

# CICS Transaction Server V3.2: Benefits of Implementing Web Services

Smart  
SOA

**Steve Ware**

Systems Coordinator, UF

Session 1311A: CICS and Enterprise Services Track at Impact 2008

Tuesday, April 8, 2008, 3:15pm, MGM Grand Las Vegas, NV, Room 305

<http://nersp.cns.ufl.edu/~sftware/impact2008/1311.pdf>



# Abstract

This session presents a case study demonstrating how a customer has deployed the IBM CICS® Transaction Server for z/OS® Version 3.2 as a core feature of their service oriented architecture (SOA) environment. Session attendees will discover how this customer streamlined their business process, reused their existing resources, and integrated with new and existing business processes.



# Disclaimer

Standard disclaimers apply. Any opinions expressed are the opinion of the author only.

Any mentioned brand names, trademarks, registered trademarks, service marks, etc., are the exclusive property of their owners.

No warranties are either expressed or implied, your mileage may vary, etc.





# Agenda/Topics

- Introduction
- What is SOA? What is Smart SOA?
- Why CICS?
- Why the mainframe?
- UF CICS Web Services Case Study
- Other "Case Studies"
- Summary and Q&A
- Appendix and Additional Information
- Abbreviations (and a bit of Glossary)



# Introduction - UF CNS



- UF **CNS**, University of Florida Computing & Networking Services (formerly known as **NERDC**), is the primary data center at the Gainesville, FL campus.
- Currently utilizing an IBM z9 BC 2096-Q02 with 16GB and 1 zAAP, running **z/OS 1.7**, **CICS TS 3.2**, **DB2 7.1**, **RACF**, **JES2**, etc.
- 3 LPARs - 1 internal "sysprog sandbox", 1 test "alternate", and 1 production or "primary".
- We have 12 CICS regions configured, and run ~1M prod. transactions/weekday, and ~2M on peak load days.
- 2 internal/test sandbox, 2 development/test, 4 test/QA, and 4 production CICS regions currently configured.





# Introduction - UF

- Founded in **1853**, became the **University of Florida** in **1905**.  
(East Florida Seminary -> Florida Agricultural College -> University of Florida)
- **UF** is a member of the **AAU**, the Association of American Universities.  Association of American Universities  
1200 New York Avenue, NW, Suite 550 Washington, DC 20005 202.408.7500  
*An association of 62 leading research universities in the United States and Canada*
- UF is one of the fourth largest universities in the U.S., public or private.
- **~50K enrolled** and **~250K alumni**.



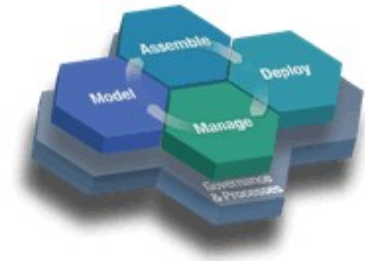


# Introduction - UF CICS

- We're considered a "**Classic**" CICS site.  
("**Legacy = It Works!**")
- Web access to CICS is via the **CICS Socket Interface**, in use at our site since ~1997.
- ~**60%** of local CICS tasks utilize **sockets**.
- All locally developed CICS applications are **Assembler** and/or **COBOL**. We have ~8K CICS application load modules, and ~32 have CICS sockets API (for file/data transfer, email, web enablement, etc.).
- Several internal CICS applications written in **C/C++** and **REXX**. **Java** in CICS has only been IVP tested, but with our System z9 zAAP, we're looking at exploiting Java in CICS in the future.



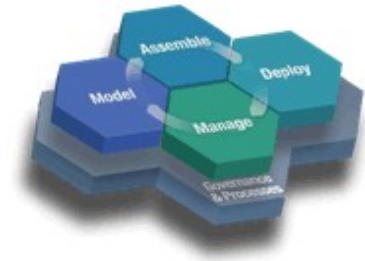
# Introduction - UF CICS WS



- Are we now considered a **Nouveau** CICS site?
- CICS Web Services in production since Sept. 2006:  
<http://docweb.cns.ufl.edu/update/u0610cics/u0610cics.html>
- CNS & UF Registrar Implement "MyStudentBody.com" Requirement using CICS Web Services.
- "On Friday, September 22nd, 2006, UF CNS CICS systems staff and UF Office of the University Registrar application staff implemented a new, secure (https) CICS Web service, with CICS acting as the service requester, for the MyStudentBody.com UF health requirement. The new capabilities introduced to support this initiative pave the way for implementation of encrypted Web services accessing real-time student data, making applications more accurate, serving the UF community better."



# Introduction - UF CICS WS



- As of March, 2008, we now have eight CICS Web Services (WS) in production.
- Two of these WS are providers (where CICS is acting as the server), and the other six are requesters (where CICS is acting as the client).
- We've been able to utilize the IBM supplied CICS Web Services Assistant for all of these WS. The CICS WS Assistant is actually two Java programs, which allow you to either take a language structure, such as a COBOL copybook, and create WSDL (DFHLS2WS), or start with WSDL, to create a language structure (DFHWS2LS). This allows the creation of a WS in CICS in a matter of minutes!
- Special thanks to IBM for providing these great CICS WS tools.

# What is SOA? xml.com says...

- SOA is Service Oriented Architecture. Let's try to define SOA with a couple of examples.
- "SOA is an architectural style whose goal is to achieve loose coupling among interacting software agents. A service is a unit of work done by a service provider to achieve desired end results for a service consumer. Both provider and consumer are roles played by software agents on behalf of their owners." As quoted on O'Reilly xml.com:

<http://webservices.xml.com/pub/a/ws/2003/09/30/soa.html>





# What is SOA? ibm.com says...

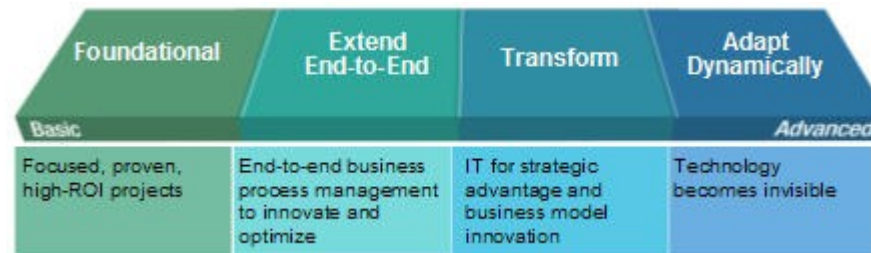
- "Service Oriented Architecture (SOA) is a business-centric IT architectural approach that supports integrating your business as linked, repeatable business tasks, or services. SOA helps users build composite applications, which are applications that draw upon functionality from multiple sources within and beyond the enterprise to support horizontal business processes" As quoted on ibm.com:

<http://www-306.ibm.com/software/solutions/soa/>



# What is Smart SOA?

- Agile. Nimble. Quickly and effectively respond to requirements, and/or changes and/or opportunities and/or threats.
- Use tested and proven techniques.
- Eliminate complexity - demand simplicity and robustness in IT projects.
- The IBM Smart SOA continuum model:



# Why CICS? Premier.

Stick With CICS!

<http://www.share.org/cics>



- IBM's CICS is the planet's premier OLTP (On-Line Transaction Processing) system. CICS Rocks! Stick With CICS! Anyway...
- CICS is sometimes referred to as an "Application Server" and/or "Middleware". IBM has positioned CICS Transaction Server in the WebSphere "application and transaction infrastructure".
- Enterprise caliber, high performance, fast response time, exceptional throughput and reliability, renowned maintainability and serviceability, resilient, large installed base, capabilities continually enhanced, large variety of applications and tools, superior technical support from IBM, expansive API, feature-rich application processing environment - and so much more!



# Why CICS? Current and modern.

- From the IBM United States Announcement 207-051, dated March 27, 2007, for CICS TS V3.2:

"CICS® is a modern transactional application server designed to execute demanding mixed-language application workloads, and to enable those applications easily to be integrated with enterprise solutions. CICS Transaction Server for z/OS® (CICS TS) V3.2 provides open standards-based connectivity, enabling CICS applications to be integrated within a service-oriented architecture (SOA), while preserving the long-established CICS qualities of security, reliability, data integrity, and optimal application responsiveness."



# Why CICS? Current WS support.

CICS TS V3.2 WS enhancements include:

- Support for WSDL 2.0
- Support for WS-Trust in WS-Security
- Support for MTOM (Message Transmission Optimization Mechanism) and XOP (XML-binary Optimized Packaging)
- Support for WS-I (Web Service Interoperability) Basic Profile 1.1 and Simple SOAP Binding Profile 1.0
- Older SOAP for CICS feature still supported (same for CICS TS V3.1) - integrated CICS WS support obviously recommended
- Data mapping enhancements in the CICS Web Services Assistant



# Why CICS? Well positioned.

- Over a decade ago, IBM said that CICS will continue to be well positioned to be a key component of their internet and open standards strategies. Some examples include:
  1. CICS Internet Gateway
  2. CICS Socket Interface
  3. CICS Web Interface
  4. CICS and EJBs, Java, and JVMs
  5. CICS Web Support
  6. SOAP for CICS
  7. CICS Web Services
  8. CICS Information Center (Eclipse based)

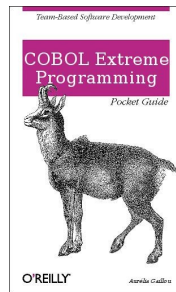


# Why CICS? Extensive support.

- Support for "modern" technologies, including TCP/IP, SSL, Web services, SOA, SOAP, Java, JVMs, EJBs, C/C++, XML, and much more.
- Support for "heritage" technologies such as VTAM and SNA networking and 3270 devices. Some things are still better via a CHUI (CHaracter User Interface) than a GUI (Graphical User Interface). Data entry and scripting are examples where a CHUI shines.
- Support for most operating systems - our focus will be z/OS.

# Why CICS? Creative programming.

- Programming language support includes Java, C/C++, Enterprise COBOL, Assembler, PL/1, and REXX. Object oriented programming support, even in Assembler, with IBM's High Level Assembler. High speed XML parsers and CICS translator integration available with some compilers.





# Why CICS? Expansive API.

- Expansive API (Application Programming Interface), SPI (System Programming Interface), and XPI (eXit Programming Interface). You can truly "make CICS dance" anyway you'd like.
- The CICS **API** helps isolate the CICS application programmer from the operating system, allowing application programmers to focus on application development and business solutions.
- The CICS **SPI** allows the CICS system programmer to enhance the local CICS environment with SET and INQUIRE system capabilities.
- The CICS **XPI** allows the CICS system programmer to enhance and expand upon the delivered CICS capabilities in the many supplied CICS exit points.



# Why CICS? Application support.

- Additional programming APIs for CPSM, Java (including JCICS), etc.
- CICS ships with integrated debugging (CEDF/CEDX) and support tools (CEBR, CECI, CEDA, CEMT, CETR, CICSplex SM, etc.), a large sample library, and sample applications with source code.
- IBM and other vendors also have very capable optional CICS application development, testing, debugging, monitoring, and other tool products.



# Why CICS? Comprehensive.

- CICS offers flexible intercommunication facilities allowing it to be supported and configured across a variety of networks. New product function in CICS TS V3.2 includes "IP interconnectivity for DPL". DPL in CICS is "Distributed Program Link".
- CICS data management is comprehensive and includes support for major database management systems such as DB2 - this also includes a high performance CICS DB2-Attach facility. Data can be in databases, OS datasets, datatables, or even within CICS itself. Other database support includes Oracle, IMS, etc.
- CICS and VSAM continue to be enhanced, such as with the VSAM RLS function of DFSMS, and DFSMStvs (Transactional VSAM Services), for CICS and batch. Note that a coupling facility is required for these optional VSAM components.



# Why CICS? SOAP for CICS.

- Many IBM CICS SupportPacs, vendor products, freely available source code, and helpful web sites and discussion lists. Speaking of SupportPacs, SOAP for CICS is a good example of how quickly IBM added this support to CICS:
  1. A free download was made available by IBM. The download included code, documentation, and samples. IBM also provided a good SOAP for CICS discussion list.
  2. An optional no-charge feature was added to CICS TS 2.2 and CICS TS 2.3, using a CALLable interface.
  3. It's fully integrated starting with CICS TS 3.1, via new and enhanced commands, such as EXEC CICS INVOKE WEBSERVICE, and Channels and Containers for enhanced inter-program data transfer - CICS COMMAREAs are still supported.



# Why CICS? Excellent design.

- CICS is designed and developed for high performance, serviceability, availability, and capability. Examples include:
  1. CICS domain architecture
  2. CICS storage protection and transaction isolation
  3. CICS and Language Environment (LE)
  4. CICS and the MVS Logger
  5. CICS-DB2 Attach Facility
  6. CICS-WebSphere MQ Adapter (new in CICS TS 3.2)
  7. CICS and Java/JVMs
  8. CICS Web Support
  9. CICS Web Services



# Why CICS? The future looks bright!

- CICS TS 3.2, CICS Transaction Server for z/OS V3.2, is the latest (and greatest) GA release from IBM, available as of June 29, 2007. The IBM 3.2 announcement letter also mentions "a future release of CICS TS".
- UF ordered and installed CICS TS 3.2 at GA and completed our migration in September 2007.
- Our experience with CICS TS 3.2 has been very near 100% scheduled availability. Exceptional reliability, with sub-second response time from the web! What's not to like?
- The latest CICS Information Centers for CICS TS 3.1 and 3.2 are (open standards) Eclipse based, with support for z/OS, Linux, AIX, and Windows. The Linux support has been thoroughly tested and utilized by the presenter.



# Why CICS? New in CICS TS 3.2.

CICS TS V3.2 delivers a wide range of new capabilities, which are grouped in three areas:

- **CICS application connectivity.** New Web services capabilities include support for security specifications and large message size optimizations. Communication is introduced between CICS regions over TCP/IP for Distributed Program Link. There are enhancements for HTTP and TCP/IP workloads.
- **CICS application reuse.** Enhancements to channels and containers include 64-bit exploitation. Integrated translator support is provided for C/C++.



# Why CICS? New in CICS TS 3.2.

CICS TS V3.2 delivers a wide range of new capabilities, which are grouped in three areas:

- **CICS service management.** Improvements include easier management of program libraries; support for enterprise-wide workload management; enhancements to CICSplex System Manager and its Web user interface; and threadsafe enhancements. Larger capacities are introduced for shared data tables and VSAM entry sequenced data sets; there are improvements to monitoring and statistics.



# Why the mainframe? Intro.

- What is a mainframe?
- Executive summary:  
An enterprise class of modern, flexible, scalable, and resilient computing servers. The current IBM System z systems are true mainframes. On Tuesday, Feb. 26, 2008, IBM announced the new z10 EC mainframe - 100 models, with up to 64 processors.
- More detailed summary:  
Enterprise computing system, with lots of processing power, continually being enhanced, with very wide I/O bandwidth, comprehensive instruction set, efficient resource sharing, very capable resource management capabilities, 64-bit architecture, supporting many simultaneous processes/programs, all efficiently managed with an enterprise caliber OS.



# Why the mainframe? System z.

- Latest mainframes from IBM include:

System z10 EC. IBM says: "The Future Runs on System z". Up to 64 processors, all 64-bit enabled (with 24-bit and 31-bit still supported). Lower total cost of ownership, lower power consumption, a growing ecosystem of skills, solutions, and services.

- Do some enterprises not disclose use of the mainframe, due to "competitive advantage"?



# Why the mainframe? Capable.

- "Mainframe renaissance" (once again?) in recent years. "The legacy lives on!" "Big Iron Staying Power."
- MTBF in decades (mainframe) vs MTBF in years (non-mainframe).
- OS choices include z/OS, zNALC (replaces z/OS.e), Linux on System z (zLinux, Linux/390), z/VM, TPF, z/TPF, VSE, and z/VSE.
- Processor (CPU) capabilities include General CPs, ICFs, IFLs, zAAPs, and announced early in 2006 - zIIPs for z9. Also, special tamper-proof cryptographic coprocessor cards.



# Why the mainframe? Availability.

- Extremely high availability - "five nines", 99.999%, with Parallel Sysplex. Our S/390, zSeries, and System z experience for over a decade has been 100% scheduled mainframe availability.
- World renowned security and cryptography, including RACF (or other z/OS ESM) and ICSF for z/OS key management facility and crypto API set.
- Very fast, sub-second response time, high transaction volumes, even from the web!
- Think modern, innovative, scalable, resilient, comprehensive, flexible, secure, community (SHARE ;-), value, integration, very capable, etc.
- A very robust and viable component of today's IT infrastructure.



# Why the mainframe? Virtualization.

- PR/SM (based upon VM), LPARs and LPAR clusters, etc.
- Virtualization, via z/VM and PR/SM, has about a 30 year head start over VMware and other virtualization technologies. Note that the speaker occasionally utilizes VMware for Linux.
- Even with a merging of technologies in servers, the capabilities that others are striving for are the capabilities already available today with the mainframe and in some cases, have been available for possibly decades.



# Why the mainframe? Throughput.

- Easily supports thousands of interactive users, and a large batch workload. You did consider both the interactive (foreground) and batch (background) workload when sizing those systems, right?
- Varying workloads are all very well managed via z/OS WLM.
- Very high data throughput and I/O are well known mainframe characteristics.
- z/OS has a very capable IP stack. SSL/TLS available with very good performance. Lots of enhancements in the latest releases of z/OS, including Application Transparent TLS. CICS Socket Interface enhancements include tracing improvements, and OTE support.
- Bottom line: z/OS and z/OS components and subsystems are continually being enhanced. The mainframe is dead. NOT!



# Why the mainframe? \$\$\$ - Cost.

- Have you seen recent discussions about "transactions per watt hour", or "transactions per joule", or "transactions per BTU"? One cost benefit of the mainframe is a typically much lower cost of power and cooling per transaction.
- A recent UF IT enterprise infrastructure cost analysis of mainframe vs non-mainframe indicates that non-mainframe costs are approximately three times more, for hardware, software, and system administrators.
- What is the benefit/value of owning the source code to your core business application(s)? Invaluable. What is the cost if you become dependent upon a core business software vendor that doesn't meet your business needs, or doesn't fix critical flaws, or drops support for your purchased application?

# Why CICS and the mainframe now?

- IBM has reported CICS and mainframe revenue growth. Both CICS and mainframe capabilities and features continue to be enhanced, seemingly faster than some customers can keep up! In my opinion, this is all good news.
- Why not invest in CICS and the mainframe? Remember, it's not an expense, it's an investment! And when choosing, be sure to "choose wisely"!
- Why not now? The future is bright for CICS and the mainframe!



# Why the mainframe? (cont.)

- Never trust a computer you can lift ;-)





# UF CICS Web Services Case Study

- We have 2 processors with 16GB main memory in our z9 BC mainframe, which has simultaneously run all of UF and UNF financial and student administration, all of the State of Florida universities LUIS (Library User Information System) and FACTS (Florida Academic Counseling and Tracking for Students).
- Compare this with the non-mainframe "solutions" with literally hundreds of processors, near terabytes of main memory, many times more disk space, power consumption, floor space, system administrators, etc. Add it up - which is more cost effective? Which consistently provides better response time? Which is more reliable? Which is more serviceable? Which is more secure? Which is easier to recover in a disaster recovery scenario?



# UF WS Case Study - Timeline.

## Project timeline:

- 2006-04-10:  
Ordered CICS TS V3.1, via Norm Leafe and Associates, our IBM zSeries/System z software business partner.
- 2006-04-14 (Fri. morning):  
CICS TS V3.1 package and tapes received from IBM.
- 2006-04-17 (Mon. afternoon):  
First UF internal CICS region running CICS TS V3.1 - migration was from CICS TS V2.3 to CICS TS V3.1.



# UF WS Case Study - Conversion.

Project timeline:

- 2006-04-17 through 2006-05-23:

All but 2 production regions converted to CICS TS V3.1 (waiting on client/customers preferred conversion schedule). A very easy and quick migration from CICS TS V2.3 to V3.1.



# UF WS Case Study - MSB Project.

## Project timeline:

- 2006-08-08:  
UF Registrar's Lead Application Developer/Architect received initial email regarding the MyStudentBody.com (MSB) project.
- 2006-08-09:  
Received "heads up" notification of the UF MSB project.
- 2006-08-11:  
Departed for SHARE in Baltimore, MD.
- 2006-08-14:  
Getting data into CICS (requester) from MSB web service (provider), using MSB WSDL and EXEC CICS INVOKE WEBSERVICE API.



# UF WS Case Study - Teamwork.

Project timeline (cont.):

- 2006-08-14 through 2006-09-19:

UF Registrar's Lead Application Developer/Architect designed and created end user interface, DB2 tables and VSAM interfaces, debugged code (local and MSB), added error handling, etc.

- 2006-08-31 through 2006-09-06:

Last 2 production regions converted to CICS TS V3.1. (Just in time for the MSB project to go production.)



# UF WS Case Study - Production.

Project timeline (cont.):

- 2006-09-21:

Production CICS installation of local WS code and definitions.  
Only six weeks from concept to production.

- 2006-09-25:

CNS News (/Update Newsletter feature) article published:  
<http://docweb.cns.ufl.edu/update/u0610cics/u0610cics.html>

- 2007-01-08:

UF accepts delivery of IBM z9 BC mid-afternoon, and plays and wins an important football game later in the evening ;-).



# UF WS Case Study - Success.

Project success components:

- CICS TS V3.1 justification, pre-planning, and installation.
- CICS WS configured and IVP tested early.
- CICS and WS preventive maint. kept fairly current.
- CICS Information Center (Eclipse based and Linux support). Good reference, task oriented info., and search capabilities.
- Excellent IBM Redbooks - some are invaluable.
- MVS sysprog assistance with CICS WS requirements such as Java 2 SDK 1.4.2, z/OS UNIX, z/OS Language Environment (LE), and z/OS support for Unicode.



# UF WS Case Study - Community.

Project success components:

- Local CICS application development tools such as EAGLE, along with good CICS application developer skills and willingness to learn.
- IBM and OEM CICS tools such as the IBM CICS Web Services Assistant (Java based), IBM Tivoli Omegamon for CICS, and CA-InterTest for CICS.
- IBM and UF SOA discussions and meetings.
- IBM UF System z Advocate discussions and meetings.
- SHARE and the CICS and mainframe communities.



# UF WS Case Study - Benefits.

## Project benefits:

- Once again showcases ("proves") capabilities and features of CICS and the mainframe - modern, fast (deployment and response time), secure, cost effective, code reusability/extendability, etc.  
  
(Fast: Are you sure it's working? I can't believe how fast the response time is! WS connection is Gainesville, FL, to Boston, MA, for this MSB WS.)
- First major production UF SOA/Web Services implementation, utilizing "best of breed" - CICS and the mainframe!
- Helped justify mainframe and disk subsystem upgrade from IBM z800 to IBM z9 BC and Shark to DS8100?



# UF WS Case Study - Opportunities.

Project "opportunities" ("gotchas?"):

- Typical "moving target" application specifications. UF Registrar's Office originally told UF students could only take one specific MSB online health test. Turns out, there are five tests allowed to satisfy the UF health requirement.
- Codepage conversion "opportunities" (easily resolved).
- First "end-user" production use of z/OS Unix file system.
  - Did not want to use IBM supplied `/usr/lpp/cicsts/cicsts31/`.
  - Initially used personal `/u/sfw/cics/ws/` HFS space.
  - Now using HFS `/local/cics/ws/` in production, and testing zFS.
  - Non-root mount/unmount and read/write vs readonly "opportunities".



# UF WS Case Study - Opportunities.

Project "opportunities" ("gotchas?"):

- It is sometimes difficult to find UF entities willing/able to consume CICS Web Services.
- Several new integration projects recently completed and in production, utilizing CICS Web Services on the mainframe.
- These "messaging" integration projects involve CICS Web Services, and a Microsoft BizTalk hub with WS connected to the UF Oracle PeopleSoft ERP implementation at UF.
- New and existing CICS applications are being built and transformed into even more capable and strategic services.
- The integration project managers are impressed - it just plain works, and works well!



# Other "Case Studies" - Financial.

- Why is it ok to spend \*more\* money on non-mainframe solutions, and in addition, provide \*poorer\* service? What follows is a recent personal example of such nonsense...
- A recent letter addressed to "Dear ... Participant" that I received from a large "Financial Services" company, included the text :

"To bring you these and other benefits aligned to your needs, we have been transforming virtually all aspects of our organization. Along the way, we have occasionally and inadvertently inconvenienced some participants with processing problems and long wait times to speak with our consultants on the phone. We regret these problems, and we are working day and night to resolve them and prevent their reoccurrence..."



# Other "Case Studies" - Airline.

- Here's text from a recent email I received from a major US airline:

"Most airlines were built on "legacy" mainframe systems from the 60's and 70's. These systems are deeply embedded in everything from reservations, to flight operations, to airport operations, to accounting. They are very reliable, but very inflexible. As our business changes, it's as though we're fighting with one hand tied behind our back."

- They don't want "very reliable"? And they want me to fly in their airplanes? They're "inflexible"? Baloney! People might be inflexible, or their technology knowledge might be "inflexible" (lacking), but mainframes are *\*very\** flexible/agile/capable!

# Other "Case Studies" - Dino Myth.

- The Dinosaur Myth - 2004 update (Arcati pdf available):

<http://www-306.ibm.com/software/info/zseries/>

"Simple cost of ownership comparisons between the mainframe and distributed platforms are often misleading - dangerously so. Many distributed costs are surprisingly well hidden within the enterprise; but with the mainframe data center, what you pay is what you get. And what you get is unparalleled scalability, very high utilization levels, and mature centralized management."





# Summary

- Computing pioneer Seymour Cray once said, "What would you rather have to plow a field - two strong oxen or 1,024 chickens?"
- Billions and billions of transactions processed daily - CICS is truly a software "star" ;-).
- UF continues to **exploit** the many inherent **advantages** of **CICS**, **z/OS**, and the **mainframe**.
- The **CICS Web Services** enhancements starting in CICS TS V3.1, and enhanced in CICS TS V3.2 have been invaluable at UF - thanks, IBM.
- IBM says: "**CICS** Transaction Server V3.2 – Continuing to put the **S** in **SOA**"



# Summary (cont.)

- **Do all the math** when making computing decisions - you just might calculate that the mainframe costs are very competitive, and that the capabilities of the mainframe are much better than other platforms. **TCO** and **TCU** are both important.
- Personally, some business relationships are based upon whether or not the business utilizes a mainframe. Ask, and let the business know what your preferences are!



# Summary (cont.)

- I really like things that work, and **work well** - like **CICS** and the **mainframe** (ok, **Linux**, too ;-).
- IT work should be productive, enjoyable, and **fun**. Ok, it doesn't always work out that way! My experience is that working with CICS and the mainframe, from a Linux workstation, is productive, enjoyable, and yes, even fun (most of the time ;-).
- The **future** looks very **bright** for **CICS** and the **mainframe!**  
(Sunglasses are optional ;-)



# Summary (cont.)

- **Thanks!** Have a great time for the remainder of the conference, and have a safe trip home.
- **Questions? Comments? *Random thoughts?***



# Appendix and Additional Information

- IBM CICS (Customer Information Control System):  
*<http://www.ibm.com/cics> (which recently resolved to)  
<http://www-306.ibm.com/software/htp/cics/>*
- IBM Announcement Letters for CICS Products:  
*<http://www-1.ibm.com/support/docview.wss?uid=swg21227090>*
- IBM CICS Transaction Server - Features and benefits:  
*<http://www-306.ibm.com/software/htp/cics/tserver/features.html>*
- IBM CICS SupportPacs:  
*<http://www-306.ibm.com/software/htp/cics/support/supportpacs/info.html>*





# Appendix (cont.)

- IBM Academic Initiative:  
*<http://www.ibm.com/university/>* (which recently resolved to)  
*<http://www-304.ibm.com/jct09002c/university/scholars/>*
- IBM Redbooks:  
*<http://www.redbooks.ibm.com/>*

# Appendix (cont.)

- SHARE ("It's not an acronym, it's what we do.")  
<http://www.share.org/>
- SHARE CICS Project:  
<http://www.share.org/cics>





# Appendix (cont.)

- The **University of Florida (UF)**:  
*<http://www.ufl.edu/>*
- **UF Computing & Networking Services (CNS)**:  
*<http://www.cns.ufl.edu/>*
- **CICS at UF**:  
*<http://cics.ufl.edu/>*
- **EAGLE at UF**:  
*<http://eagle.ufl.edu/>*



# Abbreviations

- ADCD: Application Developer Compact Disk (CD)
- API: Application Programming Interface
- CICS: Customer Information Control System
- CIS: Computer and Information Sciences
- CISE: Computer and Information Science and Engineering
- CMF: CICS Measurement Facility (via SMF)
- CNS: Computing & Networking Services (formerly NERDC)
- COD: Computing On Demand
- CP: Central Processor - see CPU
- CPSM: CICSplex Systems Manager
- CPU: Central Processing Unit



# Abbreviations (cont.)

- DFSMS: Data Facility Storage Management Subsystem
- DP: Data Processing
- DPL: Distributed Program Link
- EAGLE: UF Enhanced Application Generation Language for the Enterprise
- EGL: IBM Enterprise Generation Language
- ESM: External Security Manager
- FACTS: Florida Academic Counseling and Tracking for Students
- GA: General Availability
- Heritage: See Legacy



# Abbreviations (cont.)

- IBM: International Business Machines, Inc.
- I/O: Input/Output
- ICF: Integrated Coupling Facility
- ICSF: Integrated Cryptographic Service Facility
- IFL: Integrated Facility for Linux
- IP: Internet Protocol
- IT: Information Technology
- LE: Language Environment
- Legacy: It Works!
- LPAR: Logical Partition



# Abbreviations (cont.)

- MSB: MyStudentBody.com
- MTBF: Mean Time Between Failures
- MVS: Multiple Virtual Storage
- NERDC: Northeast Regional Data Center (now CNS)
- OS: Operating System
- OTE: Open Transaction Environment
- PDF: Portable Document Format
- PR/SM: IBM Processor Resource/Systems Manager
- RMF: Resource Monitoring Facility
- RLS: Record Level Sharing



# Abbreviations (cont.)

- SCRT: Sub-Capacity Reporting Tool
- SMF: System Monitoring Facility
- SNA: Systems Network Architecture
- SOA: Service Oriented Architecture
- SOAP: Simple Object Access Protocol (a component of web services)



# Abbreviations (cont.)

- TCO: Total Cost of Ownership
- TCU: Total Cost per User
- TS: Transaction Server
- UF: University of Florida
- UNF: University of North Florida
- VSAM: Virtual Storage Access Method
- VTAM: Virtual Telecommunications Access Method
- VM: Virtual Machine
- WLM: Workload Manager
- WS: Web Services



# Abbreviations (cont.)

- z/OS: The "zero downtime" and "a to z" Operating System
- zSeries: The "zero downtime" and "a to z" Enterprise Servers
- zAAP: zSeries Application Assist Processor (for Java)
- zIIP: zSeries Integrated Information Processor (for DB2)
- zNALC: System z New Application License Charges

# Presentation Information

- The **Slackware Linux Project**:

*<http://www.slackware.com/>*



- **OpenOffice.org 2.3.1 "Impress"**:

*<http://www.openoffice.org/>*

(File -> Export as PDF)



- IBM (Lenovo) **ThinkPad T40 2379-D5U**:

*<http://www.lenovo.com/think/us/en/>*

