



# Enterprise Workload Manager Overview and Implementation

Silvio Sasso  
IBM ITS Delivery for z/OS  
[sisas@ch.ibm.com](mailto:sisas@ch.ibm.com)

# Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.

CICS*	GDPS*	NetView*	System z9
DB2*	HiperSockets	OMEGAMON*	SystemPac*
DB2 Connect	HyperSwap	OS/390*	Tivoli*
DB2 Universal Database	IBM*	Parallel Sysplex*	TotalStorage*
DFSMSdss	IBM eServer	PR/SM	VTAM*
DFSMSrmm	IBM logo*	Processor Resource/Systems Manager	WebSphere*
DFSMSshsm	IMS	RACF*	z/Architecture
DFSORT	InfoPrint*	RMF	z/OS*
DRDA*	Language Environment*	S/390*	z/VM*
Enterprise Storage Server*	Multiprise*	Sysplex Timer*	z/VSE
ESCON*	MVS	System z	zSeries*
FICON*			

\* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.

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

Linux is a trademark of Linus Torvalds in the United States and other countries..

UNIX is a registered trademark of The Open Group in the United States and other countries.

Microsoft and Excel are registered trademarks of Microsoft Corporation in the United States and other countries.

\* All other products may be trademarks or registered trademarks of their respective companies.

## Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

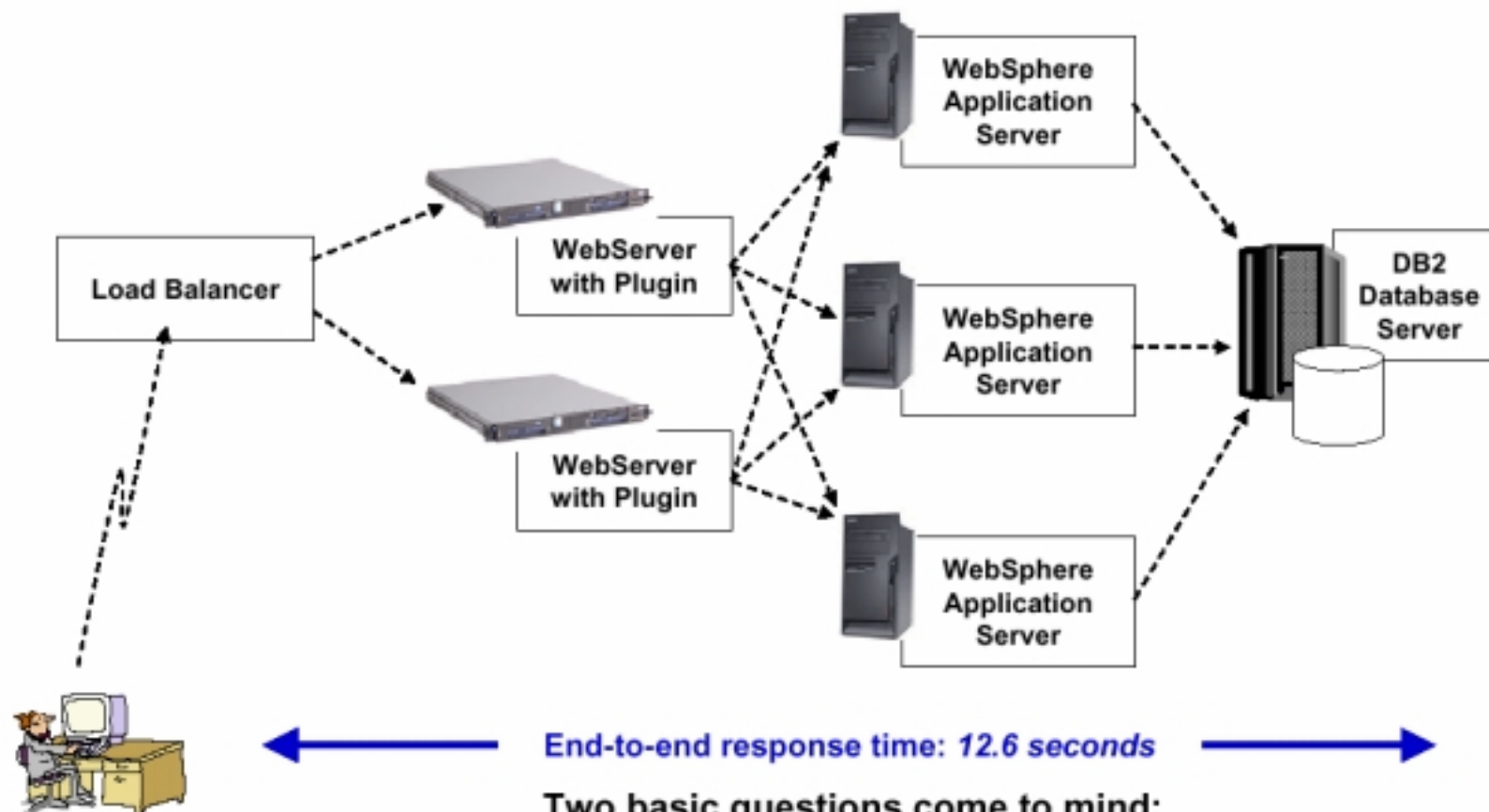
# Agenda

- ▶ **EWLM Introduction and Overview**
- ▶ **Implementation and Customization**
- ▶ **Documentation and Reference Material**
- ▶ **Summary**



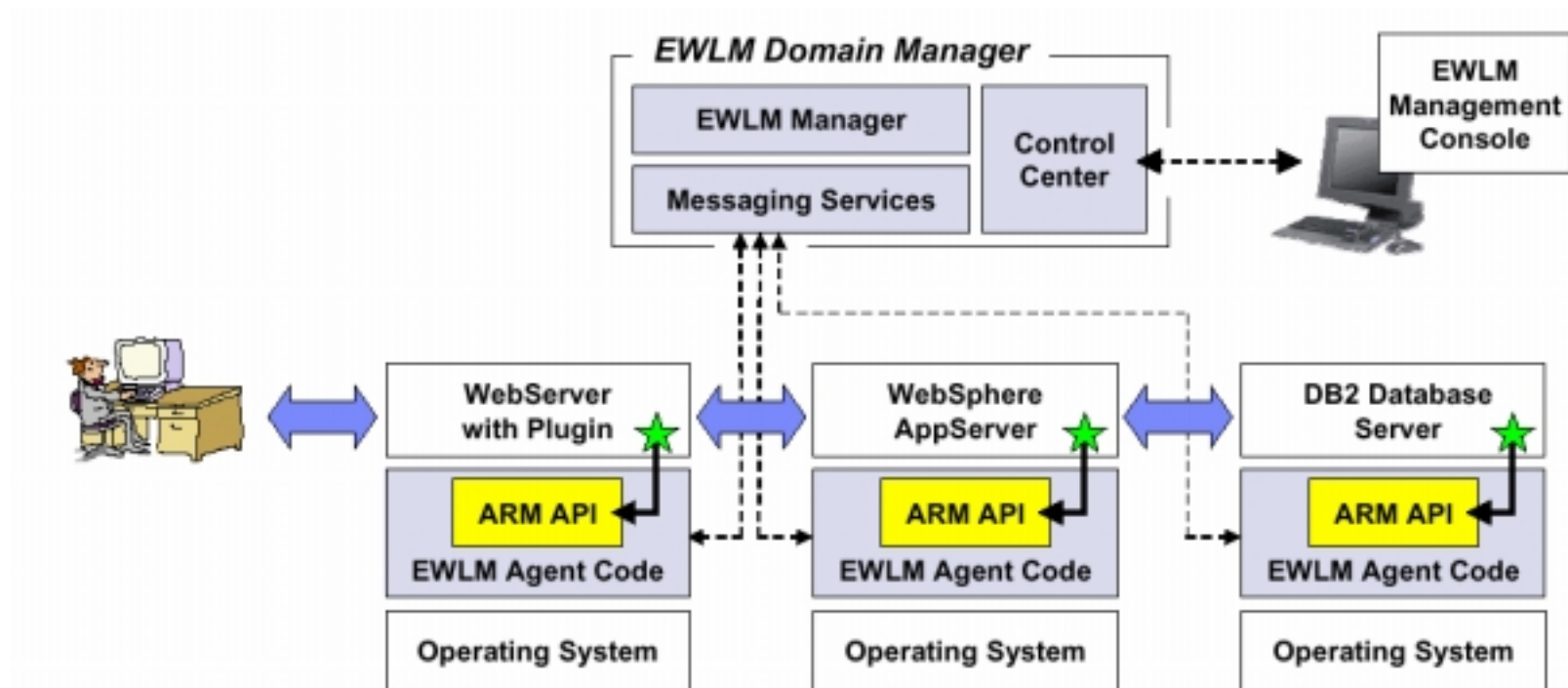
## **EWLM Introduction and Overview**

# Why EWLM? – The basic Issue



Two basic questions come to mind:  
 1. Is that a problem?  
 2. If so, *where's* the problem?

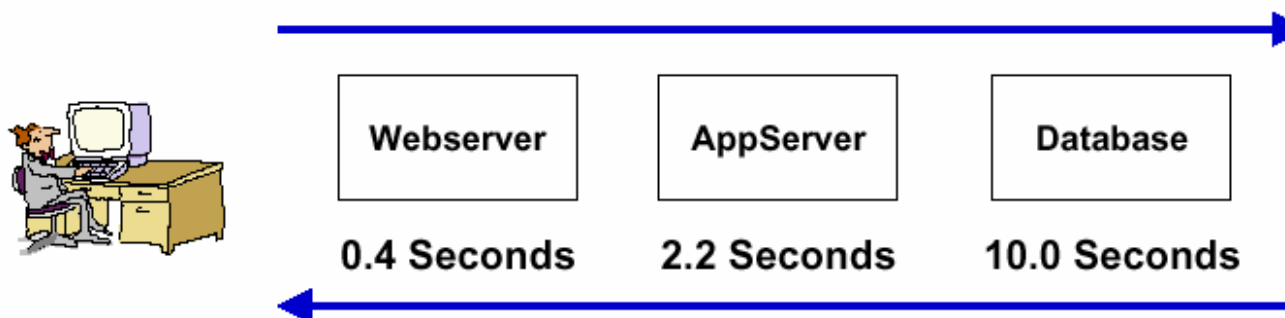
# EWLM – High Level View



## Key Points:

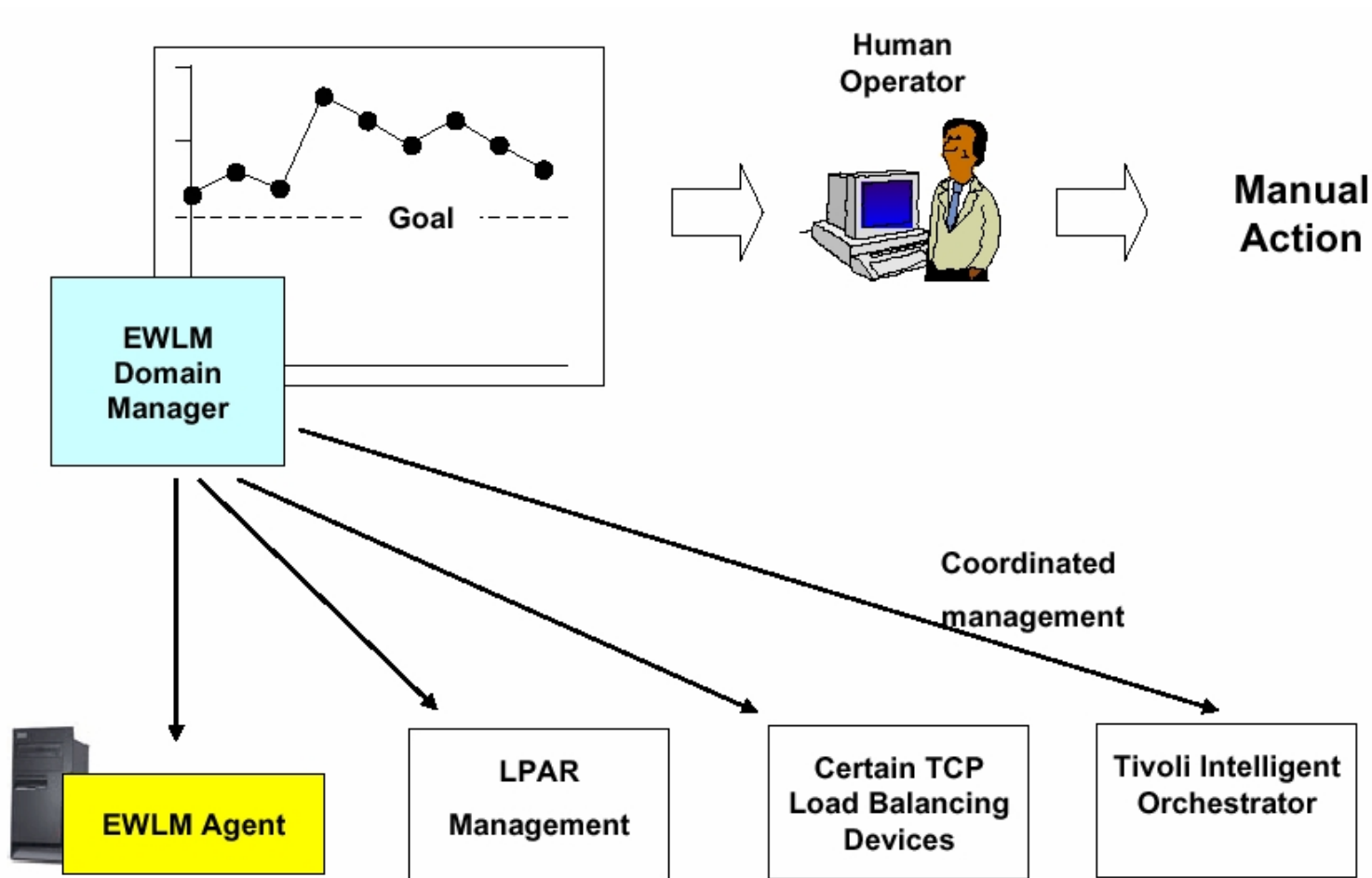
- **EWLM Agent code installed on the servers**
- **Industry standard ARM API part of agent code**
- **Middleware "instrumented" to make use of ARM API**
- **Application "hops" timestamped**
- **Agents feed collected information up to Domain Manager**
- **Domain Manager collects and groups for reporting**

## Information gathered ... do we have a Problem?



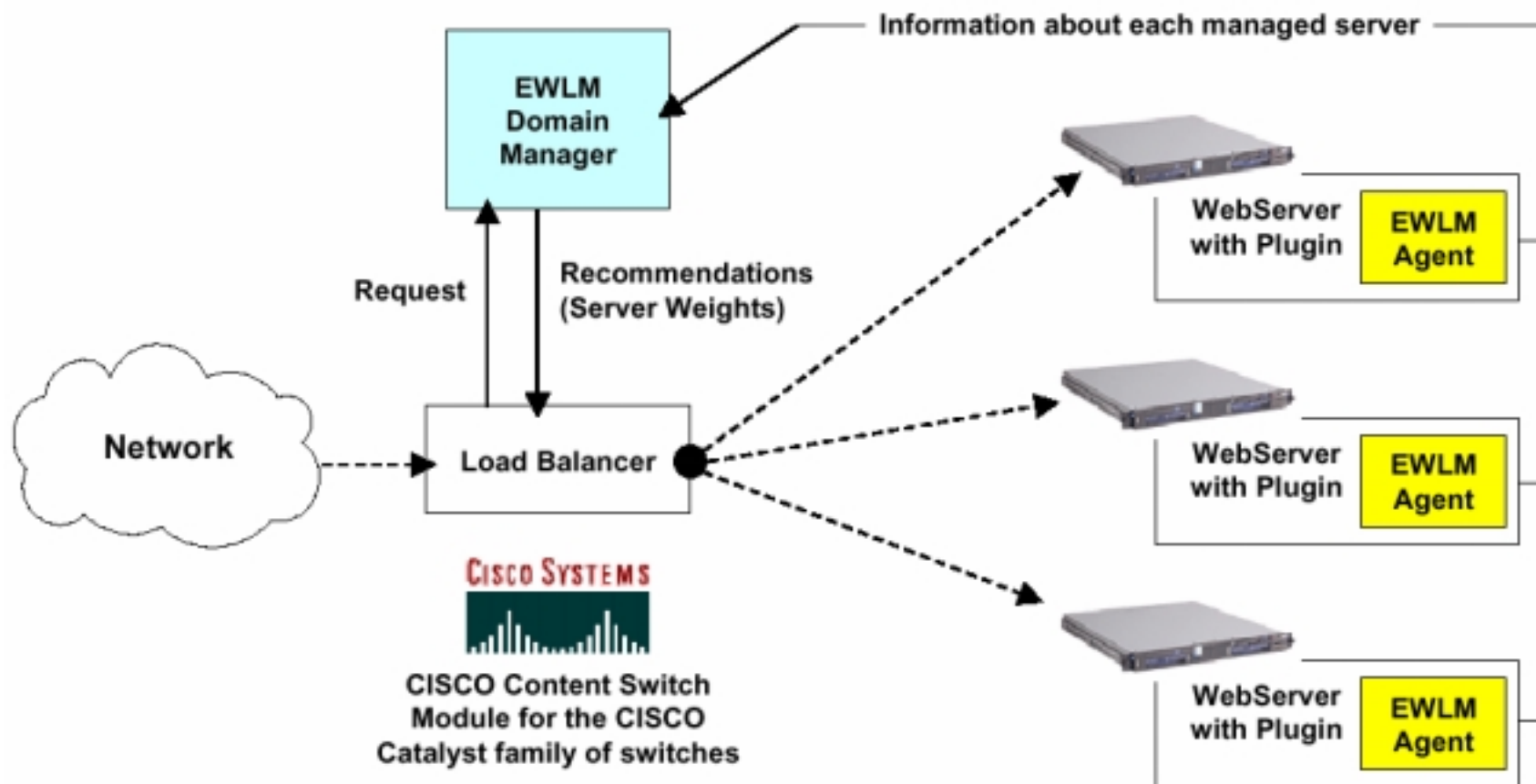
- **It depends...**
  - ▶ Is it a critical application?
  - ▶ Is 12.6 seconds good or bad for this application?
  - ▶ Is this 12.6 second response time typical?
  - ▶ Or does this usually run quickly but had a one-time bad response?
  - ▶ Is there something else running that's even more important than this application?
  - ▶ Is that more critical thing consuming resources?
  
- **To answer these questions we have to compare the response times seen to defined goals to know whether there's a problem or not**

# Problem identified – simply report it or do something about it?





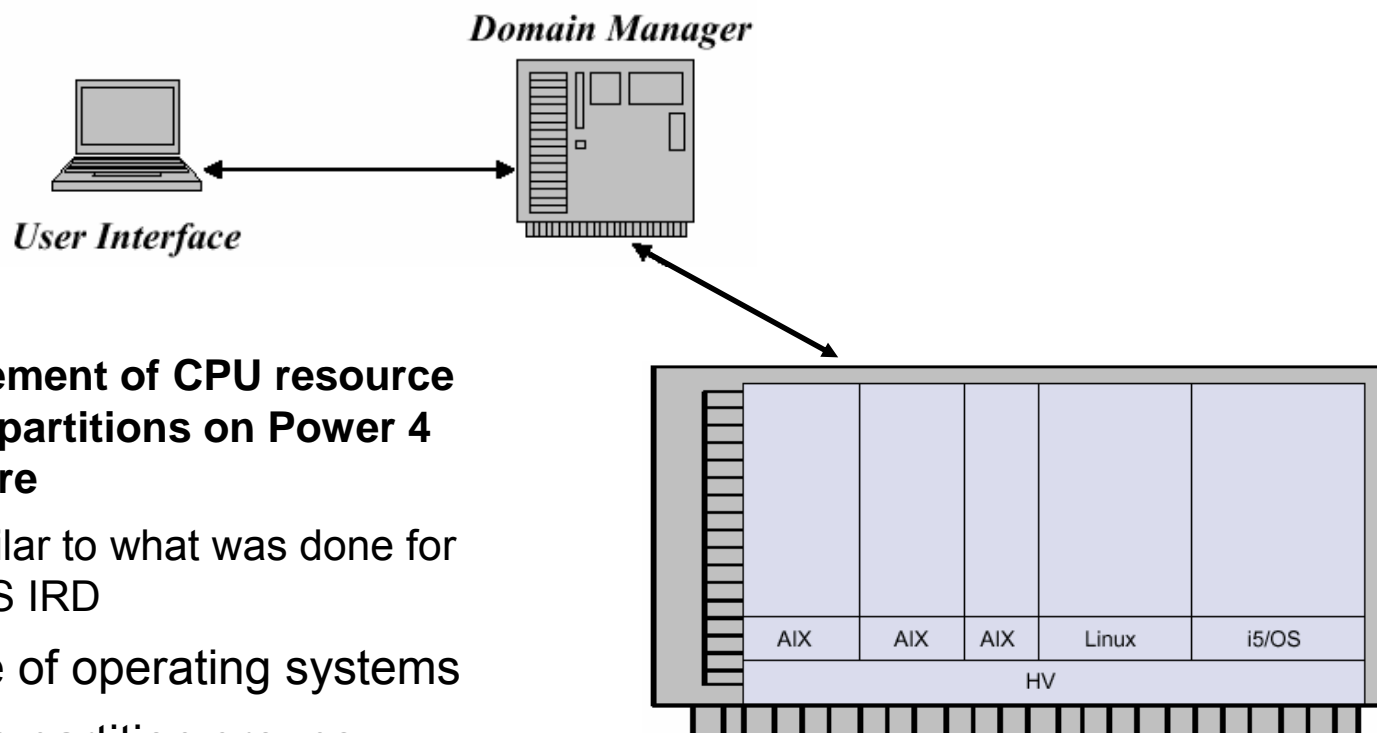
# EWLM Workload Management – Load Balancing



**EWLM provides recommendations**

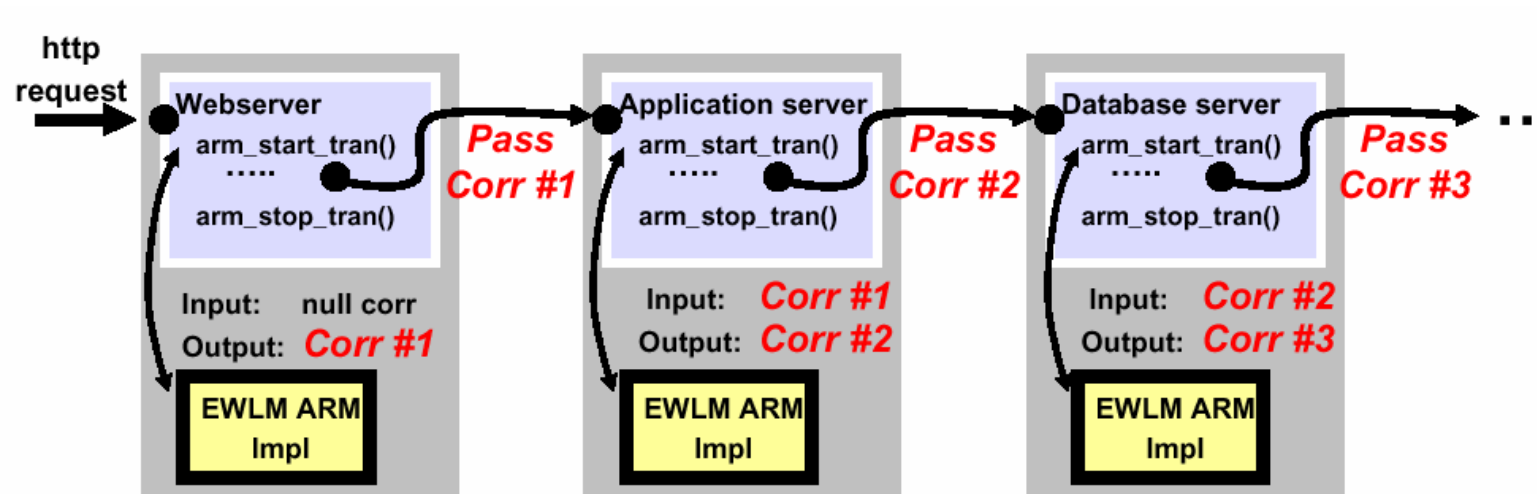
**Load balancer modifies behavior based on the recommendations provided**

# EWLM LPAR Management



- **Management of CPU resource across partitions on Power 4 hardware**
  - ▶ Similar to what was done for z/OS IRD
- Mixture of operating systems
- Multiple partition groups
- Multiple EWLM domains allowed

# EWLM ARM Correlation Implementation



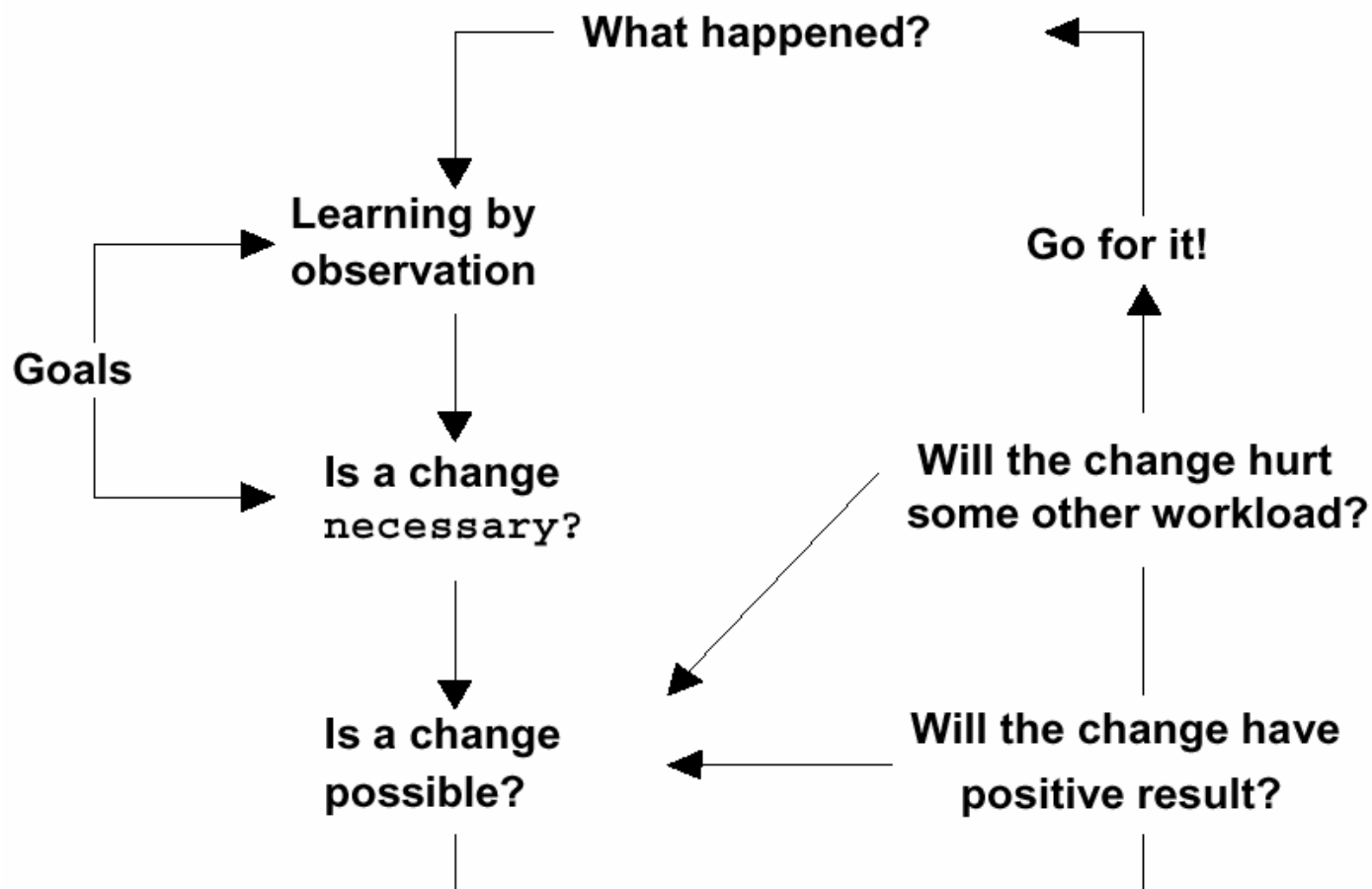
- EWLM arm\_start\_transaction() will:

- ▶ Accept an input **parent** ARM correlator – reflects transaction upstream processing
- ▶ Construct a **current** ARM correlator to represent the transaction instance to downstream applications and middleware

Application Response Measurement (ARM):

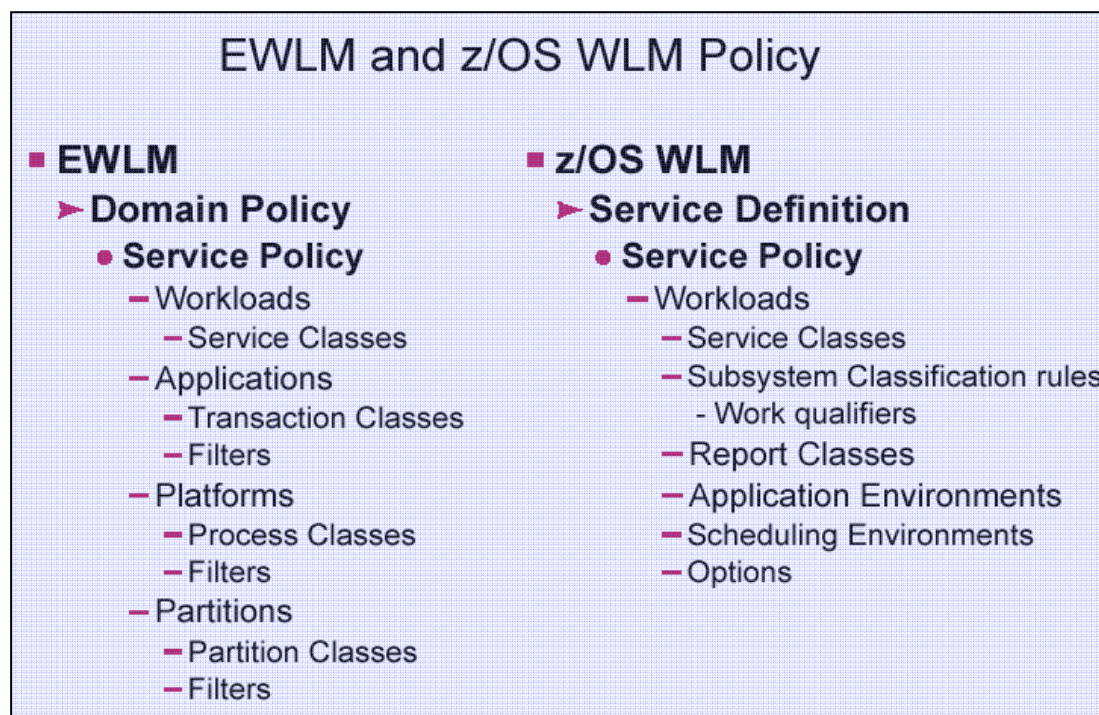
ARM is an Open Group standard composed of a set of APIs that can be used to collect response time information for enterprise applications

# Learning from z/OS WLM – Goal oriented Resource Management



# Workload Management on zSeries

- **EWLM and z/OS WLM policy structures are conceptually the same**
  - ▶ The main difference is related to the work managers and to the logical components that receive and submit the work requests
  - ▶ As work managers, EWLM supports applications and z/OS WLM platforms, and zWLM supports several subsystems, including CB, CICS®, DDF, and IMS™

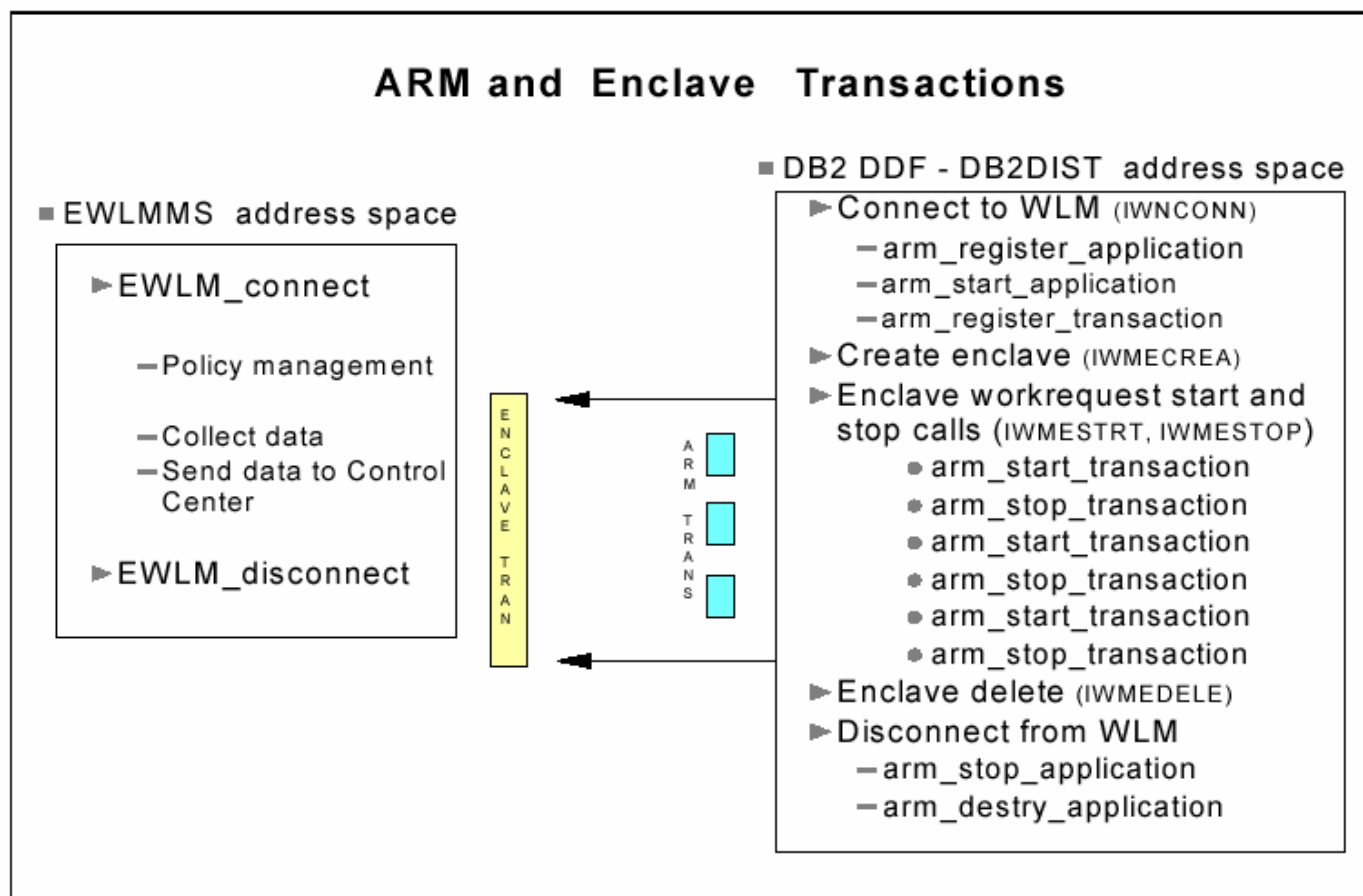


# Workload Management on zSeries

- **To associate a work request to a service class, a work manager passes some attribute values that EWLM matches against filters and /OS WLM matches against subsystem work qualifiers**
- **This process is called *classification***
- **The attribute values for work requests received by the entry application of an EWLM Management Domain might be different from those passed to z/OS WLM**
  - ▶ especially if the classification on the distributed platform happened in an application environment different from the entry point on z/OS
  - ▶ For this reason it may not always be possible to have a one-to-one mapping for EWLM service classes and z/OS WLM service classes
- **ARM enablement on z/OS**
  - ▶ Workload Manager on z/OS has provided workload management services to middleware such as DB2, WebSphere, CICS, and IMS for a long time
  - ▶ For example, DB2 DIST uses WLM enclave services to schedule and process received work requests
  - ▶ In the current implementation, EWLM on z/OS leverages the existing WLM services and maps them to ARM calls, so that an EWLM managed server can simply pick up the performance data and send it to the domain manager

# Workload Management on zSeries

## ▶ ARM enablement on z/OS - General Flow



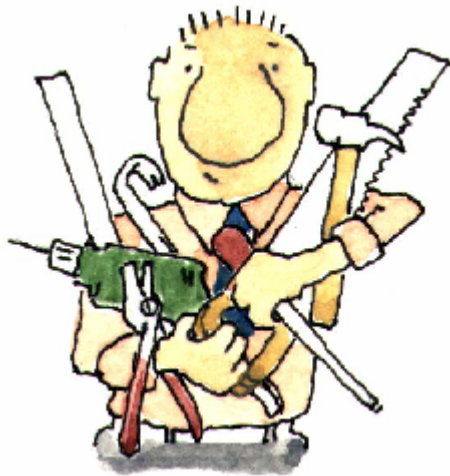
# Workload Management on zSeries

- **When the EWLM address space is started it connects to WLM. Its main functions are:**
  - ▶ Manage the EWLM policy received from the domain manager
  - ▶ Collect the performance data and send it to the domain manager
- **When DB2 is started it connects to WLM and specifies that it will participate in EWLM**
  - ▶ The DB2 DIST address space is then registered to ARM on z/OS as an EWLM application
- **ARM data is captured only if a correlator is sent in from an ARM-enabled distributed application such as WebSphere**
  - ▶ otherwise, DB2 continues processing distributed requests as it usually does
- **When DB2 receives the correlator from a distributed application, DB2 DDF creates an enclave and starts a work request**
  - ▶ Attribute values such as PLAN, PACKAGE, and CONNECTION are passed with the correlator to zWLM to classify the work request and associate it to a service class



## Workload Management on zSeries

- **When an enclave is created by DDF, an `arm_start_transaction` is implicitly invoked to mark the beginning of a transaction, and then an `arm_stop_transaction` gets called at the end of processing a database request message**
  - ▶ A pair of start and stop transaction is associated with each database request message from a distributed application or DRDA® requester until a commit is received and the enclave is then deleted
  - ▶ In other words, in an EWLM environment, a DB2 transaction no longer has the same meaning as one enclave transaction, or a unit-of-work between two DB2 commits
  - ▶ In the case described, a transaction becomes an ARM work request, and its elapse time is simply the duration of receiving and processing a database request message to DB2 from the DRDA requestor



## **EWLM Implementation and Customization**

© 2006 IBM Corporation

# IBM Systems

# Platform Support in EWLM V2.1

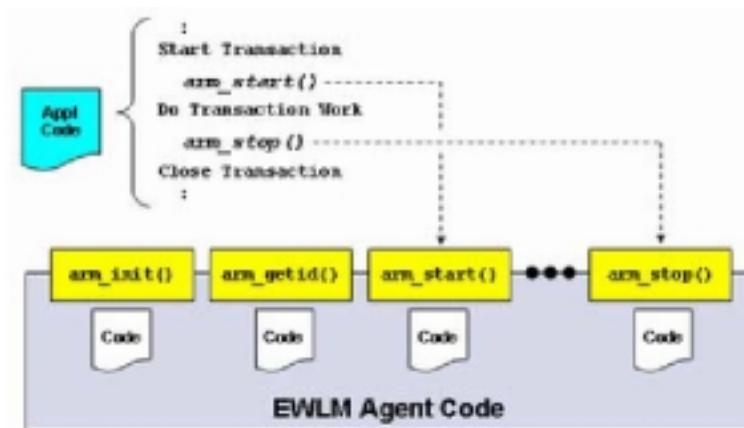
- **Domain Managers**

- ▶ RedHat / SuSe Linux
- ▶ AIX 5L and 5D
- ▶ i5/OS 5.3
- ▶ Windows 2000 / 2003
- ▶ z/OS 1.6

- **Managed Servers**

- ▶ AIX 5L Version 5.2
- ▶ OS/400 5.3
- ▶ Windows 2000 / 2003
- ▶ Solaris 8 / 9
- ▶ HP-UX 11i for PA\_RISC
- ▶ Linux SuSe SLES 9 SP2
- ▶ z/OS 1.6

# EWLM Agent Code provides Implementation of ARM APIs



An application coded to drive ARM API's is said to be ARM „instrumented“

Not all application are ARM instrumented

## Home-grown applications

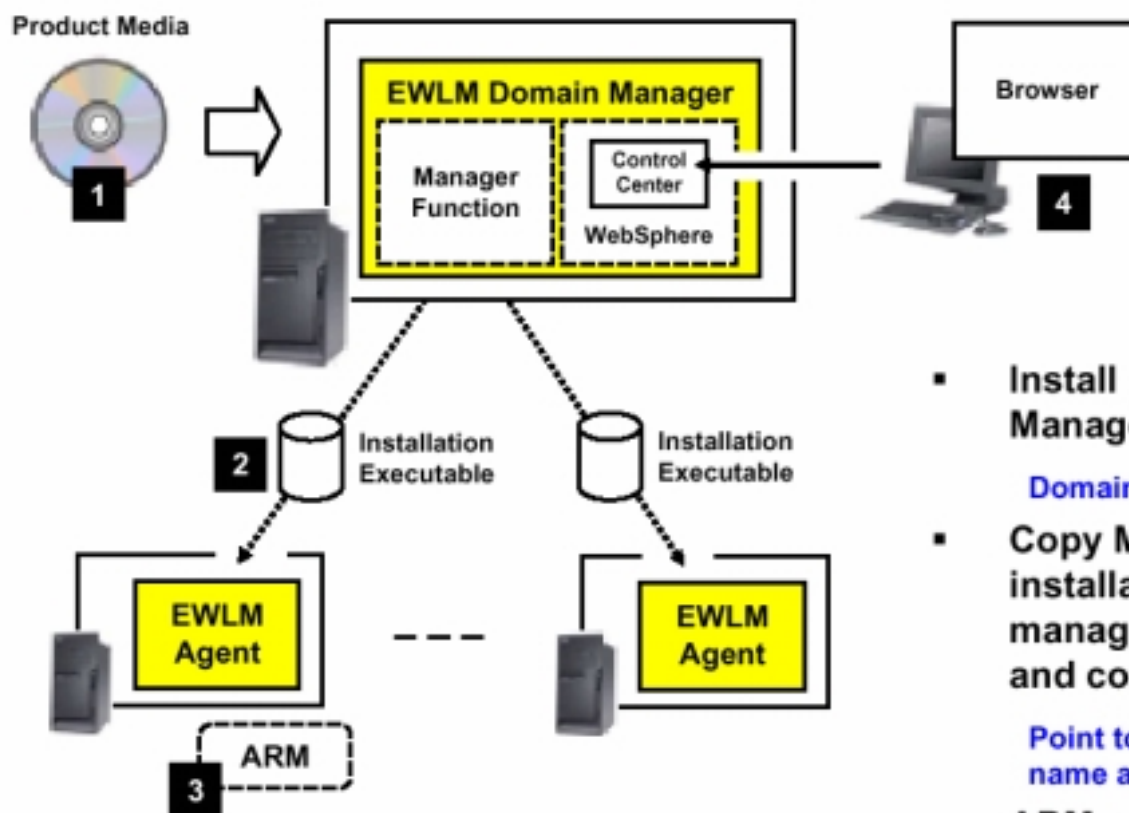
Essentially a programming issue – how to instrument an application to make use of the API's

## Vendor applications

Three possible states:

- **Uninstrumented**  
Some management available
- **Instrumented, not enabled**  
Need to follow instructions to enable ARM
- **Instrumented, enabled**  
Ready to go

# High Level Description of the Installation Process



- Install EWLM Domain Manager code  
     Domain name, ports
- Copy Managed Node installation images to managed servers, install and configure  
     Point to Domain Manager host name and port
- ARM-enable applications
- Configure Domain Policy

# Implementing an EWLM Domain Manager on z/OS

- **The Domain Manager is the central point of control for an EWLM management domain**
  - ▶ The domain manager coordinates the activation of policies on the managed servers that are defined in the EWLM domain
  - ▶ In addition, the domain manager receives data from the managed servers periodically
  - ▶ The domain manager provides this data to you through the EWLM Control Center monitors and reports via a WebSphere application
  
- **There are several places from which you can get information on EWLM for z/OS:**
  - ▶ The IBM Infocenter at URLs:
    - <http://publib.boulder.ibm.com/infocenter/eserver/v1r2/index.jsp>
    - <http://publib.boulder.ibm.com/infocenter/eserver/v1r2/topic/ewlminfo/eicaa.pdf>
    - The program directory which comes with the product when ordered
    - EWLM V2R1 Redbook, SG24-6785, which lists more references in the appendix

# Implementing an EWLM Domain Manager on z/OS

- **EWLM V2R1 is packaged and delivered with different product numbers based on the platforms supporting the domain manager and the managed servers**
- **For z/OS use the following product number:**
  - ▶ **5655-EWM** IBM Virtualization Engine Enterprise Workload Manager for z/OS V2.1
- **Includes EWLM domain manager, common components, and z/OS agent**
- **EWLM for z/OS is distributed as z/OS priced feature at no additional cost**
- **EWLM for z/OS consists of the following FMIDs:**
  - ▶ HJVA140, IBM SDK for z/OS, Java 2 Technology Edition, V1.4
  - ▶ HVE1210, IBM Virtualization Engine Enterprise Workload Manager for z/OS
  - ▶ HDM1210, IBM Virtualization Engine Domain Manager for z/OS
  - ▶ HVW1210, IBM Virtualization Engine WebSphere for z/OS
  - ▶ HCJ1210, IBM Virtualization Engine Runtime Support for ICU4J/iLog Jviews
  - ▶ HGC1210, IBM Virtualization Engine Runtime Support for GCR Toolkit/CMR
  - ▶ EWLM product can also be ordered as CBPDO or ServerPac

# Configuration and Setup Considerations

- **It is highly recommended that you read and follow the EWLM Redbook (SG24-6785)**
- **The installation can seem daunting especially for those not familiar with installing programs that are USS (OMVS) in nature**
- **In addition to consulting the Redbook during the installation process, please also refer to the Techdocs mentioned under “EWLM Documentation and Reference Material”**
  - ▶ *These Techdocs further emphasize a few items from the Redbook which might be overlooked and may cause you a lot of problems*
- **Security may well be the biggest challenge you face**
  - ▶ Authorizations are required for many components so particular attention to this area will save time and trouble
- **You should also plan for a couple of user IDs that will be use to install the EWLM domain manager and managed server**
  - ▶ For details see the Redbook (SG24-6785) and the Techdocs (TD102355 and TD102861)





## **EWLM Documentation and Reference Material**

## EWLM Documentation and Reference Material

- **EWLM Workload Manager V2.1, SG24-6785-00**
- **Implementing an EWLM managed node on z/OS, Techdoc TD102355**
- **Implementing an EWLM Domain Manager on z/OS, Techdoc TD102861**
- **Enterprise Workload Manager Overview, REDP-3989-00**
- **Enterprise Workload Manager z/OS Support, REDP-3993-00**

## Summary

- **z/OS and zSeries hardware provide a robust platform for EWLM**
- **The IBM Virtualization Engine Enterprise Workload Manager for z/OS is a separately-orderable package and includes an EWLM domain manager and managed server**
- **The EWLM Control Center is installed as part of the domain manager installation which provides an embedded WebSphere instance**
- **The managed server installation image includes a firewall broker installation image**
- **The capacity of the server that you install the domain manager on is dependent on the number of managed servers and the domain policy complexity**
- **The ability to use the zAAP processor on the zSeries hardware may also benefit performance**
- **The techdocs article “EWLM Domain Manager and Managed Server on z/OS: Exploiting zAAPs!” discusses this functionality”**
  - ▶ See <http://w3-03.ibm.com/support/techdocs/atsmastr.nsf/Web/Techdocs>