CICS Transaction Server for z/OS V4.1
Technical Overview

Steve Zemblowski

zem@us.ibm.com

Notes

CICS Transaction Server for z/OS® (CICS TS) is a modern, dependable, and cost-effective application platform. CICS TS V4.1 will deliver new business event capabilities, Web 2.0 programming support, and new operational tooling that will enable enterprise agility when responding to business opportunity and risk in rapidly changing markets. Its traditional qualities of security, reliability, availability, scalability, data-integrity, and application responsiveness make CICS TS the perfect transaction-server partner for Web application servers.

CICS TS V4.1 will build on the solid foundations, mature service oriented architecture (SOA) support, and open connectivity of previous CICS versions, bringing new levels of ease-of-use and flexibility to meet the needs of the next generation of business users. It will assist its users in their activities directed at:

Competing for new opportunities by gaining insight into business processes and responding by modifying key business applications quickly and with confidence

Complying with corporate, industry and government policies to manage business risk of critical business applications

Controlling costs by simplifying IT infrastructure and improving development and operations productivity through easier-to-use interfaces and functions

CICS TS V4.1 is the appropriate upgrade path for all CICS customers

- CICS TS V1.3 service was discontinued 30 April 2006
- CICS TS V2.2 service was discontinued 30 April 2008
- CICS TS V2.3 service will be discontinued 30 September 2009

Session Agenda

- **CICS TS V4.1**
  - Strategy and Themes

  - Compete
    - Event Processing
  - Control
    - CICS Explorer
  - Comply
    - Management of CICS resource definitions
  - Architectural enhancements
    - Improvements to XML parsing

- **Summary**

Notes

This presentation will discuss the key features of CICS Transaction Server V4.1.

Strategy and Customer Trends

- **SOA continues to be major driving force**

  – Driving increased IT flexibility

  – Faster time to market for new solutions

  – Greater reuse of existing assets

- **Maturing of new programming architectures**

  – WEB 2.0

  – Event Based Processing

- **Skills and expertise shortages**

- **Increased Governance requirements and regulations**

- **Interoperability and synergy with other SOA products**

Notes

There are no notes for this slide.

CICS Transaction Server V4 Themes

Compete *for new opportunities by gaining insight into business processes and responding by modifying key business applications quickly and with confidence*

Comply *with corporate, industry and government policies to manage business risk of critical business applications*

Control costs *by simplifying IT infrastructure and improving development and operations productivity through easier-to-use interfaces and functions*

Architectural Enhancements *to relieve constraints on processing, configuration or data capacities allowing for continued application and system growth*

Notes

CICS Transaction Server for z/OS, Version 4 Release 1 delivers a set of capabilities which provide customer value by enabling business flexibility through IT simplification.

These capabilities are represented in three principal themes.

Compete for new opportunities by gaining insight into business processes and responding by modifying key business applications quickly and with confidence

Comply with corporate, industry and government policies to manage business risk of critical business applications

Control costs by simplifying IT infrastructure and improving development and operations productivity through easier-to-use interfaces and functions

There is a fourth theme that spans the CICS product:

Architectural Enhancements to relieves constraints on processing, configuration or data capacities allowing for continued application and system growth.

CICS Transaction Server V4.1 Key Enhancements

Comply

Resource signatures

WebSphere Registry & Repository Support

Support for distributed identities

Compete

Support for event processing

Atom feeds from CICS

Application Bundles

Service Component Architecture

Java 6

Web Services Addressing

Improvements to data mapping

Control

CICS Explorer

IPv6

IPIC Transaction Routing

MQ Group attach

Improvements to CPSM workload management

New SPI commands for managing the CSD

Discovery Library Adaptor

Architectural Enhancements

Improvements to XML parsing in CICS

Large file hosting

Performance Improvements

Notes

The key capabilities delivered by CICS Transaction Server V4.1 grouped by themes.

Event Processing

- **An event is something that happens that is relevant to the business**
  - "**simple**" event: meaningful in itself
    - Order placement, stock trade
  - "**complex** event processing": detect and respond to patterns of events over time
    - 3 orders from a customer in 2 days, suspicious pattern of ATM activity
  - "Business Event Processing" extends event processing capabilities to business users

- **CICS can be significant source of events**
  - Focus is on events relevant to the Line-of-Business
  - *CICS will emit simple events*
  - Events emitted by CICS could:
    - Drive another CICS transaction
    - Be written to a WebSphere MQ queue
    - Be written to a temporary storage queue
    - Be input to a monitor or business manager's dashboard
      - e.g. WebSphere Business Monitor
    - Be sent to a "complex event processing" engine such as WebSphere Business Events
    - …

Notes

You can specify, capture, and emit business events from a CICS® application. These business events can be consumed by another CICS application, or placed on a WebSphere MQ queue for consumption in a variety of ways including by a complex event processing engine such as IBM WebSphere Business Events.
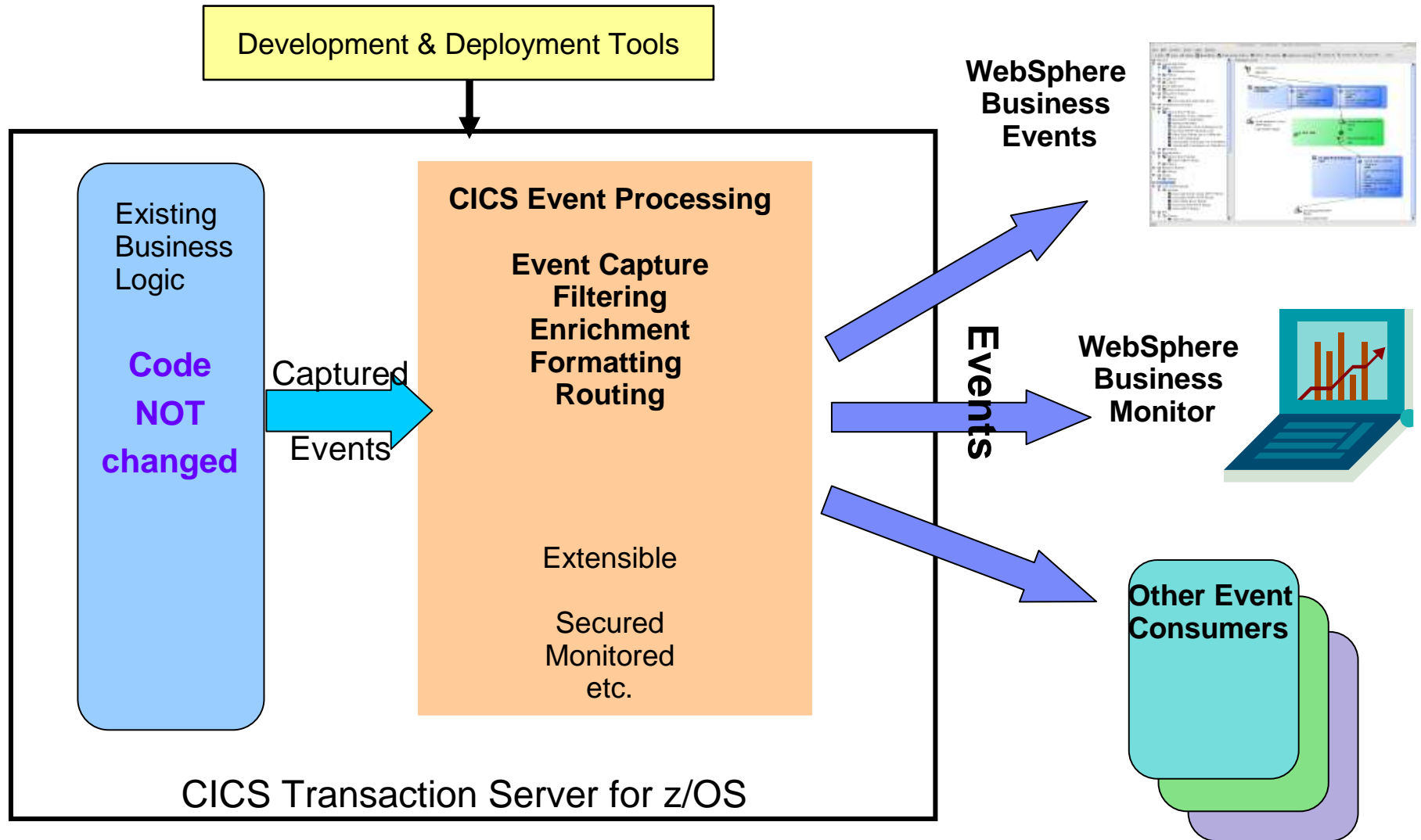
A business event is anything that happens that is relevant to your business. An individual stock trade or the placement of an order are both examples of business events. CICS allows business events to be captured in an application and emitted for processing, with typically no change to the emitting CICS application.

After CICS has captured and processed a business event, it is passed to an EP (Event Processing) adapter for formatting and routing. CICS provides EP adapters to allow business events to be emitted to WebSphere MQ, either in XML format for consumption by WebSphere Business Events, in CBE (Common Business Event) XML format to the Common Event Infrastructure (CEI) for consumption by business monitors, or in a non-XML character format, as well as to a CICS transaction or to a CICS temporary storage queue.

You can also write your own custom EP adapter. CICS supports custom EP adapters written in the following languages:

COBOL

PL/I

C

C++

Assembler language

Event Processing…

Development & Deployment Tools

WebSphere Business Events

Existing Business Logic

Code NOT changed

Captured Events

CICS Event Processing

Event Capture
Filtering
Enrichment
Formatting
Routing

Extensible

Secured
Monitored
etc.

Events

WebSphere Business Monitor

Other Event Consumers

CICS Transaction Server for z/OS

Notes

An overview of event processing in CICS.

Event Processing

- **Non-intrusive instrumentation of events**

  – No requirement to change existing business logic

- **EXEC CICS SIGNAL EVENT for explicit instrumentation of events**

- **Tooling to create event specifications**

  – Event binding editor in CICS Explorer

  – Deployed to CICS via event bindings in BUNDLE resources
    - Specifies event and its payload, and how it can be detected/captured by CICS
    - Specify event capture points as EXEC CICS command
    - Filtering on command parameters and data

- **Events dispatched to specified EP adapter for formatting and emission to event consumer**

  – CICS-supplied EP adapters

  – Capability to write custom EP adapters

Notes

A CICS application can capture and emit business events with no change to the application itself, using noninvasive capture points, before and after selected EXEC CICS API calls, and at program start.

For situations in which the noninvasive capture points are not sufficient to capture a specific business event, a new EXEC CICS API call, SIGNAL EVENT, allows events to be captured anywhere in a CICS application.

The CICS Explorer includes the CICS event binding editor, a tool that helps you to define simple business events and create event bindings for your CICS applications. You deploy event bindings to a CICS system from the CICS Explorer.

After CICS has captured and processed a business event, it is passed to an EP (Event Processing) adapter for formatting and routing. CICS provides EP adapters to allow business events to be emitted to WebSphere MQ, either in XML format for consumption by WebSphere Business Events, in CBE (Common Business Event) XML format to the Common Event Infrastructure (CEI) for consumption by business monitors, or in a non-XML character format, as well as to a CICS transaction or to a CICS TS queue.

You can also write your own custom EP adapter. CICS supports custom EP adapters written in the following languages:

COBOL

PL/I

C

C++

Assembler language

Dynamic Scripting Feature Pack

- **Newly announced for CICS TS V4.1**

- **Embed Zero programming model into CICS on z/OS**

  – Use the entire sMash feature set to Web 2.0 enable CICS.

  – Script CICS assets using Groovy or PHP.

  – Exploit sMash tools and modules – ZRM, Flow, App Builder

  – Build a web 2.0 AJAX presentation layer onto CICS Programs.

  – Expose CICS assets as RESTful services.

- **Tight integration with existing CICS application assets and data**

  – Easy and efficient access to COBOL assets

  – Inheriting CICS run time qualities of service

Notes

CICS Dynamic Scripting integrates Project Zero technology into CICS TS V4.1, for a robust environment for PHP and Groovy situational reports, dashboards, and web feeds.

CICS Dynamic Scripting helps clients meet the long tail of application demand and exploit business opportunities by rapidly creating and deploying situational applications which access CICS assets, while complying with key IT processes like manageability, scalability, and security.

CICS Transaction Server Feature Pack for Dynamic Scripting embeds and integrates technology from Project Zero to open up CICS to make it accessible to more than four million PHP and Groovy developers, and reduce the time and cost of CICS application development.

CICS Dynamic Scripting provides a robust, managed environment for situational applications written in PHP and Groovy. The Feature Pack is built on scalable CICS TS V4.1 JVMServer technology that is easily distinguished from other production CICS workload, and where Zero Resource Model (ZRM) data can be managed by DB2 for z/OS.

CICS Dynamic Scripting allows you to quickly try out new business ideas, build productivity applications like reports, dashboards, and widgets, or application front-ends for IT and Line of Business users.

You can introduce new IT staff to CICS via PHP and port many existing unmanaged PHP and WebSphere sMash applications into CICS.

The CICS Dynamic Scripting Feature Pack will be available as an optional product extension to customers with CICS Transaction Server for z/OS, V4.1 on June 22nd 2010.

Dynamic Scripting Feature Pack…

- **ProjectZero Features for CICS**

  - Script in PHP, Groovy, Java

  - Respond to HTTP requests

  - Call CICS commarea programs from PHP or Groovy

  - Access DB2 databases from PHP, Groovy, ZRM

  - Call any JCICS interface from PHP or Groovy

    - Manage units of work (commit/rollback)

  - Use event handlers to easily create RESTful Web Services

  - Debug PHP scripts with Rational Developer for Z.

  - Access any Java class from PHP code using the PHP/Java Bridge

Notes

WebSphere sMash based on IBM's Project Zero incubation effort. Project Zero is the development and incubation community. All released versions are called WebSphere sMash.

The CICS Dynamic Scripting Feature Pack provides an agile web application platform for developing and running modern web applications. You can take advantage of web technologies and improve your productivity by using CICS Dynamic Scripting to create, assemble, and run applications that meet your specific needs or the needs of your clients. You can use the feature pack to take advantage of the following web technologies:

A dynamic scripting CICS-based run time for Groovy and PHP applications

Java as a system programming language

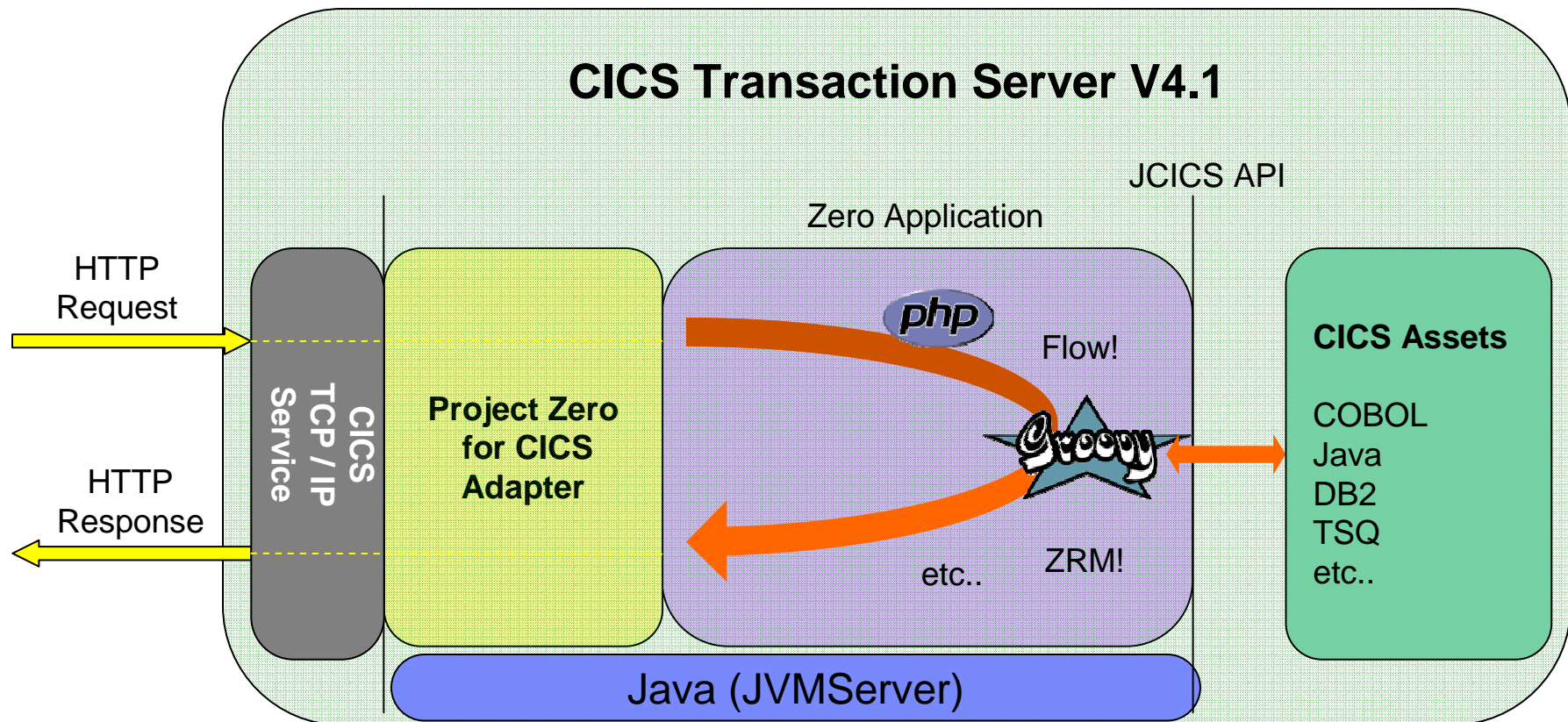Application programming interfaces that use Representational State Transfer (REST) services

Rich Asynchronous JavaScript and XML (Ajax) web user interfaces

Integration of mash-ups and feeds

The CICS Dynamic Scripting Feature Pack is developed using technology from WebSphere sMash v1.1.1.3. You can find out more about developing dynamic scripting applications by visiting projectzero.org, the community website for WebSphere sMash.

Dynamic Scripting Feature Pack…

- **Project Zero on CICS Architecture**

CICS Transaction Server V4.1

JCICS API

Zero Application

HTTP Request

HTTP Response

CICS TCP / IP Service

Project Zero for CICS Adapter

php

Flow!

Groovy

ZRM!

etc..

CICS Assets

COBOL
Java
DB2
TSQ
etc..

Java (JVMServer)

Notes

The WebSphere sMash architecture in CICS Transaction Server V4.1

Atom feeds from CICS

- **What is an Atom Feed?**

  - Protocol and XML format for content publishing
    - Provide XML based feed of updated content
    - Process is known as syndicating a feed
      - > Follow-on to Real Simple Syndication (RSS)
    - Simple publish/subscribe implementation
      - > Polling model
      - > Based on http support

  - Described by two Internet Request for Comments
    - The Atom Syndication Format
      - Targeted at producing feeds
      - RFC4287: (Dec 2005)   **http://tools.ietf.org/html/rfc4287**

    - The Atom Publishing Protocol
      - Targeted to creating and updating resources
      - RFC5023: (Oct 2007)   **http://tools.ietf.org/html/rfc5023**

Notes

Atom is both a protocol and an XML format for content providers to provide XML based web feeds of updated content. An Atom feed is a web feed provided using the Atom protocol and format. This provision of updated content is known as syndicating a web feed. Web users can subscribe to a feed allowing them to see new content as soon as it is made available.

Atom is described by two proposed standards defined by two Internet RFCs. Consult the following RFCs for complete, and authoritative, information about Atom.

The Atom Syndication Format

The Atom Publishing Protocol

Atom feeds from CICS…

- **Enables CICS applications to:**
  - Provide live information for Web 2.0 consumption
  - Integrate with related data
  - Give full picture in a single holistic view

- **Create new applications based on up-to-date content and information**
  - Decision-support tools for knowledge workers
  - Composite user interfaces for expert workers
  - Information feeds & widgets to consumers for use in their own mashups

- **Develop using WebSphere sMash or RD/z with EGL**

Mainframe

z/OS

CICS Transaction Server

ATOMSERVICE

DB2 for z/OS

Customers & Insurance Policies

Atom feeds

Unix

IBM Lotus Mashups Server

Servlet

Map API

PC Browser
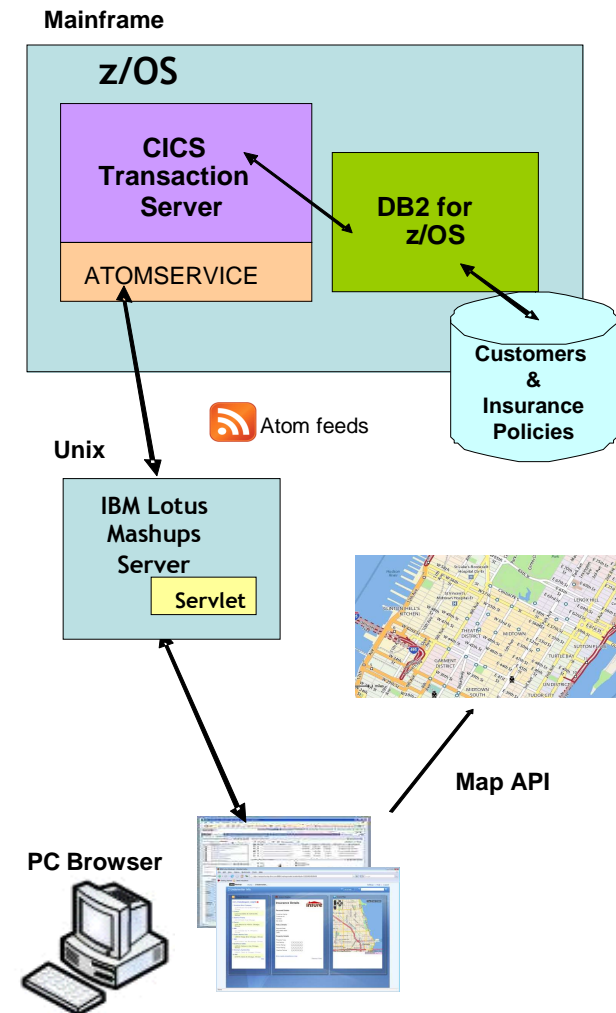
High level architecture: CICS Atom feeds

Notes

Business users can take advantage of CICS Atom feeds to quickly create mashups, Web 2.0 applications that pull information from a variety of sources and aggregate it into a single, dynamic application. Businesses can exploit mashup technology in a variety of ways, including these four:

Create decision-support tools for knowledge workers who require access to a variety of commercial and public, structured and unstructured, data and content.
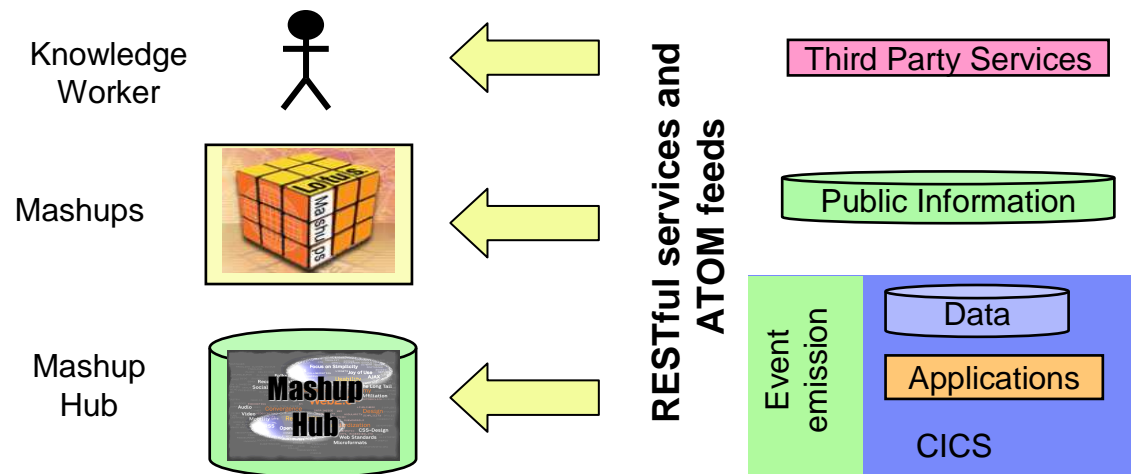
Build composite, dynamic user interfaces for workers whose daily activities involve accessing data and functions in multiple, disparate applications and systems.

Make new location-aware applications combining enterprise data with presence information from mobile providers and mapping and geo-spatial information from third parties.

Deliver information feeds and widgets to consumers to use in their own mashups,

CICS Mashup example: Insurance Underwriting

- **Insurance underwriting decisions require access to rich information from a broad variety of sources**

- **Many decisions need to be taken**
  - Identification of risk levels
  - Policy approval
  - Co-insurance and re-insurance decisions etc.

- **Information is needed from a wide variety of sources:**
  - Operational systems, Public sources, Third party sources

- **Access to these information sources, has typically involved using separate tools to access a large number of different systems**

Knowledge Worker

Mashups

Mashup Hub

RESTful services and ATOM feeds

Third Party Services

Public Information

Event emission

Data

Applications

CICS

Notes

An example of using a Mashup with CICS Atom Feeds.

CICS Mashup example: Insurance Underwriting…

Customer list Atom feed from CICS
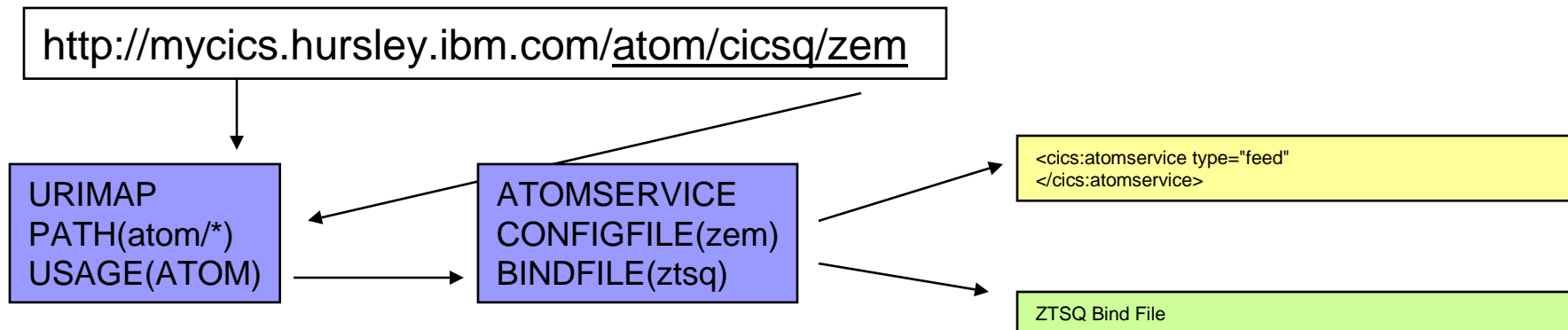
Insurance policy list Atom feed from CICS

Maps feed

Notes

An example of using a Mashup with CICS Atom Feeds.

Atom feeds from CICS…

- **Definitions for Atom Feeds**
  - URIMAP definition
  - ATOMSERVICE definition
    - Describes the type of Atom document returned
      - FEED, SERVICE, COLLECTION, CATEGORY
    - Specifies the type of CICS resource that provides the data for this Atom feed or collection
      - FILE, PROGRAM, TSQUEUE
  - Atom Service Configuration file
    - Specifies metadata/field names for the returned document
  - XML Binding file
    - Describes the CICS resource format
    - Created by the CICS XML Assistant

```
http://mycics.hursley.ibm.com/atom/cicsq/zem
```

URIMAP
PATH(atom/*)
USAGE(ATOM)

ATOMSERVICE
CONFIGFILE(zem)
BINDFILE(ztsq)

```
<cics:atomservice type="feed"
</cics:atomservice>
```

ZTSQ Bind File

Notes

The new ATOMSERVICE resource definition defines an Atom service, feed, collection, or category document, and identifies the Atom configuration file, CICS resource or application program, and Atom binding file that are used to supply the data for the feed. URIMAP resource definitions handle the incoming requests and point to the appropriate ATOMSERVICE resource definition.

The Atom configuration file contains XML that specifies metadata and field names for the Atom document that is returned for this resource definition.

The XML binding file specifies the data structures used by the resource named in RESOURCENAME, which supplies the data for the Atom document that is returned for this resource definition. You create an XML binding file using the CICS XML assistant program DFHLS2SC.

Application Bundles

- **New resource type: BUNDLE**

- **Defines a unit of deployment for an application**
  - The collection grouping is maintained for the life of the bundle install
    - All resources are enabled or disabled as a group
    - Managed by a new Resource Lifecycle domain (RL)

- **A collection of:**
  - A manifest file that describes the contents
    - Imports, exports, defines
    - Imports
      - Things the application uses
    - Defines
      - Event Binding
      - XSD Bindfile
      - SCA Composite

- **Provide an extension point for user artefacts to be deployed and managed**
  - Registration Program
    - Name (register) the callback program for a resource
    - URI that describes the resource that the callback program can manage
  - Callback Interface
    - Creates and manages the life cycle of a user resource
      - Create, enable, disable, or discard the user resource

Notes

You can now deploy applications into CICS using bundles. A bundle is a collection of CICS resources, artifacts, references, and a manifest that represent an application. Use bundles to more easily manage the availability of an application and the life-cycle of its resources.

Bundles are created by an application developer using a tool such as Rational Developer for System z or the CICS Explorer. A bundle contains only the resources that are required by the application. The system resources that the application requires might be defined as prerequisites, but they are not included in the bundle. This separation means that you can install the same application into multiple CICS regions without having to repackage or redeploy the bundle.

A bundle is defined in CICS using a BUNDLE resource.

The BUNDLE resource is different from an RDO group because it maintains a relationship with all of the resources after they are installed, so that you can manage all of the related resources as a single entity; for example, if you disable a BUNDLE resource because you want to stop an application from running, CICS disables all of the related application resources for you. To view the contents of a bundle and the state of its resources, use the IBM CICS Explorer.

Use the bundle registration interface to register a program as the owner of a resource type that is deployed in an application bundle. The program is responsible for managing the life cycle of any resources of a particular type that are installed in a CICS region as part of a bundle deployment.

To define your own user resources in a bundle, write a callback program to handle the management of the user resources when the bundle deploys into the CICS region. CICS provides a registration interface to register the callback program as the owner of a user resource type. The registration interface is channel-based and provides containers that your registration program must populate.

When a bundle contains a resource type that the callback program is responsible for handling, CICS calls the program to create the resource using a channel-based callback interface.

Service Component Architecture

A new service-oriented programming model for IBM middleware to simplify the development of applications and integration of applications into solutions

- **Service Component Architecture**
  - A concrete manifestation of an SOA way of thinking
  - Designed for building agile service oriented applications
  - A framework for implementing, assembling, composing and deploying services
  - Supports loose or tight coupling of coarse or fine grained services
  - Language, Application Environment, Framework and Vendor neutral
  - Supports Java and Web Services, and more
- **An extensible set of:**
  - Protocol bindings (eg. SCA, WS, RMI, …)
  - Implementation languages (eg. Composite, Java, …)
  - Interface definitions (eg. WSDL, Java, …)
  - Pluggable Data bindings (eg. PoJo, JAXB, …)
  - Policies and Intents (eg. Integrity, Confidentiality).

Notes

"Service Component Architecture (SCA) provides a programming model for building applications and systems based on a Service Oriented Architecture. It is based on the idea that business function is provided as a series of services, which are assembled together to create solutions that serve a particular business need. These composite applications can contain both new services created specifically for the application and also business function from existing systems and applications, reused as part of the composition. SCA provides a model both for the composition of services and for the creation of service components, including the reuse of existing application function within SCA compositions.

SCA is a model that aims to encompass a wide range of technologies for service components and for the access methods which are used to connect them. For components, this includes not only different programming languages, but also frameworks and environments commonly used with those languages. For access methods, SCA compositions allow for the use of various communication and service access technologies that are in common use, including, for example, Web services, Messaging systems and Remote Procedure Call (RPC)."

Specifications:

http://osoa.org/display/Main/Service+Component+Architecture+Specifications

Service Component Architecture…

- **Provide capability to easily develop flexible and reusable CICS application components**
  - Rapid assembly and deployment of new Services
  - Express existing applications as re-usable components

- **Separation of bindings from application code allows flexible infrastructure changes**

- **Reduce skills and effort required to view and manage business applications**
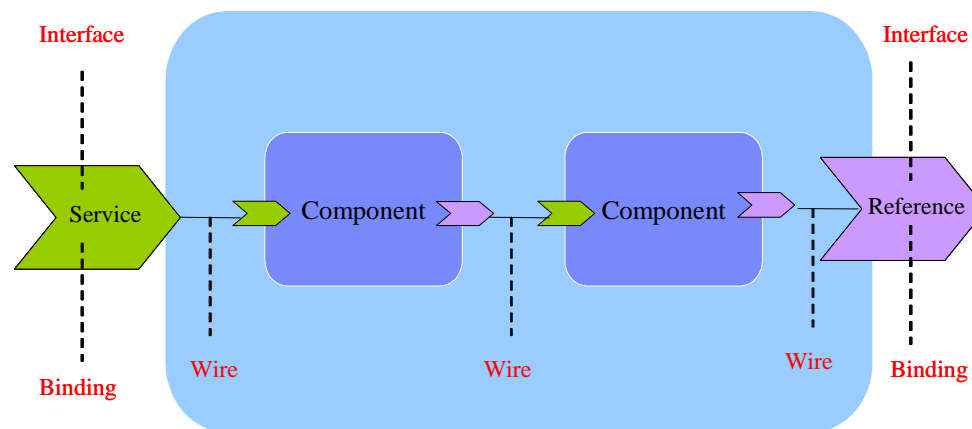
Notes

CICS provides support for applications that conform to the Service Component Architecture (SCA) specifications. SCA supports rapid deployment of new applications to meet changing business requirements by promoting the reuse of existing application assets in a component model.

In SCA, applications are composed of services, where each service is described using one or more components. These components are assembled together into a composite application that can participate in a service-oriented architecture (SOA) as part of a business service. A business service is a service that is aligned with business processes and models rather than a technical implementation.

The SCA component model is flexible, so you can reuse the same components in different composite applications without needing to change the underlying business logic.

To create a composite application, use the SCA tooling in Rational Developer for System z. The SCA tooling deploys composite applications to CICS as bundles and can be enabled and disabled using the BUNDLE resource.

You can create two types of service from your CICS applications:

Channel-based services

These services use the Service Component Architecture (SCA) support to describe applications as components. The interface for these services is a channel. Channel-based services can be called only by other CICS applications using the INVOKE SERVICE API command.

XML-based services

These services are typically Web service provider or requester applications that use XML to interface with other applications and use a binding to transform the data. Web service applications can also be described as components using SCA. The interface for these services is XML with a binding to transform the data. XML-based services can be called by other CICS applications using the INVOKE SERVICE API command or by an external client.

Service Component Architecture…

- **CICS SCA Infrastructure**
  - Types of services
    - **Channel based services**
      - Allow CICS LINKable assets to be defined as a component interface
        - > Channel and container support
        - > COMMAREA support
      - Available to other CICS programs that use the INVOKE SERVICE command
    - **XML based services**
      - Available to CICS applications that use the INVOKE SERVICE command
      - Available to business services on an external network

  - Use SCDL to describe and deploy a composite
    - RDz SCA tooling
      - Wizards for CICS Component and Composite creation
      - Composite editor
      - Wizard for Bundle creation/deployment

  - New EXEC CICS INVOKE SERVICE command
    - INVOKE WEBSERVICE command now a synonym of INVOKE SERVICE

Notes

Channel-based services are CICS applications that are described as components and assembled together using a tool such as Rational Developer for System z. These services are available only to other CICS applications that use the INVOKE SERVICE API command and pass binary data in containers on a channel.

In a service component architecture, the application program that you want to expose as a channel-based service is the implementation of the business logic. The application program is defined in the <Implementation> element of a component. The service definition of the component describes the interface to the application program in WSDL. There is one interface for a component, which can contain operations and bindings. For a channel-based service, the binding is described in the binding.cics section of the SCDL. There are no data mappings for a channel-based service, because the application interface expects a channel and binary data in containers.

XML-based services are Web service provider or requester applications that use XML to interface with other applications and use a binding to transform the data. XML-based services are available to CICS applications that use the INVOKE SERVICE API command or to business services that are on an external network.

You can either create Web services using the Web services support in CICS or you can use Rational Developer for System z. If you use Rational Developer for System z, you can also create a service architecture component from your Web service. There are some advantages to creating a component from a Web service:

You can more reuse existing components to rapidly develop new composite applications using Rational Developer for System z.

You can use SCDL to describe the Web service, moving the configuration information out of the application and into metadata that is easier to change without having to change the application. For example, if you want to run a Web service under different transaction and user IDs, you can change the SCDL without having to regenerate the Web service binding file.

In a service component architecture, the application program that you want to expose as a Web service is the implementation of the business logic. The application program is defined in the <Implementation> element of a component. The service definition of the component describes the interface to the application program in WSDL. There is one interface for a component, which can contain a number of operations and bindings. For a Web service, the binding is described in the binding.ws section of the SCDL. The SCDL also includes the data mappings that transform the XML to the appropriate high-level language.

Java 6

- **Support for provided the IBM 31-bit SDK for z/OS Java Technology Edition, V6**

    - Pre-requisite for CICS Web services and XML assistants, Java programs in CICS
    - Compliant with the Java SDK 6 compatibility tests
    - Can utilise System z Application Assist Processors (zAAPs)
    - Support for Java 1.4.2 and Java 5 has been removed
        - Java SDK 6 supports upward compatibility

Notes

CICS TS V4.1 supports the JVM provided by the 31-bit version of the IBM SDK for z/OS, Java Technology Edition, Version 6. With Java 6, you continue to benefit from the new Java language features, improved garbage collection, and simpler class sharing, first introduced in Java 5.

CICS TS V4.1 does not support the 64-bit version of the IBM SDK for z/OS, Version 6.

XML & Web Services

- **CICS Web Services Addressing**
  - Defines transport neutral mechanisms to address Web Services
    - Facilitates message transmission through networks
  - Consists of a SOAP Header describing
    - Endpoint Reference
      - Uniquely identify the service requested
    - Message addressing properties
      - Convey information about message relationships
      - Provide information on where messages are to be directed
  - WS-Addressing Specifications
    - Recommended
      - http://www.w3.org/TR/ws-addr-core/
      - http://www.w3.org/TR/ws-addr-soap/
      - http://www.w3.org/TR/ws-addr-metadata/

    - Submission
      - www.w3.org/Submission/ws-addressing

Notes

CICS supports services that use the Worldwide Web Consortium (W3C) Web Services Addressing (WS-Addressing) specifications. This family of specifications provides transport-neutral mechanisms to address Web services and facilitate end-to-end addressing.

Web Services Addressing (WS-Addressing) is a Worldwide Web Consortium (W3C) specification that aids interoperability between Web services by defining a standard way to address Web services and provide addressing information in messages. The WS-Addressing specification introduces two primary concepts: endpoint references, and message addressing properties
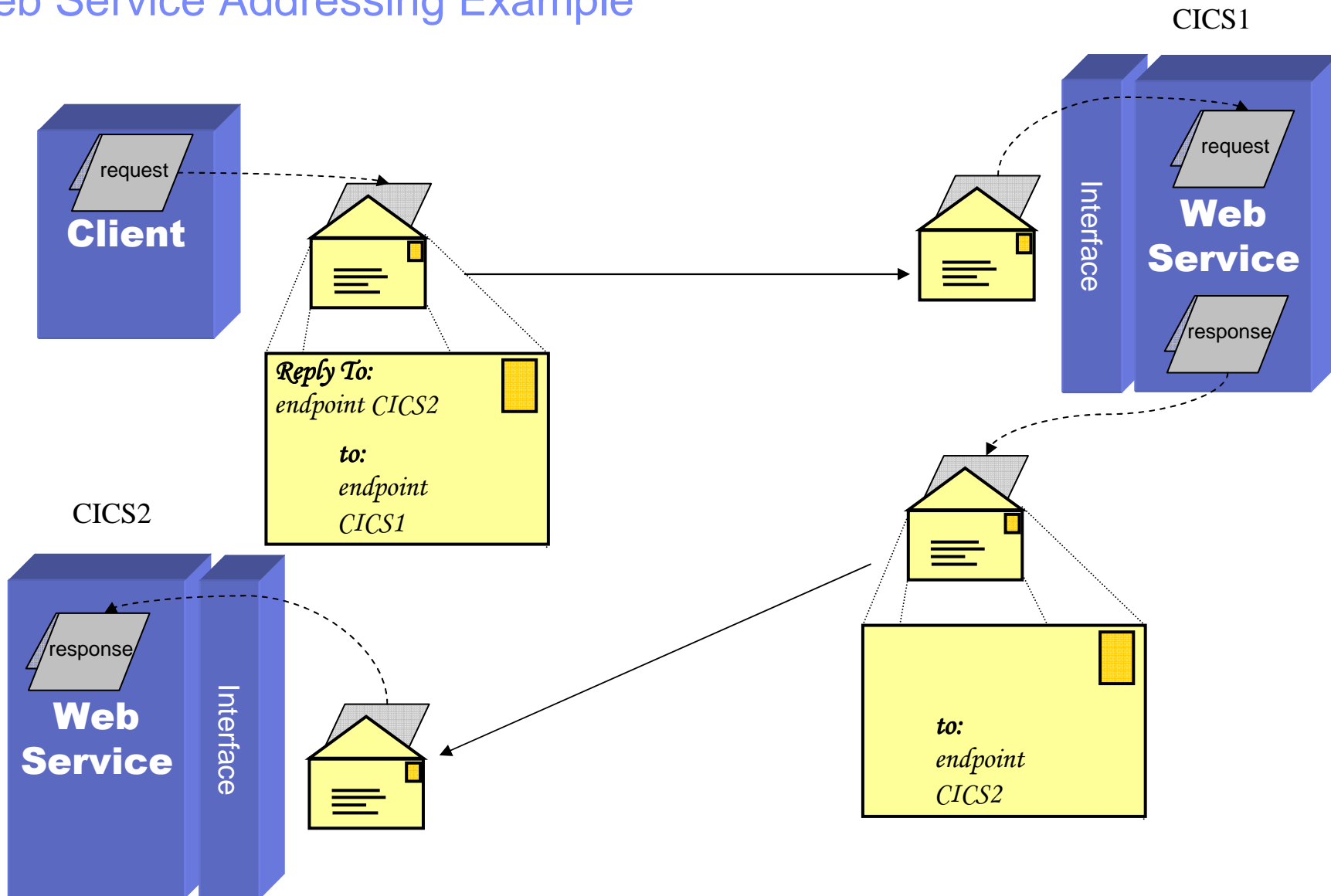
Endpoint references provide a standard mechanism to encapsulate information about specific endpoints. Endpoint references can be propagated to other parties and then used to target the Web service endpoint that they represent.

Message addressing properties (MAPs) are a set of well defined WS-Addressing properties that can be represented as elements in SOAP headers and provide a standard way of conveying information, such as the endpoint to which message replies should be directed, or information about the relationship that the message has with other messages.

By default, CICS supports the W3C WS-Addressing 1.0 Core and W3C WS-Addressing 1.0 SOAP Binding specifications that are identified by the http://www.w3.org/2005/08/addressing namespace. Unless otherwise stated, WS-Addressing semantics that are described in this documentation refer to these specifications.

For interoperability, CICS supports only the W3C WS-Addressing Submission specification with the namespace http://schemas.xmlsoap.org/ws/2004/08/addressing.

Web Service Addressing Example

CICS1

Client

request

Web Service

Interface

request

response

Reply To:
endpoint CICS2

to:
endpoint
CICS1

CICS2

Web Service

Interface

response

to:
endpoint
CICS2

Notes

An example of how WS Addressing may be used.

XML & Web Services…

- **CICS Web Service Addressing…**

  - CICS Pipeline Configuration (Requester and Provider)

    - New SOAP header handler to processing WSA constructs

  - New CICS commands

    - EXEC CICS WSAEPR CREATE

      - Create an endpoint reference (EPR) to represent a Web service or Web service resource

    - EXEC CICS WSACONTEXT BUILD

      - Build an addressing context

    - EXEC CICS WSACONTEXT GET

      - Get the message addressing properties (MAPs) of the service requester
      - Get the MAPs of a service provider

Notes

CICS service provider and requester programs need not be changed to use the WS Addressing functions. A CICS provided SOAP pipeline header handler can process the WSA elements and shield the application from the need to know about WS Addressing.

Web Services Addressing (WS-Addressing) introduces new commands to manipulate addressing contexts and endpoint references (EPRs).

XML & Web Services…

- **New Markup Language Domain (ML)**
  - Uses the z/OS systems services parser
    - Eligible for off-load to zAAP engines
    - Parsing storage acquired from 64 bit storage

- **Generic XML Mapping**
  - EXEC CICS TRANSFORM command
    - XML to Data
    - Data to XML

  - New XML Assistants
    - Language structure to schema
    - Schema to language structure
    - Generates artifacts necessary to define a BUNDLE
      - New XMLTRANSFORM resource definition

Notes

z/OS XML System Services (z/OS XML®) parser is a system level XML parser that is integrated with the base z/OS operating system. It is intended for use by system components, middleware, and applications that need a simple, efficient, XML parsing solution.

A new CICS domain is introduced, the CICS Markup Language Domain (ML). This domain will invoke the systems services parser as required. Storage necessary for the parsing process will be acquired from the 64 bit DSA above the bar. If the z/OS LPAR is configured with zAAP engines then systems XML parser activity will be offloaded to those engines.

The CICS XML assistant is a set of batch utilities that can help you to transform XML into high-level language structures and vice versa. The assistant supports rapid deployment of applications that perform XML processing with the minimum amount of programming effort.

Using the XML assistant for CICS reduces the amount of code that you must write to parse or construct XML; CICS transforms data between XML fragments and the data structure of an application program.

The XML assistant can create an XML schema from a simple language structure, or a language structure from an existing XML schema, and supports COBOL, C/C++, and PL/I. It also generates metadata that CICS uses at run time to automatically convert XML data to binary application data or vice versa; the metadata is defined in an XML binding file and stored on z/OS UNIX.

CICS provides a new EXEC CICS TRANSFORM command to transform application data to XML or XML to application data. The XML bind file produced by the XML assistants is used in the transformation.

CICS Explorer

- **Intuitive and common tooling for CICS architects, analysts, developers, administrators**
  - Quick and easy to install via Web browser
  - Eclipse based, runs on Linux and Windows
  - Fast, highly customizable and extensible
- **CICS Explorer provides operations, workload, resource management and application deployment**
  - Rich set of views, tasks and editors
  - Supports a single CICS region to large CICSplex
- **Supports CICS tools and the CICS Transaction Gateway**
  - Other IBM tools
  - Easy to link and perform tasks across products
  - Unified resource representation and terminology
- **Transfer skills, knowledge and best practice to new CICS technical staff**
- **Powerful, context-sensitive resource editors**
  - Show only applicable attributes and tabs
  - Clear explanation of options and context help
  - Parameter case sensitivity, length and relationship to other parameters handled by intelligent controls
  - Instant feedback for errors
- **Create your own dashboard to quickly perform common tasks**
 - Save your windows, views and filters

Notes

The CICS Explorer is a system management tool that is designed to provide a simple, easy to use way of managing one or more CICS systems. Based on an Eclipse platform, it provides a view of some of the CPSM functions in a CICS Transaction Server and a platform for the integration of future CICS tools.

Eclipse is a platform for building and deploying client applications in which the majority of data manipulation is done by the client application rather than the server. This platform is known as a Rich Client Platform (RCP). The platform provides the ability to deploy native graphical user interface (GUI) applications to a variety of desktop operating systems. The CICS Explorer is an Eclipse RCP based GUI application that runs on the Windows and Linux operating systems.

Today, you manage CICS regions with a variety of stand-alone graphical and non-graphical user interfaces, and with no common way of performing tasks. The CICS Explorer provides a base set of CICSPlex SM functions to manage CICS regions, and a platform on which future CICS tools that are integrated in the Eclipse RCP, can perform tasks and present the information in a common way.
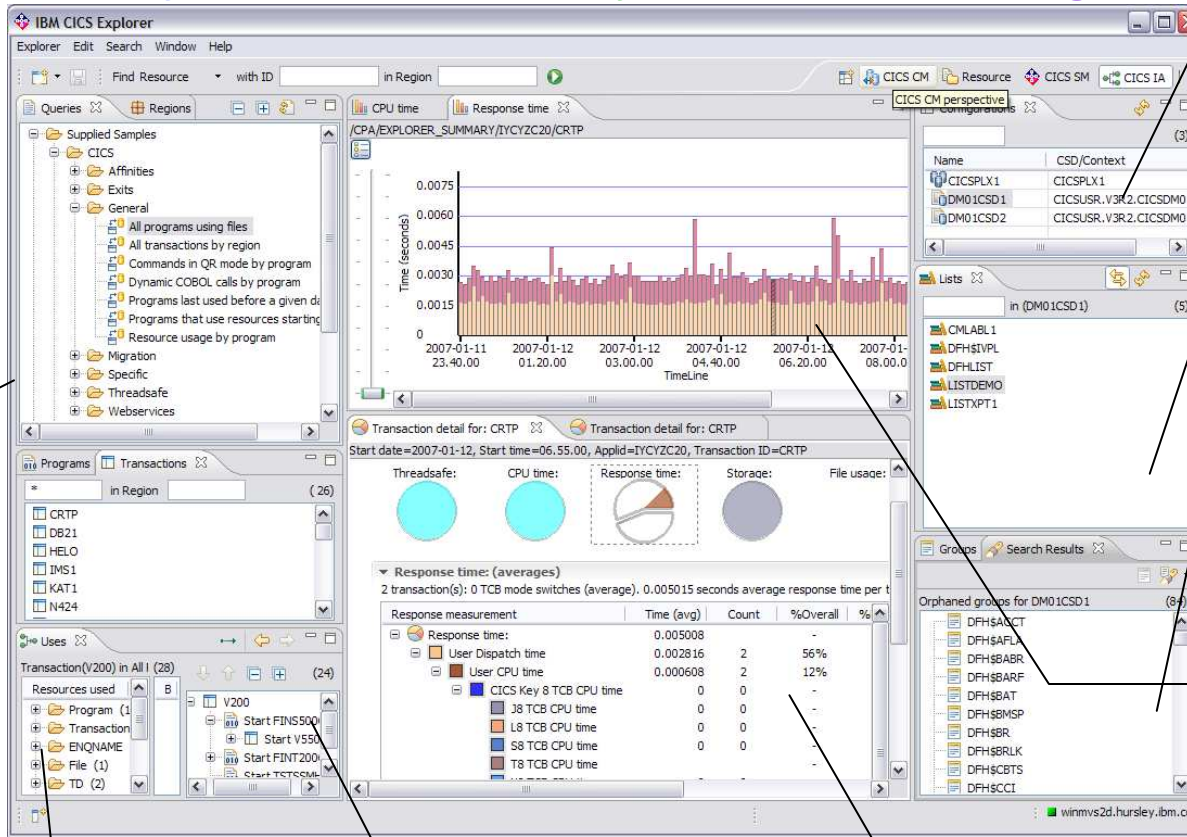
CICS Explorer…
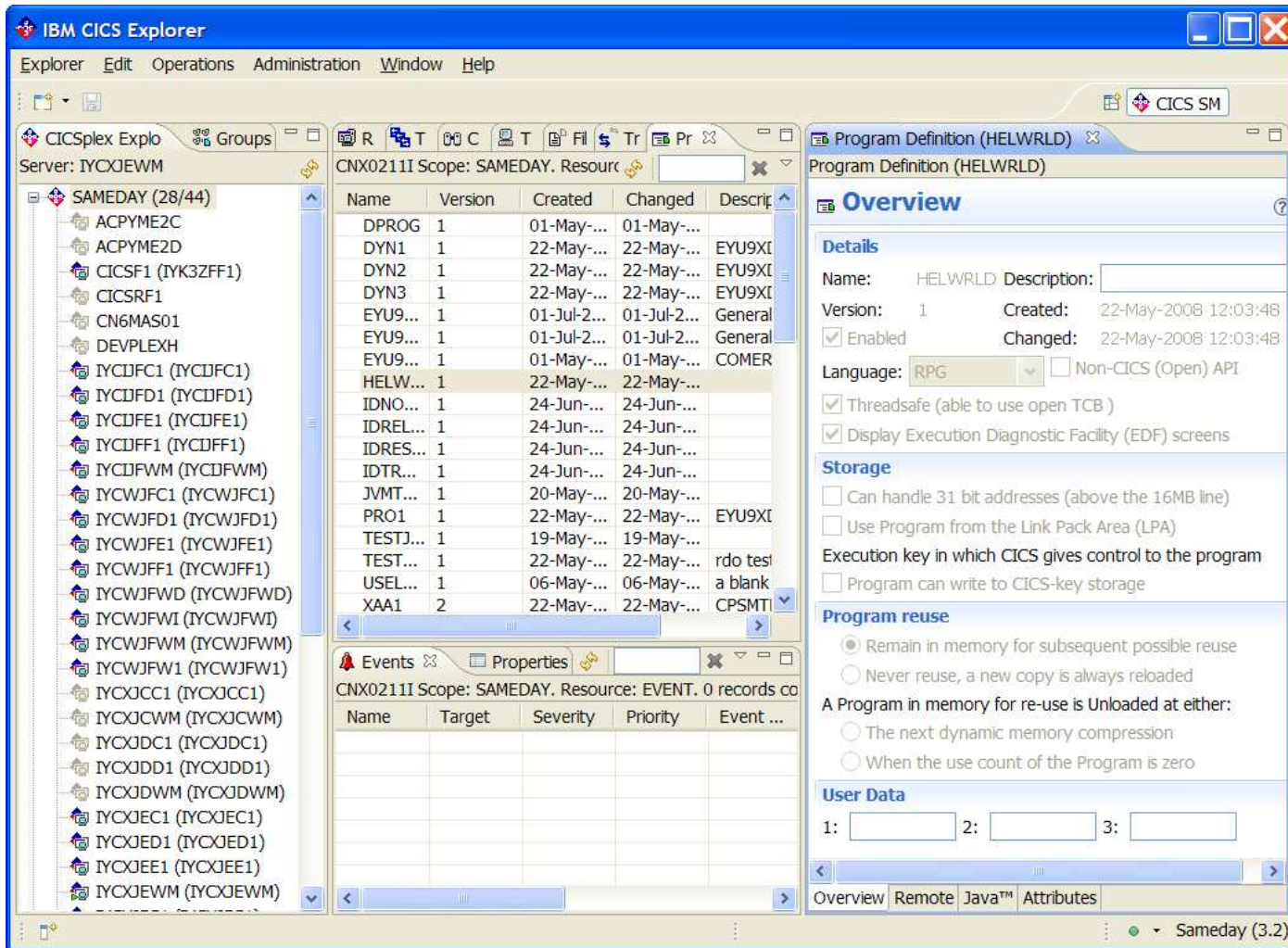
Notes

An example of a CICS Explorer view.

CICS Explorer…

Interdependency Analyzer **Performance Analyzer** **Configuration Manager**

Single point of control for CSDs and DREPs

Lists and ResDescs

Search for Orphaned groups

Timeline of response times

Shipped Sample Queries

View tree of resources used

Resources used by a transaction

Drilldown into transaction

Notes

An example of a CICS Explorer view illustrating the CICS Tools plug-ins.

IPv6 Background

- **Evolution of the current version of IP (IPv4)**
  - Work started on this in the early 90's

- **IPv4 has <u>32 bit</u> addresses**
  - IPv4 address – 10.67.122.66
  - Practical limit: less than 1 billion useable global addresses

- **IPv6 has <u>128 bit</u> addresses**
  - IPv6 address – 2001:0db8:0000:0000:0000:0000:1428:57ab
  - No practical limit on global addresses
    - 2^{128} addresses
    - 5×10^{28} addresses for each of the 6.5 billion people alive today

- **IPv4 format**
  - Native   1.2.3.4   with each element 0-9, 00-99 or 00-255

- **IPv6 format to represent IPv4**
  - Compatibility   ::1.2.3.4   with each element 0-9, 00-99 or 00-255
  - Mapped   ::FFFF:1.2.3.4   with each element 0-9, 00-99 or 00-255

- **IPv6 format**
  - 1:2:3:4:5:6:7:8   with each element 0-F, 00-FF, 000-FFF or 0000-FFFF
  - If one or more 4 digit groups are 0000 the zeros may be replaced by two colons

Notes

IPV6 Background

From Wikipedia:

"Internet Protocol version 6 (IPv6) is a network layer for packet-switched internetworks. It is designated as the successor of IPv4, the current version of the Internet Protocol, for general use on the Internet. The main change brought by IPv6 is a much larger address space that allows greater flexibility in assigning addresses. The extended address length eliminates the need to use network address translation to avoid address exhaustion, and also simplifies aspects of address assignment and renumbering when changing providers. It was not the intention of IPv6 designers, however, to give permanent unique addresses to every individual and every computer.

It is common to see examples that attempt to show that the IPv6 address space is extremely large. For example, IPv6 supports 2^{128} (about 3.4×10^{38}) addresses, or approximately 5×10^{28} addresses for each of the roughly 6.5 billion (6.5×10^9) people alive today. In a different perspective, this is 252 addresses for every star in the known universe – more than ten billion billion billion times as many addresses per star as IPv4 supported."

You can specify an IPv4 address in one of the following formats:

As a native IPv4 dotted decimal address (for example, 1.2.3.4)

The following two formats are the IPv6 way of representing an IPv4 address

Using IPv6 syntax for an IPv4 migrated address (for example, ::FFFF:1.2.3.4)

As an IPv6 compatible address (for example, ::1.2.3.4)

You can specify an IPv6 address in the following format:

As a native IPv6 colon hexadecimal address (for example, ::a:b:c:d) without square brackets. IPv6 address syntax is described in RFC 4291, IP Version 6 Addressing Architecture, available from http://www.ietf.org/rfc/rfc4291.txt.

IPv6 addresses are normally written as eight groups of four hexadecimal digits, where each group is separated by a colon (:). For example, 2001:0db8:85a3:08d3:1319:8a2e:0370:7334 is a valid IPv6 address.

If one or more four-digit group(s) is 0000, the zeros may be omitted and replaced with two colons(::). For example, 2001:0db8:0000:0000:0000:0000:1428:57ab can be shortened to 2001:0db8::1428:57ab. Following this rule, any number of consecutive 0000 groups may be reduced to two colons, as long as there is only one double colon used in an address. Leading zeros in a group can also be omitted (as in ::1 for localhost)

CICS Support for IPv6

- **Allow for IPv4, IPv6 or host names in:**

  - Resource definitions

  - Application Programming Interface

  - Systems Programming Interface

  - User Replaceable Modules

  - Global User Exits

  - Monitoring Records

Notes

With CICS TS V4.1, you can now operate in an Internet Protocol version 6 (IPv6) network without the need for address translation. CICS still operates in an IPv4 network as before.

IPv6 is the protocol designed by the IETF to replace the current version Internet Protocol, IP Version 4 (IPv4). The IPv6 format has a much larger address space than IPv4 (128 bits compared to 32 bits), which provides greater flexibility in allocating addresses and for routing traffic.

A dual-mode environment operates in both IPv4 and IPv6 networks and always attempts to communicate using IPv6 before using the IPv4 network. A single-mode environment operates in an IPv4 network only. You can communicate using IPv4 and IPv6 if you have a CICS TS V4.1 region running in a dual-mode (IPv4 and IPv6) environment that is communicating with a CICS TS V4.1 region that is also running in a dual-mode environment.

IP Interconnectivity

- **Continuation of the CICS IP interconnectivity strategy**
  - Provide a new transaction IP communications protocol for connectivity between and into CICS
  - Long term plan to provide CICS with IP choice for most of the CICS programming model
  - Performance benefits:
    - Reduced CPU usage compared to VTAM/SNA
    - Exploitation of 10 Gigabit Ethernet through OSA and QDIO hardware
- **Enhancements to support 3270 transaction routing**
  - BMS, security and monitoring support
  - Shippable terminals supported
  - Restrictions
    - Routable(YES) on transaction definition
    - No EDF
    - No routing for APPC devices
- **Enhancements to Asynchronous Starts**
  - ATI over IPIC supported

Notes

CICS has a multi-release plan to provide a new transactional IP protocol between CICS regions and into CICS systems. This strategy will allow CICS to realize value from advances in TCP/IP, exploit System z technology (e.g. hipersockets) and provide for a standard IP CICS infrastructure for access into CICS.

IPIC supports these intercommunication functions and releases:

Distributed program link (DPL) calls between CICS TS 3.2, or later, and other CICS TS 3.2, or later, regions.

Transaction routing between CICS TS 4.1 and other CICS TS 4.1 regions.

Asynchronous processing of START, START ATTACH, START CHANNEL and CANCEL commands between CICS TS 4.1 and other CICS TS 4.1 regions.

ECI requests from CICS Transaction Gateway Version 7.1 or later.

Up to two send and two receive sockets are now available for each IPIC connection, replacing the single send and single receive socket. The increase in the number of sockets can improve your transaction throughput. Additional performance enhancements include improved performance for two phase commit processing and CPU consumption better or equal to VTAM usage.

MQ Group Attach

- **Connect to any active member of a WMQ Queue Sharing Group**

  – Allows common resource definitions for CICS regions

  – Connect to only one QMGR at a time
    - RESYNCMEMBER attribute for in-doubt resolution
  – QMGR must be on the same LPAR

- **Changes to CICS externals**
  – Resource definition
    - New MQCONN resource

  – SPI
    - EXEC CICS SET MQCONN
      – Quiecse or forceclose the connection
    - CPSM WUI views
    - CICS Explorer support

Notes

You can now specify a WebSphere MQ queue-sharing group for the CICS-WebSphere MQ connection, so CICS uses any eligible queue manager in the group when it reconnects to WebSphere MQ, rather than waiting for a single queue manager. Queue-sharing groups increase reliability when you reconnect to WebSphere MQ, and help you standardize this aspect of CICS setup across CICS regions and z/OS images.

Instead of defining default settings for the CICS-WebSphere MQ connection in the DFHMQPRM operand of an INITPARM system initialization parameter, you must now use the new MQCONN resource definition. You can use the MQCONN resource definition to specify a queue-sharing group, or you can use it to specify the name of a single queue manager.

If you have specified a queue-sharing group for the connection, you can select appropriate resynchronization actions for CICS using the RESYNCMEMBER attribute of the MQCONN resource definition. Resynchronization works in the same way as it does for the group attach function for DB2. Resynchronization takes place when the connection to WebSphere MQ is lost and CICS is holding outstanding units of work for the last queue manager. You can choose whether CICS waits to reconnect to the same queue manager, or whether CICS makes one attempt to reconnect to the same queue manager, but if that attempt fails, connects to a different eligible queue manager in the group. A queue manager is eligible for connection to a CICS region if it is currently active on the same LPAR as the CICS region.

An MQCONN resource defines the attributes of the connection between CICS® and WebSphere MQ. You can install or discard an MQCONN resource only when CICS is not connected to WebSphere MQ.

Only one MQCONN resource can be installed at a time in a CICS region. When you define an MQCONN resource, CICS checks to ensure that there is only one MQCONN resource defined in the group or list. If more than one is found, even one with a different name, a warning message is issued. If you do install a second MQCONN resource, CICS implicitly discards the existing MQCONN resource before proceeding with the installation, unless you are reinstalling an MQCONN resource with the same name.

When you have installed the MQCONN resource, you can use the CEMT or EXEC CICS SET MQCONN command, CICSPlex SM, or the CICS Explorer to start the connection between CICS and WebSphere MQ.

Workload Management and CICSPlex SM

- **Sysplex-optimization to significantly reduce workload batching effects**

  – Exploitation of z/OS coupling facility

    - "Near real time" Sysplex-wide focus on target region status
    - No impact to "non-optimized" WLM

  – Optimized WLM routing enabled by configuring a Region Status Server

    - Uses CF Data Table to hold Region Status information
      – SOS, MaxTask, System or transaction dump in progress, Current Tasks?
    - Shared by all routing regions (in the Sysplex)

- **Percentile goals**

  – CICSPlex SM WLM support for percentile goals

- **Support in CICSPlex SM for all new resources, statistics, etc**

Notes

In CICS TS V4.1, workload throughput is improved through a more efficient workload management optimization function. This function is most effective for distributed workloads, for which the routing and target regions are managed by different CMASs. With new CPSM WUI views, you can monitor the distribution of dynamic workloads through your CICSplex.

Sysplex optimized workload routing is enabled at the z/OS coupling facility level by a region status server. If the coupling facility is not available, workload routing is managed by CICSPlex SM Workload Manager (WLM) using z/OS dataspaces owned by a CMAS to share cross-region load and status data.

CICS TS V4.1 provides support for z/OS workload manager percentile goals.

A percentile goal sets the percentage of transactions to be completed in the target response time. For example, consider a percentage goal set to 80% and response time set to 1 second. If a period contains 200 transactions, 160 of them must complete with a response time less than or equal to 1 second to meet the percentile goal.

Before CICS TS V4.1, any percentile goals were ignored by CICSPlex SM. So customers who use the CICSPlex SM goal algorithm and have z/OS WLM percentile goals will see different routing behavior when they migrate to CICS TS V4.1.

CPSM WLM: Optimized versus Non-optimized Routing

Non Sysplex Optimized Workload

Sysplex Optimized Workload
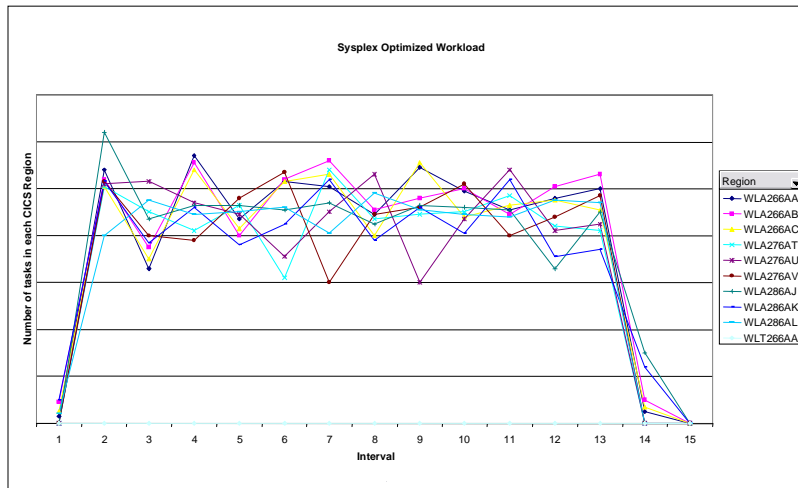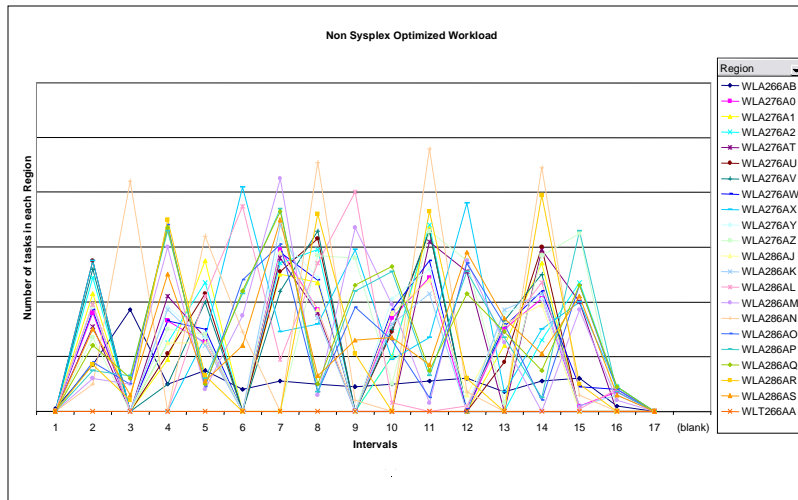
- **Non-optimized Routing**
  - 27 CICS regions on 3 LPARs

- **Optimized Routing**
  - 9 CICS regions on 3 LPARs

- **Comparison**
  - Workload is 10K started transactions
  - Number of tasks in each region measured every 10 second interval

- **Results**
  - Non-optimized environment shows "batching effects
  - Optimized results shows smoother distribution of work and higher throughput

Notes

This chart illustrates the difference between CICSPlex SM Sysplex Optimized Routing and non-optimized routing. Ten thousand (10,000) transactions were started in both cases. In the non-optimized case there were 27 CICS regions on 3 LPARs. In the optimized case there were 9 CICS regions on 3 LPARs.

What can clearly be seen in the non-optimized case (top chart) is the latency or batching that is occurring on the remote AORs. An AOR is routed to until it reaches MAXTASK and then the router discovers it is too busy and routes nothing to it. This occurs on the 15 second CPSM cycle as the heartbeat gathers the health information on the remote regions. An AOR is found to be not busy, CPSM routes to it and 15 seconds later finds it overloaded and stops routing to that AOR. This cycle is repeated over and over.

What we see in the optimized case (bottom chart) is what happens when the 15 second cycle is removed and the router has current information as to the health state of the AORs. Our results show a leveling of throughput on all the AORs and a higher throughput resulting in the 10K transaction be done in less elapsed time.

Discovery Library Adapter

- **The IBM Discovery Library facilitates a common way to share information about discovered resources and relationships**

  - A set of specifications, components, and best practices for communicating the discovery of resources and the relationships between resources within the enterprise

  - A way to exchange resource and relationship data across multiple applications.

  - XML schema specification

- **CICS is providing a Tivoli DLA for CICS resources**

Notes

The Discovery Library Adapter (DLA) for CICS is a utility that collects runtime information about CICS and CICSPlex SM resources. It generates XML files that can be used by a discovery library reader to track these resources and their interdependencies across your enterprise.

The CICS DLA uses CICSPlex SM services to generate XML files that conform to the Discovery Library IdML XML schema and Common Data Model (CDM). The books can be loaded by various products including Tivoli Change and Configuration Management Database (CCMDB) and Tivoli Business Service Management (TBSM).

The CICS DLA operates against all the CICSplexes that are associated with a specified maintenance point CMAS. The specified CMAS must be running at the current release level; however, the CICS regions discovered by the DLA can be at earlier release levels, as allowed by the cross-release support rules for CICSPlex SM.

Discovery Library Adapter…

- **CICS DLA Utility (EYUJXDDP)**
  - Runs as Batch Job or Started Task
    - Range and depth of discovery controlled by parameters
    - Select the books to create:
      - CICSPLEX, CICSREGION, CTSPLEX
  - Uses CICSPlex SM services to generate XML files
    - Writes IdML books to a PDS
  - Runs against a specified CMAS
    - CMAS must be current level of CICSPlex SM
    - CICS regions discovered by the DLA can be an earlier release

Notes

To use the CICS DLA you run EYUJXDD0 as a batch job on a z/OS system. Running the DLA creates IdML books in a PDS data set. The DLA also writes control files and debug information to this data set.

When you run the DLA, it can generate three types of IdML books:

One CTSPLEX book containing summary information about discovered CICSplexes

One or more CICSPLEX books containing information about the CICS system topology in a specific CICSplex and also the CICSPlex SM logical scopes and their associated resources.

One or more CICSREGION books containing information about CICS resources associated with a specific CICS region

The next step is to transfer the books to the discovery library file store (DLFS) located on your local system or accessible through a network connection. The DLA provides an automated mechanism for creating FTP statements and transferring files to your DLFS. From the DLFS you can load the books into CCMDB or TBSM.

You control the output of the DLA using the EYUDPARM DD parameters. These parameters control the range and depth of the discovery and specify which IdML books are created. The parameters also set general options about the operating environment, including the name of the maintenance point CMAS under which the discovery takes place. You supply these parameters in the EYUJXDD0 input file.
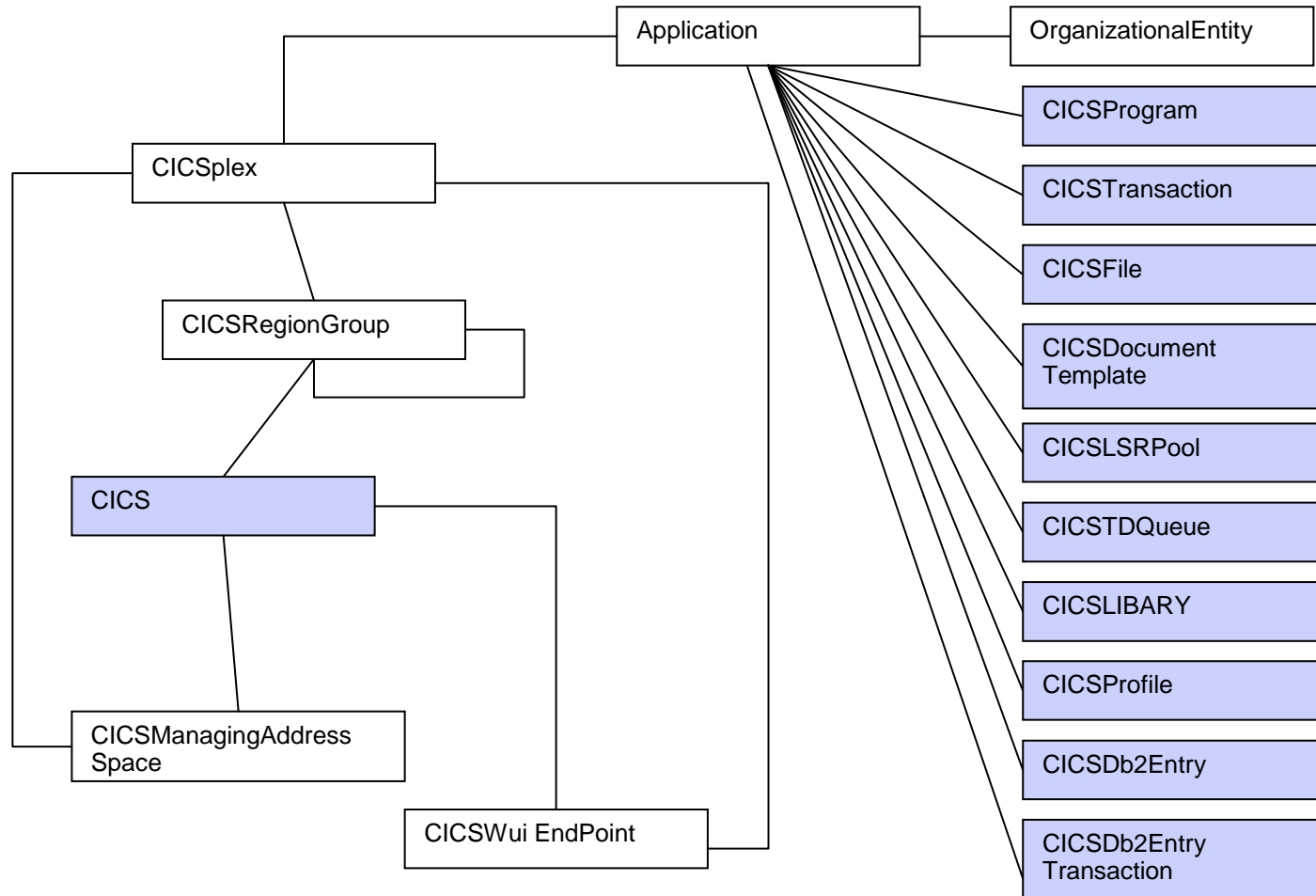
You can run the DLA in either update mode or refresh mode:

Update mode is the default. In this mode, the DLA produces a snapshot of your CICSplexes. The DLA produces books that might not contain information about all affected resources because some are not currently active.

In refresh mode, the books contain everything known about the environment. The reader treats as obsolete anything that was previously loaded from a similar book and is not in this book. CCMDB and TBSM support refresh mode by deleting these obsolete resources. Refresh mode can be useful when there has been a significant configuration change but you should use it with caution because of the large amount of processing that might be involved.
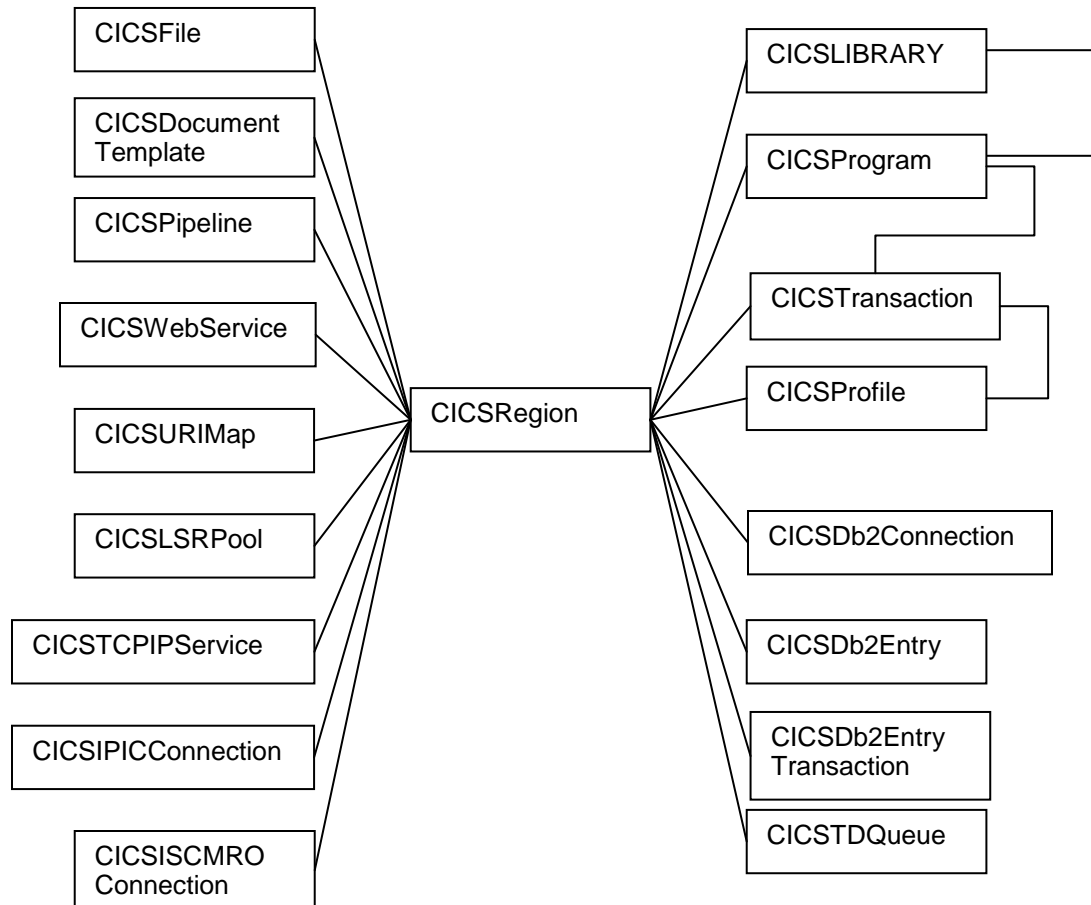
Classes and Relationships included in the CICSplex books


```
                                    Application ──── OrganizationalEntity
                                        │
                                        ├── CICSProgram
     CICSplex                           │
        │                               ├── CICSTransaction
        │                               │
     CICSRegionGroup                    ├── CICSFile
        │                               │
     CICS                               ├── CICSDocument Template
        │                               │
     CICSManagingAddress Space          ├── CICSLSRPool
        │                               │
     CICSWui EndPoint                   ├── CICSTDQueue
                                        │
                                        ├── CICSLIBARY
                                        │
                                        ├── CICSProfile
                                        │
                                        ├── CICSDb2Entry
                                        │
                                        └── CICSDb2Entry Transaction
```

Notes

An example of the CICS Data Model.

Classes and Relationships included in the CICSregion books

CICSFile

CICSDocument Template

CICSPipeline

CICSWebService

CICSURIMap

CICSLSRPool

CICSTCPIPService

CICSIPICConnection

CICSISCMRO Connection

CICSRegion

CICSLIBRARY

CICSProgram

CICSTransaction

CICSProfile

CICSDb2Connection

CICSDb2Entry

CICSDb2Entry Transaction

CICSTDQueue

Notes

An example of the CICS Data Model.

Where is this Information Used?

- **Tivoli Business Service Manager**

Notes

An example of the Tivoli Business Service Manager showing the impact of losing a CICS resource.

CICS Resource Signatures

- **Definition signature attributes added to CICS resources**
  - Signature data added when you add/alter a resource
    - DEFINESOURCE
    - DEFINETIME
    - CHANGETIME
    - CHANGEUSRID
    - CHANGEAGENT
    - CHANGEAGREL

- **Installation signature attributes added to CICS resources**
  - Signature data added when you Install a resource
    - INSTALLAGENT
    - INSTALLTIME
    - INSTALLUSRID

Notes

The resource signature, which combines the installation and definition signatures, provides specific information for resource definitions that were installed or changed in CICS® TS 4.1 or later. You can use these improved details to detect resource modifications for auditing, tracking, or for fixing problems.

The installation signature captures details as to when, how, and by whom the resource was installed. The definition signature captures details in the CICS system definition data set or the CICSPlex SM EYUDREP data repository, as to when, how, and by whom the resource was defined and the last change made to the resource definition.

The resource types that support the resource signature are: ATOMSERVICE, BUNDLE, CONNECTION, CORBASERVER, DB2CONN, DB2ENTRY, DB2TRAN, DJAR, DOCTEMPLATE, ENQMODEL, EVENTBINDING, FILE, IPCONN, JOURNALMODEL, JVMSERVER, LIBRARY, MQCONN, MQINI, PIPELINE, PROFILE, PROCESSTYPE, PROGRAM, REQUESTMODEL, TCPIPSERVICE, TDQUEUE, TRANCLASS, TRANSACTION, TSMODEL, URIMAP, WEBSERVICE, and XMLTRANSFORM.

CICS Resource Signatures…

- **Signature information**
  - CEDA
    - New panel option (PF2)
  - CEMT
  - INQ SPI
  - CICS Explorer
  - CICSPlex SM Views
  - DFHCSDUP
    - New SIGSUMM Option produces a signature report
      - LIST GROUP(group) SIGSUMM   (group can be generic)
      - LIST LIST(list) SIGSUMM
      - LIST ALL SIGSUMM

Notes

To support the definition signature, the PF2 key in the CEDA DISPLAY and EXPAND GROUP panels has a new function. To display the definition signature for an individual resource, or a group of resources, in the CEDA DISPLAY and EXPAND GROUP panels press PF2. To return to the previous CEDA command panel, press PF2 again. You can switch between the two panels while retaining scroll position.

The CEMT INQUIRE command supports the improvements in management of resource definitions for a number of resource types. The definition and installation signatures are displayed on the expanded view panel of the supported resource types. The new fields are displayed at the end of the list of attributes for the installed resource.

The INQUIRE command supports the resource signature for a number of resource types.

CICSPlex SM views support the resource signature. Detailed resource signature information can be viewed in the CICS operations views. New fields can also be displayed in the Web User Interface resource administration views.

DFHCSDUP now supports the improvements in management of resources. The definition signature fields are included in DFHCSDUP EXTRACT processing. A new option, SIGSUMM, is added to DFHCSDUP LIST to produce a summary of definition signatures for all of the specified resources

CICS Resource Signatures…
CEDA View

CICS Resource Signatures…
CEDA View

CICS Resource Signatures…
CEMT View

CICS Resource Signatures…
CICS Explorer View

WebSphere Services Registry and Repository

- **A Web service registry is the 3rd aspect of Web services**
  - Requester, provider, registry
  - WSRR provides central repository for Web services and...
    - Query/search, user defined meta-data, lifecycle, version #, relationships, dynamic service selection and binding, governance, enforce policies, federation with other repositories (e.g. UDDI)
    - Institute best practices, encourages discovery and reuse of Web services

- **CICS and WSRR**
  - Publish WSDL representing CICS Web service providers
  - Retrieve WSDL representing Web services to be used by CICS requesters
  - Complements CICS Web services assistants, WSDL editors, etc.
  - Initially made available as SupportPac CA1N for CICS TS V3

Notes

- WebSphere Service Registry and Repository provides a central repository for storing service interfaces and associated meta-data.

- It enables the management of services throughout their life-cycle, from concept through to retirement. The life-cycle is fully configurable using WebSphere Integration Developer. Thus, the SOA governance model for the service life-cycle can be represented by defining the life-cycle of the services in your organization and defining the authorization process for an application to move from one phase of the life-cycle to another. Once the life-cycle is defined, it can be installed into WSRR and used as the basis for managing the life-cycle of your services.

- Another area in which WSRR enables governance is through the use of classifications. The classification systems defined as part of your organizations governance model can be imported into WSRR and used to classify the services described in it.

- WSRR also enables documents to be tagged with properties and relationships to be defined between documents. The WSRR user interface is web based and provides search capabilities to enable services to be easily located.

CICS Support for WSRR

- **Enhance function, documentation, and provide formal support by delivering WSRR support into base CICS**

- **Support new releases of WSRR**

- **Integrate WSRR access into the CICS Web services assistants**

  - DFHLS2WS extension

    - Generate a WSDL file and WSBind file from copybooks and publish WSDL file to WSRR

  - DFHWS2LS extension

    - Extract a WSDL file from WSRR and generate copybooks and wsbind file

Notes

- The Web services assistant batch jobs DFHWS2LS and DFHLS2WS have new parameters to support interoperability with the IBM WebSphere Service Registry and Repository (WSRR) server. You can optionally secure the network connection to WSRR using secure socket layer (SSL) encryption.

- This support requires mapping level 1.2 or higher.

Identity Context Propagation

- **z/OS Identity Propagation initiative to provide asserted identity for end-to-end distributed security**
  - Logically tie together distributed end-user identities with z/OS userids
  - Enhance the ability for z/OS applications to participate centrally in SOA solutions
  - Function will require z/OS 1.11

- **ID Context Propagation allows an end user's identity to be propagated through to CICS**
  - Currently, the end users' identity is lost before the request gets to CICS

Notes

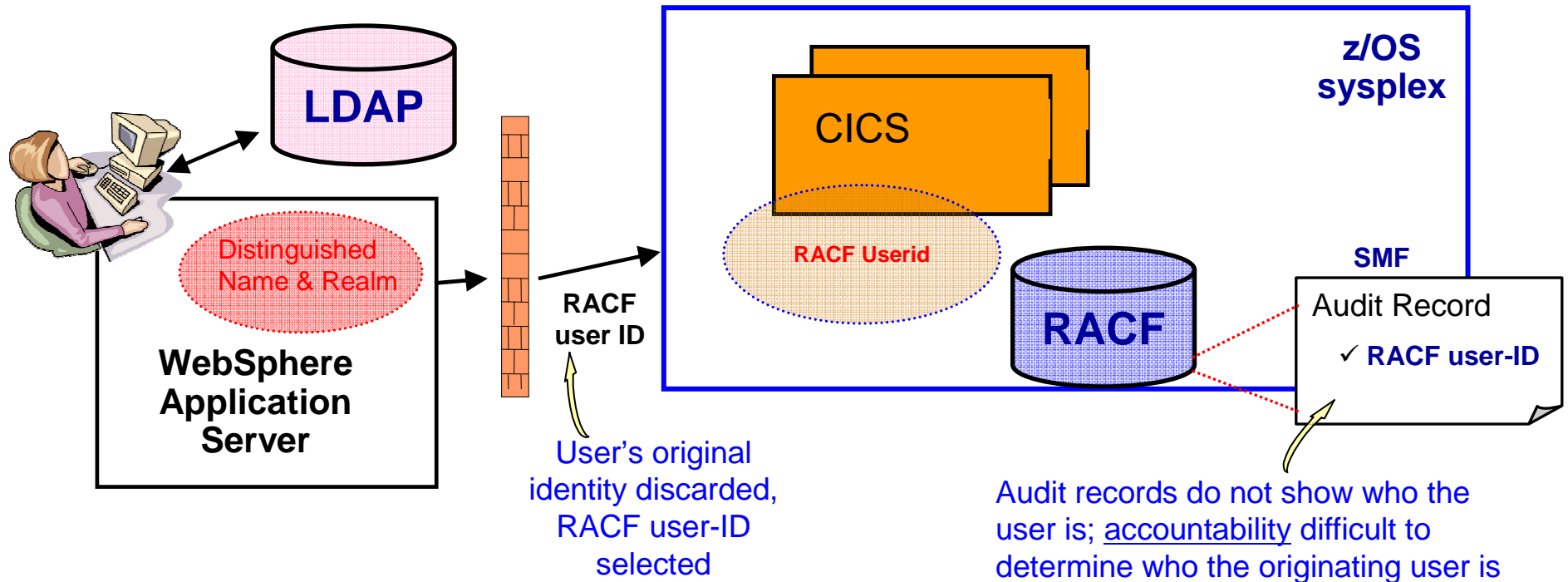**Identity context propagation (ID Prop)**

The concept of sending a user's security identity information (the distributed identity) from a client system across a network, preserving the distributed identity for use during authorization and for auditing purposes.

The distributed identity is carried with a request from the distributed client application to the server, and is incorporated in the access control of the server as part of the authorization process.

If you are using z/OS, Version 1 Release 11, you can now send a user's security identity information (the distributed identity) from a client system across a network, preserving the distributed identity for use during authorization and for auditing purposes.

Identity context propagation (ID Prop) is dependent on the z/OS Identity Propagation function that is provided in z/OS, Version 1 Release 11. ID Prop provides improved accountability and a new means of authorizing requests by associating the security information in the distributed environment with the security information used on the server.

Current Implementation of User Identity and Mapping

LDAP

WebSphere Application Server

Distinguished Name & Realm

RACF user ID

User's original identity discarded, RACF user-ID selected

z/OS sysplex

CICS

RACF Userid

RACF

SMF

Audit Record
✓ **RACF user-ID**

Audit records do not show who the user is; accountability difficult to determine who the originating user is

• Distributed identity information is lost when request sent to RACF

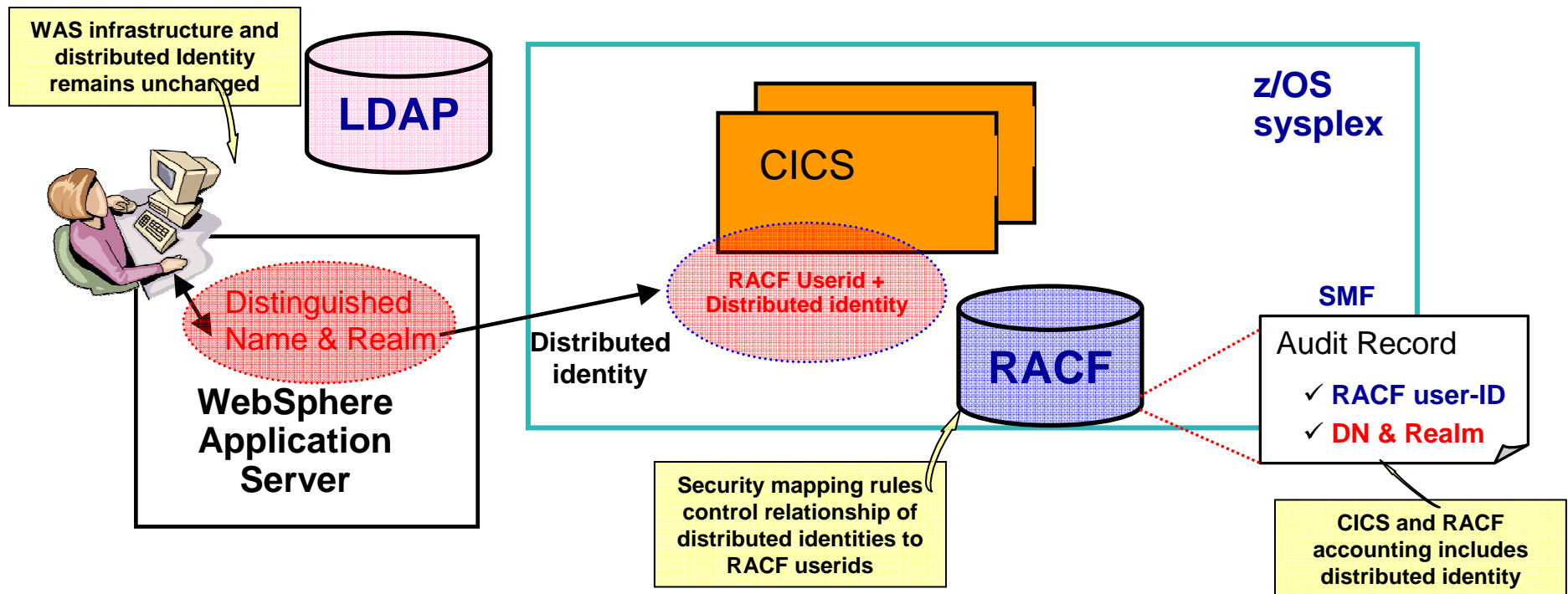Static RACF userid must be associated with the application

Identities often 'asserted' so that passwords do not need to be managed

Custom security code often created to map distributed identity to RACF userid

Notes

An example of the current CICS distributed security environment.

Implementation of Identity Context Propagation

WAS infrastructure and distributed Identity remains unchanged

LDAP

z/OS sysplex

CICS

RACF Userid + Distributed identity

Distinguished Name & Realm

Distributed identity

WebSphere Application Server

RACF

SMF

Audit Record

✓ **RACF user-ID**
✓ **DN & Realm**

Security mapping rules control relationship of distributed identities to RACF userids

CICS and RACF accounting includes distributed identity

- Propagates end user's identity from WAS through to CICS

- Scenarios for identity propagation

  ▪ Inbound to CICS from WebSphere Application Server through CICS TG

  ▪ Inbound to CICS as a WS-Security header element in a Web services request

  ▪ Propagating out across IPIC and MRO connections between CICS systems

Notes

Identity propagation provides a mechanism to allow a user identity from an external security realm be preserved, regardless of where the identity information was created, strengthening the audit trail across distributed environments.

In an external computing environment, for example, WebSphere Application Server, the identity of a user is authenticated using user information that applies to that environment. Applications like WebSphere Application Server often use a separate, shared external security manager user ID when communicating with a CICS system. The original identity of the user is not passed to CICS and therefore cannot be passed onto the external security manager, making it difficult to determine the initial user identity and impacting the audit trail of the request.

A distributed identity originates outside of CICS only; CICS is never the source of a distributed identity, but is capable of propagating the distributed identity onwards.

When a distributed identity enters the sysplex, it is automatically propagated over MRO and IPIC connections in the same way as a RACF userid.

CICS security handles the distributed identity as additional information relating to the user ID, and a distributed identity cannot exist without a user ID.

Outside the sysplex, the distributed identity can be propagated on, depending on support. Receipt of a distributed identity by another party depends on whether the party is participating in identity propagation.

Benefits of Identity Propagation

- **Cross product end to end security:**

  - Propagation of security credentials across the enterprise

    - Inbound Web Services requests
    - Inbound CICS Transaction Gateway calls
    - Outbound from CICS <u>with original identity</u>

  - Authorization via identity assertion (link security)

  - Mapping controlled by z/OS security administrator

    - Bypass need of non z/OS platforms to know about z/OS userids

- **End-to-end auditing of transactions requests:**

  - Associating additional information with the RACF identity

    - Distinguished name
    - How user entered the enterprise complex
    - Optional customer selected data

Notes

Identity Propagation provides:

The ability for the user's distributed identity to appear in z/OS run-time security context (in the ACEE) along with the RACF User ID. It also provides for the user's distributed identity to be associated with the "back-end" process providing enhanced auditing.

The ability to do identity 'assertions' using a user's distributed identity bypassing the need for the front end system to map to a specific z/OS User ID.

The control of identity mapping is within the administrative scope of the z/OS Security Administrator rather than the Application Administrator.

Additional Enhancements

- **New SPI for managing CSD definitions**

  - EXEC CICS CSD commands

- **CICS monitoring improvements**

  - Transaction resource class data for DPL requests

  - Additional data

    - Web Services (Operation name, URIMAP, SOAP request/response length)
    - Parsing data (number bytes parsed, CPU time for parsing)

  - CMF Record compression is now the default

- **Configurable VTAM Persistent Sessions**

- **Wild branch diagnostic improvements**

  - Supports Breaking Event Address Register (BEAR) in the TACB & DFHPEP COMMAREA

Notes

You can now manage CICS resource definitions in the CSD using a set of new CICS system programming interface (SPI) commands. These commands give you a command-level alternative to the CEDA transaction and the DFHCSDUP batch utility for updating the CSD.

New performance data metrics improves the performance reporting and analysis of Web and Web service applications. Use the new transaction resource class monitoring data for distributed program link requests to improve the work load management of DPL applications. Also, to help reduce the volume of monitoring output produced, the default now is to compress monitor records. New monitoring data provides information on how the XML System Services parser is performing when parsing XML documents.

You can increase the number of CICS regions in an LPAR by removing VTAM persistent sessions support where it is not required, such as in a CICS region that is used only for development or testing. You can now disable VTAM persistent sessions support for a CICS region using the new value NOPS for the PSTYPE system initialization parameter.

CICS TS 4.1 supports the Breaking Event Address Register (BEAR). This register captures the address of the instruction that generated the last successful branch before the error occurred in the program. This information helps when investigating wild branch problems in programs. The Breaking Event Address Register is held in the system diagnostic work area.

Additional Enhancements…

- **JVM Server runtime environment**

  - Provides a mechanism for CICS to use the same JVM for multiple tasks concurrently

  - New T8 TCB pool

  - Architectural enhancement that is provided for CICS to perform system processing

  - The JVM server is not available for use by Java applications

 - Architected to allow multiple JVMServers in a single CICS
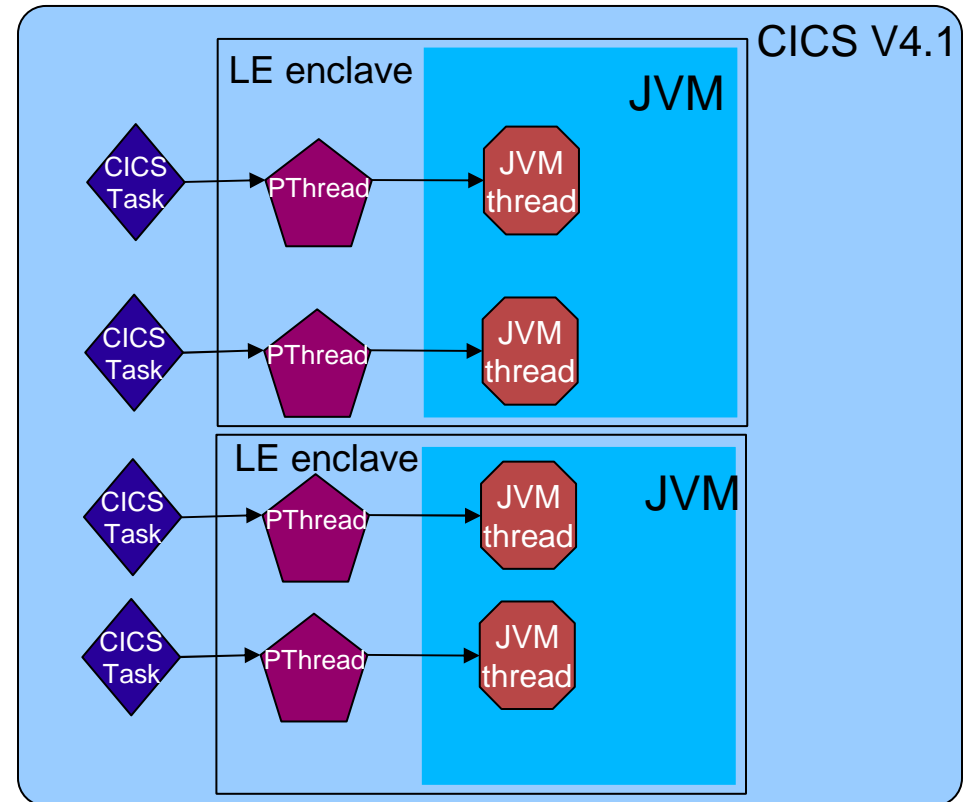
Notes

CICS includes a new runtime environment to perform Java processing. The runtime environment is called a JVM server and provides a mechanism for CICS to use the same JVM for multiple tasks concurrently. The JVM server requires a Language Environment enclave and a pool of T8 TCBs. Each task uses a T8 TCB to run as a pthread in the JVM. The JVM server is represented in CICS by a JVMSERVER resource . The JVMSERVER resource defines the runtime options for the JVM, including the location of the JVM profile and the maximum number of threads that the JVM supports.

The JVM server is not available for use by Java applications. The JVM server is an architectural enhancement that is provided for CICS to perform system processing only.

New CICS Tool: CICS Deployment Assistant

- **New Deployment perspective integrates with CICS Explorer**

- **Discovery and Modelling of CICS systems and applications**

- **Advanced Topology Visualization**

- **Automated creation, deployment, customization, and control of CICS regions**

- **Edit files, submit JCL, view output**

- **Best practice cheat sheets assist CICS version upgrades and CICSPlex SM set-up**
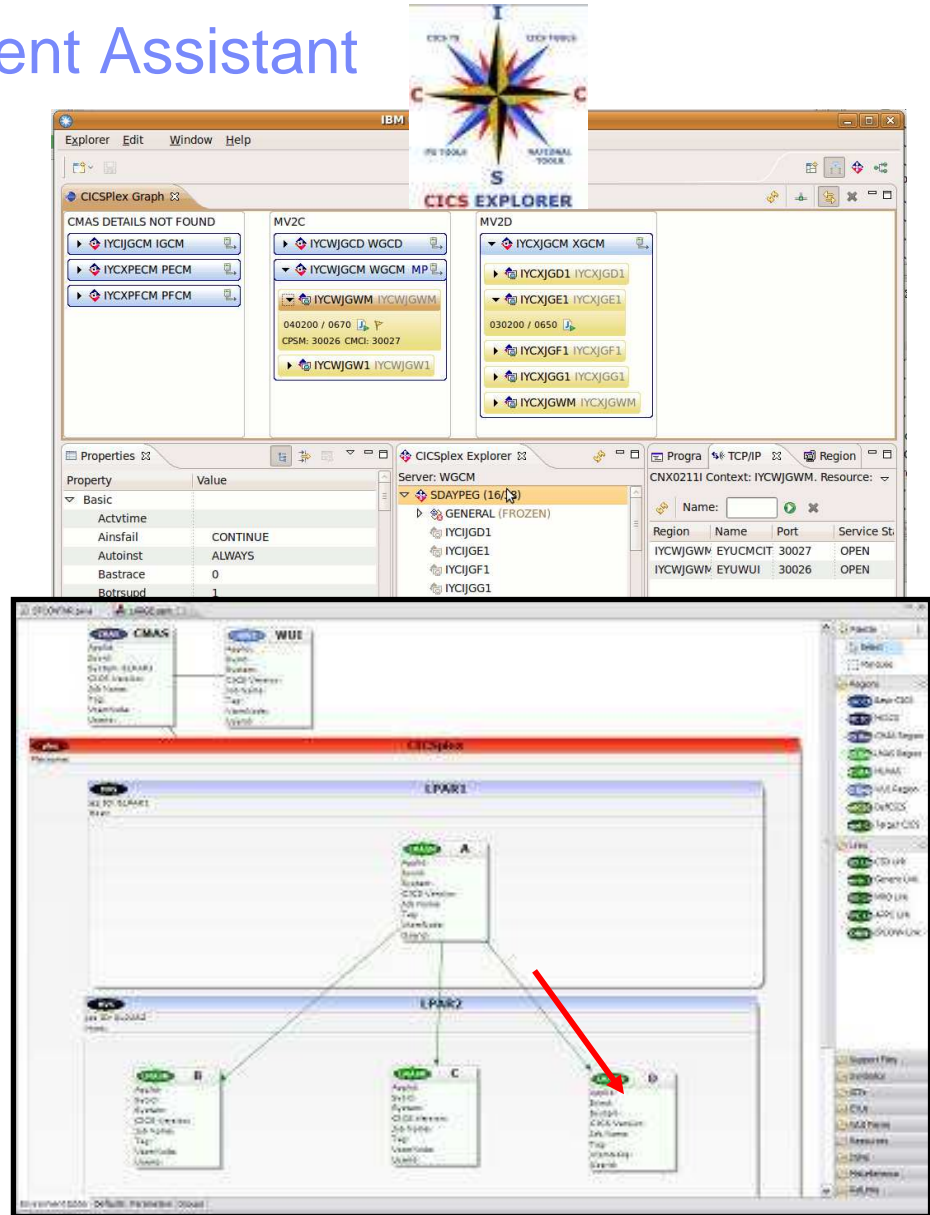
Notes

IBM CICS® Deployment Assistant for z/OS® (CICS DA) helps you use the latest IBM CICS technology faster and easier than ever before. With CICS DA you can discover, model, visualize, and deploy new and existing CICS regions, using automation and policy-driven advice built on your own and IBM's knowledge and experience.

CICS DA provides key benefits to the enterprise. By simplifying and automating complex tasks, it improves productivity and reduces costs, risks, and dependency on specialized CICS skills. By promoting standardization and reducing complexity it helps to accelerate the deployment of new CICS capabilities and lines-of-business applications, while minimizing errors and the associated risk of failure.

Key deployment tasks supported by CICS DA

include:

• Discover CICS regions and dependencies

• Check CICS topology and initialization options are as expected

• Work with JCL, spool files, and datasets

• Add a CICS to an existing CICSplex

• Clone an existing application-owning or terminal-owning region

• Clone an existing standalone test region

• Start CICS region

Eclipse cheat sheets, embedding best practices, are also included to assist with CICS version to version upgrades and a CICSPlex SM implementation.

Announced July 22nd 2010 with planned generally availability September 24th 2010, CICS DA V1.1 will support CICS Transaction Server for z/OS (CICS TS) V3.1, V3.2, and V4.1.

New CICS Support Pacs

- **CP13: CICS VSAM Record Level Sharing Performance Study**

- **CS1Q: CICS & MQ Explorer Integration**

- **CN0A: Application Performance Analyzer Explorer Plug-in**

- **CN0D: IBM Debug Tool Explorer Plug-in**

- **CN0F: IBM Fault Analyzer Explorer Plug-in**

Notes

This chart lists the new CICS Supportpacs.

Performance Improvements

- **CICS TS V4.1 performance improvements compared against CICS TS V3.2:**

  - **Optimized use of MVS timer services:**
    - When compared with CICS TS V3.2, between **1% to 5% CPU reduction** was observed due to optimizations in the use of MVS timer services on System z9 or z10 hardware

  - **Faster XML processing:**
    - CICS TS V4.1 showed a **reduction in the CPU time** used to parse XML messages, due to CICS now utilizing the z/OS XML parser.

  - **Improved capacity and faster intersystems:**
    - When compared to workloads currently using LU6.2 and VTAM for transaction routing or Dynamic Program Link (DPL), CICS TS V4.1 showed a **reduced response time and overall CPU usage** by migrating to TCP/IP and the IPIC functionality.

  - **System z9 and z10 hardware:**
    - Users could see a **reduction in CPU per transaction** for those applications running on IBM System z9 or z10, due to exploitation of this new IBM hardware.

  - **Throughput improvements with CICSPlex SM Work Load Management:**
    - Users who exploit the CICSPlex SM Work Load Management component should see **throughput improvements**, particularly for distributed workload requests when exploiting the new sysplex optimised workload management facilities.

  - **Improved efficiency and resilience management:**
    - Changes have been made to CICSPlex SM Topology that allow it to track more resource types, and provide customizable limits on the number of resources to be returned. Users should therefore benefit from **more efficient and resilient management**, particularly when using CICSPlex SM APIs, CICS Management Client Interface (new in this release), Web User Interface (WUI), and the CICS Explorer.

Notes

The following performance improvements have been observed when executing workloads on CICS TS V4.1 region in a controlled test environment:

Optimized use of MVS timer services: When compared with CICS TS V3.2, a traditional workload consisting primarily of COBOL programs accessing VSAM data running in CICS TS V4.1 provided better performance characteristics, including between 1% to 5% CPU reduction, due to optimizations in the use of MVS timer services. The saving varied according to the frequency the MVS timer service IEATTUSD was used by CICS when executing the workload on IBM System z9 and z10.

Faster XML processing: When compared with CICS TS V3.2, a workload consisting of Web services connectivity running in CICS TS V4.1 showed a reduction in the CPU time used to parse XML messages due to CICS utilizing the z/OS XML System Services parser. In addition, some aspects of XML processing can now be off-loaded from general processors to IBM System z Application Assist Processors (zAAP) possibly giving performance and cost benefits.

Improved capacity and faster intersystems connectivity: When compared to workloads currently using LU6.2 and VTAM for transaction routing or Dynamic Program Link (DPL), CICS TS V4.1 showed a reduced response time and overall CPU usage by migrating to TCP/IP and the IPIC functionality. In addition, the migration to TCP/IP can provide further performance improvements due to capabilities in the System z networking infrastructure including gigabit network exploitation provided by OSA-Express and the zIIP-Assisted HiperSockets™ available in IBM z/OS V1.10. In addition CICS TS V4.1 users also migrating to IBM z/OS V1.11 can benefit from improved performance for wide area networks through dynamic tuning of the TCP window's size. System z9 and z10 hardware: Users that upgrade to CICS TS V4.1 could see a reduction in CPU per transaction for those applications running on IBM System z9 or z10 due to the exploitation of new hardware.

Throughput improvements with CICSPlex SM Work Load Management: Users who exploit the CICSPlex SM Work Load Management component should see throughput improvements, particularly for distributed workload requests when exploiting the new sysplex optimised workload management facilities.

Improved efficiency and resilience management: Users of the CICSPlex SM APIs, CICS Management Client Interface (new in this release), Web User Interface (WUI) and CICS Explorer should see benefits from more efficient and resilient management, with the introduction of CICSPlex SM Topology changes that now track more resource types, and provide customizable limits on the number of resources to be returned to the requestor. This reduces cycles and dataspace consumption, including backing storage.

Summary

- **CICS Transaction Server V4.1 allows you to:**

  Compete for new opportunities by gaining insight into business processes and responding by modifying key business applications quickly and with confidence

  > **Business Flexibility and Innovation**

  Comply with corporate, industry and government policies to manage business risk of critical business applications

  > **Governance and compliance**

  Control costs by simplifying IT infrastructure and improving development and operations productivity through easier-to-use interfaces and functions

  > **IT Simplification**

Notes

CICS Transaction Server for z/OS V4.1 is a modern, dependable, and cost effective application platform.

This version brings new levels of ease-of-use and flexibility to meet the needs of the next generation of business users. It will assist its users in their activities directed at:

Competing for new opportunities by gaining insight into business processes and responding by modifying key business applications quickly and with confidence. Business flexibility: Making it easier to create, extend, and reuse applications quickly, to meet changing business needs

Complying with corporate, industry and government policies to manage business risk of critical business applications. Governance and compliance: Ensuring, and demonstrating, effective management control over business applications and IT facilities

Controlling costs by simplifying IT infrastructure and improving development and operations productivity through easier-to-use interfaces and functions. IT Simplification: Helping IT staff to perform their tasks more effectively, while assuming a mix of skill levels

New technologies can enable businesses to respond quickly and effectively, allowing them to capture emerging opportunities as markets recover. CICS TS V4.1 will deliver important new function to help businesses meet the demands of increased regulatory scrutiny, address changing demographics, and position themselves to take advantage of new growth opportunities