Pulse2011

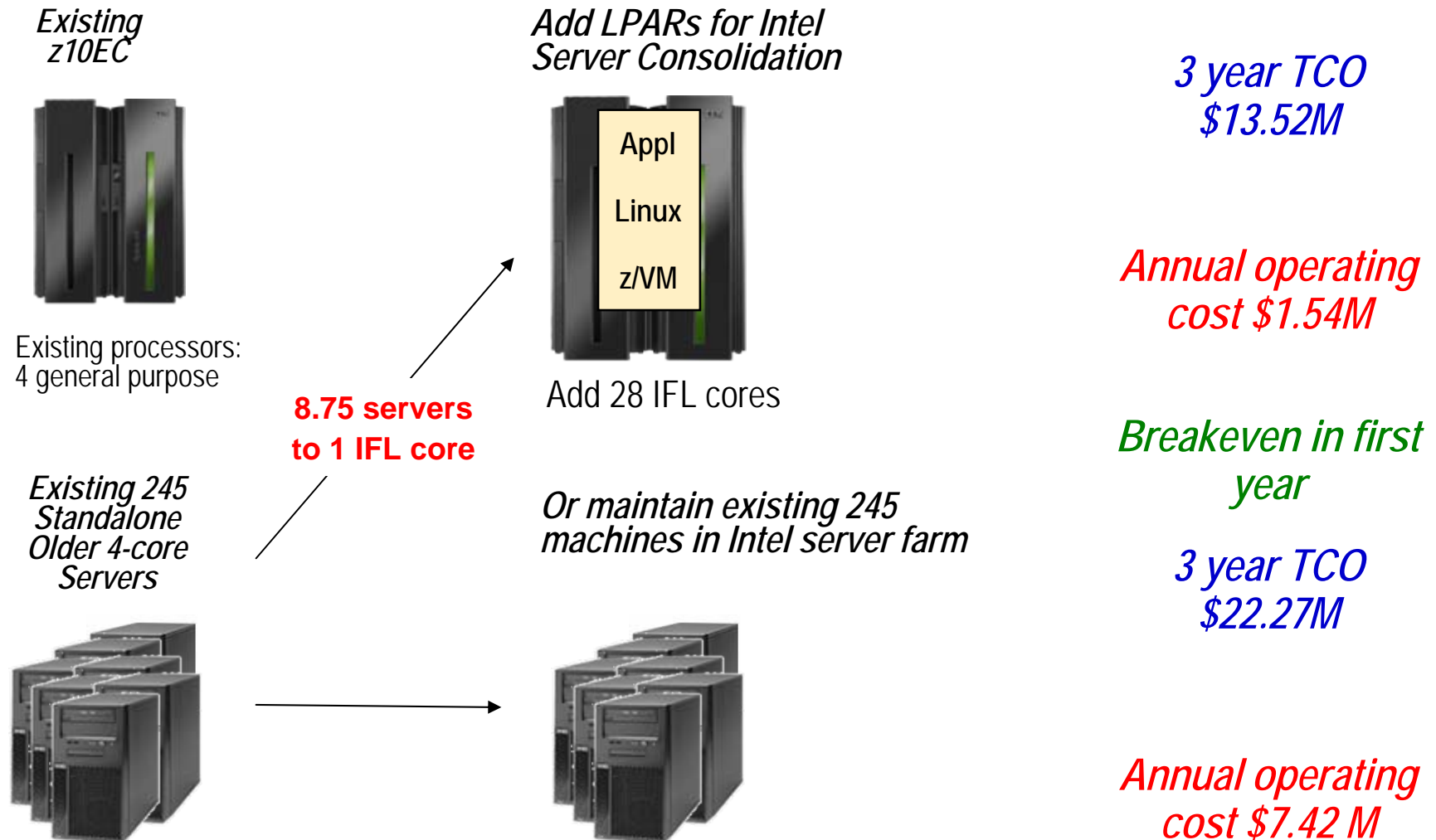# Ways To Save On Software License Costs

John J Thomas

jothomas@us.ibm.com

Ways To Cut Software License Costs

- Use virtualization to consolidate onto fewer physical processor cores
    - ▶ Immediate returns

- Use workload optimized systems to deliver same amount of work, but with less amount of resources

- Eliminate redundant software
   - ▶ Harder to do, but bigger impact

Use Virtualization To Consolidate Onto Fewer Physical Processor Cores

2009 Case Study: Consolidate On Mainframe vs. Keeping Existing Dedicated Servers

Existing z10EC

Existing processors: 4 general purpose

Add LPARs for Intel Server Consolidation

Appl

Linux

z/VM

Add 28 IFL cores

8.75 servers to 1 IFL core

Existing 245 Standalone Older 4-core Servers

Or maintain existing 245 machines in Intel server farm

3 year TCO $13.52M

Annual operating cost $1.54M

Breakeven in first year

3 year TCO $22.27M

Annual operating cost $7.42 M

Workload Optimized Systems Drive Down Cost Per Workload

24 mission critical web facing applications

Web Facing

24 workloads each driving 3080 tps
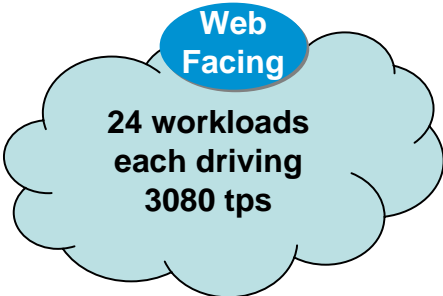
High availability
Workload isolation

Competitor

24 12-core Xeon servers

2 servers
(for ESB)
312 cores total

Competitor System

$433K
Per workload
3yr TCA
Front end HW+SW
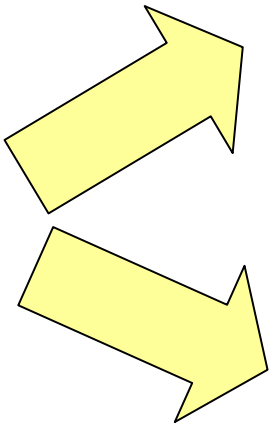
WebSphere App Server
24 POWER7
8-core blades
2 DataPower XI50z
in zBX
192 cores total

Power Blades in zBX

$177K
Per workload
3yr TCA
Front end HW+SW

Source: IBM internal benchmarks. 3 yr TCA calculation includes hardware acquisition, maintenance, software acquisition and S&S. U.S. list prices. Prices will vary by country.

Eliminating Redundant Software Is Harder To Do Than Infrastructure Consolidation

… but potential savings are greater!

To minimize software redundancy:

▶ **Find redundant software** in your infrastructure,
then consolidate it

▶ **Drive down redundant code**,
and prevent it from coming back

Identifying Existing IT Assets Is Key To Eliminating Redundant Software

Only 40% of IT assets are discovered and are understood.

-- Finance Week

If 60% of our IT assets are NOT understood, **How Can We Know:**

▶ *What is deployed in our data center?*

▶ *How are deployed items interrelated?*

▶ *How are they configured? How are they changing?*

▶ *Which business services run on which servers?*

▶ *Do the servers support the business services properly?*

▶ *How to avoid changes that cause problems, requiring a costly rollback?*

Tivoli Application Dependency Discovery Manager (TADDM)
can help answer these questions to eliminate redundancy

TADDM Shows Existing Components And Finds Dependencies Between Them

Universal Discovery Engine

- ▶ Discovers
  - – Servers
  - – Operating systems
  - – Middleware
  - – Applications
  - – Network devices
- ▶ Reconciles names
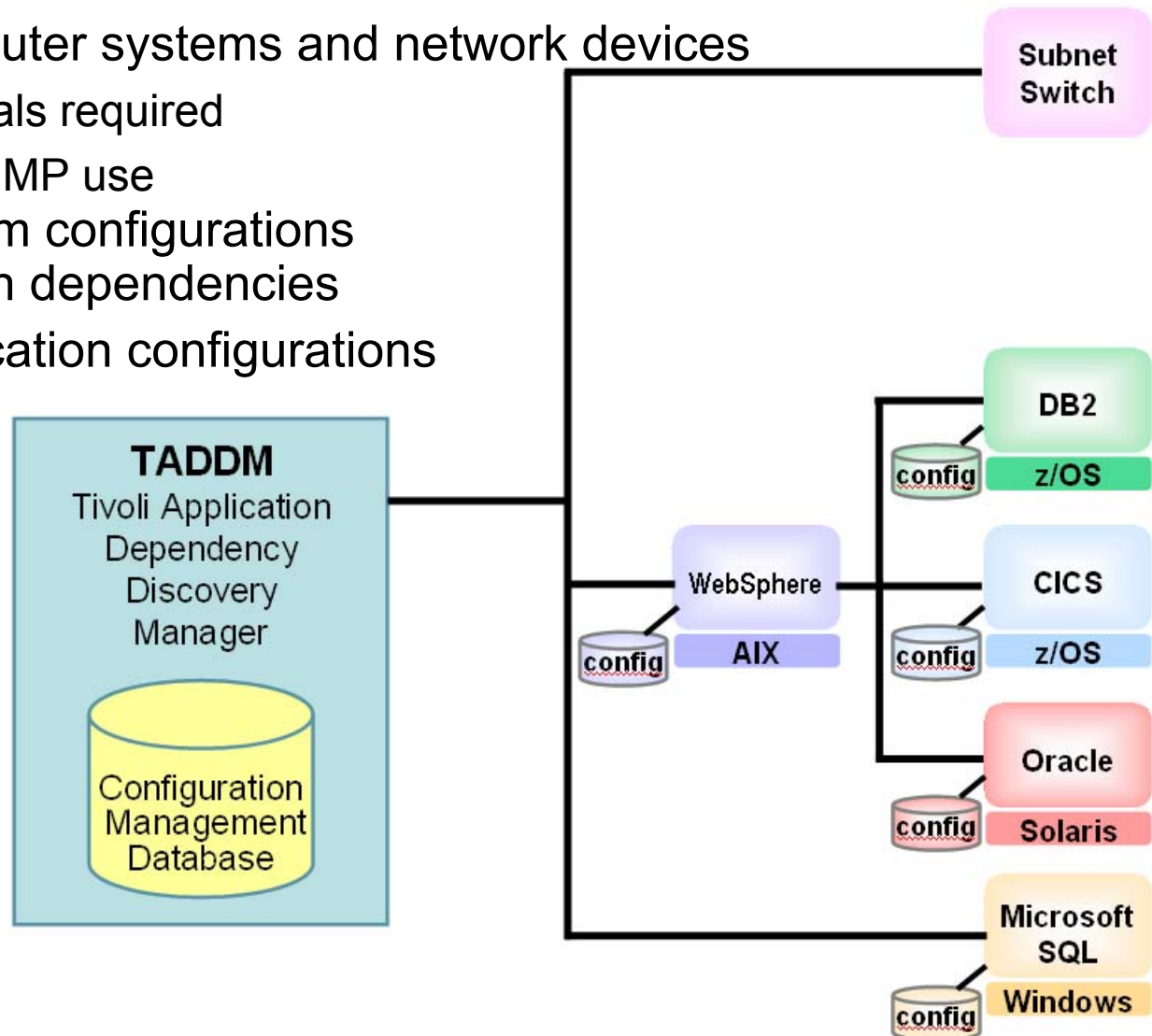- ▶ Normalizes data
- ▶ Creates topology views
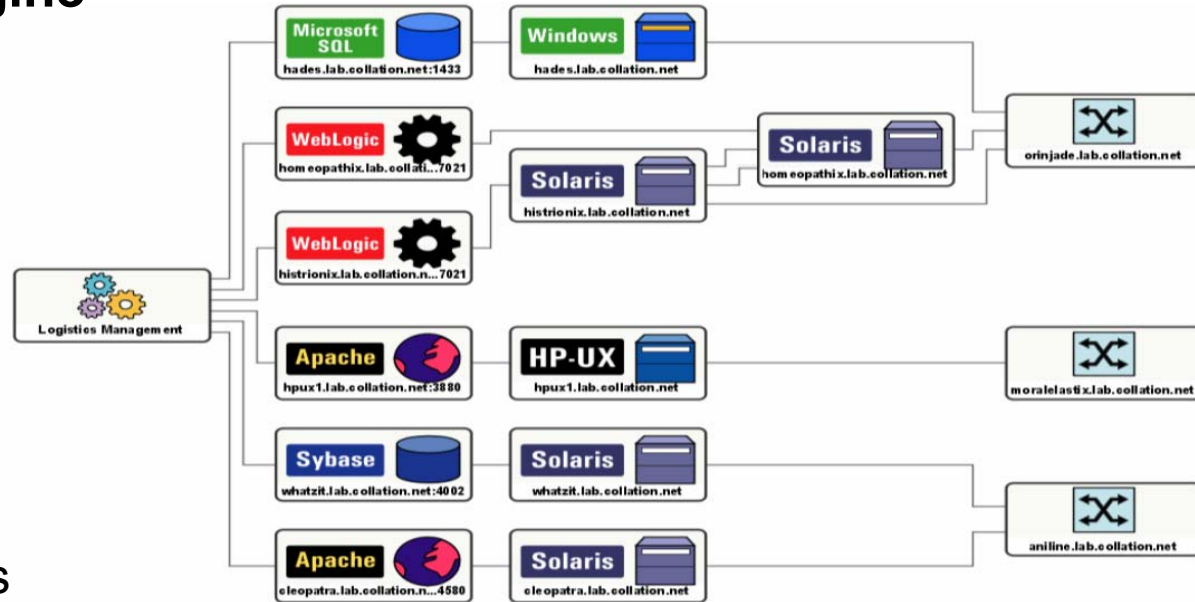
Agent-less Application Mapping
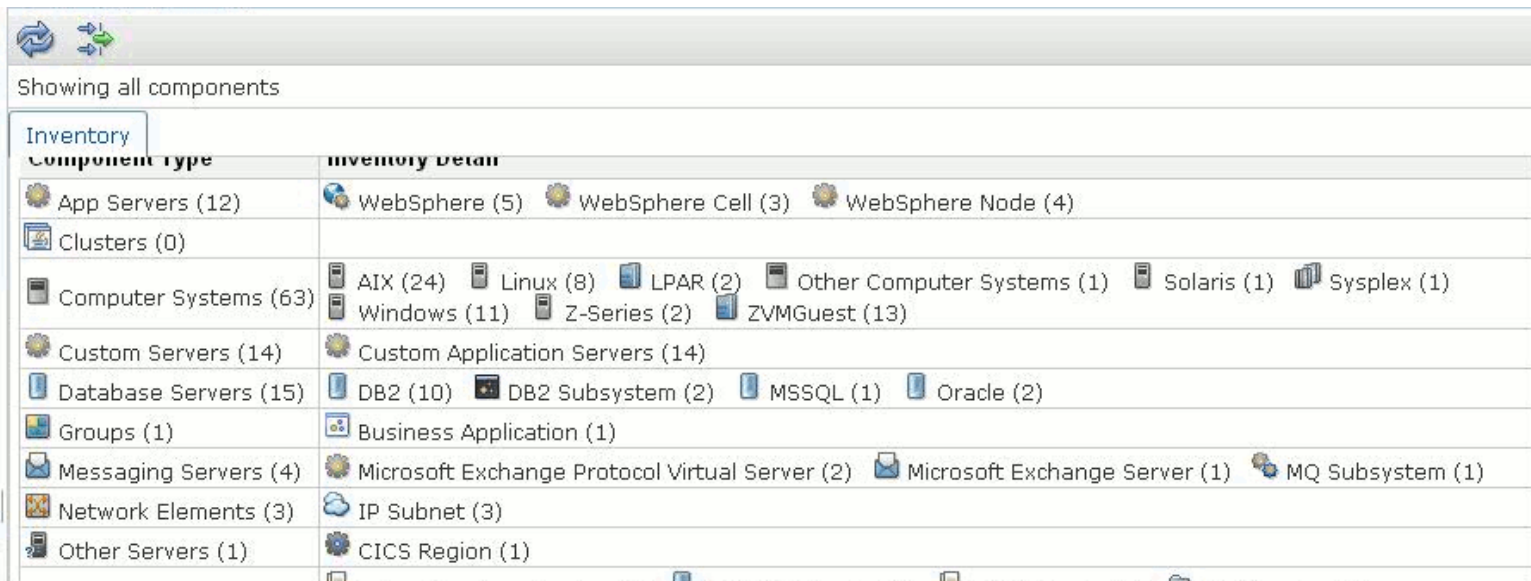
- ▶ Finds dependencies between discovered items

TADDM Has A 3-Step Discovery Process

1. Discover computer systems and network devices
   ▶ No credentials required
   ▶ Optional SNMP use
2. Discover system configurations and application dependencies
3. Discover Application configurations

DEMO: Using TADDM To Identify Redundant Software

1. The IT department knows there are multiple database types and instances in use. License costs can be reduced by consolidating the various databases
2. Use TADDM to discover database servers, software and instances
3. Use TADDM discovery results to identify potential software consolidation targets

TADDM Output Helps Identify Redundant Servers And Software For Consolidation

- **A quick view of server usage for server consolidation**
  - ▶ What is this server used for?

- **A complete view of installed software**
  - ▶ Which servers support a specific application or middleware?
  - ▶ Highlights opportunities for consolidation or simplifying backups

- **Detailed views of application interactions**
  - ▶ What would be the impact of moving or eliminating this application?

- **Locate department 'islands' or clusters of redundant software that is not part of the overall IT plan**
  - ▶ Identify assets that are 'flying under the radar'

- **TADDM output is a key resource for quickly identifying software redundancies from mergers or acquisitions**

Typical TCO savings of 30% to 70%

IBM Tivoli Asset Discovery for z/OS

- Automatically discovers and identifies IBM and third party software running on the z/OS platform

- Monitors software usage and related growth to proactively plan for future capacity needs

- Generates interactive reports that provide convenient hyper-link drill down for more details

- Use with IBM Tivoli Asset Management for IT to manage the full lifecycle of hardware and software assets including software license and contracts management
  - ▶ manage the complete IT asset management lifecycle from requisition to disposal

TADz Provides Key Details

- Which products are installed — Vendors, Product Titles, Features, Modules—Versions, Releases, Maintenance levels—PID, S&S PID, Entitlement ID

- Where the products are installed &used—Regions, Machines, LPARs, SMFID, Storage Volumes, Datasets

- Who is using the products—Job names, Users, Account Codes

- When the products are being used—Monthly periods

- What products can potentially be consolidated—Trend graphs with drill down to show who is using the product

Identify Products That Have Low Usage And Drill Down To See Who Is Using It
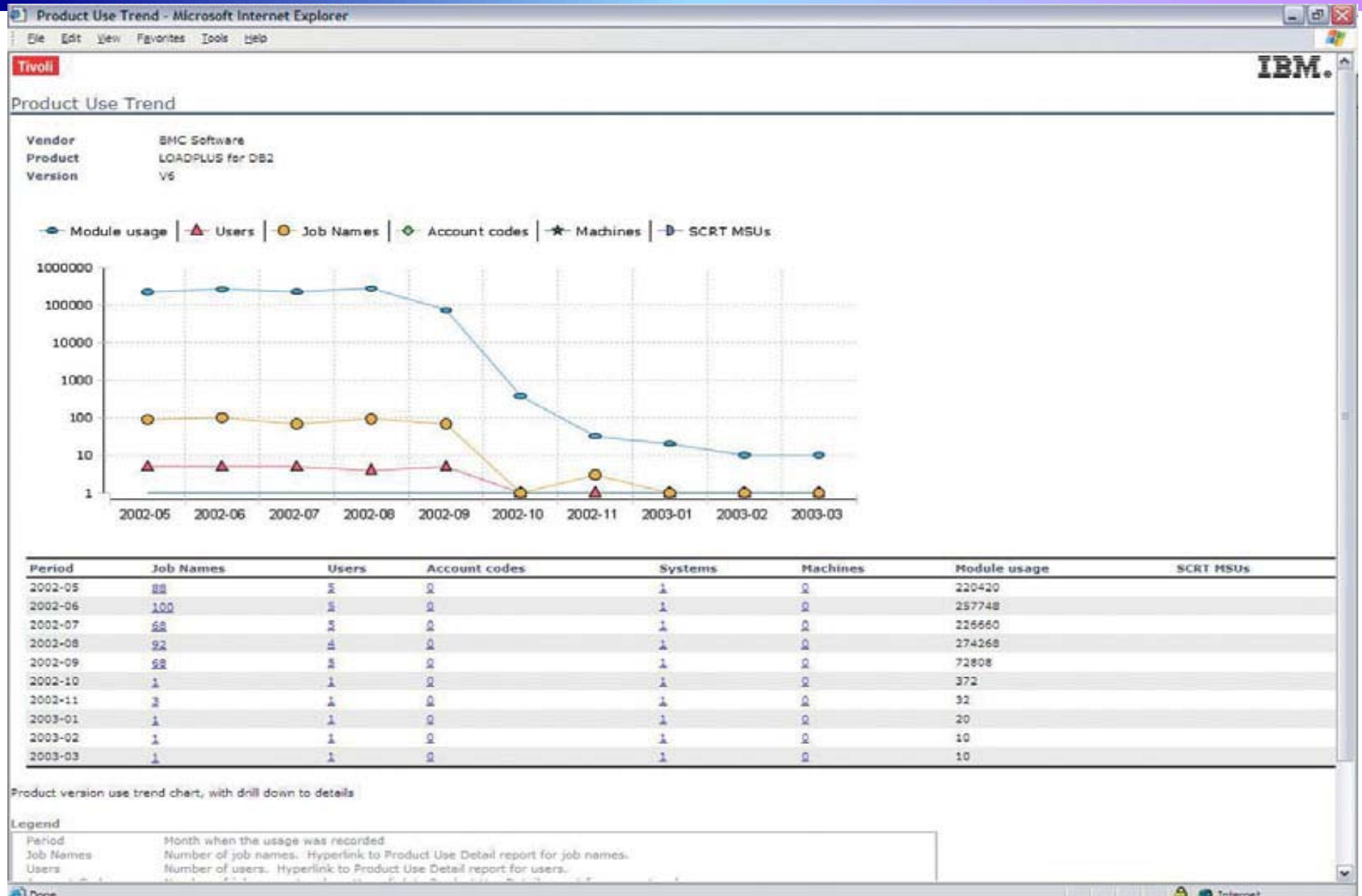
Service Oriented Architecture Can Help Drive Down Redundant Code By Pooling Services

Many businesses re-implement the same function over and over again

- ▶ Leads to software redundancy
- ▶ Drives up software licenses
- ▶ Creates software sprawl

Redundancy can also be caused by mergers and acquisitions

To drive down redundancy…

- ▶ Find the functions
- ▶ Categorize them
- ▶ Rationalize them

…then **pool** services by identifying repeated functions, and

- ▶ Choosing one as the strategic function
- ▶ Deploying it as a Web service
- ▶ Modifying each solution to use the strategic service
- ▶ Eliminating the non-strategic functions
- ▶ Ensuring that all new solutions use the strategic service instead of creating a new one

Fewer lines of code means lower maintenance costs, so more resources are available for new strategic development!

Rational Asset Analyzer (RAA) Helps Find Reusable Functions In Source Code

- RAA scans source code to determine program and data flow, and show dependencies between code elements
  - ▶ COBOL, PL/I for CICS, IMS, DB2
  - ▶ z/OS JCL and High Level Assembler
  - ▶ Java, Java EE, C/C++ applications

- RAA creates an application knowledge base showing relationships, program structure, flows, and run times
  - ▶ Provides browsable view of architectures and artifacts
  - ▶ This forms the basis for intelligent SOA-based design using services built from existing code
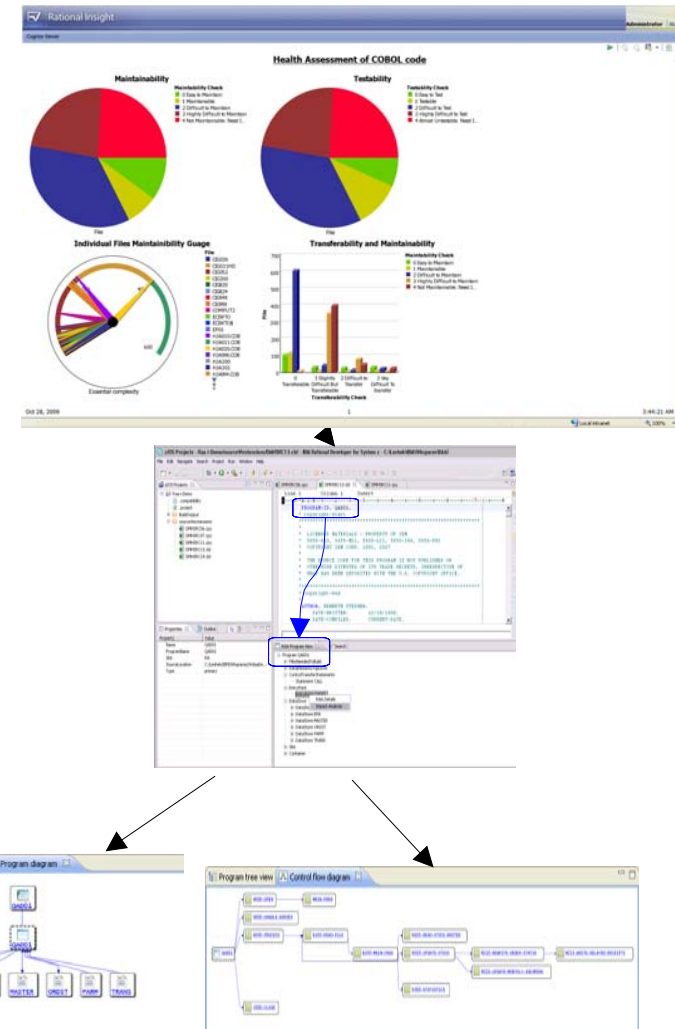
RAA Analyzes CICS, IMS And DB2 Assets For Potential Reuse As Web Services

- Some assets in particular have potential for reuse as services:
  - ▶ CICS online regions and transactions
  - ▶ Existing Web services in CICS
  - ▶ IMS transactions
  - ▶ DB2 access and stored procedures

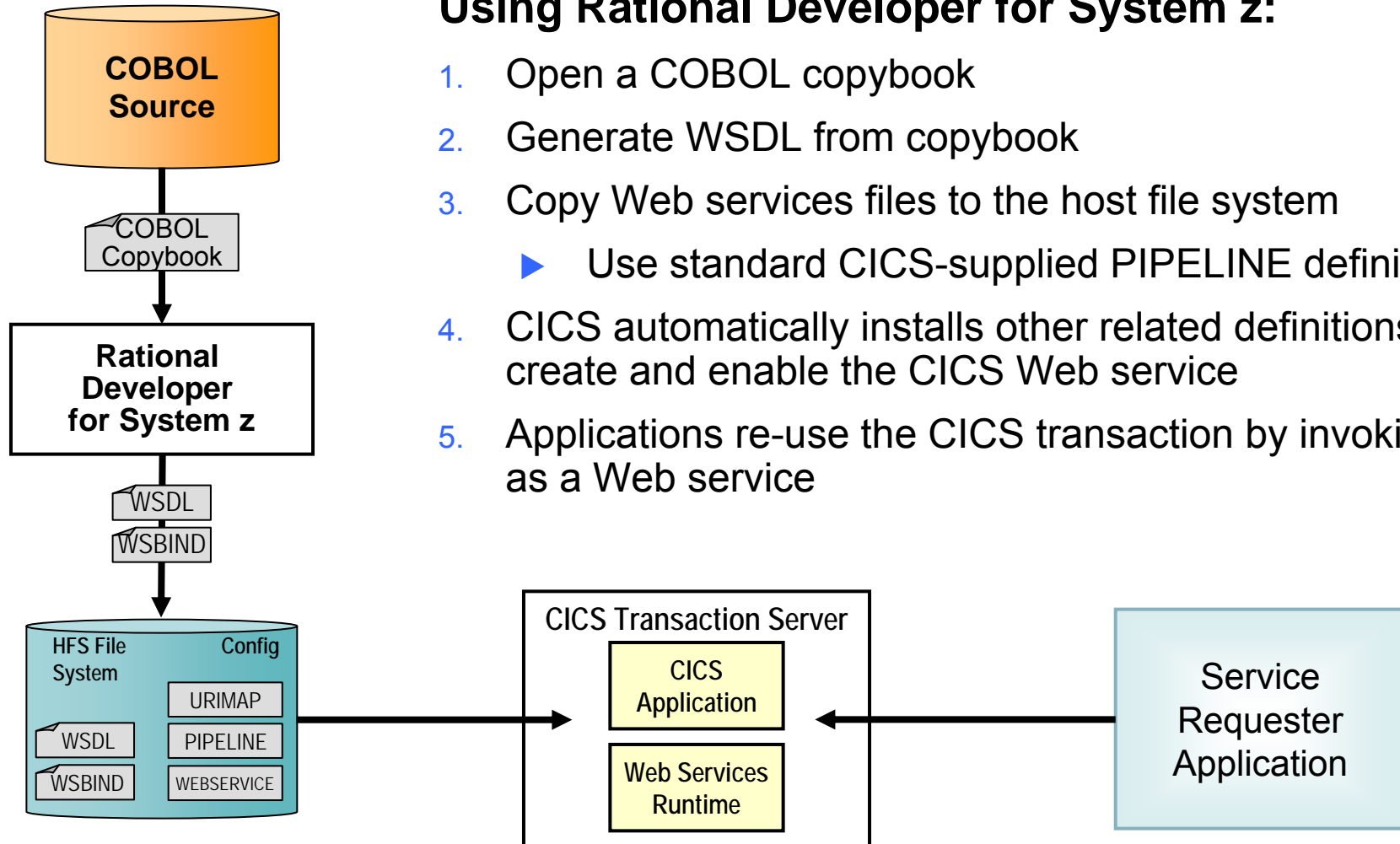- Accessing existing functions as services makes modernization easy and flexible

Source Code
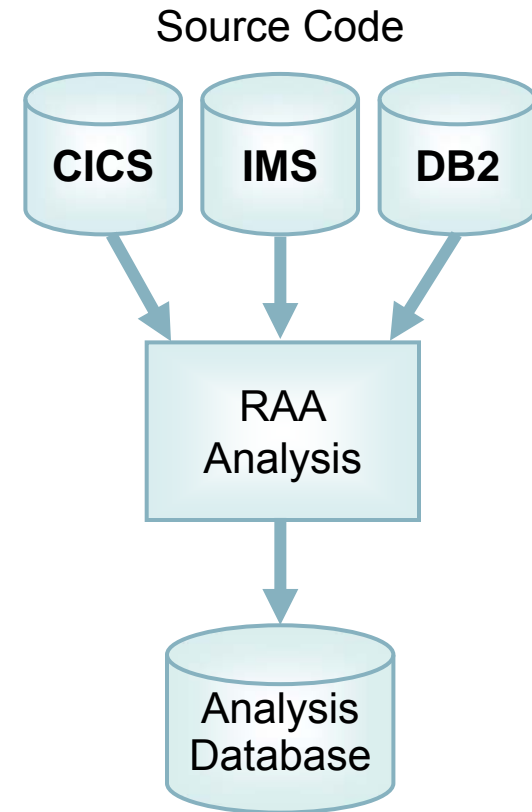
Rational Developer for z makes it easy to create Web services from any of these – without writing any code!

Use Rational Developer For z To Create Services From Existing CICS Transactions Without Coding

Using Rational Developer for System z:

1. Open a COBOL copybook

2. Generate WSDL from copybook

3. Copy Web services files to the host file system

   ▶ Use standard CICS-supplied PIPELINE definition

4. CICS automatically installs other related definitions to create and enable the CICS Web service

5. Applications re-use the CICS transaction by invoking it as a Web service

COBOL Source

COBOL Copybook

Rational Developer for System z

WSDL
WSBIND

HFS File System — Config
URIMAP
WSDL   PIPELINE
WSBIND   WEBSERVICE

CICS Transaction Server
CICS Application
Web Services Runtime

Service Requester Application

DEMO: Use RDz To Expose CICS Program As Web Service

- Generate WSDL and WSBIND files
- Deploy files to host file system
- Test using Web Services Explorer

Generate Web Service
files using RDz

Generate client proxy
using RDz

COBOL
Source

COBOL
Copybook

Java
client

XML / SOAP / HTTP

CICS
Web Service

Rational Developer
for System z (RDz)

HFS   CSD

WSDL

WSBIND
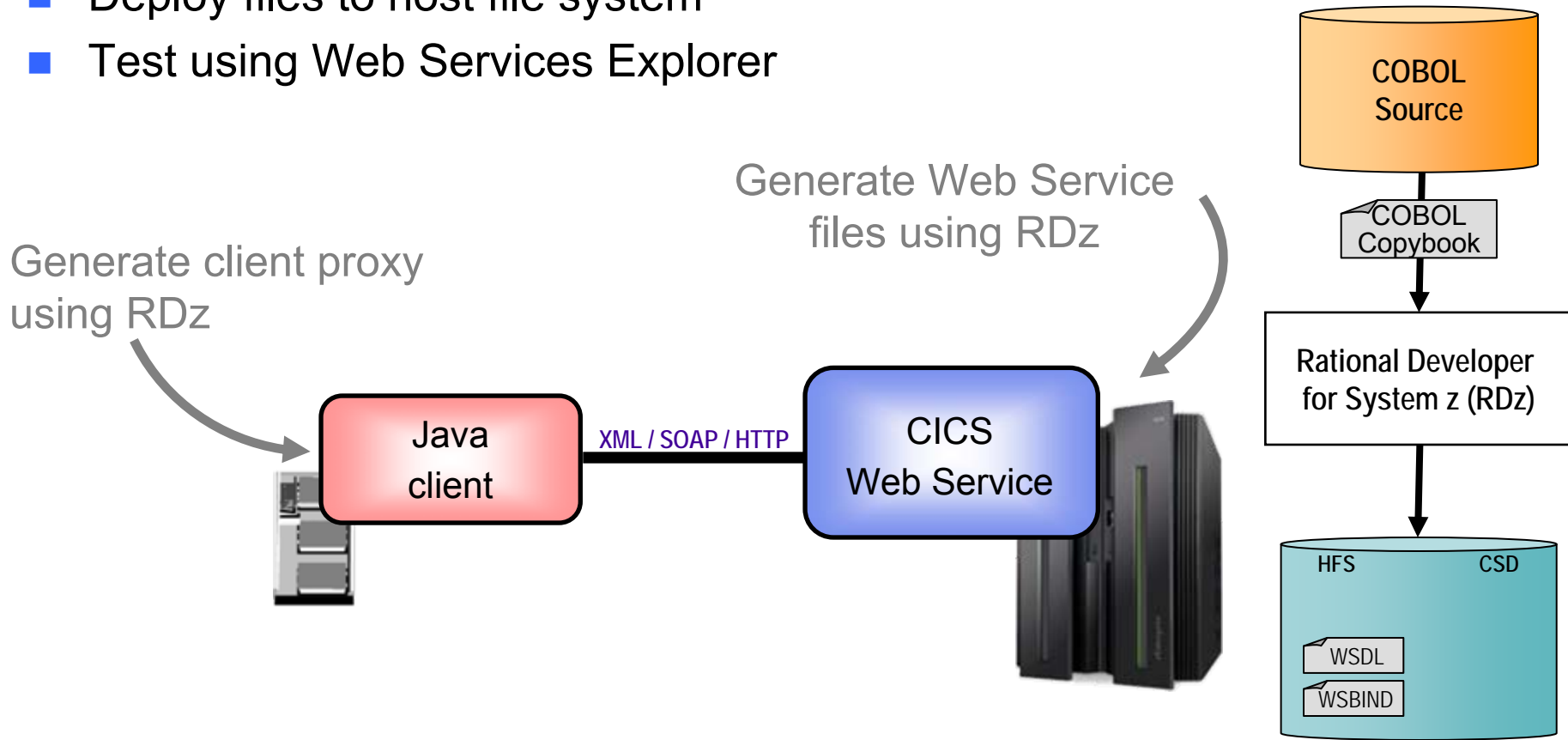
Use WebSphere Service Registry And Repository To Prevent Redundancy From Returning

Define An SOA Governance Plan

▶ Create a development process for reviewing new application and service deployment proposals

▶ Enable enforcement of policies

Encourage Reuse

▶ Publish new services for future reuse

▶ Quickly locate required services directly from development tools for new applications and business processes

Enhance Connectivity

▶ Locate required services for runtime requests, including alternates

▶ Enable dynamic and efficient interactions between services at runtime with an Enterprise Service Bus

Enterprise Service Bus

IBM BPM Process Designer

IBM BPM Integration Designer

WebSphere Service Registry Repository
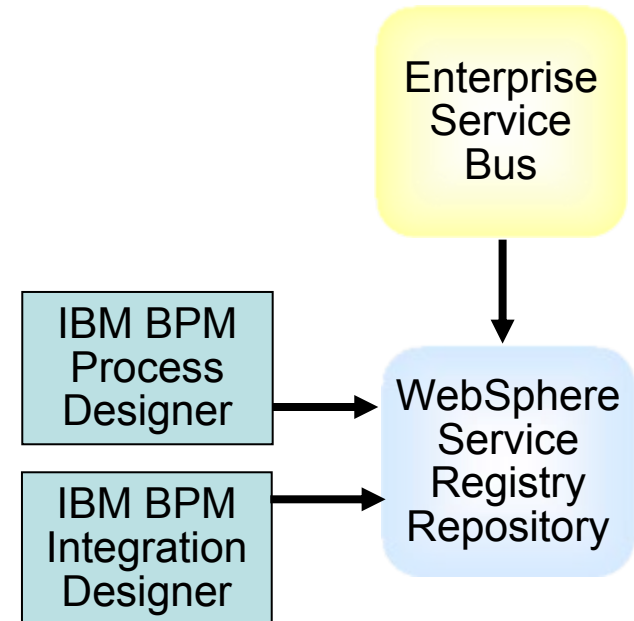
zEnterprise Supports All SOA Components
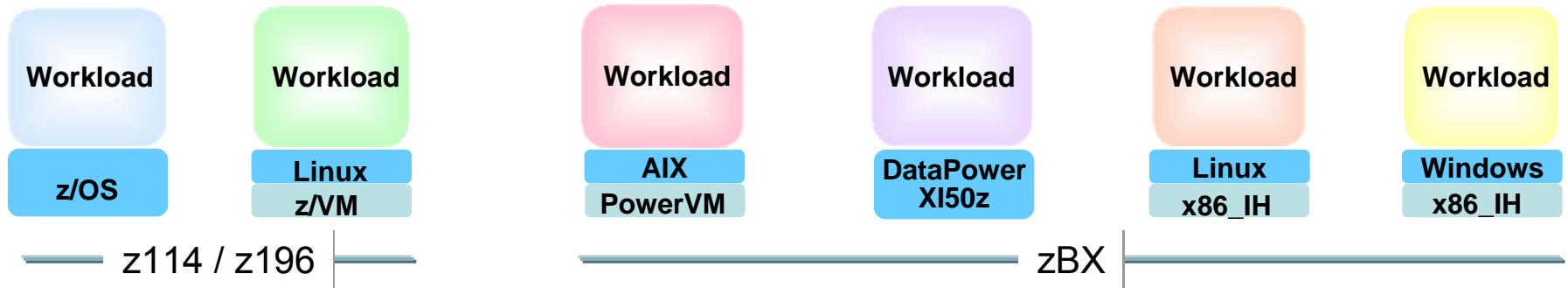
Service Requesters

BPM Process Server

WebSphere Application Server

CICS Applications

DB2 Stored Procedures

IMS Applications

COBOL applications

Java batch applications

z/OS Batch Jobs

Service Intermediaries

Enterprise Service Bus

WebSphere Service Registry Repository

Web service front ends to CICS/DB2

Service Providers

BPM Process Server

WebSphere Application Server

CICS Transactions as services

DB2 Stored Procedures as services

IMS Transactions and Data as services

COBOL procedures as services

z/OS batch jobs

Java batch applications

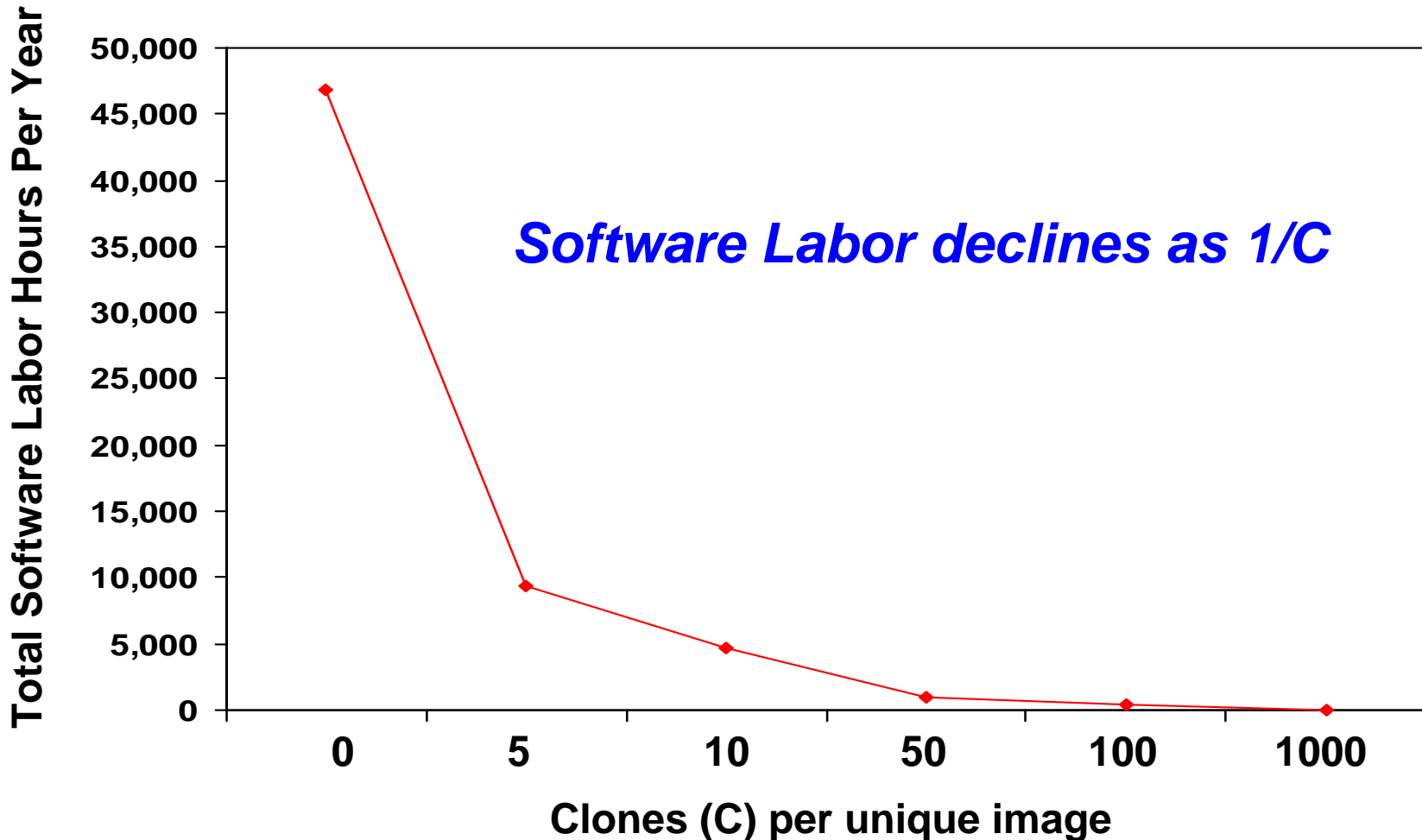| Workload	Workload	Workload	Workload	Workload	Workload
z/OS	**Linux z/VM**	**AIX PowerVM**	**DataPower XI50z**	**Linux x86_IH**	**Windows x86_IH**

z114 / z196 ———————————— zBX

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

Standardization Of Software Stacks Can Also Eliminate Redundant Software

- A server needs a full set of software to run a workload
  - ▶ Operating System, Middleware, Applications
  - ▶ Patches, configuration specifications

- The combination of all this software is called a "software stack"

- Without controls, the variety of software stacks tends to proliferate, driving up labor costs
  - ▶ Different levels, patches, product selections, etc

- Standardization of software stacks can reduce labor costs
  - ▶ Uniformity reduces the number of unique stacks to manage
  - ▶ Re-using a standard software stack is called "cloning"

Benefit Of Cloning On Software Labor Costs In A Virtualized Environment

Software Labor declines as 1/C

(Chart: X-axis "Clones (C) per unique image" with values 0, 5, 10, 50, 100, 1000; Y-axis "Total Software Labor Hours Per Year" from 0 to 50,000)

Save Money By Eliminating Redundant Software

Start by assessing current environment

- ▶ Discover hidden "islands" of underutilized applications under departmental control

- ▶ Identify candidates for eliminating or consolidating applications and servers

Use Service Oriented Architecture to pool services and reduce redundant application code

- ▶ Reducing code yields reduced maintenance costs, enabling investment in new strategic development

- ▶ Use SOA Governance and a service registry to ensure redundant code does not return

Clone software stacks to significantly reduce labor costs