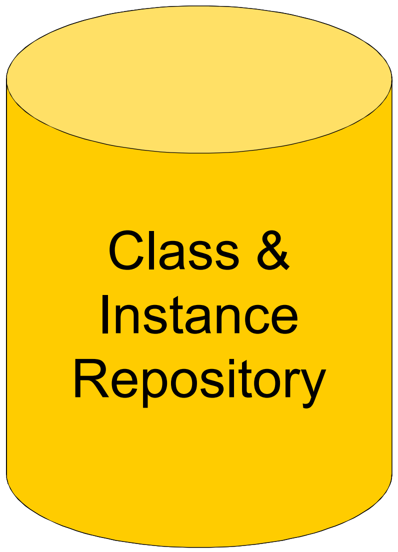
Query Language Support

- WQL
  - Complete support (remember no spec)
    - Dynamic parser & evaluator
  - Minor extensions for SNIA specials
  - Primary objective is Indication Subscription
- CQL
  - Support for most required functions
    - Dynamic parser and evaluator
  - Defined and implemented from early preliminary spec.
  - Indication Subscription Support

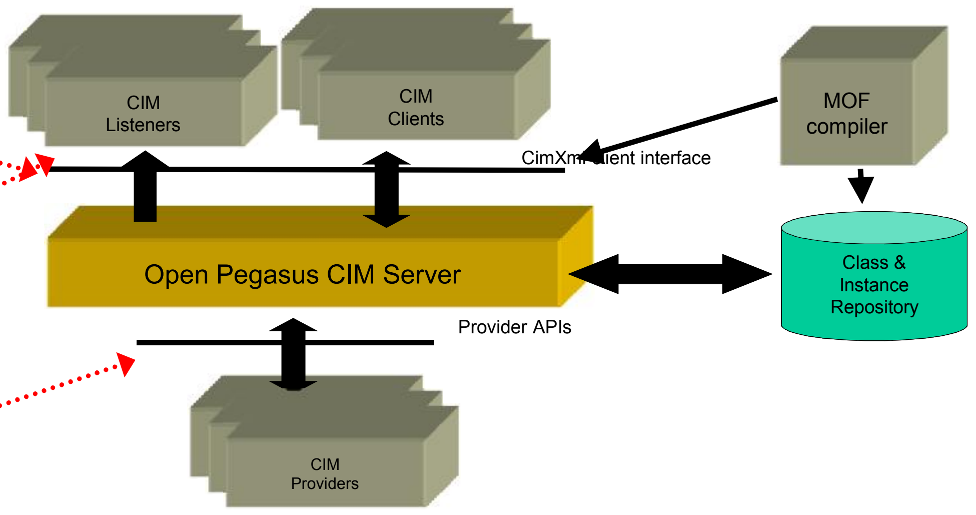


OpenPegasus Repository

- Characteristics
  - Class and Instance repositories
  - Supports all CIM operations
    - except query and life cycle indications
  - Default Disk File based repository
    - XML, binary, compressed encodings
  - Alternate DB Based respository
    - SQLite based repository
  - Off-line and on-line MOF compilers
    - Cimmof – online communicates to server
    - Cimmofl – offline communicates directly with repository
  - Optional Memory Resident Repository
    - MOF classes compiled into a c++ file which is compiled
  - Internal cache for performance



OpenPegasus Public Interfaces

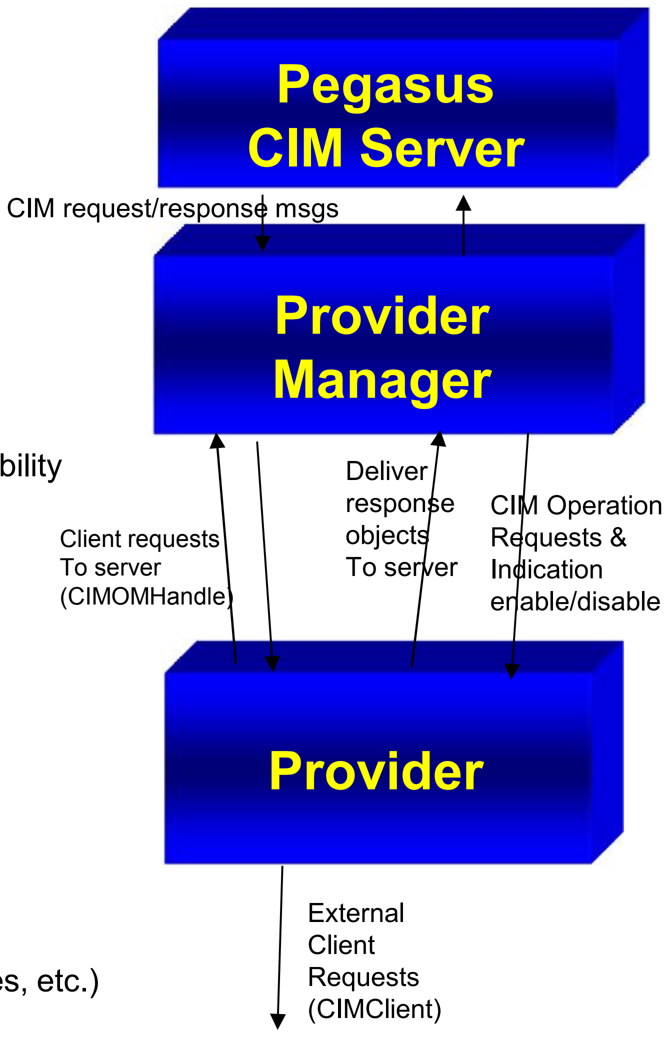


- **CIM Client Public Interfaces**
  - Implement CIM Operations
  - Implement Server Connection
  - CIM Objects*
  - SLP Discovery
- **CIM Listener Interfaces**
  - Listener setup
  - Indication reception
  - Indication consumers
  - CIMObjects*
- **CIM Provider Interfaces**
  - Implement Provider Types (Instance, Method, Association, Query)
  - Mimic Client Operation APIs
  - Extend with Context container for security, etc.
  - CIM Objects*
  - Multiple language bindings through multiple provider managers
  - C, C++. Java, etc.
- **CIM C++ Objects**
  - Manipulate CIM Objects
  - Class, instance, property, method, Value, etc.
- **Selected CLI Functions (ex. Compiler, admin tools)**
  - CLI cmd line interfaces maintain compatibility between versions

•Public Interfaces
- **Frozen**
- **Versioned**
- **Backwards Binary Compatibility**
- **Available through SDK (rpms)**

OpenPegasus Provider Interface Characteristics

- **OpenPegasus Provider Types**
  - **Instance** (get, enumerate, create, delete, modify instances)
  - **Method** (invokeMethod)
  - **Association** (References, referencenames, associatiors, associatornames)
  - **Indication**
    - Enable, disable, create, modify, delete subscriptions
    - Indications generated through the same interface as operation responses
  - **InstanceQuery** (ExecQuery)
  - **IndicationConsumer** (Sink for indications)
- **Provider Control**
  - Initialize() terminate() functions
  - Providers are dynamically loaded **AND** unloaded
  - Provider normally unloaded when not used but can override unloadability
- **Provider Access to Other Providers**
  - CIMOMHandle
    - Allows all CIM operations back to Cimom binary interface
    - Access point provided with initialize
- **ProviderOperationContext**
  - Part of every operation request to provider
    - User information, etc.
- **Provider Security**
  - out of process providers
  - Run as (server permissions, user permissions, etc.)
- **Providers can also be Clients**
  - Use client library
- **OpenPegasus operation response interface is incremental**
  - Deliver partial responses (individual objects, subset of total responses, etc.)
  - Important to control memory usage.
  - Generated indications delivered through this interface
- **CMPI provider manager implements Remote Providers**
- **OpenPegasus Providers dynamically registered**


Pegasus CIM Server
CIM request/response msgs
Provider Manager
Client requests To server (CIMOMHandle)
Deliver response objects To server
CIM Operation Requests & Indication enable/disable
Provider
External Client Requests (CIMClient)

Provider Characteristics (cont)

- OpenPegasus implements Out-Of-Process providers
  - Provider failure does not cause CIM Server failure
  - Implements a failed provider recovery algorithm (2.12)
- Provider operation calls are multithreaded
  - Every Operation call is a new thread
  - MultiThread protection is the provider’s responsibility
  - The Pegasus thread classes are NOT considered public.
- Providers & Modules
  - Provider Module
    - Loadable component (dll, sh, etc.)
    - Contain one or more providers
  - Provider
    - Implementation of methods for a single class
    - May be grouped into Provider Modules
- Incremental Response Interface
  - Every multiobject response interface allows
    - Return array – May be partial array
    - Return single object
    - Complete call closes the response
  - Return small groups of response objects
    - Pegasus must work with the array size you return

Provider APIs

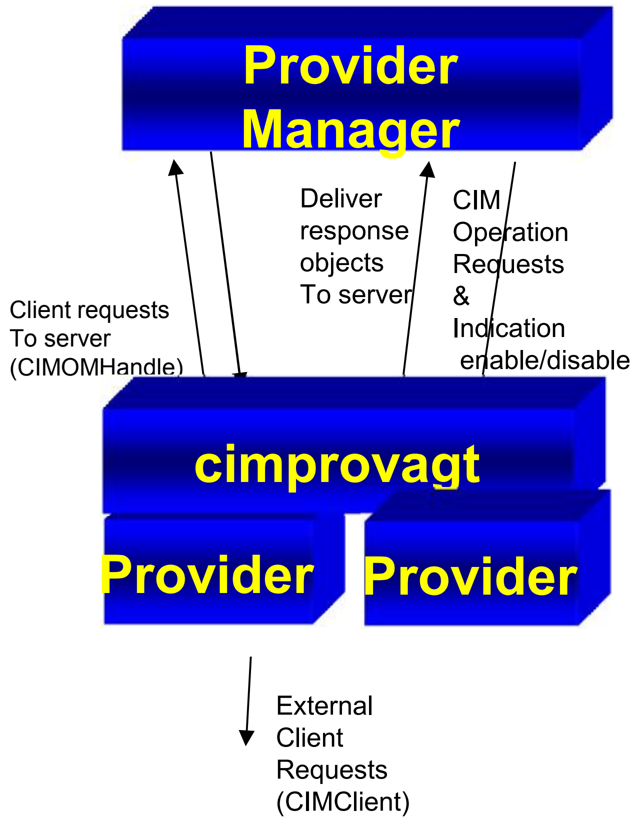
- General Functionality
  - Initialize Provider
  - Operation Request (getInstance, etc.)
  - Indication enable/disable (activate, etc. in CMPI)
    - Indication filter information (CMPI only)
  - Unload Provider
  - Status change (i.e. nounload())
  - Each operation request includes an operationContext container
    - Selected information (user, etc.)
- C++
  - Similar to C++ Client support APIs
- CMPI
  - Support current version of OpenGroup CMPI specification
  - Provide functions defined by CMPI specification
- JMPI
  - Similar to JSR48

Internal Providers

- Internal Providers (Control Providers)
  - Linked to CIM server
    - See pegasus/src/Pegasus/ControlProviders directory
  - All are C++ providers
  - Internal registration
    - Registration defined in a server internal table
  - Direct calls to communicate with Server modules
  - Direct access to Repository
  - Control Provider functionality today
    - __Namespace, CIM_Namespace, interop classes, usr/auth mgt, statistics, DMTF Indications Profile, and DMTF profile registration profile

Out-of-process Providers

- **Execute Providers in separate processes**
- **Objectives**
  - Prevent providers from damaging CIMOM
  - Binary compatible for Providers
  - Run providers within different security contexts
  - Run existing providers off all types
- **Configuration based**
  - Set at provider registration with:
    - PG_ProviderCapability:userContext
    - PG_ProviderModule:ModuleGroupName
  - Dynamic modification through modulegroup parameter
- **Authorization defined by user-context**
  - Only enabled for OOP and root permission svr
- **User contexts are permissions oriented**
  - Requestor, Designated, Privileged, CIMServer
- **Number of process determined by modules, user-context definition & module grouping**
- **NOTE:** Significant performance improvement in 2.9 (~ 300%)
- Significant further performance increase in 2.10
- Improve error recovery in 2.11 and 2.12


Provider Manager
Client requests To server (CIMOMHandle)
Deliver response objects To server
CIM Operation Requests & Indication enable/disable
cimprovagt
Provider
Provider
External Client Requests (CIMClient)

OpenPegasus Provider Management

- **Provider Installation**
  - Put provider into Pegasus provider directory
  - Register provider to OpenPegasus
- **Provider Registration**
  - Create instances of provider registration classes (providermodule, provider, provider capabilities
  - Registration can be static or dynamic
- **Dynamic provider state control**
  - Enable / disable (`cimprovider` utility)

OpenPegasus Security

- Security
  - SSL (uses OpenSSL)
  - Implements HTTP basic authentication
  - Local Authentication
  - PAM Authentication (where available)
  - Authentication Management
  - Cert based Authentication
  - SSL Certificate Management
    - Cmd line tool (cimtrust)
  - Privilege Separation (optional)
    - All privileged functions separated to one component
  - Audit Logging
    - Log all operations that modify server

SLP

- OpenPegasus provides capability for:
  - SLP Service Agent
    - Manages DMTF compatible SLP advertisement
  - UA and UA interface
    - Generating and processing client side SLP queries
- OpenPegasus allows alternate SLP SA implementations
  - Internal Pegasus SLP libraries (SA and UA)
    - Started and controlled by OpenPegasus server
  - OpenSLP
  - Supplier specific SLP libraries (ex SunSLP)

MANAGEMENT DEVELOPERS CONFERENCE

Client Infrastructure Support

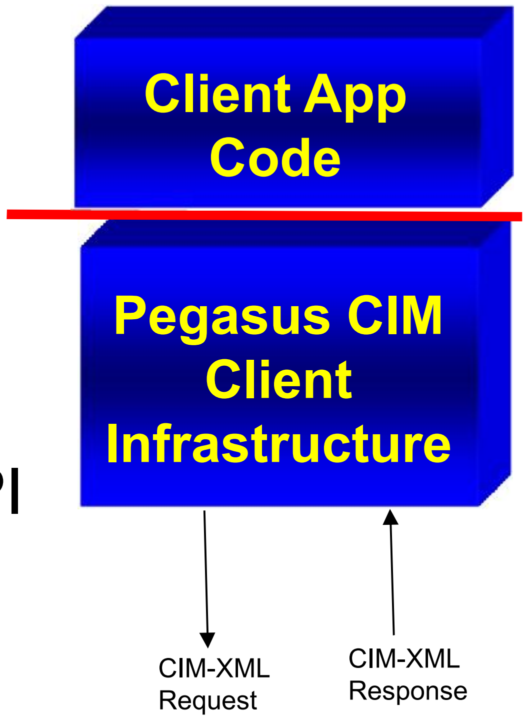
- CIM-XML
  - Supports all DMTF defined Operations
  - Provides
    - HTTP/HTTPS
    - Encoding/Decoding
    - Authentication
    - SLP User Agent
  - WS-MAN
    - No client support today
    - Reviewing possible commitment for V 2.12

Client Infrastructure Support

•CIM Operations
•Connectivity
•XML
•HTTP/HTTPS
•Authentication
•SLP UA

OpenPegasus CIMClient API

- Multi-Thread C++ Client API
  - CIM-XML
  - Provides all DMTF defined operations
  - Local Domain socket connection option (localconnect) (OpenPegasus specific)
  - Supports basic authentication, SSL with client side certificates.
  - Released public C++ Client Interface API
  - Limited to CIM/XML today
    - Experimental ws-man client (see pegasus_unsupported). Early Discussion of
    - Integrated ws-man client infrastructure.
- Java Client
  - JMPI API



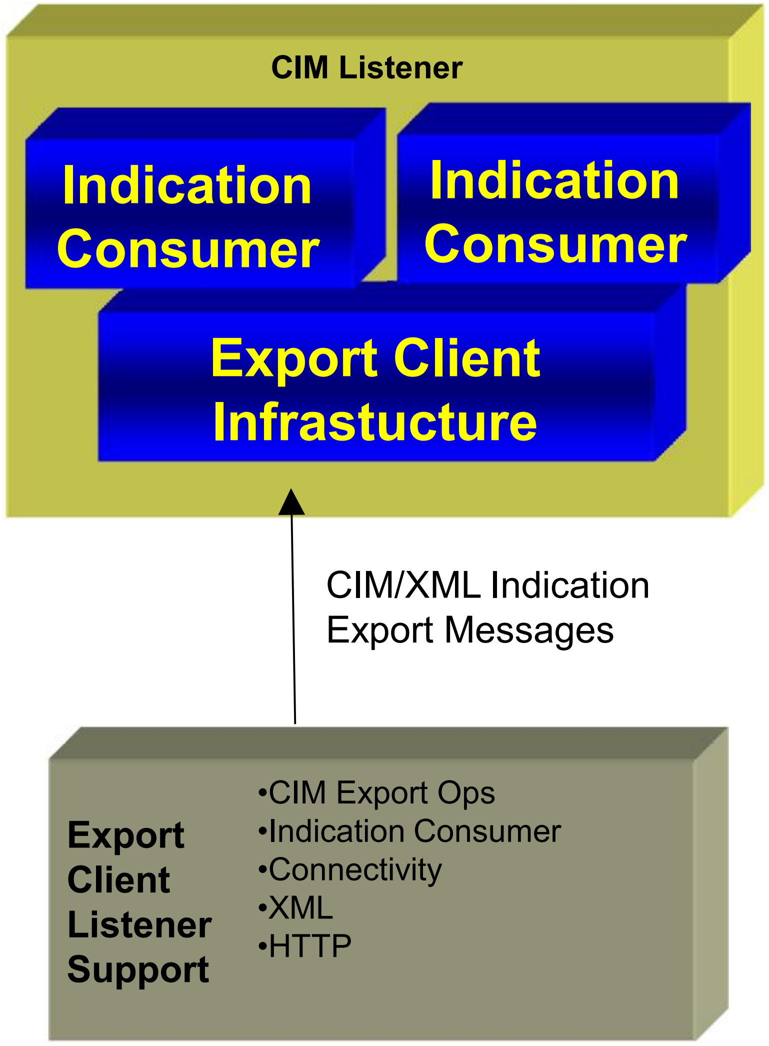
C++ Client API

- API methods match CIM-XML operations
  - Ex. getClass, etc.
- Methods for connect, disconnect, http language negotiation, authentication
- Parameters similar to CIM-XML operations
- Response Errors handled as Exceptions
  - CIMException, Exception

```
CIMClass getClass(
    const CIMNamespaceName& nameSpace,
    const CIMName& className,
    Boolean localOnly = true,
    Boolean includeQualifiers = true,
    Boolean includeClassOrigin = false,
    const CIMPropertyList& propertyList = CIMPropertyList());
```

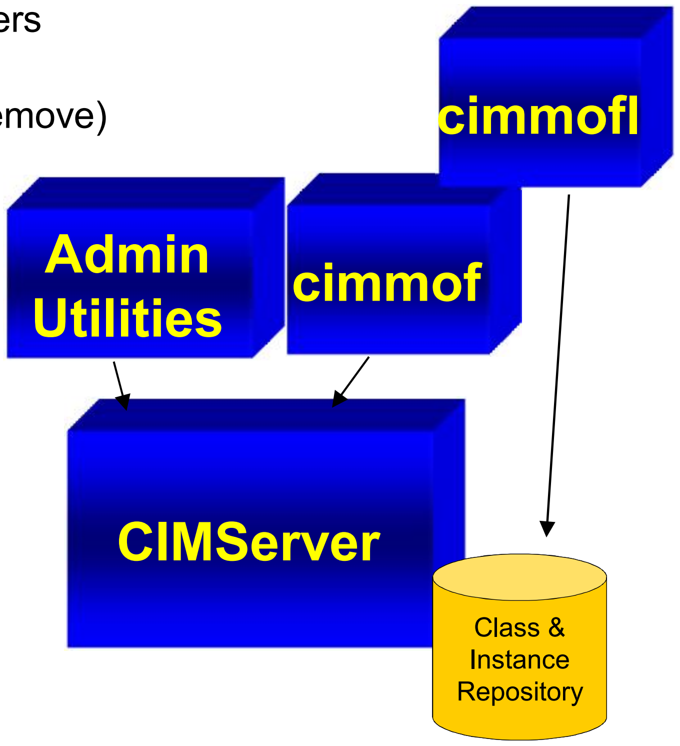
Indication Listeners

- Client Infrastructure
  - CIM-XML today
  - Providers
    - HTTP/Encoding, connectivity
- Static Listener
  - Statically defined Indication consumers to allow routing indications
- Dynamic Listener
  - Dynamically add indication consumers to route indications



OpenPegasus Admin Utilities

- **Admin tools are separate command-line utilities**
  - Included in production release
  - Security controlled to limit access to adminstrator
  - Communicate with server using localconnect
  - Provide off-line view options where possible
- Major Admin Tools today
  - **`cimconfig`**
    - Modify static and dynamic server configuration parameters
  - **`cimprovider`**
    - Determine and set state of providers (enable, disable, remove)
  - **`cimuser`**
    - Set user information (only selected environments)
  - **`cimtrust`**
    - Manage certificates
  - **`cimauth`**
    - Manage user authorizations (effectively obsolete)
  - **`cimmof`**
    - On-line MOF compiler. Uses client interface
  - **`cimmofl`**
    - Off-line MOF compiler. USE WITH CAUTION
  - **`repupgrade`**
    - Utility to upgrade repository in installed system
  - **`cimsub`**
    - Manage/display Indication subscriptions



OpenPegasus Provided Providers

- OpenPegasus includes a number of Providers with the source distribution
  - Control providers
    - Server functions for Admin and certain Profiles
    - Considered part of server
  - Sample Providers
    - demonstrate coding
  - Test Providers
    - Test Pegasus functionality
  - Limited Server Management Providers
    - Unique to certain OS
  - Profile Providers
    - Support selected profiles (They may be Control Providers)

Profile Support

- Support several generic profiles for DMTF and SNIA
  - DMTF Indication Profile
  - SNIA WBEM Server profile
  - DMTF Provider Registration Profile

Section 1.4

TECHNICAL SUBJECTS

- •Provider Module Grouping Function
- •Build Environment
- •Embedded System Support

Provider Module Grouping

- Added OpenPegasus 2.11
  - See PEP 356
  - Backported to 2.10 and 2.9.2
- Functionality
  - Allows execution of multiple provider modules under single out-of-process agent process
  - Grouping can be defined as part of the provider registration or dynamically
  - New option in cimprovider (-g) sets provider module group for a provider module
  - New property in PG_ProviderModule Class
    - `string ModuleGroupName`

Pegasus Build Environment

- Distributed in source form
- Supports Debug and Release Building
- Make fully integrated
  - Gnumake on all platforms
- Uses default CIMModel
  - Default version updated for each release
- Build controlled by env. variables
  - See Source files:
    - `doc/BuildAndReleaseOptions.html`

Build/test from tar

- Expand tar
- `> cd pegasus`
- Set configuration variables
- `>make world`

`OR`

- `>make clean;make`
- `>make tests`
- `>make servertests`

Build environment variables

- Env Variables control
  - Component location
  - Compile platform
  - Server functionality
    - Ex. SSL support, CQL, WS-Man, out-of-process providers, cmpi
  - Server Alternative implementations
    - Ex. Repository type (xml, binary, SQLLite)
  - Build type
    - Release, debug, etc.
  - Internal Parameters
    - Cache sizes, etc.
  - Security
    - Provider security levels
  - Test Options
    - Parameters for post-build tests
- Env variable Presets
  - Files control some presets for particular platforms.
    - Ex. env_var_Linux.status
- There are a lot of options today

OpenPegasus and Embedded Systems

- Embedded System Significant Characteristics
  - Resources (cpu, memory, disk)
    - Limited resources
    - Hard limits rather than soft limits
  - Administration Issues
    - Often limited
    - Typically remote
    - Often Specialized
    - Sometimes OS Limited
  - Deployment model
    - Software Deployed with hardware
    - Complete Deployment (no add-ons post delivery
    - Minimal updates (replace everything)
  - High Availability
    - Expected to run without restarts, etc.
  - Management Integrated with OS and other Apps
  - Support a limited set of profiles
    - Specific management goals
  - Tied to specific hardware
  - Deterministic operation
    - Embedded systems want to be sure everything works.
  - OS's are often limited
    - Simplified Interfaces
    - Simplified concepts of users and security

- OpenPegasus Issues
  - Server Resource Utilization
    - Static - big
    - Dynamic – No limits
  - Disk utilization
    - extensive
  - Server Performance
  - Administration
    - Based on local admin model
  - Deployment model
    - Server based deployment
  - Modularity and Flexibility
  - Supporting split development environment

Embedded CIM Server Size

- Static Object Code Size
  - Issue
    - Server was 7 – 9 MB
    - Multiple Shared Libraries
  - Solution
    - Static Build
      - Reduce server to 3–5 MB (With memory repository)
    - Function build configurability. Eliminate unused Server components
      - Not everybody requires the complete server

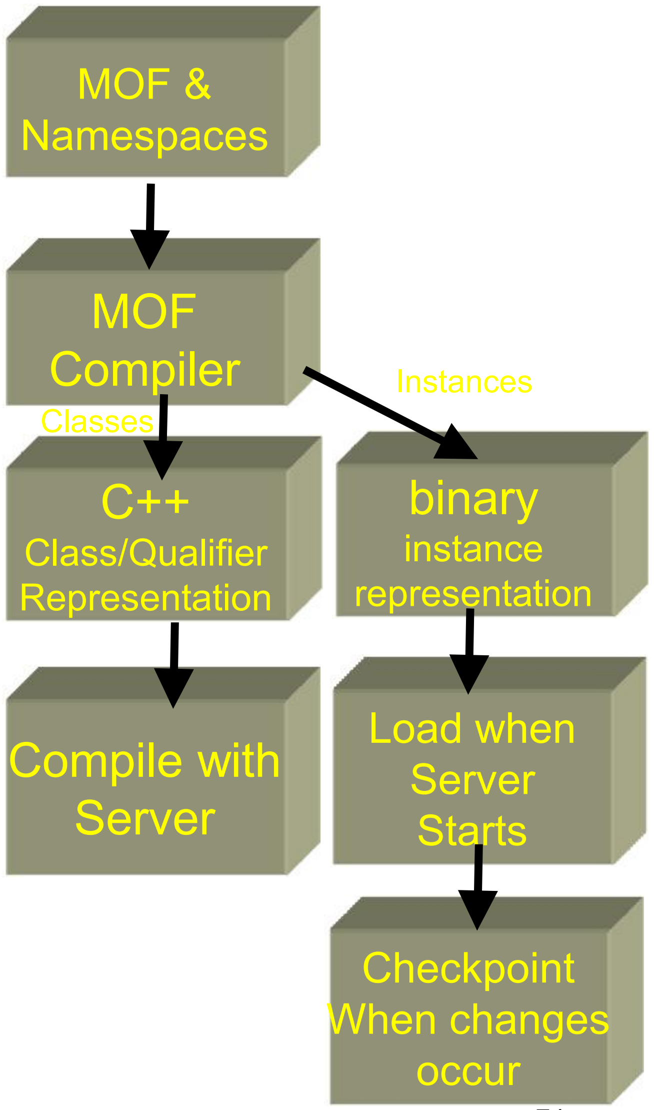
- Dynamic Memory Usage
  - Issue
    - Limit dynamic memory use
    - Control limits of memory use
  - Solution
    - Add memory limits to allocator
    - Control execution of operations / indication flow
  - Note: Pull operations will help this also

Disk Footprint

- Issue
  - Currently large footprint with many shared libraries
  - Difficult to separate server components from other build components
  - Large disk footprint for repository (~ 20 MB)
- Solutions
  - Reduce footprint by building a single image server (on single file)
  - Modify build process to allow build of components rather than simply the whole environment
  - Create much smaller repository representation
    - i.e. memory-resident repository

Memory Resident Repository

- **Goal**
  - Class and instance repository independent of disk files
  - Significantly reduce size of class repository
  - **Disallow** schema modification (no create class …)
- **Implementation**
  - Class repository
    - MOF compiler compiles c++ code representing class repository.
    - Code linked into embedded system
    - Memory-resident repository implementation converts to internal CIMClass form
    - Class closure filtering.
      - Compile from leaf classes using only required superclasses
  - Instance Repository
    - Instance repository is memory only.
    - Load and checkpoint functions to restore and save memory-resident instance repository
      - Implement as user definable callbacks
    - Initial instances can be created with MOF compiler
    - Potential to reduce size by maintaining internally in serialized form.
- **Performance**
  - Class repository size about 5% of disk repository.
    - 1.2 MB for complete repository vs. 20MB on disk
    - < .5 MB with Description Qualifiers removed
  - Performance – Faster but no real metrics to date



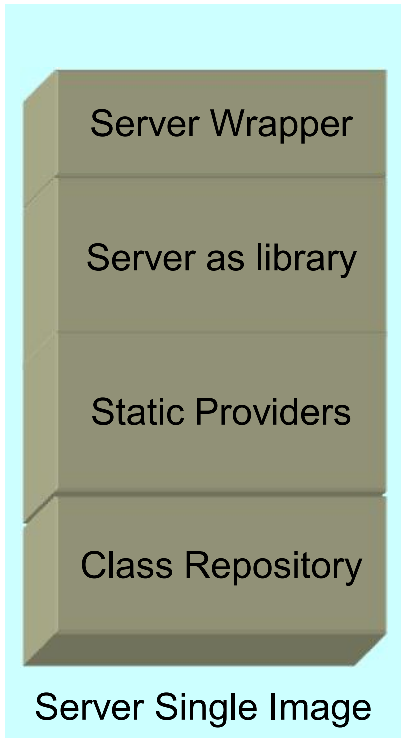
MOF & Namespaces
MOF Compiler
Instances
Classes
C++ Class/Qualifier Representation
binary instance representation
Compile with Server
Load when Server Starts
Checkpoint When changes occur

Server Performance

- Issues
  - Embedded CPUs often very slow
  - Performance issues become much more obvious with embedded systems
- Solutions
  - Continuous work on performance improvement
    - 15+ times speed up starting with Version 2.5.1
    - Additional performance increases in 2.6 and 2.7, 2.8
- Goal
  - Continued work on performance
    - Code improvement, algorithm improvement

Server Size Reduction

- Static server linking
  - Eliminate unused code
  - Static code size is smaller
- Move unused functionality to conditional compile
- Today
  - Capable of 5.5 MB server image with memory-resident repository (~ 4 MB without repository) (10 MB with multiple providers)
- Embedded system developer
  - Writes wrapper
  - Compiles classes with memory resident repository option
  - Modifies Make to build the static structure



Static linked Providers

- Goal
  - Deterministic Providers
    - No loading / unloading
  - Single Image with no dynamic libraries
  - No dynamic provider installation/registration
- Implementation
  - CMPI / C++ providers integrated into static build.
  - Provider registration integrated into server startup
    - Eliminates at least some of registration functionality

Limited File System Support

- Issue
  - Embedded systems often have limited file systems and/or very little disk space
- Goal
  - Greatly reduce server dependence on file systems
  - Lower limit is no file system support
- Implementation
  - Memory resident repository
  - External management of Certificates, passwords, etc.
  - Callback functions for getting info on Certificates, passwords, instance persistence, etc.
  - Provide user based functions output functions for other file issues such as logging output, trace output, etc.
    - Embedded system developer handles I/O from the callbacks

CIM Server Management

- Issue
  - OpenPegasus administration today is extensive
    - Includes both configuration and dynamic parameterization
  - Based largely on local user interface
    - Root based administration and OpenPegasus admin tools
- Goals
  - Limit administration of the server
  - Move some functionality from CIM Server to environment
  - Fix most parameterization (build time)
- Typical dynamic functions in embedded system
  - User setup
  - SSL certificate mgt
  - Minimal dynamic parameters (ex. Traces, log levels, etc.)
- Move all dynamic admin functions to:
  - Adopter responsibility (ex. User management, cert management)
  - Remote administration (ex. Setting trace levels, etc.)

Externalize main()

- Issue
  - Embedded system additional and configuration functionality built-in rather than configured or parameterized
- Goal
  - Improve modifiability without integrator developer having to modify Pegasus released components
- Externalize main
  - Pegasus becomes library
  - Main is created by the integrator developer
    - Outside the Pegasus source release
  - Includes functions like:
    - Load memory-resident repository
    - Install call backs for log, trace, instance persistence, configuration, etc.
    - Provider static registration

Build Environment

- Extend build environment for split development (host and embedded system targets)
  - Selective component builds
    - i.e. Server build for target
    - MOF clients built in host
    - MOF compiled in host
  - Test functions in both host and target
  - Tests Run in combination of target and host

Things we would like to do

- Performance
  - Indication Processing
  - Association Operation Processing
- Footprint
  - More compile options on major components
- Functionality
  - Update to next CMPI spec version
  - Implement more Profiles
  - Enhance Compiler
    - Error Detection
    - Build repository from tail
- Usability
  - Linux type build configure
  - Reduce number of config variables
  - Improve provider debugging
- Miscellaneous
  - Improved adminstration

See wiki for working list of suggestions. Contribute to this list.

MANAGEMENT DEVELOPERS CONFERENCE

Part 1.5
Working With OpenPegasus And the Pegasus Project

Section 1.5

WORKING WITH OPENPEGASUS AND THE PROJECT

Working With the Pegasus Project

- **Using OpenPegasus Source Code**
  - Free for use. Multiple and growing number of sources for access to Pegasus
- **Contributing to the Project**
  - Outside contributors
    - In Company
    - Specific financed projects
    - Contribute via patches or authorized developers
  - Join or follow the PEPs and Architecture Team
    - No commitment to join required to participate
    - There is no free lunch.
  - Join the Steering Committee
    - Influences priorities, commitments, releases.

Sources for access to OpenPegasus

- OpenPegasus CVS
  - All Releases source code (By CVS tag)
  - Current unreleased work (head of tree)
- Integrated into specific OS releases
  - ZOs, HPUX, AIX, etc.
- Linux Source RPM's for releases
  - Pegasus web site
- Release source tarballs
  - Pegasus web site (tar and zip)
- Redhat AS (and Fedora)
  - Binary rpms

Getting Support

- Ask the Pegasus mailing Lists
- File Pegasus Bugs
  - And follow up
- Attend the Pegasus calls
  - Sqeaking wheels and all that blah
- Contract 3rd Party for support/maintenance

OpenPegasus Community structure and participants overview

OpenPegasus project participation is based on a meritocracy model with ballots for bugfix and design approvals by “recognised” voters

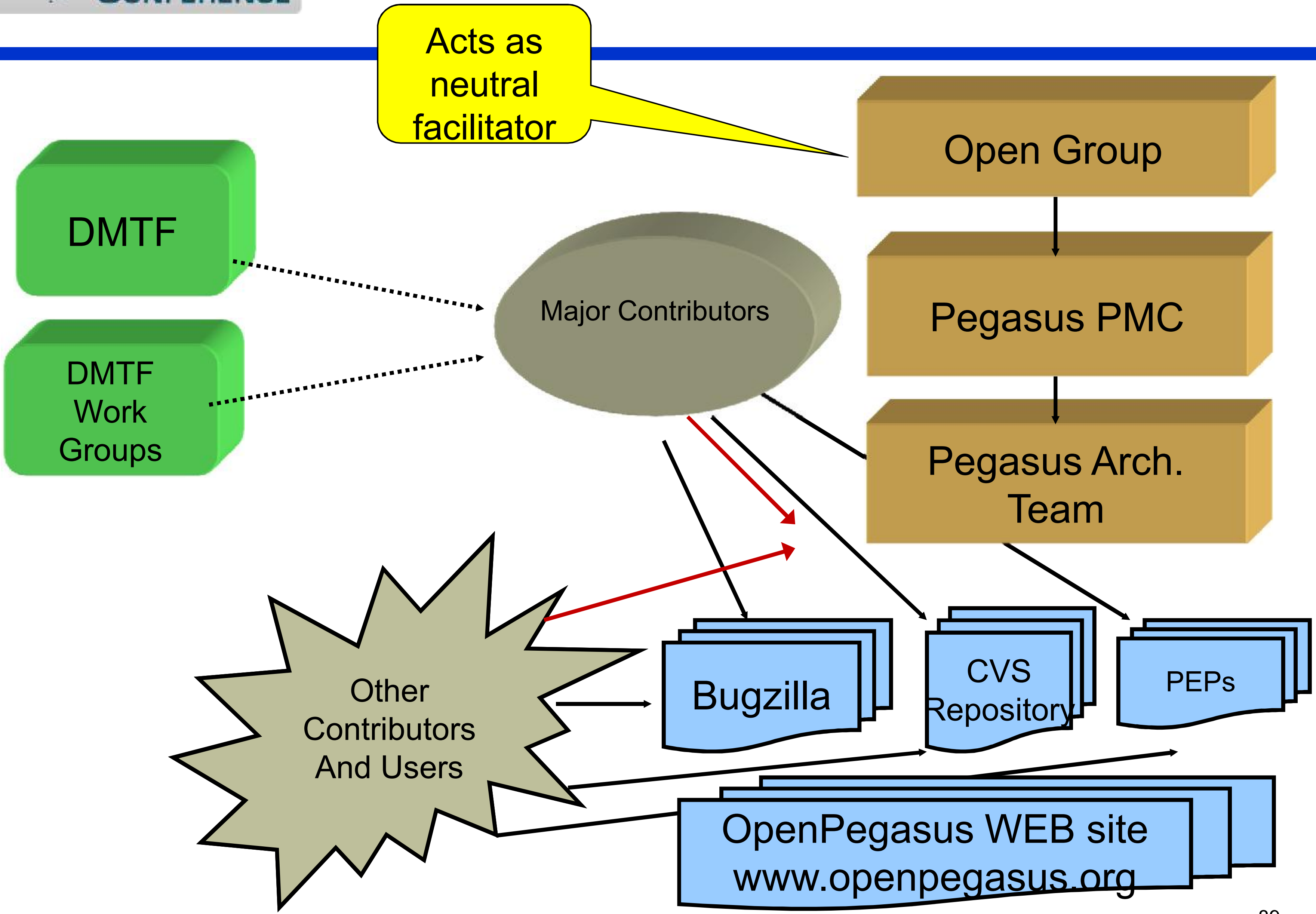
PMC (Project Management Committee)	•responsible for all technical aspects of the project •grants recognition by inviting contributers to become Committers •<u>Joining:</u>Invitation of new members to the PMC through agreement of existing PMC members (takes several years activity in the project and tech.expertise in several areas)
Committers	•responsible for sponsoring Bug fixes •have binding voter rights on design and bugfix decisions •heavy influence on future and direction of the OP project •<u>Joining:</u>Invitation of new members through PMC in recognition of expertise and commitment to the project (takes usually at least a year full activity in the project and tech.expertise in at least one area)
Architecture Team	•Design and Architecture team actively working on the OpenPegasus strategic and design decisions •regular Team meeting call •discusses fixes for Bugs (Bugzilla) •create and discuss Design (PEP) •<u>Joining:</u>Free, through joining “pegasus-architecture@openpegasus.org” mailing list
OpenPegasus Users	•Community of OpenPegasus users •mostly used for “self-support” between the group and release announcement •<u>Joining:</u>Free, through joining “pegasus-l@openpegasus.org” mailing list

Groups are based on each other, i.e. PMC members are always Committers etc.

<u>More details see:</u> http://www.openpegasus.org/pmc/
<u>as well as:</u> PEP#336 / PEP#337

©2012 Marek Szermutzky(IBM)

The Pegasus Project

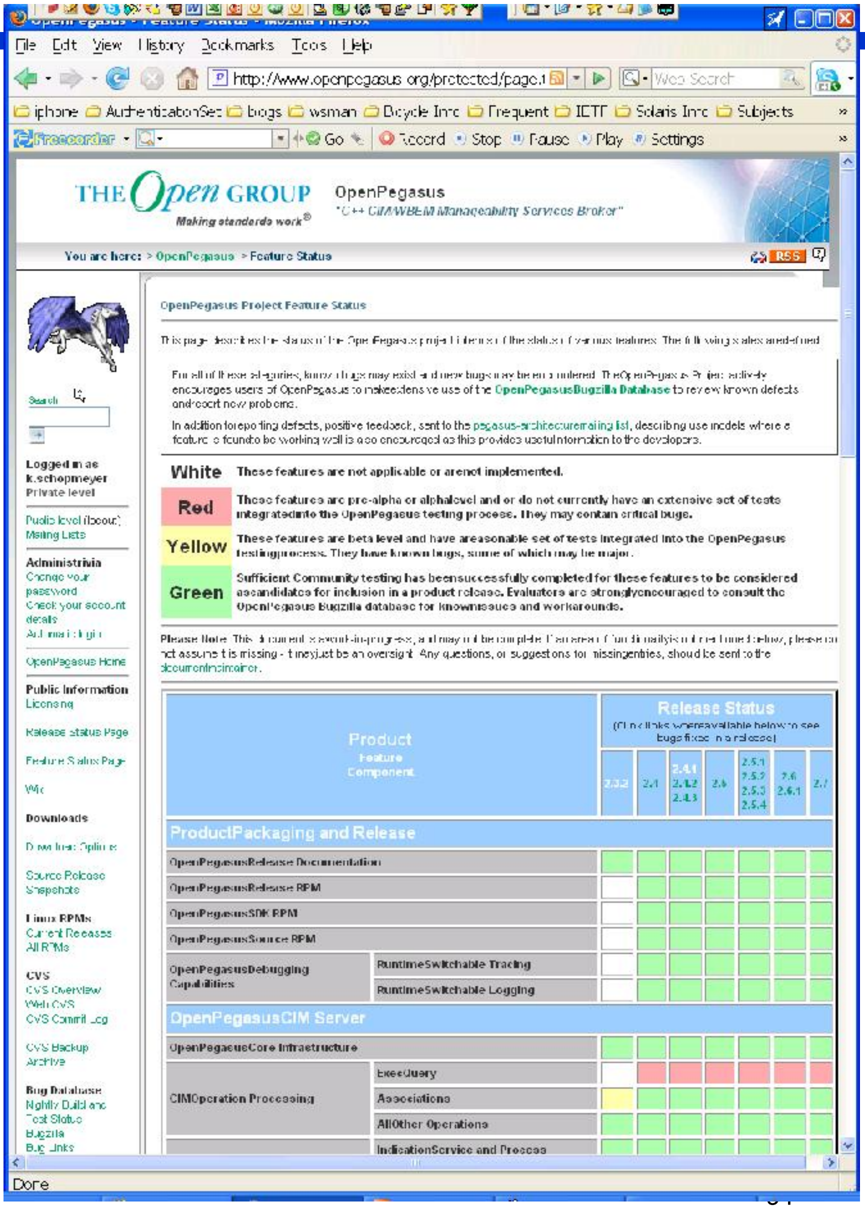


Relating Pegasus to work inside companies

- Pegasus is not a hacker project
- License accepted by major IT suppliers
  - We use MIT license for a reason
- Code investment by major IT suppliers
- Function and schedule driven by user needs
- Function driven by contributors
  - There is no magic set of hidden developers here

Pegasus Feature Status Information

- Documented in Features Page for each release
  - www.opengroup.org/ -> Feature Status Page
- Goal - summarize Features and Status
  - Status - functionality and Quality
    - Red – Alpha level not extensively tested
    - Yellow – Beta level, reasonable tests, outstanding bugs
    - **Green** – Candidates for inclusion in production release
    - White – Status Unknown
- Major Feature Categories Today
  - CIM Server
  - Repository
  - Provider Interface
  - Providers
  - Client Interfaces
  - Indication Listeners
  - Indication Handlers
  - Security
  - SLP
  - WMI Mapper
  - Packaging and Releases



You are here: > OpenPegasus > Feature Status

OpenPegasus Project Feature Status

This page describes the status of the OpenPegasus project in terms of the status of various features. The following scales are defined

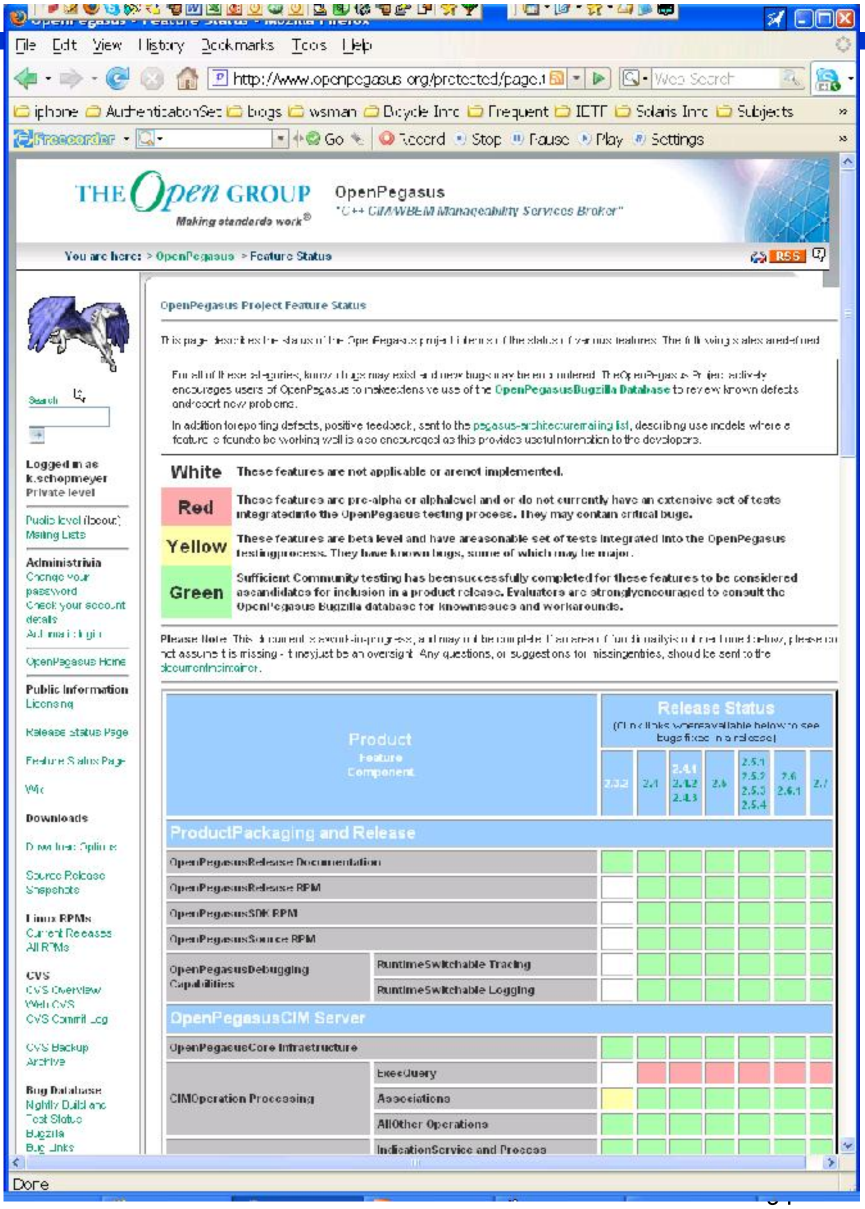
For all of these categories, known bugs may exist and new bugs may be encountered. The OpenPegasus Project actively encourages users of OpenPegasus to make extensive use of the OpenPegasus Bugzilla Database to review known defects and report new problems.

In addition to reporting defects, positive feedback, sent to the pegasus-architecture mailing list, describing use models where a feature is found to be working well is also encouraged as this provides useful information to the developers.

White	These features are not applicable or are not implemented.
Red	These features are pre-alpha or alpha level and or do not currently have an extensive set of tests integrated into the OpenPegasus testing process. They may contain critical bugs.
Yellow	These features are beta level and have a reasonable set of tests integrated into the OpenPegasus testing process. They have known bugs, some of which may be major.
Green	Sufficient Community testing has been successfully completed for these features to be considered as candidates for inclusion in a product release. Evaluators are strongly encouraged to consult the OpenPegasus Bugzilla database for known issues and workarounds.

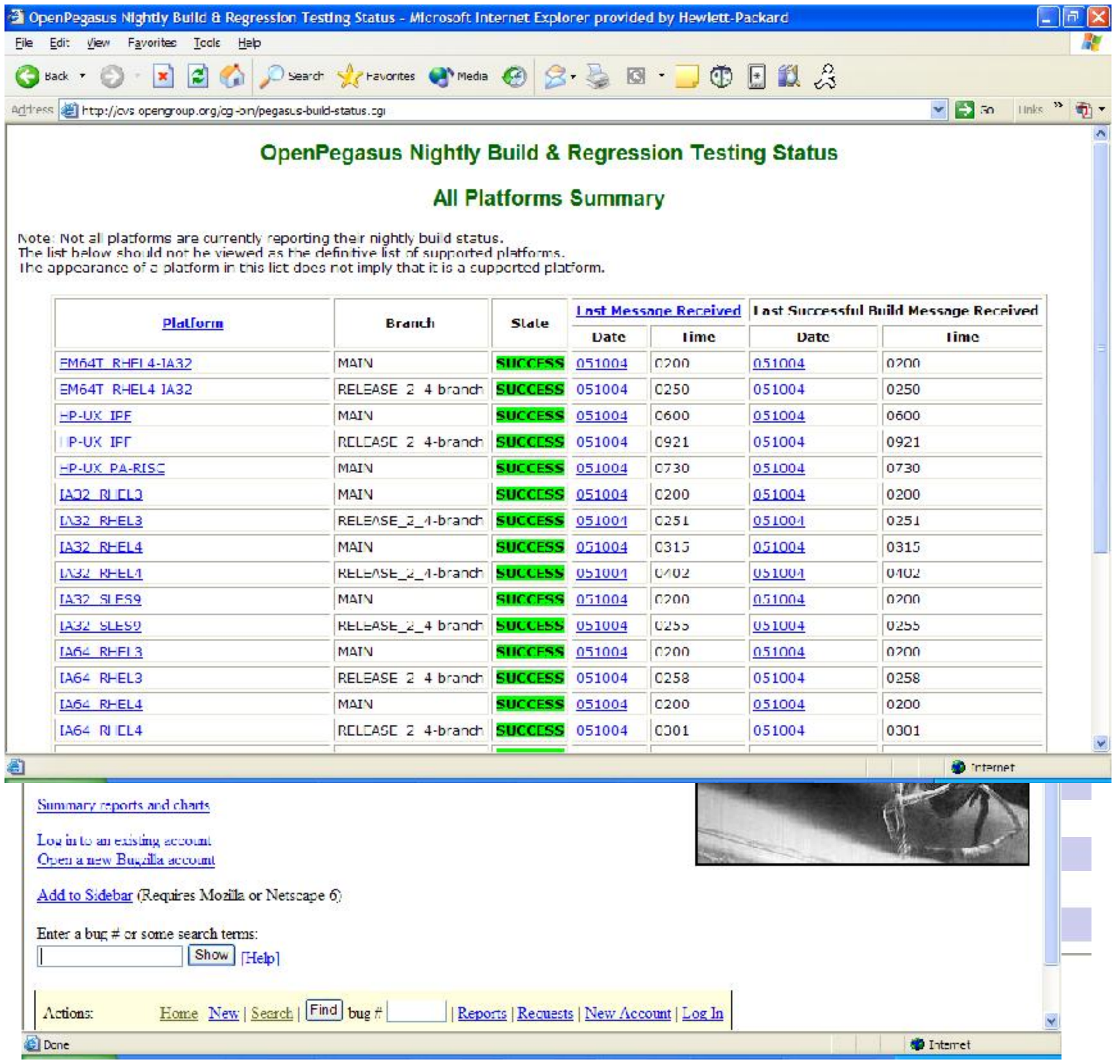
Please Note: This document is a work-in-progress, and may not be complete. If an area of functionality is not mentioned below, please do not assume it is missing - it may just be an oversight. Any questions, or suggestions for missing entries, should be sent to the document maintainer.

Product Feature Component		Release Status (Click links where available below to see bugs fixed in a release)						
		2.3.2	2.4	2.4.1 2.4.2 2.4.3	2.5	2.5.1 2.5.2 2.5.3 2.5.4	2.6 2.6.1	2.7
ProductPackaging and Release								
OpenPegasusRelease Documentation		🟩	🟩	🟩	🟩	🟩	🟩	🟩
OpenPegasusRelease RPM		⬜	🟩	🟩	🟩	🟩	🟩	🟩
OpenPegasusSDK RPM		⬜	🟩	🟩	🟩	🟩	🟩	🟩
OpenPegasusSource RPM		⬜	🟩	🟩	🟩	🟩	🟩	🟩
OpenPegasusDebugging Capabilities	RuntimeSwitchable Tracing	⬜	🟩	🟩	🟩	🟩	🟩	🟩
	RuntimeSwitchable Logging	⬜	🟩	🟩	🟩	🟩	🟩	🟩
OpenPegasusCIM Server								
OpenPegasusCore Infrastructure		🟩	🟩	🟩	🟩	🟩	🟩	🟩
CIMOperation Processing	ExecQuery	⬜	🟥	🟥	🟥	🟥	🟥	🟥
	Associations	🟨	🟩	🟩	🟩	🟩	🟩	🟩
	AllOther Operations	🟩	🟩	🟩	🟩	🟩	🟩	🟩
	IndicationService and Process	🟩	🟩	🟩	🟩	🟩	🟩	🟩



Working on the Pegasus Project

- Working with the Code
  - CVS, snapshots
- Documentation
  - API documentation
  - PEPs
  - Readme documents
- Understand releases & state of Pegasus
  - Nightly build status, bugs, release definition PEPs, ViewCVS, Blocker bug list
- Understanding and future directions
  - Release Definition PEPs
- Contributing Bugs and Corrections
  - OpenPegasus bugzilla
  - Team Reviews
- Contributing New Functionality
  - Define with PEPs
  - Team Review
- Defining future "Requirements"
  - Get Involved


OpenPegasus Nightly Build & Regression Testing Status - Microsoft Internet Explorer provided by Hewlett-Packard
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://cvs.opengroup.org/cgi-bin/pegasus-build-status.cgi
Go Links
Internet

OpenPegasus Nightly Build & Regression Testing Status

All Platforms Summary

Note: Not all platforms are currently reporting their nightly build status.
The list below should not be viewed as the definitive list of supported platforms.
The appearance of a platform in this list does not imply that it is a supported platform.

Platform	Branch	State	Last Message Received		Last Successful Build Message Received	
			Date	Time	Date	Time
EM64T_RHEL4-IA32	MAIN	SUCCESS	051004	0200	051004	0200
EM64T_RHEL4_IA32	RELEASE_2_4 branch	SUCCESS	051004	0250	051004	0250
HP-UX_IPF	MAIN	SUCCESS	051004	0600	051004	0600
HP-UX_IPF	RELEASE_2_4-branch	SUCCESS	051004	0921	051004	0921
HP-UX_PA-RISC	MAIN	SUCCESS	051004	0730	051004	0730
IA32_RHEL3	MAIN	SUCCESS	051004	0200	051004	0200
IA32_RHEL3	RELEASE_2_4-branch	SUCCESS	051004	0251	051004	0251
IA32_RHEL4	MAIN	SUCCESS	051004	0315	051004	0315
IA32_RHEL4	RELEASE_2_4-branch	SUCCESS	051004	0402	051004	0402
IA32_SLES9	MAIN	SUCCESS	051004	0200	051004	0200
IA32_SLES9	RELEASE_2_4 branch	SUCCESS	051004	0255	051004	0255
IA64_RHEL3	MAIN	SUCCESS	051004	0200	051004	0200
IA64_RHEL3	RELEASE_2_4 branch	SUCCESS	051004	0258	051004	0258
IA64_RHEL4	MAIN	SUCCESS	051004	0200	051004	0200
IA64_RHEL4	RELEASE_2_4-branch	SUCCESS	051004	0301	051004	0301

Summary reports and charts

Log in to an existing account
Open a new Bugzilla account

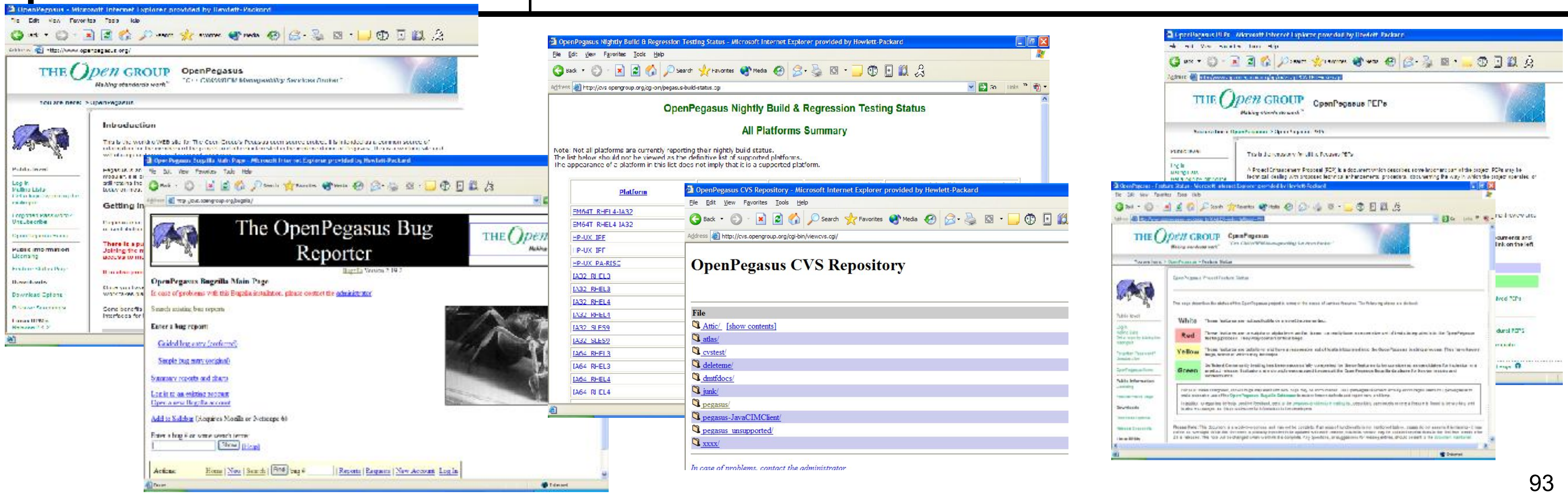
Add to Sidebar (Requires Mozilla or Netscape 6)

Enter a bug # or some search terms:

Show
[Help]
Actions: Home | New | Search | Find bug # | Reports | Requests | New Account | Log In
Done
Internet

Getting More Information

OpenPegasus Home	http:// http://www.openpegasus.org
OpenPegasus CVS	http:// cvs.opengroup.org/cgi-bin/viewcvs.cgi/
OpenPegasus Bugzilla	http:// cvs.opengroup.org/bugzilla/
OpenPegasus Build Status	http:// nbat.openpegasus.org
OpenPegasus Documentation	http://www.openpegasus.org/pp/index.tpl
OpenPegasus Email Lists	http://www.openpegasus.org
OpenPegasus Feature Status	http://www.openpegasus.org/page.tpl?ggid=799


The OpenPegasus Bug Reporter

OpenPegasus Nightly Build & Regression Testing Status

All Platforms Summary

OpenPegasus CVS Repository

Section 1.6

ISSUES

Known Technical Issues

- Pegasus Provider Registration proprietary
  - Will fix when DMTF provider registration profile completed
- Statistics Model broken
  - Does not work with pull operations
- Some behavior differences (discussed above)
- Does not comply with hidden property requirement
- No Client for WSMAN

Issues List we keep hearing

- **We react/move too slowly**
  - Only through process can we control quality, schedules, etc.
  - Pegasus is a project that must meet user demands and schedules if it is to continue
  - It is the level of involvement that drives Pegasus, not the level of wishes
- **Releases are not frequent enough**
  - Trying to balance of quality releases with reasonable development groups
  - Train release mechanism costs time but imposes quality control
- **Too much process**
  - Without process we don't know where we are or where we are going
- **Pegasus is too:**
  - *Slow, big, incomplete, small*, etc.
    - Continuous a) refactoring, b) performance work, c) new functionality
    - We can only implement what someone commits to do.
- **Pegasus does not do what I want**
  - Things only get done through people that do them (see below)
- **Pegasus not true open source**
  - Work with us. You can contribute. You can vote.
  - Openness takes time also
  - Moving to Open Source PMC, meritocracy based model now
- **Somewhere there is a magic set of developers**
  - Effectively a volunteer organization. What you see is what you get
- **Documentation sucks**
  - Again we only get done what someone will do.
- **There is a magic group somewhere (i.e. OpenGroup) developing for Pegasus**
  - Whoops, Pegasus is Open Source and volunteer among interested parties

All of these are open for discussion

How we decide what gets done

- **Somebody needs it**
- **Somebody is willing to do it**
  - Document the requirement and function
  - Do the code
  - Integrate it
  - Provide test environment
- **It is consistent with the project goals**
  - Architecture, risk, quality, . . .

OpenPegasus in one page

- All major WBEM components
  - (server, client/listener infrastructure, compilers, some providers, test suite, CQL, WQL, Indication Support, security
- Project
  - Community project under auspices of The Open Group
  - Major contributors, HP, IBM, Symantec, EMC, Novell, Sun, Microsoft
  - Project Lead– The Open Group
- Regular Releases
  - ~ 9 month cycle
- Availability
  - Source (cvs, rpms, tar balls)
  - Binaries for Linux (RedHat and SUSE distributions)
- Major users
  - HP, IBM, Symantec, EMC
  - Multiple other SNIA SMIs server implementers.
- Platform Target
  - Initially broad set of OS/Platforms
  - Now adding embedded system support

- Platforms Supported
  - Linux, Unix, Mac, Windows, VMS, ZOS, VxWorks (planned)
- License
  - MIT License
- Provider Types
  - Pegasus C++
  - CMPI
  - Java (SNIA Provider Interface today)
- Development Language
  - C++
- Client API Language
  - C++
  - Java
- Client Protocols
  - CIM/;XML
  - WS-Man

Questions & Discussion

?

We would like to get your feedback on issues, priorities, users/usage, requests for OpenPegasus.
Email, Attend Architecture Meeting, bugs, etc.