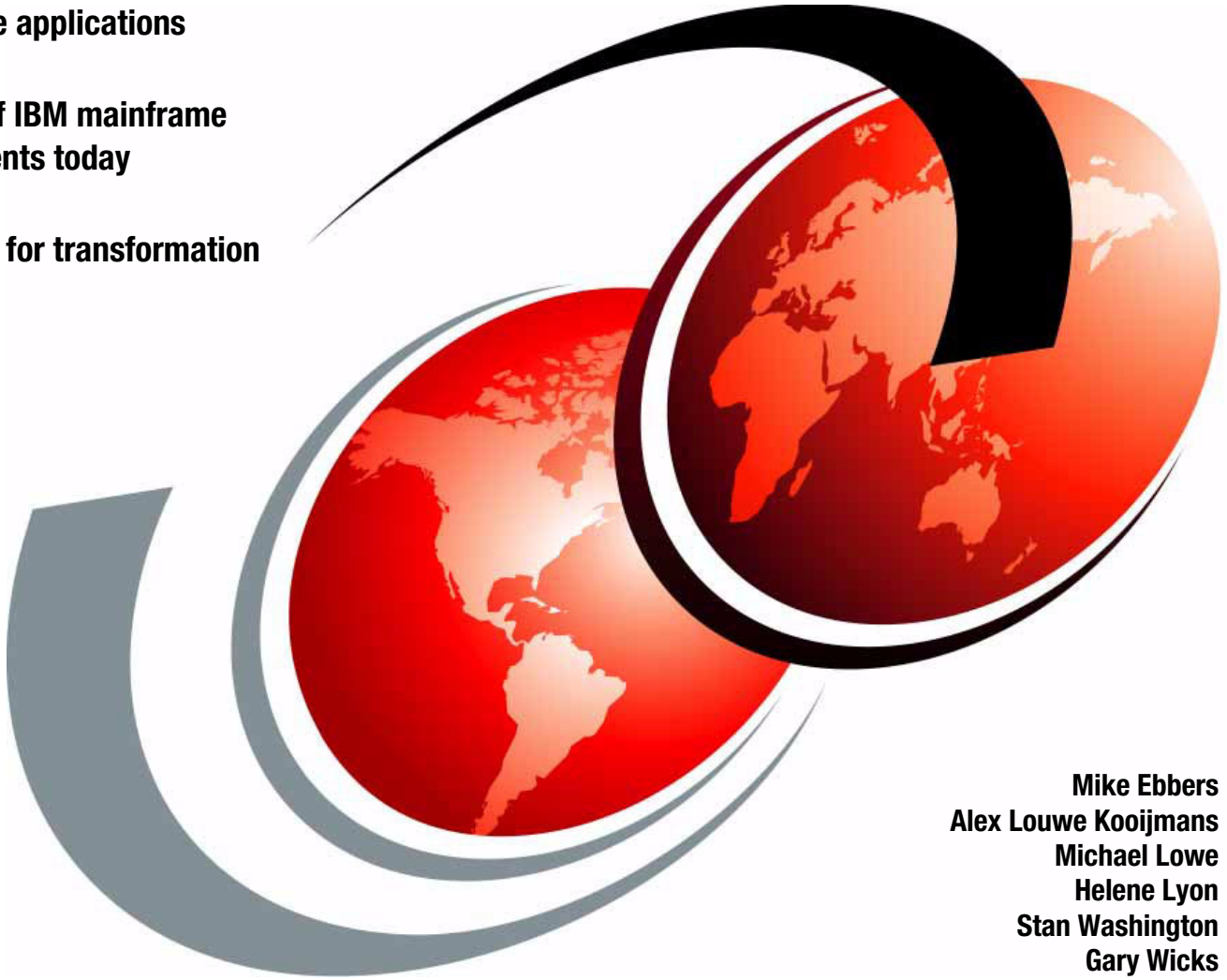
IBM

Rethink Your Mainframe Applications

Reasons and Approaches for Extension, Transformation, and Growth

- Why to transform existing IBM mainframe applications
- Benefits of IBM mainframe environments today
- Strategies for transformation



Mike Ebbers
Alex Louwe Kooijmans
Michael Lowe
Helene Lyon
Stan Washington
Gary Wicks

Redpaper

ibm.com/redbooks

International Technical Support Organization

Rethink Your Mainframe Applications: Reasons and Approaches for Extension, Transformation, and Growth

May 2013

REDP-4938-00

Note: Before using this information and the product it supports, read the information in “Notices” on page v.

First Edition (May 2013)

This edition applies to IMS Version 12 and CICS Transaction Server for z/OS Version 4.2.

Contents

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

This document, REDP-4938-00, was created or updated on May 4, 2013.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at
http://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

CICS®	IMS™	Redbooks (logo) ®
Cognos®	InfoSphere®	SPSS®
CPLEX®	MVS™	System z9®
DataPower®	Optim™	System z®
DB2®	OS/390®	Tivoli®
developerWorks®	Parallel Sysplex®	WebSphere®
Distributed Relational Database Architecture™	RACF®	z/OS®
	Rational Team Concert™	z/VM®
DRDA®	Rational®	z/VSE®
IBM®	Redbooks®	z9®
ILOG®	Redpaper™	zEnterprise®

The following terms are trademarks of other companies:

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

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

Other company, product, or service names may be trademarks or service marks of others.

Preface

Today there are new and exciting possibilities available to you for creating a robust IT landscape. Such possibilities include those that can move current IT assets into the twenty-first century, while supporting state-of-the-art new applications. With advancements in software, hardware and networks, old and new applications can be integrated into a seamless IT landscape.

Mobile devices are growing at exponential rates and will require access to data across the current and new application suites through new channels. Cloud computing is the new paradigm, featuring anything from SaaS to full server deployment. And although some environments are trying to virtualize and secure themselves, others such as IBM® zEnterprise® have been at the forefront even before cloud computing entered the scene.

This IBM Redpaper™ publication discusses how transformation and extensibility can let you keep core business logic in IBM IMS™ and IBM CICS®, and extend BPM, Business Rules and Portal in IBM WebSphere® on IBM z/OS® or Linux on IBM System z® to meet new business requirements. The audience for this paper includes mainframe architects and consultants.

The team who wrote this paper

This paper was produced by a team of specialists from around the world working at the International Technical Support Organization, Poughkeepsie Center.

Mike Ebbers is a Consulting IT Specialist and Project Leader at the International Technical Support Organization, Poughkeepsie Center. He has worked with IBM mainframe hardware and software products since 1974 in the field, in education, and in the ITSO.

Alex Louwe Kooijmans is a Senior Architect at the Financial Services Center of Excellence in the IBM Systems & Technology Group. Prior to this position he spent almost 10 years in the International Technical Support Organization leading IBM Redbooks® projects, teaching workshops and running technical events with a focus on using the IBM mainframe in new ways. Alex also worked as a Client Technical Advisor to banks in The Netherlands, and performed various job roles in application development. His current focus is on modernizing core banking systems and the role of IBM mainframe technology.

Michael Lowe is a System z Client Architect in the UK who works with clients to maximize their return on investment in mainframe technology. He has 25 years of experience in this field, and his areas of expertise include IBM DB2® for z/OS and infrastructure architecture. Michael recently achieved TOGAF 9 certification. He has been a speaker at the International DB2 User Group.

Helene Lyon is an IBM Distinguished Engineer from France who has almost 30 years of experience in the mainframe area. She is now the technical executive of the European IMS Architecture team whose focus is to develop database and transaction management solutions that fully integrate and collaborate with existing IT systems to perform a business function. Helene's main responsibility is to ensure the "right fit" positioning of the IMS database manager and transaction manager capabilities within those systems.

Stan Washington is a System z Client Architect in the United States, working with clients to maximize their return on investment in mainframe technology. He has 40 years of experience in this field. His areas of expertise include z/OS, Linux, zEnterprise, DB2 for z/OS, and infrastructure architecture. Stan has led architectural workshops, solution development designs, and deployment projects for clients in the Health and Financial industries. He also led the initial implementation of disaster recovery services for Sungard Computer Services.

Gary Wicks is a IBM Certified Consulting IT Specialist reporting to the IMS laboratory in San Jose, California, USA who works in the worldwide IMS Specialist group. He has 38 years of experience in the mainframe software and service support field, supporting IMS Clients around the world. His areas of expertise include IBM IMS Parallel Sysplex® enablement and performance. Gary has written extensively about IMS Block Level Data Sharing and SOA implementations for IMS environments.

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our papers to be as helpful as possible. Send us your comments about this paper or other IBM Redbooks publications in one of the following ways:

- Use the online **Contact us** review Redbooks form found at:

  ibm.com/redbooks

- Send your comments in an email to:

  redbooks@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- Find us on Facebook:

  http://www.facebook.com/IBMRedbooks

- Follow us on Twitter:

  http://twitter.com/ibmredbooks

- Look for us on LinkedIn:

  http://www.linkedin.com/groups?home=&gid=2130806

- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm

- Stay current on recent Redbooks publications with RSS Feeds:

  http://www.redbooks.ibm.com/rss.html

1

Business and IT trends and how they affect existing applications

Investment in the transformation of existing IBM CICS or IMS applications in most cases calls for a clear return on investment. In some cases transformation is inevitable because mandatory new functions must be added, requiring the application to receive a technology refresh or architectural makeover. Generally, the business reasons for transformation include the following areas:

- Avoidance of risk, by improving the maintainability of the application
- Providing better agility, resulting in a better time to value, by improving the application architecture and benefitting from middleware wherever possible
- Reduction of cost, by introducing higher productivity tooling for development and deployment
- Increasing functionality, by incorporating new technologies such as Web 2.0, and moving to rich programming languages such as Java

Tip: Transformation of your technology to state of the art can fully take place while keeping CICS or IMS in place as the transaction management middleware.

1.1 Trends in technology and business

There are a variety of technology and business trends occurring that can impact the way CICS and IMS applications are currently implemented. Being able to properly respond to them might require some level of transformation. One way to discover the trends is by taking a look at the 2011 IBM Global CIO Study survey results[1]. Figure 1-1 illustrates the focus areas.

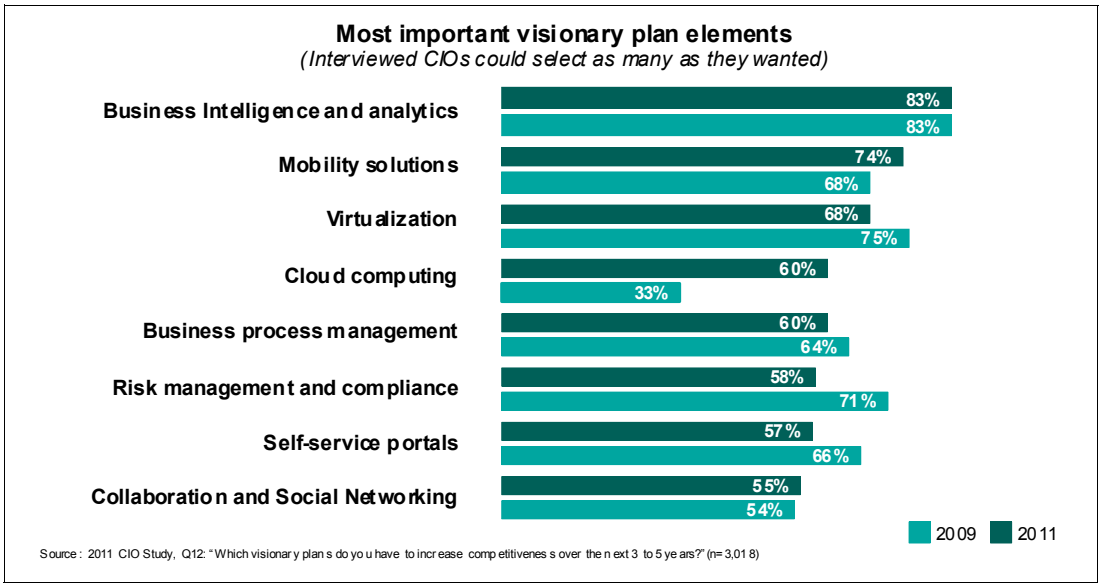


Figure 1-1 2011 IBM Global CIO Study

Although not all of the CIO focus areas shown in Figure 1-1 have a direct impact on CICS and IMS applications, those areas that can benefit from an improved time to value if the existing applications are modernized are examined here.

1.2 Business intelligence and analytics

Analytics are increasingly being performed in real time, as part of transactions. The benefit of this is that actions can be taken immediately instead of after the fact. Realtime analytics are typically added to existing business logic programs by invoking a scoring engine such as IBM SPSS®.

Here are two examples of applying realtime analytics to existing applications:

- A bank performs realtime fraud detection.

  Algorithms are executed directly from the application programs that process the payment transactions.

- A retail store determines buying behavior in realtime by executing analytics directly as part of e-commerce transactions over the Internet.

  With this approach, the store can suggest other products immediately during the client's session.

[1] The landing page for the 2011 IBM Global CIO Study can be found at http://www.ibm.com/services/c-suite/cio/study.html

1.3 Mobility solutions

The mobile platform is leading to a re-establishment of the industry ecosystem and the emergence of new business models, as indicated by the following examples:

- Branchless financial services companies are emerging, providing a new set of services, sometimes combined with other commercial propositions, purely based on the mobile platform.
- Mobile devices facilitate new and enhanced products and services, extending the “electronic wallet,” integrating diverse services, serving as a distribution channel, and providing many other functions.[2]
- Mobile network operators will diversify into financial services, media services, and citizen services, and will integrate with brick and mortar businesses.

Mobile applications require a fast and agile deployment environment because they tend to respond to business needs much faster than their more traditional predecessors. Many of these mobile applications integrate at some point with existing CICS and IMS back-end business logic and data access programs. It is important that this integration can take place in a flexible manner and can use a fast deployment model. CICS and IMS applications might need some level of modernization to ready them for mobile integration.

Web 2.0 has become a popular standard for web applications, and it is common these days to use so-called “mash-ups” to tie together information from many different sources into a single web page. As illustrated in Figure 1-2, these sources could well be CICS and IMS programs on z/OS.

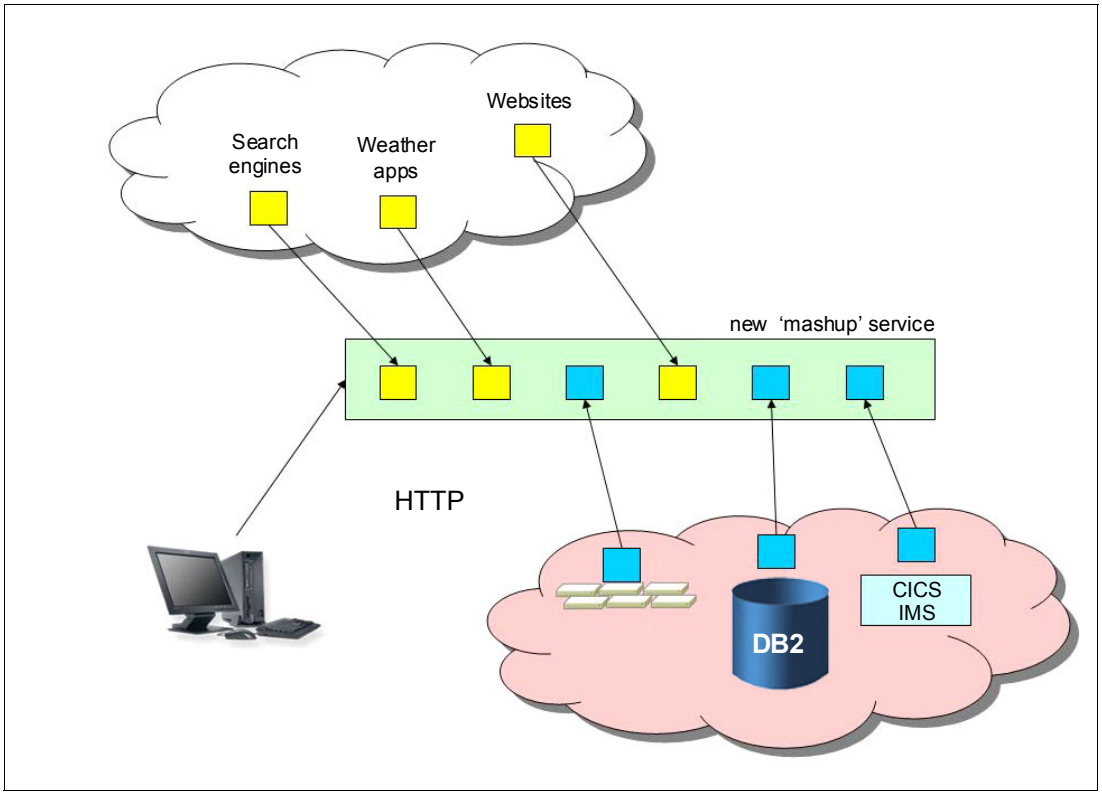


Figure 1-2 Accessing CICS and IMS programs on z/OS as part of a Web 2.0 mash-up

[2] For more detail, see:
http://www.ibm.com/developerworks/rational/library/enterprise-teams-mobile-application-projects/index.html#author1

1.4 Business process management

Business process management (BPM) is a discipline combining software capabilities and business expertise to accelerate process improvement and facilitate business innovation. The business process management lifecycle consists of design, modeling, execution, monitoring, and optimization. BPM enabled by service-oriented architecture (SOA) is a discipline enhanced by a flexible IT architecture to accelerate the creation and reuse of business services to facilitate business innovation. Bringing together the most advanced SOA-based software capabilities along with broad expertise will provide a higher value BPM solution.

BPM drives business and IT alignment around the following functional and transformational business objectives:

- Collaborate to predict and optimize process outcomes through modeling and simulation.
- Rapidly customize processes with business users using policies instead of code.
- Sense and respond to business events in real time for automated response or human decision support.
- Rapidly deploy new solutions from reusable building blocks that can be changed dynamically.

BPM with SOA provides process flexibility by improving the way you design, manage, and optimize the business processes and reuse existing assets. BPM enabled by SOA is based on creating agile and dynamic processes today that serve as the foundation for greater innovation in the future. It provides the key alignment between business architecture and IT infrastructure and keeps this alignment flexible and continuous. BPM offers a service-oriented approach to application development, based on orchestration of services. You can take existing software assets and quickly define how those assets are used in a new integration application.

A successful BPM implementation marries business processes with both new and existing services or composite services. Both types of services may be implemented as CICS or IMS services. The appropriate scope and granularity of these services has a strong influence on the agility, effectiveness and maintainability of the BPM processes. Ideally, each service or composite service involved in the business process performs a clearly defined step in the business process, for example “Confirm loan request”. See Figure 1-3 on page 5.

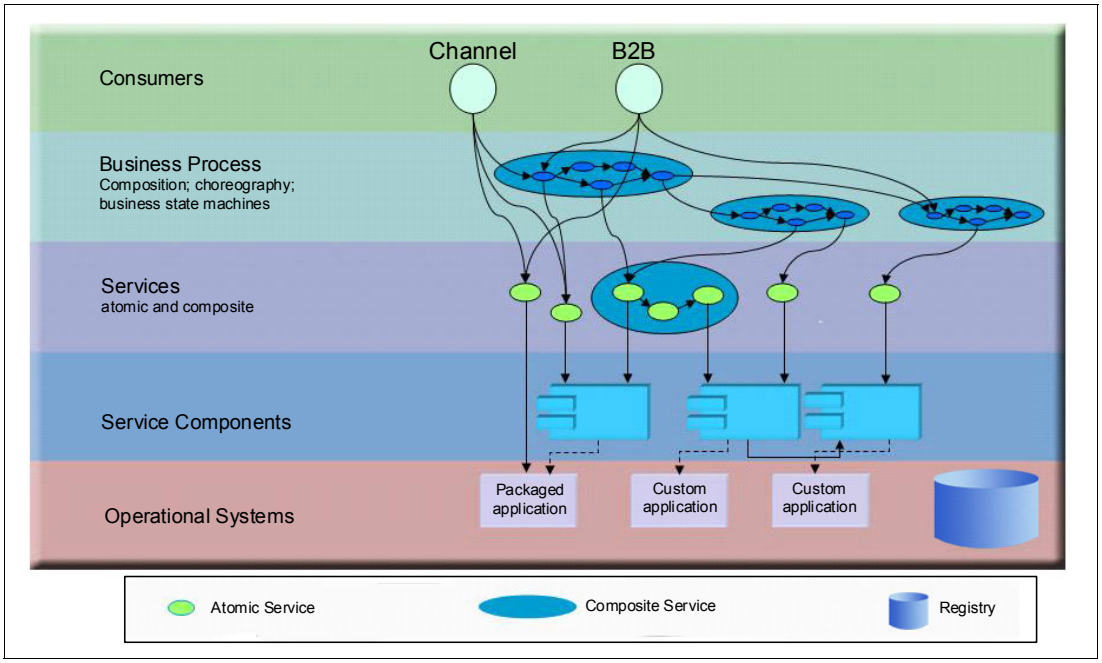


Figure 1-3 How business process management drives services

1.5 Risk management and compliance

Especially in financial services, companies are frequently forced to implement mandatory functions to comply with regulations. One of the latest sets of requirements that will affect the core business systems of most financial institutions worldwide derives from the “Foreign Accounts Tax Compliance Act” (FATCA). This is a new set of regulations from the United States government that are designed to curtail tax evasion through foreign accounts.

Compliance with such regulations usually means changes throughout existing core systems, which often are implemented in CICS or IMS. Making these changes can be easy or difficult, depending on the architecture of the application and to what extent existing applications already exploit specialized middleware, such as middleware for events processing, business rules, and reporting engines.

1.6 Driving down IT cost while gaining flexibility

Ideally, the IT budget is used for the most part to add new business functionality, while expenses for maintenance are kept to a minimum. There are a number of factors that influence the cost of application maintenance. They can be placed into three main categories:

- Application lifecycle management tooling. A well-integrated application lifecycle management factory provides productivity and quality and has a positive impact on cost. The following questions should be asked:
  - How labor intensive is the tooling? For example, are there helpful syntax checkers, code generation wizards, and impact analysis tools?
  - How integrated are the design, development, and deployment tools? Are artifacts generated in one tool automatically available in another tool?
  - Are tools in place for testing, debugging and quality management?

  - Do the tools have direct access to the z/OS test systems?
  - Can all development and deployment tasks be performed from a GUI-based environment?
- Application architecture. The application architecture determines how easily new business functionality can be added and thus influences application maintenance cost. The following questions should be asked:
  - At a minimum, have presentation, business, and data access logic been separated?
  - Is the application implemented according to a design pattern?
  - Are business logic programs well-scoped and granular?
  - Does the presentation layer allow multiple channels, such as web and mobile?
- Middleware exploitation. Middleware is designed to facilitate the work of the application developer and in many cases even take over duties. Proper exploitation of middleware reduces application development and deployment cost, for example:
  - Transaction management middleware (such as CICS, IMS TM, and WebSphere Application Server) takes care of running large volumes of transactions without resource conflicts, and manages automatic recovery.
  - Batch management middleware such as IMS batch manager and WebSphere batch container provide a runtime environment co-located with data. They provide efficient checkpoint and restart logic, freeing applications to manage that programmatically.
  - Messaging middleware (such as WebSphere MQ) takes care of standardized and guaranteed delivery of messages sent across systems and applications.
  - A business rules management system (BRMS), such as IBM Decision Manager, provides a rules engine and associated tooling that facilitate deployment of highly volatile business rules.
  - Enterprise service bus (ESB) middleware, such as WebSphere Message Broker (WMB), provides an intelligent integration layer and associated tooling, decoupling application components.
  - A process engine, such as IBM BPM Server, allows you to run orchestrated services as an end-to-end business process.

Programming languages: You might be surprised to notice that there is no discussion of programming languages here. That is because the only thing that really matters with regard to the programming language is whether the requested functionality can be built using a specific language. Assuming an ideal tooling environment is used, the productivity in developing Java, COBOL and perhaps even ASSEMBLER should not be a decision factor.

1.7 Discussions associated with reusing technologies

Taking existing mainframe technologies that were designed for application execution, and converting and modernizing them for application runtime processing on distributed environments, is a topic that is regularly discussed. However, the business case for enterprise innovation projects is difficult to quantify for the following reasons:

- IT benefits are not seen as end-user benefits.
- Potential benefits accrue over multiple years rather than every quarter in line with revenue and profit reporting.

- High delivery risks might exist in the conversion, because these applications are typically complex in design and functionality.

From these themes related to business cases, the following discussion points have been raised regarding application and system support extensibility.

1.7.1 Discussion point: flexibility

One concern that has been raised relates to the flexibility of legacy platforms to support business strategies in adapting quickly to new technologies such as service oriented architecture, cloud, and mobile control over and use of “big data” for business analytics.

Our response

As an example of the flexibility of an enterprise legacy application enablement middleware product, we highlight IMS. The availability of integration and access points now enable IMS assets to provide high value as part of a distributed, multitier architecture. Consider the following points:

- Since 30 October 2009, when IMS Version 11 became available, IMS Open Database has enabled SQL access to IMS data from any distributed platform.
- The IMS Catalog centralizes the storage of artifacts and metadata that can be utilized in application development and deployment in distributed environments.
- The IMS Connect facility within IMS allows both IMS data and applications to be accessed from distributed environments through TCP/IP protocols.

Every major IBM transaction system (IMS, CICS, WebSphere) is quite capable of delivering applications and transactions as services.

With respect to cloud computing, the mainframe has been a cloud-capable environment for decades. Cloud computing is essentially the provisioning of computing resources (computers and storage) as a service. Along with that comes the ability to dynamically scale the service to additional computers and storage in a simple and transparent way.

For more information about this topic, see *Cloud computing with Linux* by Tim Jones at `http://www.ibm.com/developerworks/linux/library/l-cloud-computing`

Regarding the mobile discussion, IBM has been demonstrating mobile enablement of CICS and IMS transactions for years. Much of this capability is predicated on the ability to enable existing applications as services (SOA, for example). Existing mobile tools such as Worklight can invoke a service anywhere, mainframe or otherwise. For data management, the IBM Cognos® mobile front-end can be used against mainframe SQL data sources.

Big data is a focus area that encompasses all data sources in the enterprise, whether it be System z or other. Transactional data (such as IMS, DB2, and ISV databases) is part of what feeds a big data repository. z/OS has become a much more desirable platform for data warehousing and business analytics with the advent of data warehouse technologies such as IBM DB2 Analytics Accelerator and the ISAS 9700 offering for z/OS and the Cognos and SPSS tools that provide the query and analysis functions. Placement of data warehouses on System z avoids much of the architectural and operational complexity in shifting data from platform to platform, and allows you to benefit from the scalability, security, and IT cost benefits that System z famously provides.

1.7.2 Discussion point: integration

A further discussion point involves integration: is there a lack of integration and connectivity capabilities from enterprise architectures to “open” platforms?

Our response

Architectural elements such as the enterprise service bus (ESB), workload management and routing, application containers, web session management, the database environment, messaging, and business intelligence tools all must be integrated so that data can be delivered where, when, and how it is needed throughout the organization.

Figure 1-4 is derived from the IBM white paper “What every enterprise architect needs to know about the evolution of IMS”[3]. The figure maps enterprise architecture components to application support facilities, clearly illustrating that a full range of runtime services and connectors is available.

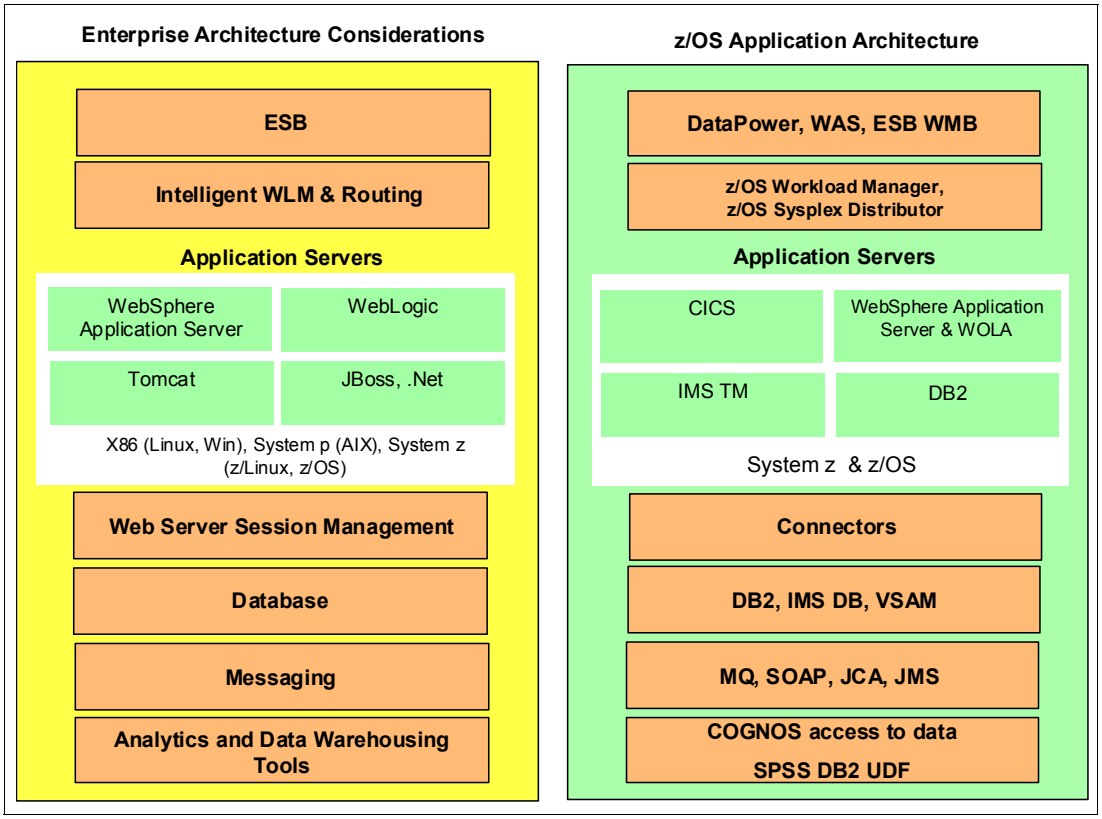


Figure 1-4 Enterprise architecture components mapped to application support facilities

1.7.3 Discussion point: skills

Another area of discussion involves concern that an aging subject matter expert (SME) base is placing the System z environment at risk for continued installation, diagnostics, performance, and general maintenance in-house skills.

Our response

In the mid-2000s, IBM began a strong push for System z skill development to fill a looming “skill gap” by providing resources to assist colleges and universities in developing and

[3] http://public.dhe.ibm.com/common/ssi/ecm/en/imw14633usen/IMW14633USEN.PDF

promoting education in mainframe technology. These programs are helping to develop a new generation of SMEs with areas of expertise in this field.

There are also effective steps that you can take to ensure that your business continues to be supported by skilled staff:

- Use the IBM job board on http://www.systemzjobs.com, which is available at no charge. It provides access to mainframe jobs and talent.
- Review the list of schools that are already teaching mainframe courses. There are hundreds of institutions currently investigating or actively teaching mainframe technologies. As your starting point, access http://www.ibm.com/university/systemz
- Identify additional schools where you would encourage IBM to assist in providing resources and assist professors. Send a note to zSkills@us.ibm.com
- Perform a skills assessment of your existing mainframe in-house skills and identify areas where you will be at future risk regarding SMEs.
- Train employees using online certificate programs or traditional lecture/lab classroom education. One online option is to use the Marist College Certificate Program, which contains systems and application programming tracks with seven certificate programs composed of 17 unique courses. You can obtain more information from http://www.idcp.org
- Assign mentors and perform job rotation for new hires, to align their individual skill set strengths with groups where there is the greatest opportunity for success.

Another consideration is to mobilize existing System z skilled employees and expand their horizons with skills in web development, mobile, big data, and other newly emerging technologies. Their value in supporting your expanding scope of user services will increase as they develop new skills, while maintaining their expertise of the mainframe environment.

The skills challenge is also being mitigated by IBM extensibility of the “user interfaces” to z/OS:

- In recent releases, z/OS has been augmented by the z/OS Management Facility (zOSMF). zOSMF provides a graphical user interface for many of the most common and complex z/OS administrative functions.
- CICS Transaction Server now has a number of GUI tools for administering and deploying CICS servers and applications.
- z/OS applications can be developed using an Eclipse-based tool such as IBM Rational® Developer for System z. Developers can build z/OS applications without even knowing they will run on mainframes.

1.7.4 Discussion point: cost

There is sometimes a misperception that the total cost of ownership (TCO) is higher with the System z platform.

Our response

With the System z platform as a springboard, you can reduce total cost of ownership in various ways. IMS Version 12, for example, allows you to handle 46,000 transactions per second on a single IMS image on the mainframe. This results in an unparalleled return on investment. Whether your business requires such huge throughput rates, the point is that the more workload presented to the System z, the more cost effectiveness is obtained.

A relatively inexpensive activity is to recompile applications using the latest language version to benefit from runtime efficiencies. This in itself is a positive transformation effort.

Downtime costs must also be considered when calculating TCO. System z is built around continuous availability and features the use of Parallel Sysplex application enablers such as CICS, IMS, and DB2. And thanks to ease of change management and failover support, the cost of unavailability to the user is dramatically reduced.

2

Attributes of an agile IT architecture

As previously mentioned, now there are new and exciting possibilities for creating robust IT landscapes. You can move current IT assets into the 21st century while supporting new state of the art applications. With advancements in software, hardware and networks, old and new applications can be integrated into a seamless IT landscape. Mobile devices are growing at exponential rates and will require access to data across the current and new application suites via new channels. Cloud is the new paradigm featuring anything from SaaS to full server deployment. While some environments are trying to virtualize and secure themselves, others like zEnterprise have been in the forefront even before Cloud was on the scene.

2.1 The ideal IT environment

IT decision makers are facing many issues in response to business requirements to improve current applications. New business drivers are pushing IT to deliver solutions faster, with higher quality, in a vastly changing technology. Each decision needs to be evaluated to understand how it will find, refine, and deliver business answers, while also ensuring data integrity and up-to-the-minute or even up-to-the-second information. These pressures can result in various misperceptions about what constitutes an ideal IT environment.

One misperception is that current applications are not agile. Another misperception is that only new distributed Java-based applications are the only option where agility and modern solutions are to be delivered. A third misperception is that a distributed approach requires less system management discipline thus making it more flexible, when in fact it needs more management to ensure quality of service to match the quality provided by zEnterprise.

In reality, there are many options that can protect and revitalize current IT investments and deliver on the business requirements for a new user-friendly model while exploiting current applications. Many IT shops are realizing that a “rip and replace” migration and re-engineering project is a more expensive solution with much higher risks than “upgrade in place.” By using current tools and functionality, applications written to CICS, IMS, DB2, and batch standards can provide a cost-effective investment. The result is modernized applications that deliver state-of-the-art functionality and integration.

Processes can be built to interface with new rules-based engines that can provide lines of business with a new way to engineer changes such as marketing campaigns, customer solicitations, insurance policies, and more. The reality is that applications and data must be integrated to ensure accuracy of business decisions, auditing, security, compliance and more. No matter where an application is deployed, it will need full management controls to deliver integrated application and data.

So, what does an ideal IT environment look like? What are the basic desires and results? What would the interaction be between IT and lines of business?

The ideal environment might include the following advantages:

- Reduced costs that allow effective ROI in less than a year.
- Management controls that are built in and able to be monitored at an IT level and a business level simultaneously.
- Servers and applications that are secure and compliant to set standards and procedures at all times.
- Standards that enhance IT’s service and business value.
- Architectures that simplify application delivery through clearly stated actionable structures.
- Ease of change and responsiveness that provide business-driven results.
- Integrated data and development that reduce support costs.
- Interruption-free service during normal operations and during routine changes to the IT environment.
- Applications that are designed not to be data-centric but to be flexible enough to adapt to changes in data formats in data stores and interfaces.

2.2 Costs in an ideal IT environment

Costs in an ideal IT environment would be driven to a level such that ROI can be realized in a minimum of time (months instead of years). This can be accomplished by reusing current IT assets and building an environment that can support, catalog, and manage reusable resources. Current investments can be harvested and repackaged through architectural constructs delivering integrated solutions, while leaving current assets as they are today or in some cases slightly modified. Staffing costs would be driven down by reducing the need to rewrite current applications or to write redundant functions. This would give current staff the opportunity to focus on new business issues, thus reducing costs. IT organizations would reduce current application maintenance, thereby freeing up valuable IT resources for developing new business requirements.

2.3 IT management processes

In an ideal IT environment, it would be easy to develop, test, quality assure, and put into production the changes that a business requires to be competitive. The ideal situation is an agile environment using an effective change management process to ensure quality code delivery and a stable production environment. For many decades, IT has been criticized about the time it takes to respond and deliver business change requests. In some cases, lines of business take situations into their own hands and circumvent IT departments altogether, building mini-data centers to provide the requirements they need today. In the distributed world many applications claim processing independence, but in fact that independence comes at a price due to redundant data, lower quality delivery, data integrity issues, and additional cost to keep distributed applications and data integrated. Poor data integrity from redundant inconsistent data can lead to disparate business decisions, thus causing an additional indirect cost.

2.4 Compliance and security

In an ideal IT environment, compliance issues are part of the solution, and not an afterthought. Security would be included in the design of the hardware and software starting from the operating system through the middleware into the application management layer. Today, security and compliance is increasing at an alarming rate with the advent of the financial fiasco on Wall Street and many other new and existing regulations. IT would gain significant advantages by providing iron-clad protection from hacking, internal threats, external fraud, data hijacking, and more.

2.5 Standards

Ideally, standards should be easy to understand, flexible, and supportable by the existing infrastructure (software and hardware). They should reduce risk while improving integration and manageability. Standards should be selectable and allow for implementing subsets as needed. Standards come from many sources and are adopted with varying scopes. IT organizations should be able to contribute to the formation of standards with a vote to accept new standards. Vendors should be major supporters and developers to ensure ongoing support for their new software releases.

2.6 Architecture

Architecture should be the methodology to empower the organization by providing guidance and directional information for the enterprise. Architecture should also be actionable as defined in *Combining Business Process Management and Enterprise Architecture for Better Business Outcomes,* SG24-7947. Actionable architecture includes the following characteristics:

- *Contextual* with a clearly defined purpose, motivation, priority, scope, and time horizon
- *Collaborative* with availability to and accessibility by all stakeholders to get participation and commitment, often even collaborative evolved
- *Connected* with traceable links across purposes and domains, including appropriate levels of change and configuration management
- *Consumable* that can be understood from various stakeholder perspectives and viewpoints as required for their understanding and buy-in

Architecture should not be perceived as loss of agility. Instead, it should be a means by which enterprise integration starts. The operating systems and software should provide an infrastructure that supports architectural frameworks seamlessly.

2.7 Integrated development environment

Integrated development environments (IDEs) provide consistent development support for any source language and deploy to any platform in patterns that maximize performance, code quality, and application integration. Effective IDEs free developers, support programmers, and platform tuning staff from extensive non-productive work. IDEs expand the resource pool that enterprises can draw on by maximizing the effectiveness of new employees.

Data integration should provide a well-documented data model that identifies the source of data, how replications are managed, and how integration processes are to refresh that data, update it, and synchronize it. With data integration methodology, data inconsistencies can be drastically reduced, thus providing business with valuable and consistent results.

2.8 Availability

Availability should include all planned and unplanned outages and downtime, thus providing a clear concise view of system and application operations. High availability is delivered through the coordination of operating systems, middleware, and hardware. Each component contributes to overall availability and should provide in-flight remedies or alternate server instances to fail over to. Hardware must have redundancy under the covers and components replaceable dynamically, thus eliminating downtime.

3

The way today's mainframe applications are implemented

In planning to extend your investment in CICS or IMS technology, it is important to consider the value and benefits you are realizing from your current infrastructure assets. With z/OS as the underlying operating system and CICS or IMS as the transaction processing middleware, there are a number of characteristics and qualities of service that many people in the organization might be taking for granted. We discuss these characteristics in this chapter.

3.1 z/OS-related benefits

z/OS is well known for its processing benefits. We list and describe a few of these benefits in the following sections.

3.1.1 Advanced workload management

The Workload Manager (WLM) component of the z/OS operating system is designed to support service level agreements (SLAs) that you can express in the form of a contract between the business and IT. You can define application or business service goals in terms of availability and performance targets that are meaningful to the business stakeholders.

For example, suppose an online retailer is responding to stock inquiries over the web and needs to invoke a stock inventory transaction in CICS to check the number items in stock. Business stakeholders can mandate that these stock checks must have the following service level policy:

- That 95% of all stock inquiries must be satisfied by CICS with an average response time less than 0.3 seconds.
- This service level is in effect 24 hours a day, seven days a week, 365 days a year.
- The application has to be available for 96% of the time with zero (0) unplanned downtime.

These service level policies can be defined to WLM using similar language, making communication and reporting between stakeholders and IT simpler. WLM will then balance machine resources across subsystems such as CICS, IMS, DB2, and WebSphere Application Server to meet the defined goals.

3.1.2 Centralized operations and integrated systems management

Because we can define service level policies in language meaningful to business users, it makes sense to report actual performance against those policies in similar terms. For example, business users will want to know that a card payment processing system is available and meeting its performance goals. They are less interested in the status of an individual server, operating system, or database instance.

Data centers today can take advantage of technologies that provide a holistic view of a business service with dashboards showing the status of key applications and functions. These views are available to business and IT users alike. In response to amber warning or red danger signals, IT staff can drill down to find the root cause of the problem and take prompt corrective action.

Figure 3-1 on page 17 illustrates how a stock trading business component view can be built from the underlying infrastructure and presented in terms meaningful to both business users and IT staff.

Figure 3-1 An equities trader's view of system health

One benefit of the maturity of many mainframe data centers is that the systems are well understood, and well-defined operational controls are in place. As a result, IT managers are confident they can use the technology to automate many of the responses to the dashboard alerts before they turn critical. For example, additional CICS or WebSphere Application Server regions can be started automatically to cope with unexpected increases in traffic. The resulting elimination of false positives can help reduce the amount of time that operations staff spend watching monitors and increase the time available for more productive tasks.

The fewer the number of physical platforms that make up an IT system, the fewer the points of failure and the fewer subject area skills needed to support the system. This implies that fewer support staff are required to maintain business services to the required policy levels.

3.1.3 Non-disruptive change management

Increased complexity in today's IT systems presents an increased risk of failure from insufficient governance around application changes. 24x7 uptime is becoming the norm, and many clients have a clear understanding of the hourly cost of an outage to their business.

The need for market responsiveness and agility in business today demands that application changes must be introduced without impacting availability. Effective change management is one of the keys to delivering continuous operations. Established mainframe clients have decades of experience in understanding the component level structure of their applications and the dependencies involved when introducing changes. When things go awry, failover procedures are in place to recover gracefully to prior state.

The z/OS component management utility SMP/E supports this strategy by managing elements of a system at a highly granular level. Functional enhancements and fixes are tested without disruption to production services, then either applied permanently or backed out as needed.

At the hardware level, the System z family of servers support concurrent upgrades with features such as these:

- Concurrent Book Add, allowing clients to add additional processor modules while the system is hot
- Concurrent Processor Upgrade, to activate installed but unused processors
- Plan Ahead Memory, to activate installed but unused memory

3.1.4 System integrity and security

Highly distributed computing, extensive online collaboration, explosive data growth and regulatory requirements have combined to make information security more critical and more complex than ever for organizations today. Security threats have intensified and become increasingly complex to manage. One of the many reasons why so many of the world's top organizations rely on the mainframe for securing their most sensitive data is because of the IBM commitment to system integrity, as shown by this statement:

"First issued in 1973, IBM's MVS™ System Integrity Statement, and subsequent statements for OS/390® and z/OS, has stood for over three decades as a symbol of IBM's confidence in and commitment to the z/OS operating system….. IBM's long-term commitment to System Integrity is unique in the industry, and forms the basis of z/OS' industry leadership in system security. z/OS is designed to help you protect your system, data, transactions, and applications from accidental or malicious modification".[1]

IBM publishes a similar statement for z/VM®[2].

This superior quality of service extends beyond the operating systems. In support of various encryption requirements, today's mainframe servers provide exceptional performance and functionality using specialized cryptographic coprocessors and accelerators.

3.1.5 Virtualization and scalability

Because virtualization is fast becoming the norm in the IT industry, it is easy to forget that the mainframe has been the leading virtualization platform for four decades. Levels of virtualization above 90% are possible as a result of its "virtualized everything" approach: processors, memory, I/O subsystem, network, and disk. With every release of a new processor we are seeing performance improvements as workloads scale.

Such improvements might come as a result of investment in middleware performance and in the hardware. Figure 3-2 on page 19 illustrates aggregate performance improvement as a result of new mainframe processors combined with new versions of WebSphere Application Server and the Java SDK.

[1] Source: IBM z/OS System of Integrity Statement (http://www-03.ibm.com/systems/z/os/zos/features/racf/zos_integrity_statement.html)

[2] System Integrity Statement for z/VM (http://publib.boulder.ibm.com/infocenter/zvm/v6r2/index.jsp?topic=%2Fcom.ibm.zvm.v620.hcpa0%2Fc7intst.htm)

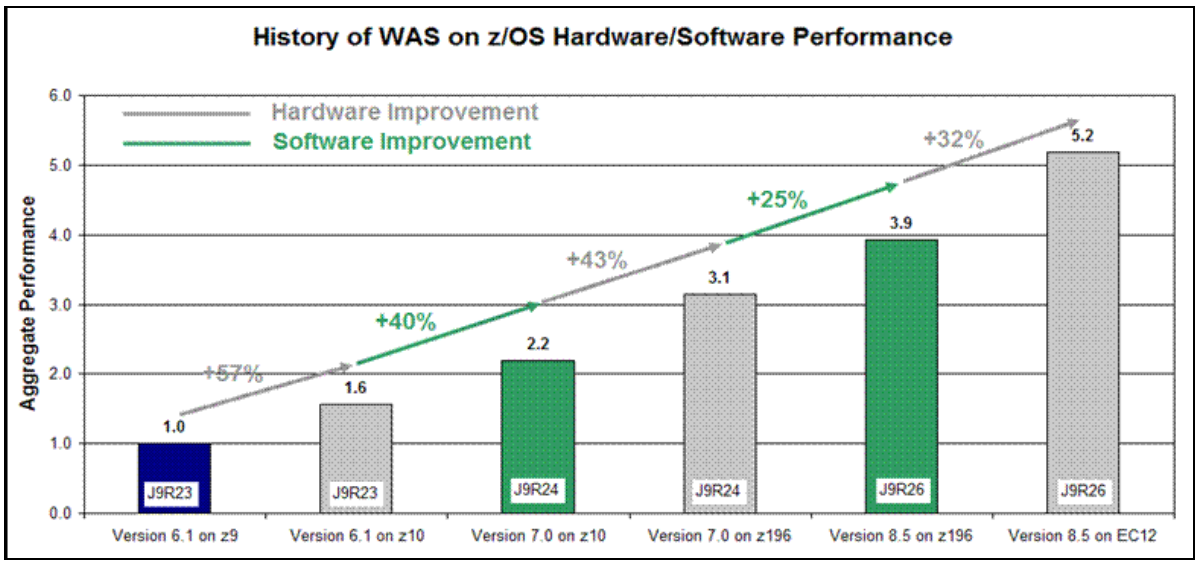


Figure 3-2 WebSphere Application Server on z/OS performance improvement

Figure 3-2 shows the history of performance improvements for WebSphere Application Server on z/OS. (This was a controlled measured environment, so your results might vary.) These results demonstrate a five-fold improvement when comparing WebSphere Application Server Version 6.1 with Java5 on an IBM System z9® to WebSphere Application Server V8.5 with Java7 on a zEC12.

To this point, the discussion has been all about z/OS. But no mention of virtualization is complete without including the growth of virtualized Linux servers running either in dedicated logical partitions (LPARs) or as guests of the z/VM hypervisor.

An IBM internal test under laboratory conditions achieved 97,543 Linux guests on a single mainframe. Although this is an extreme result, z/VM has a proven track record for overcommitting physical processors and memory to virtual machines. For the appropriate workloads, a ratio of 20 logical processors per physical processor is achievable. And because software is licensed per physical processor, this ratio can yield attractive cost savings for clients looking to consolidate workloads.

3.2 Application-related benefits

z/OS-based applications also experience a number of benefits.

3.2.1 Transaction processing

No matter what your business is, you need to ensure that your transactions are properly completed with integrity. Wrong or incomplete results can adversely affect client loyalty and company profits, and lead to claims, lawsuits, or fines. Companies need to be able to rely on computer systems that are 100% reliable and that guarantee transaction integrity at all times. The IBM mainframe is such a platform.

In the case of z/OS, transaction integrity is guaranteed by the collaboration of the middleware components. Figure 3-2 illustrates the amount of processing which takes place “under the covers” when data updates in DB2 (the Participant, shown in the lower half of the diagram)

are made under the control of CICS or IMS/TM (the Coordinator, shown in the upper half of the diagram).

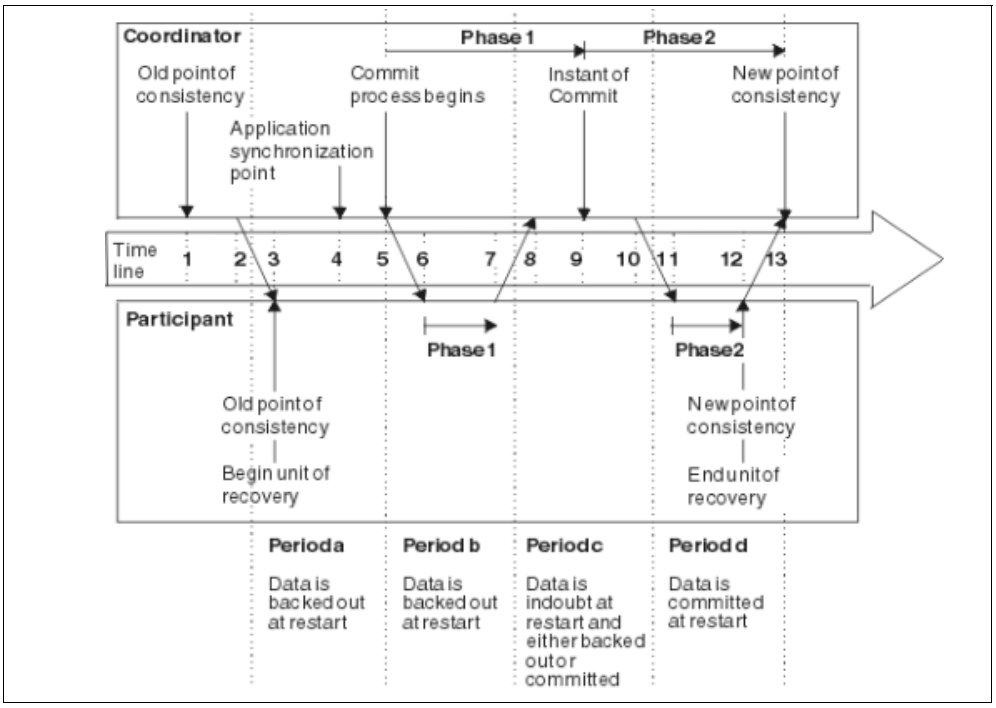


Figure 3-3 Two-phase commit processing between CICS, IMS/TM, and DB2

By following these protocols and communicating at every step in the process, the system as a whole is able to identify the status of any database update. If either the Coordinator or the Participant fails during the process, restart logic is able to deal with any data inconsistencies by backing out uncommitted updates and applying updates that were in the commit phase. And in `Period c`, which we might call the “zone of uncertainty,” the system operator will be notified that there are units of work in doubt, and the system will not allow any further updates to these data resources until they have been resolved.

3.2.2 Interoperability with distributed systems

People used to think of the mainframe as an island, cut off from the rest of the IT landscape. We know that clients today manage heterogeneous IT platforms, and by adopting the types of technology listed here, the mainframe can act as both a provider and requester of services to applications on other platforms with full support for industry standards. Such technology includes the following types:

- Connectors to and from CICS and IMS for JavaEE applications, using standards such as JCA, JMS, and Web Services.
- MQ-based connectors to CICS and IMS, either directly using MQ Bridges or indirectly using MQ trigger monitors
- Java Database connectivity (JDBC) connectors for DB2 and IMS databases
- Access through IBM Enterprise Service Bus solutions including IBM WebSphere DataPower® and WebSphere Message Broker

3.2.3 Collocation of application components

Many organizations view their data as their most precious asset and choose to store it on the mainframe, as discussed in 3.1.4, “System integrity and security” on page 18. We suggest that there is value in placing the business logic on the same physical platform as the data. The IBM Washington System Center conducts periodic investigations into the performance benefits to be gained from this architecture, which we call collocation of the application layer with the data layer[3].

Figure 3-4 illustrates this approach.

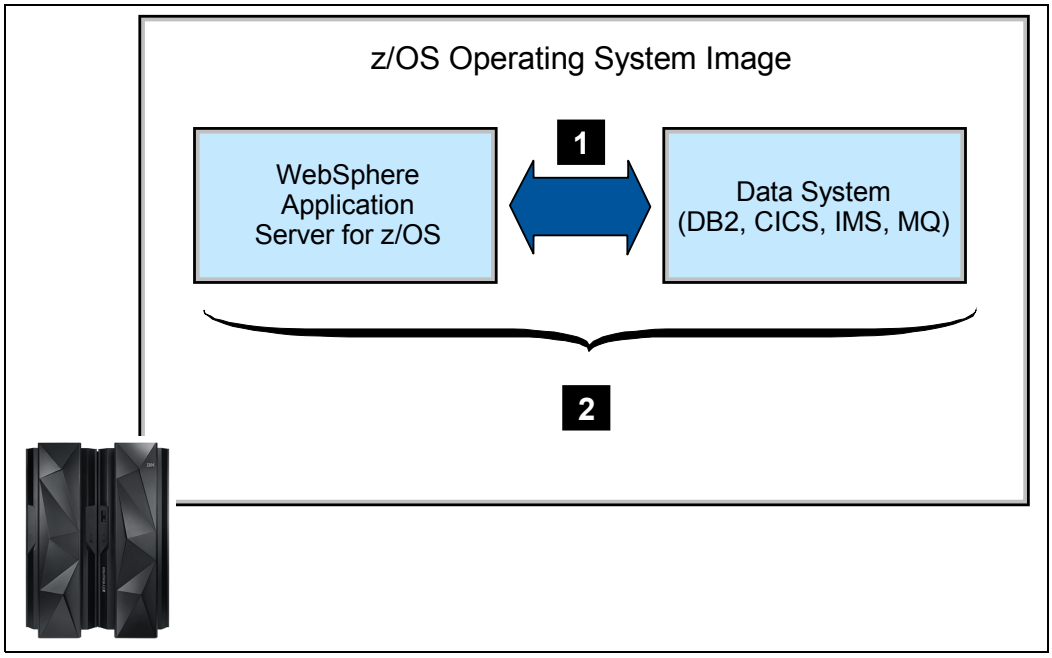


Figure 3-4 The application layer and the data layer on the same z/OS image

The benefits of collocation fall into two broad categories:

- Performance and efficiency

  Collocation within the same z/OS operating system instance enables the application to benefit from cross-memory data transfer, thereby reducing overall request latency, improving overall throughput, and reducing overall CPU utilization through the elimination of network traffic handling

- Operational benefits

  Collocation within the same z/OS operating system instance provides benefits such as the assertion of security identity, maintaining the same thread of execution, and being able to manage within a single Work Load Manager (WLM) classification.

You can also realize collocation benefits through the use of *stored procedures*, which is a common technique with DB2 for z/OS. A stored procedure is a program that is stored on the DB2 server and executes SQL statements, among other things. With this approach you move the data handling logic to the location where the data is stored.

[3] For full technical data about this topic, see “WebSphere z/OS – the value of collocation” at http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101446. It was published November 2009 and updated June 2011.

By using stored procedures, you can improve the performance of distributed applications by:

- Reducing the traffic of information across the communication network
- Splitting the application logic and encouraging an even distribution of the computational workload
- Providing an easy way to call a remote program

We illustrate these benefits in the following figures by comparing how a remote client communicates with DB2 without the use of stored procedures, and with the use of stored procedures. In Figure 3-5, a Java client communicates with the server for each SQL request.

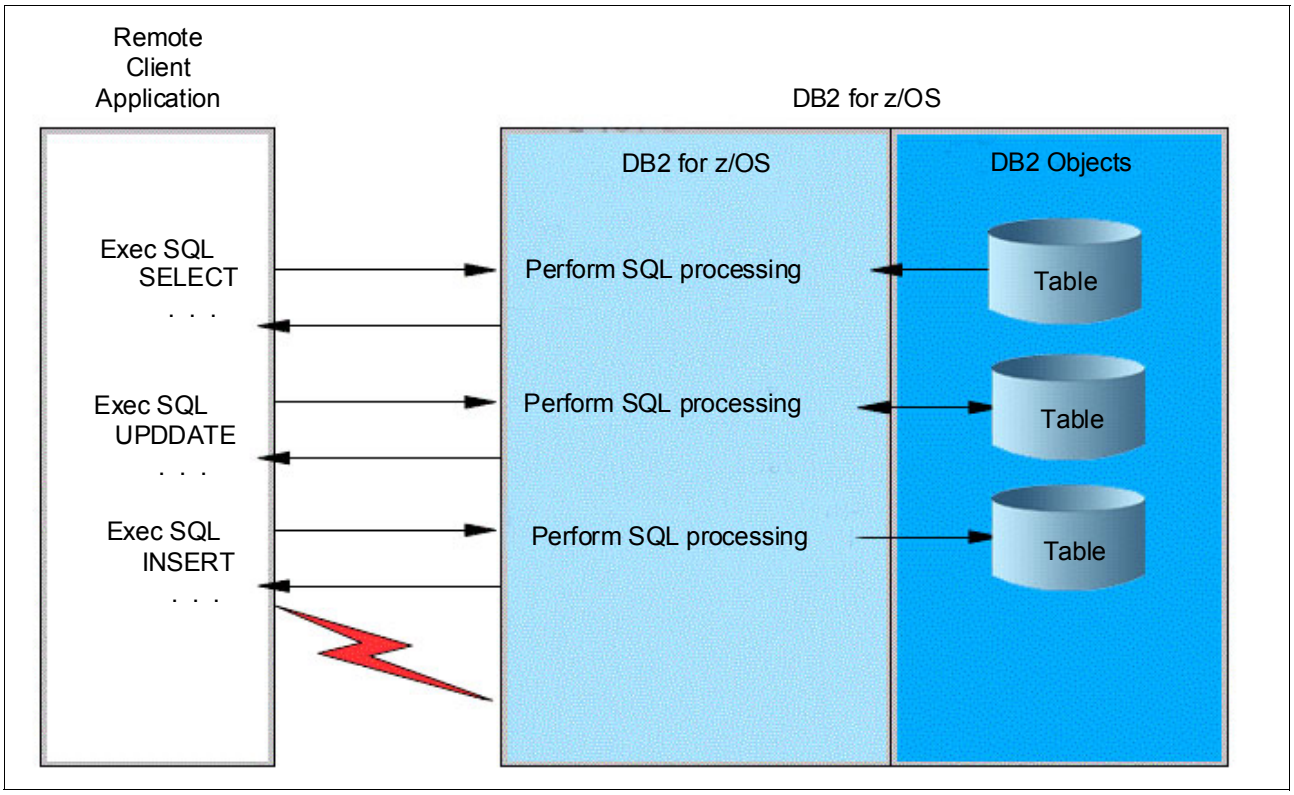


Figure 3-5 SQL processing without stored procedures

The client communicates with the server with a send and receive operation through the network for each SQL statement embedded in the program. As a consequence, the elapsed time is increased by network transmission time or Java overhead, the remote CPU path length is higher than for a local SQL cost, and DB2 locks are held until commit.

In contrast, Figure 3-6 on page 23 shows the stored procedure solution. We have moved the embedded SQL to the server, reducing the network traffic to a single call and return.

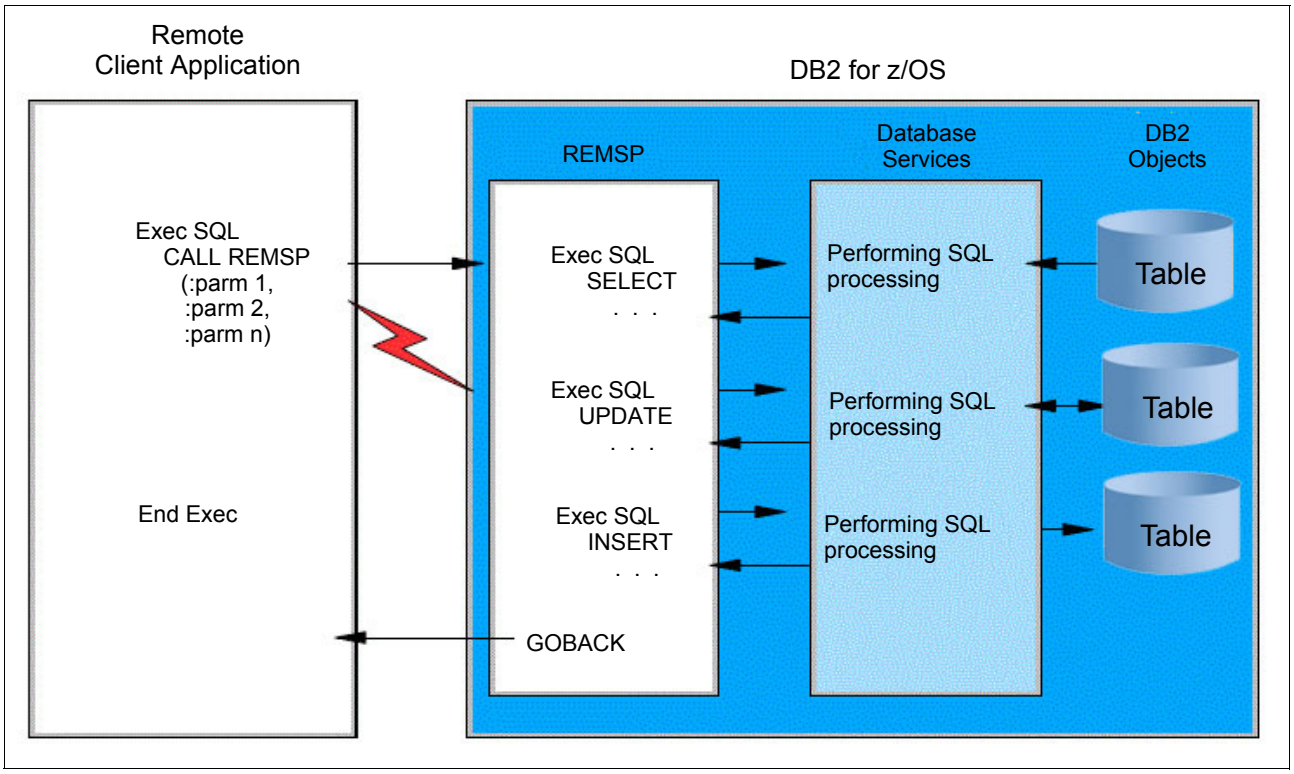


Figure 3-6 Using a single stored procedure to achieve the same result

In this case, the same SQL previously executed by the client has been stored on the server and is called by the client whenever necessary. The invocation is treated as a regular external call:

- The application waits for the stored procedure to terminate.
- Parameters can be passed back and forth.

3.2.4 A single version of the truth

Problems with data integrity have a number of impacts on an organization's business performance. Inconsistencies can develop over time with key entities such as customer and product, leading to customer dissatisfaction, inaccurate analytics, and regulatory compliance issues. Mergers and acquisitions can create challenges for data managers and database managers, who need to resolve issues of deduplication, different formats, and inconsistencies. The growth of data warehouses and data marts can lead to copies of databases becoming unsynchronized with the original operational versions.

We have seen growth in the area of master data management solutions to address these problems, and industry-leading database management systems IMS/DB and DB2 for z/OS continue to provide a trusted and reliable location for storing centralized data. IMS and DB2 are not only optimized for the mainframe platform; they inherit the qualities of service and system integrity of the operating systems and servers on which they reside. These benefits continue to appeal.

Press release: One of the leading private domestic commercial banks in the Philippines has grown rapidly over the last five years. From its current customer base of 3.6 million, the bank has set a goal of having 10 million new customers by 2014. They selected Finacle running on an IBM System z server to consolidate data and applications from across disparate systems and establish a single source of customer information. The bank will leverage this intelligence to anticipate customer needs, create innovative offerings, and enrich customer experience through increased personalization.[a]

a. https://www-07.ibm.com/sg/clientstories/cases/RCBC.html

4

Evolution paths to achieve more value

These days it is business modernization that is driving efforts for IT extensibility, and not the other way around as was the case in the nineties. IT organizations can adapt to change, be dragged along, or risk being bypassed by the business. One consequence of this tendency has been a proliferation of servers that the recent push for virtualization has been addressing.

Chapter 1, “Business and IT trends and how they affect existing applications” on page 1, describes current business and IT trends. Chapter 2, “Attributes of an agile IT architecture” on page 11, positions the attributes of an agile IT architecture, potentially the ideal IT environment that you might be targeting to fulfill your business requirements. Chapter 3, “The way today’s mainframe applications are implemented” on page 15, explains the existing situation in most mainframe IT environments.

In this chapter, we propose ways to improve and extend today’s environment. We take you through several patterns of evolution, based on the current maturity level of the IT environment, with the goal of delivering higher value in the following ways:

- Being easier to maintain
- Responding faster to business requirements
- Reducing costs, such as operating staff and server resources
- Providing better integration between applications and data

We explain four evolution paths to consider, depending on the balance you are trying to achieve between cost reduction and optimal agility. These paths are not necessarily chronological; any one can be used as a starting point. For each path we explain reasons for adopting it; briefly describe proposed solutions; and provide links to more technical details where possible. The paths are:

- Application runtime optimization

  This path focuses on enhancing existing IT infrastructure without changing application code or application architecture. We cover the following activities:

  - Running regular system health checks
  - Implementing agile development and test environments

  - Optimizing the batch window
  - Benefitting from a highly available infrastructure
- Application development foundation extensibility

  This path focuses on improving the business service provided by applications by enhancing the application development tools and processes. We cover the following activities:

  - Using the latest compilers
  - Using a single Integrated Development Environment (IDE)
  - Benefitting from an end-to-end view of development
- Business application modernization

  This path focuses on taking a new look at existing CICS and IMS applications, allowing them to become a client of new types of services. This path might require code changes, but not necessarily in the business logic. For example, the outbound integration of external services can be in the same z/OS partition, in another z/OS partition, or in a distributed environment. We cover the following activities:

  - Calling out to services running in WebSphere Application Server on z/OS
  - Integrating external business rules
  - Incorporating direct access to master data management services
  - Publishing business events
  - Accessing real-time scoring capabilities
  - Integrating local IBM CPLEX® mathematical algorithms
- End-to-end application infrastructure extensibility

  This path, which has been considered by many companies for years through diverse service-oriented architecture (SOA) implementations, has mainly focused on reusing existing CICS and IMS programs as services in other applications. We cover the following activities:

  - Integrating CICS and IMS programs as services in a service-oriented architecture
  - Augmenting the integration architecture with business process management
  - Expanding the multi-channel architecture and integrating the mobile channel
  - Accessing mainframe data directly

We start by explaining how to determine the maturity level of your application architecture.

4.1 Introduction to maturity levels

We assume that you have mainframe applications running today. Based on our experience working with many clients, we have created several simple categories of IT evolution levels related to the ability to accommodate the latest technology and respond to new business needs. We call this *maturity levels*.

At the start of the mainframe computing era and during the first 20 years until the advent of the personal computer, applications typically contained a presentation layer, a business logic layer, and a data access layer. Because these layers were included in the same application code, some applications were locked into a specific 3270 data stream flow. Remaining functions such as logging, locking, scheduling, recovery, and restart were performed by z/OS middleware products such as CICS and IMS.

Over time, users began to request graphical interfaces; this is considered to be the first evolution phase, often referred to as *revamping* or *screen scraping*. Today many revamping solutions exist, allowing a change in user navigation without changing the mainframe application. All the business logic continues to run on the mainframe inside CICS or IMS TM using 3270 logic. High Level Language Application Program Interface (HLLAPI), supported by 3270 terminal emulators running on the newly introduced personal computer, initiated this first phase in the mid-1980s.

The next wave of change was client/server architecture introducing an additional level of business logic, typically running outside the mainframe environment, on top of existing applications. This new business logic layer, in conjunction with new communication standards, was the catalyst for service-oriented implementation. SOA was born, although it was not named as such.

By this time many companies had embarked on client/server projects, which often resulted in increased complexity and increased management complexity for the multiple layers of code and infrastructure. In the early to mid-1990s, the Internet age began to evolve. The standard user interface quickly became the browser, and application architectures with “light clients” gained support. Many applications were becoming a combination of client/server, with multiple business logic tiers and a browser-based user interface.

We can call these distributed applications “service-oriented” because application components would call each other as services. The parts that were missing, however, were standards and sophisticated middleware to flow these service requests. SOA has begun to take off with the publication of the Web Services standard, and middleware and development tooling supporting this standard.

SOA has been implemented in different degrees by companies. At first the mainframe was used as a service provider, but many have further expanded the SOA concept on the mainframe by providing access to application services and data services in the following ways:

- Adding an application server acting as a gateway to existing applications
- Adding an enterprise service bus as an integration layer to support protocol conversion; to allow synchronous and asynchronous interactions; and to allow dynamic changes in service invocation
- Implementing a business process management (BPM) layer

In parallel, architects expressed the need to call external services from CICS, IMS, and batch applications. In this way, the mainframe acquired a role as service requester.

Based on these evolutionary steps, we propose a basic maturity level table (Table 4-1) against which you can position your IT environment. The same terminology is reused later in the chapter.

Table 4-1 Maturity levels

Maturity level	Runtime and connectivity	Application	Data
Terminal emulation	► Stabilized z/OS environment ► Poorly integrated with outside world ► 3270-based terminal emulation	► Host-based transactional and batch applications, written in procedural languages and running natively under z/OS, or under CICS or IMS, but continue to provide core business value ► Lack of insight about the application portfolio and code ► Languages used are assembler, COBOL or PL/I ► Development tools: ISPF-based methodology	► Access to local data managed by VSAM, IMS DB, or DB2
Revamping	► Revamping z/OS applications with a GUI	► Same as above	► Same as above
Client services	► Using a web application server and Java EE standards ► Or using a tightly coupled client/server approach from any server	► Effective split between presentation and business logic layers ► For new applications, decision to privilege distributed environment to obtain “faster” implementation	► Same as above with the following facilities: - ETL processes to allow siloed applications on distributed to share z/OS data - Starting direct access to z/OS DB2 data
Integration	► Flexible integration layer that allows easier orchestration of services ► An enterprise service bus layer, custom-built or using products from IBM or other vendors ► High availability provided by z/OS Parallel Sysplex	► Same as above with the following facilities: - Call out capabilities: Loosely coupling of z/OS applications with outside environment using MQ messaging or Web Services protocol - Development tools: adoption of Eclipse-based solutions	► Only one version of the data, managed by DB2 or IMS DB on z/OS and accessed by distributed application components using the IBM DRDA® protocol ► Simplification of ETL processes
Further evolution	► Specific architectural constructs and specialized middleware in place for business process management (BPM), business rules management, and event processing	► Application fully developed using service-oriented approach, exploiting middleware and Web Services and Web 2.0 standards ► Development tools: full suite of Eclipse-based tools, wrapped inside one workbench, to develop business process flows, ESB mediations, componentized services, business rules, and streams for analytics	► Adoption of analytics by using query acceleration on specialized hardware, and real-time analytics by using scoring engines such as SPSS

Figure 4-1 on page 29 shows the levels graphically. The `P` box represents the presentation layer. The `B` box represents the business logic layer. The `D` box represents the resource data access layer, where resources are managed by a z/OS resource manager such as DB2 or IMS. The `I` box is the integration layer as proposed in service-oriented architecture.

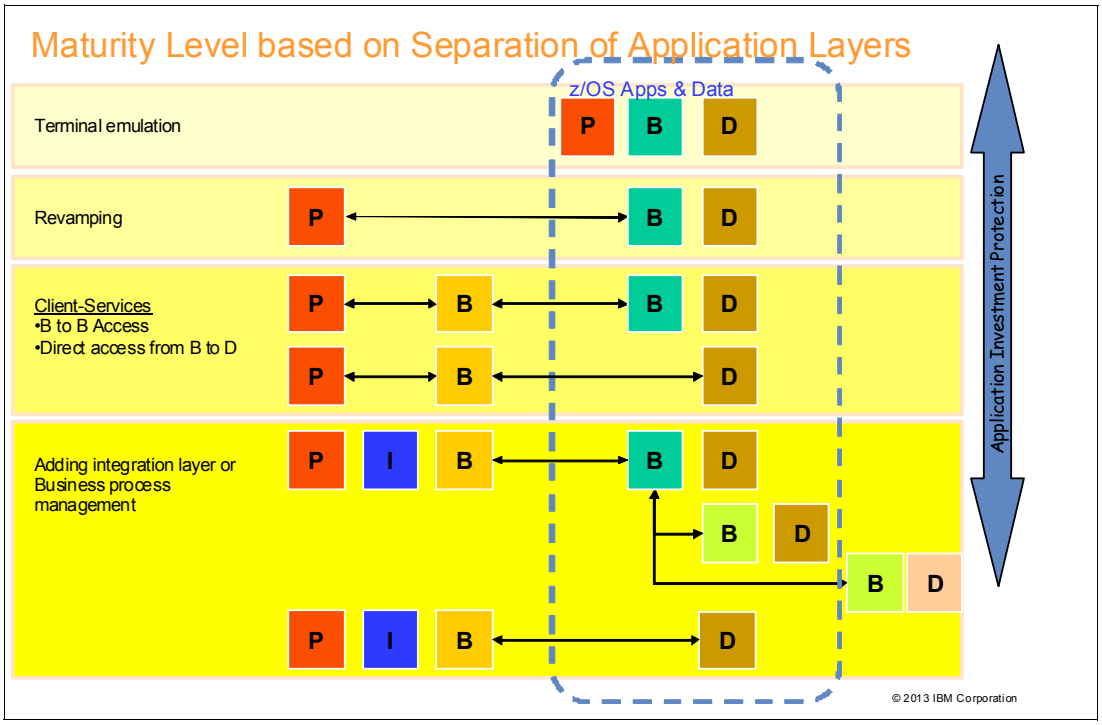


Figure 4-1 Maturity levels based on separation of application layers

4.2 Application runtime optimization

This path focuses on enhancing the existing IT infrastructure without changing application code or application architecture. We discuss the following tasks:

- Running regular system health checks
- Implementing agile development and test environments
- Optimizing the batch window
- Benefitting from a highly available infrastructure

If you have experienced only “some” evolution, you probably have not implemented many changes in your IT infrastructure to respond with more agility to business needs. Does this mean that your infrastructure is out of date? Are you using back-level releases of software? Outdated hardware? We propose several tasks that can be performed at the IT infrastructure level to improve service delivery and responsiveness to business needs.

4.2.1 Running regular system health checks

Perhaps you have been running the same applications for years without performing health checks or tuning activities, due to a lack of time, budget, or skills. You might well have seen an increase in inbound traffic to applications, resulting in an increase in mainframe resource utilization.

If the mainframe environment is a strategic part of your IT infrastructure, then it is surely worthwhile taking a look at exploiting the new functionality provided by z/OS and the middleware software designed to reduce CPU utilization, remove bottlenecks, better manage buffer pools, and optimize operating system and hardware utilization.

A useful way to start is with a health check of your system. There are different levels of health checks available, depending on the focus required. Generally a top-down approach is indicated, starting with general performance indicators before moving to more specific metrics. At the top is the operating System z/OS; in the middle are the transaction managers CICS, IMS/TM and WebSphere Application Server; and at the bottom are the database managers DB2 and IMS/DB.

You can also take a granular approach within the layers. For example, with DB2 you can look at the overall health of each subsystem and then drill down into the individual application packages, SQL statements, and queries.

IBM offers a range of tools to help you perform health checks on each layer in your system and application infrastructure. Explaining them all in detail is beyond the scope of this paper. Instead, Table 4-2 lists the commonly used tools for the layer under investigation.

Table 4-2 Examples of tools to be deployed in a health check

Area of investigation	Suggested tool for analysis
Overall z/OS health	Omegamon for z/OS
CICS transaction performance	CICS Performance Analyzer
Overall IMS performance	IMS Performance Analyzer
DB2 subsystem health	DB2 Performance Expert
DB2 application performance	IBM DB2 Optim™ Query Workload Tuner

These tools can be supplemented by expert advice and guidance from IBM service professionals, the cost of which can easily be repaid from the reduction in resource utilization you can achieve.

4.2.2 Implementing agile development and test environments

Have you considered getting a z/OS test environment on demand to reduce new business function testing time dramatically? Have you thought about providing new system programmers with a testing environment where they can make mistakes without impacting other users of the system?

The design of the IBM mainframe supports up to 60 logical partitions (LPARs). Each LPAR can run one copy of a supported operating system, namely z/OS, z/VM, IBM z/VSE®, z/TPF, or Linux on System z. In a purely z/OS environment, you can have 60 z/OS partitions divided between test, development, integration, and production uses. After reaching this limit, another server would be added.

In practice, z/OS development LPARs are shared using TSO, but suppose you want greater isolation between users? How can you give everyone their own LPAR without hitting the architectural limit? One way is to create multiple z/OS guests under z/VM. But suppose you do not want to install and maintain z/VM? Or suppose you want to free development MIPS for production use?

You can, by deploying Rational Development and Test Environment for System z (RD&T). RD&T provides your system programmers and developers with a much more flexible system, application development, application test, or education environment, which helps to eliminate the cost differentials between distributed and existing mainframe development environments.

Running under Linux on x86-based hardware, the RD&T package comes with the latest versions of z/OS, CICS, IMS, WebSphere Application Server, DB2, and WebSphere MQ.

RD&T usage: IBM Rational Development and Test Environment for System z (RD&T) may only be used for certain development and test tasks. For example, it is not allowed to perform production workloads of any kind, or robust development workloads including production module builds, pre-production testing, stress testing, or performance testing.

4.2.3 Optimizing the batch window

The problem of the “shrinking batch window” is hardly new. The global nature of business today and the need for 24x7 availability of consumer-facing applications are two factors driving the need for the IT department to reduce to a minimum, if not eliminate, the traditional batch window in which online systems are taken down for an agreed number of hours.

Optimizing the batch window is about more than ensuring that work completes in the shortest amount of time with the resources available. New business requirements might demand techniques and technologies not available in your current batch environment. For example, creating documents in PDF format is easier do in Java compared with PL/I or COBOL owing to the public availability of the needed libraries.

In discussing the opportunity for improvement, we distinguish between three types of workload typically running in the batch window:

- Day-to-day business operations

  This workload involves applications that process large amounts of data as part of day-to-day business operations. These can execute home-grown programs or vendor-supplied utilities such as sort and merge.

- Bulk data movement

  This workload involves jobs that run to move data from one location to another. This could be the Extract-Transform-Load (ETL) processes associated with data warehouses, or the bulk file transfer programs operating between organizations.

- “Housekeeping” jobs

  This workload involves database reorganization, pruning, and backup, which are common examples of batch work required to optimize performance and enable recovery, but which fall outside the scope of business processes.

We now consider each of these in turn and list a number of ways in which the processes can be improved to meet the needs of today’s requirements.

Day-to-day business operations

After 45 years of coexistence between transactional workload and batch workload, the granularity of batch processing has changed to provide more parallel processing, thus allowing transactional and batch business workload to run simultaneously, sharing the same resources.

Other significant changes are described here:

- XML processing

  There is support for XML documents in PL/I and COBOL. DB2 for z/OS (from V9 onwards) and IMS DB can store XML as a data type. If applications need to be modified to exchange data in XML format with applications on other platforms, this could be a viable alternative to a rewrite in a different language.

- Interoperability with Java and C / C++

  PL/I and COBOL compilers provide support for interacting with these languages. As a result, you can choose to develop a program in Java to produce documents in pdf format and have them integrated into existing programs.

- Adopt the use of Java batch in your z/OS environment

  This strategy combines the benefits of using Java (functionality, larger skills pool, tooling) with the benefits, as previously outlined, of using z/OS as your runtime platform. Special Java APIs have been developed for Java in batch that facilitate functions such as handling traditional sequential files, submitting jobs from Java programs, and interacting with the system console. IBM RACF® APIs are available to integrate Java into the z/OS security model.

Here is an example of how adopting Java in a batch environment can address a business problem. Eliminating paper invoices and statements and replacing them with pdf documents that clients can elect to download instead is a common cost-saving exercise. Figure 4-2 illustrates how Java can be introduced into an existing batch process.

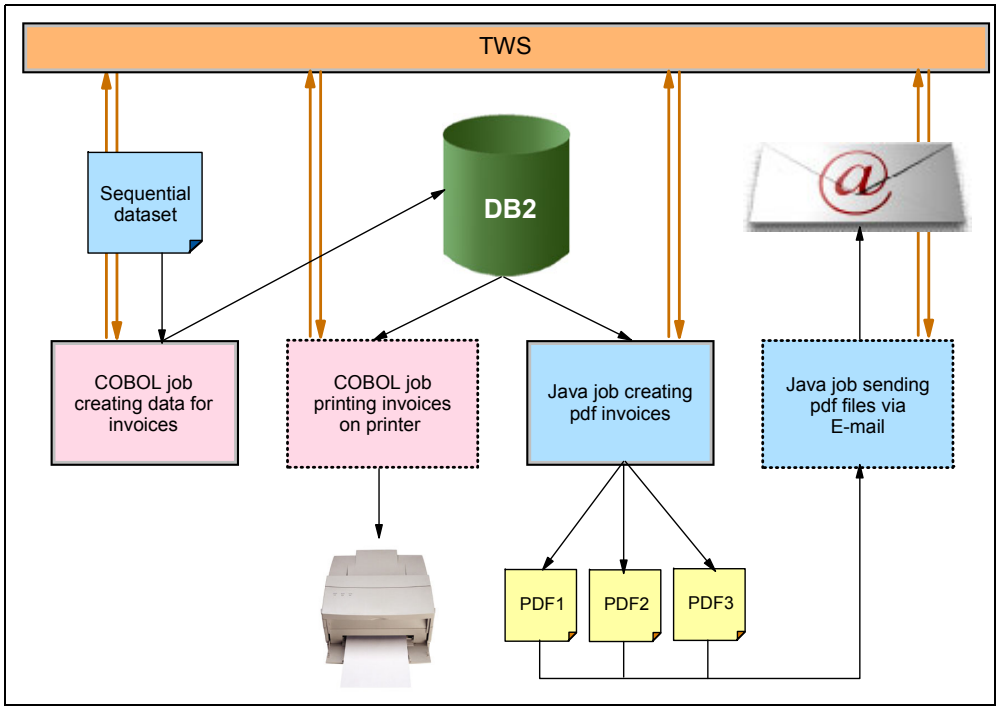


Figure 4-2 Incorporating Java programs to transform a business process

Creating the PDF files in COBOL and PL/I is possible, although more difficult and time-consuming than doing it in Java. The Java job uses the same DB2 input data as the COBOL printing job, but instead of real printed invoices, it creates PDF files using a Java PDF library. A second Java job sends those invoices directly to the recipient through email.

COBOL and Java programs can be executed under IMS Batch Management functions provided by IMS Transaction manager or IMS Database Manager. For 45 years, this product has been providing capabilities to manage application checkpoint and restart logic outside of applications, thereby allowing the repositioning of sequential files and ensuring data integrity with DB2 and IMS databases resources.

More recently, and for clients not using IMS, WebSphere Application Server for z/OS has been extended to provide a batch container. It offers a robust environment for managing

suites of batch processes with a range of functions including job scheduling, job control language, execution environment, and a job management console.

Bulk data movement

The use of data warehouses (DWs) or data marts to store information for analytical purposes is pervasive today. A traditional way to populate a DW is by a serial process known as Extract-Transform-Load (ETL). There are a number of issues with this approach:

- Data currency

  Users are increasingly demanding accurate real-time, or at least near real-time data for analytics. As operational databases grow over time, it takes longer to complete (ETL) processing. In the past it might have been acceptable to make business decisions based on data one *week* old; today it might not be good enough to have data just one *day* old.

- Concurrency

  Depending on the method used to perform the extract, the operational data might not be available for update. And conversely, during the load of the target database, the data might be unavailable for query. These outages, which tend to grow in time as more data accumulates, might no longer be acceptable to the business users.

- Cost

  ETL processing can be expensive in processor time and can tie up resources that would otherwise be available to other business processes.

To address this issue, you can reconsider your whole approach to this process. Instead of unloading or reading the data in bulk every time, changed data can be captured from the database log and moved into a queue for downstream processing. Figure 4-3 illustrates this technique.

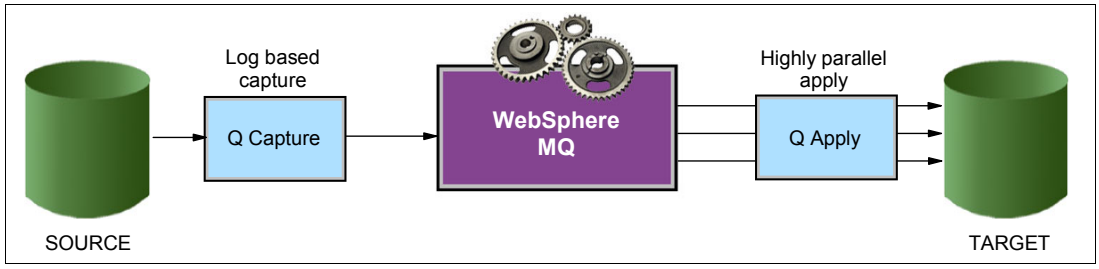


Figure 4-3 Replacing traditional ETL with log-based capture and apply

Log-based capture substantially reduces the amount of time required to maintain the currency of the target data. A global banking client recently told us that 90% of all transactions against operational retail banking data were read-only. Using this capture technique is a more efficient technique for capturing the 10% of data that has changed.

Technologies such as the IBM InfoSphere® Server family of products can enhance this approach with continuous replication, transform, and apply techniques to move this entire process out of the traditional batch window so that it becomes part of continuous operations.

Housekeeping jobs

Successive versions of the IBM DB2 for z/OS and IMS resource managers have incorporated features to reduce the amount of time needed for data reorganization and backup. One example of this is in-line copy. With the right toolset, DB2 and IMS can now take an image copy while a reorganization is running, thus reducing the amount of time the job takes to complete and thereby reducing the period of unavailability. At the same time, a copy is made available for recovery purposes.

For IMS databases, unnecessary housekeeping jobs can be eliminated by adopting tools such as IMS Database Reorganization Expert. This tool collects statistical information about the state of IMS databases and determines reorganization needs based on the policies defined.

More detailed technical information about these and other techniques to optimize the batch window can be found in *Batch Modernization on z/OS,* SG24-7779.

4.2.4 Reaping the benefits of a highly available infrastructure

Businesses can barely afford any outages these days. We are often talking about application availability of 24 hours a day, 7 days a week, and 365 days a year. If your service level agreements impose this type of availability for your critical workloads, you need to consider ways of providing that to the business. Possibly you have already implemented some features of z/OS Parallel Sysplex, which is a clustering technology allowing the highly reliable, redundant, and robust System z architecture to achieve continuous availability for z/OS-based workloads. Both hardware and software tightly cooperate to achieve this result.

Architecturally there are three ways to share data, as illustrated in Figure 4-4.

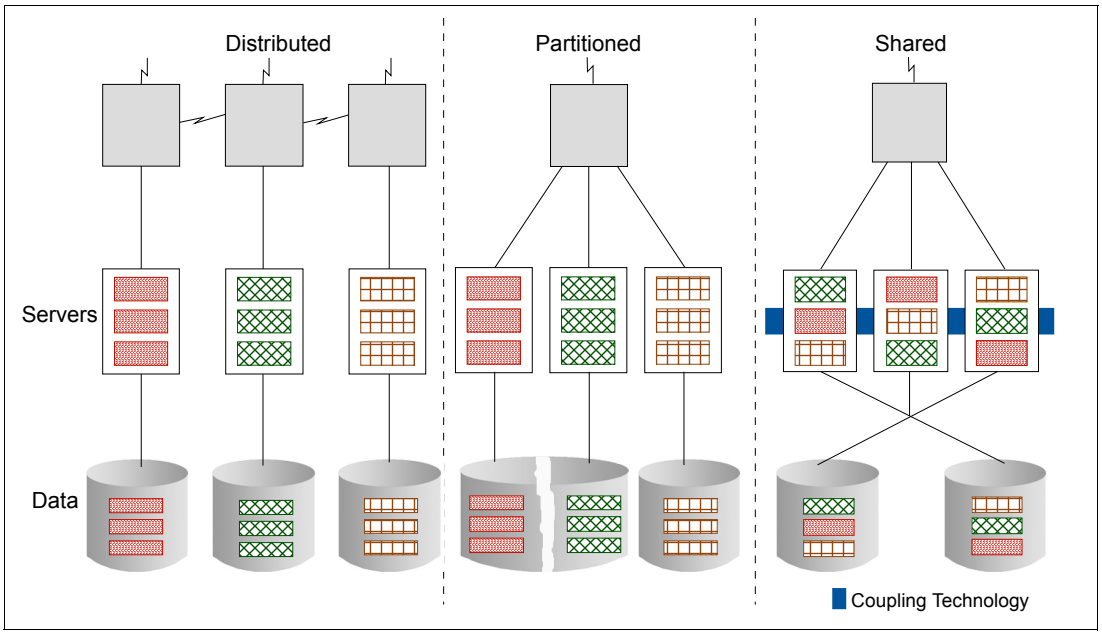


Figure 4-4 Three architectures to share data

z/OS Parallel Sysplex is based on a shared data concept. All data is accessed by different z/OS application execution environments linked together by coupling technology.

This architecture is based on external components to the application code that can remain unchanged:

- Hardware and z/OS components provide the coupling technology and base services
- Data sharing implementation in all key database management systems
- Dynamic workload balancing capabilities provided by z/OS transaction managers
- For users and operators, a single system image view with highly available connectors

The business value of z/OS Parallel Sysplex can be summarized as follows:

- Single points of failure are eliminated.
- Peer instances of a failing subsystem can take over recovery responsibility for resources held by the failing instance.
- The loss of a server is transparent to the application.
- Server workload is redistributed automatically, with little performance degradation.
- Software upgrades can be rolled through one system at a time on a sensible time scale for the business.

z/OS Parallel Sysplex also provides benefits regarding capacity and incremental growth, and workload balancing. The entire Parallel Sysplex cluster can be regarded as a single logical resource to users and business applications. Workload balancing also allows you to run diverse applications, while maintaining response times critical to the business.

CICS, IMS, and DB2 have implemented features to help achieve application and data availability. For further details, consult IBM Redbooks publications, particularly *System z Mean Time to Recovery Best Practices*, SG24-7816[1].

4.3 Application development foundation extensibility

Companies that have experienced only some or moderate technology evolution will likely not have made many changes to mainframe application development tools and processes. By implementing the suggestions in this section, it is possible to improve the performance of existing applications and reduce delivery time for new applications and changes in existing applications.

Note that in this evolution path, we do not consider investment in changes to the architecture of the core applications. We simply focus on reducing the cost of maintenance and improving performance.

4.3.1 Using the latest compilers

In 3.2.4, “A single version of the truth ” on page 23, we use the phrase “single version of the truth” to describe the benefits of holding a single copy of master data in DB2 or IMS. We can extend this idea to the z/OS environment as a whole. Millions of transactions per day, accessing terabytes of data per day, execute within a single operating system version and a single middleware stack. A single program version is used to achieve this.

The positive thing is that when you upgrade the hardware, operating system, and middleware products such as CICS or IMS, you do not need to recompile those application programs. Compatibility with earlier versions has been a long-standing commitment of IBM. Programs written 30 to 40 years ago will still execute on the latest machines and versions of the operating system.

So although it might be tempting to leave these programs untouched, doing so means that you would not be benefitting from compiler enhancements. IBM compilers improve developer productivity by exploiting advances in hardware and advances in performance optimization, without requiring special coding or source code changes. New compiler releases are tested against new versions of the hardware. Internal test and measurements showed a 30% improvement in IMS workloads running on the new zEC12 server, as a result of faster

[1] http://www.redbooks.ibm.com/redbooks/pdfs/sg247816.pdf

processors, cache, and the improved compliers. Reduction in processor cycles by recompilation can be a compelling option if you are under pressure to increase performance or delay a hardware upgrade.

Implementing the latest compilers requires a “compiler campaign” across the lines of business. We often encounter the objection that this is a disruptive exercise requiring more MIPS to achieve. However, with IBM Rational Developer and Test for System z (RD&T), though, the effort can be offloaded from the mainframe.

The RD&T environment can be assigned to a single developer in a small system configuration, or it can support small-scale team environments on a server platform. These environments can be tailored to a single developer’s or a team’s testing needs without altering existing mainframe testing environments. In this way you can run the compiler campaign testing without burning additional mainframe MIPS.

And while on the topic of compilers, be aware that there extensions in COBOL and PL/I to support interoperability with Java programs and support for data in XML format. This provides you with new options for integrating applications across platforms.

The following compiler product pages provide more detailed information about this activity:

- Enterprise COBOL for z/OS

  http://www-01.ibm.com/software/awdtools/cobol/zos/
- Enterprise PL/I for z/OS

  http://www-01.ibm.com/software/awdtools/pli/plizos
- z/OS XL C/C++

  http://www-01.ibm.com/software/awdtools/czos/

You can also refer to the white paper “Using the latest compiler technologies on System z - Capitalizing on the hardware and software that runs your business” which is available at:

http://public.dhe.ibm.com/common/ssi/ecm/en/zsw03065usen/ZSW03065USEN.PDF

4.3.2 Using a single integrated development environment

Some studies indicate that typically up to 70% of an IT budget is locked in maintenance. At the same time, mainframe development skills need to be refreshed as experienced staff retire. So you might be hiring new college graduates and, if so, you want them to become productive as quickly as possible.

Today’s application development environment can help to control maintenance costs by simplifying and shortening maintenance tasks and reducing the time it takes to make new developers productive. Thus, it makes sense to use a common set of development tools across all platforms so that newly acquired skills are reusable from one project to the next.

The IBM single integrated development environment for mainframes is Rational Developer for zEnterprise (RDz). It offers a modern Eclipse-based multiplatform development environment attractive to developers regardless of their programming background. For new hires, there is no need to learn ISPF and similar tools long associated with “green screen” technology. Figure 4-5 on page 37 shows the RDz workbench COBOL development mode.

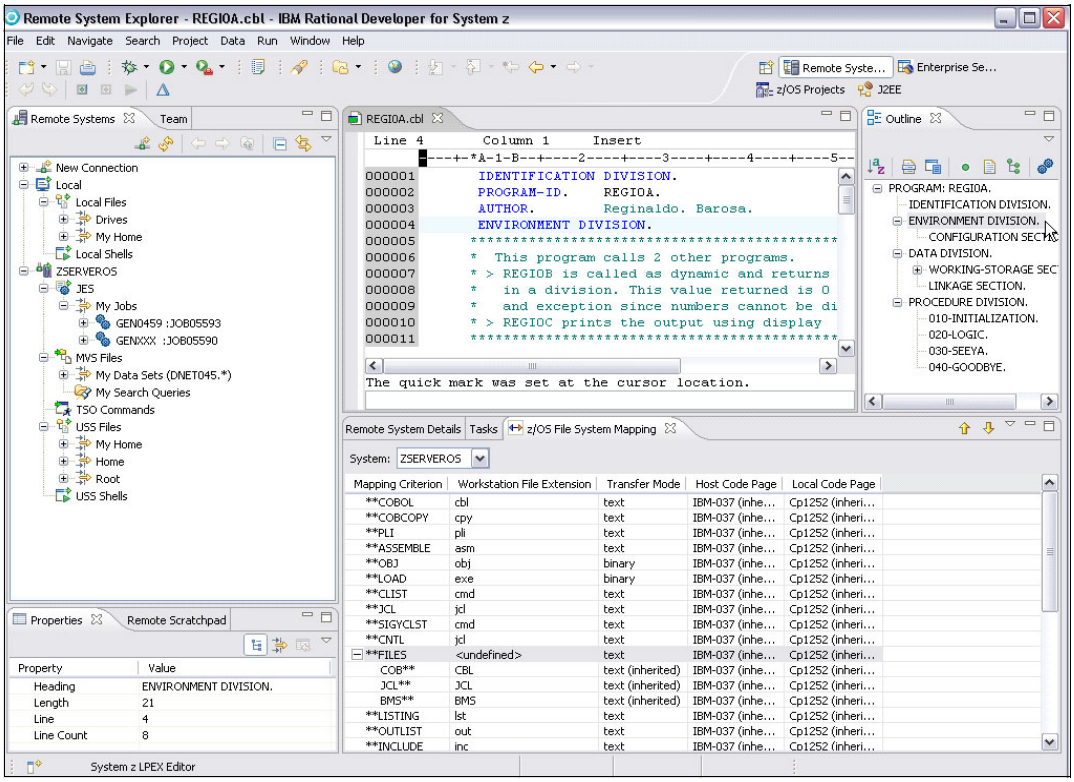


Figure 4-5 RDz - an alternative to “green screens”

RDz also provides a common view of tiered applications that support multiple platforms, languages, and technologies. For example, a developer maintaining a COBOL program can also work on Java code built for WebSphere Application Server using the same tooling.

Today’s demand for mobile computing means that the IDE has to support mobile development, which brings a new set of challenges for developers:

- Which smartphones and tablets am I going to support?
- What new skills do I need? HTML5, Java, Objective C?
- Am I going to maintain separate applications or software stacks for each of the mobile operating systems?
- What about integration with my back-end systems?

The IBM Worklight Mobile Platform is designed to meet these challenges. RDz Version 8.5 includes IBM Worklight for quick deployment of mobile applications to the IBM Worklight Server. We further discuss this challenge as a separate variation of an evolution path in 4.5.3, “Expanding the multichannel architecture and integrating the mobile channel” on page 53.

4.3.3 Benefitting from an end-to-end view of development

Software development organizations are facing increasing pressure to promptly deliver high-quality software aligned with continuously evolving business objectives. However, this is becoming increasingly difficult to achieve because of several factors:

- Teams are often expected to deliver more applications with the same or even fewer resources.
- Many teams are geographically dispersed, which can make collaboration challenging.
- Many organizations find it difficult to apply consistent processes and standards across disparate projects and groups.

RDz is well integrated with IBM Rational Team Concert™ (RTC), which is a collaborative lifecycle management solution with integrated planning, work item tracking, version control, build management, and reporting. RTC is designed to increase individual and team productivity for teams that follow agile or traditional development processes. The Web 2.0-based browser interface makes it possible for users to access project areas, browse repository information, update tasks and plans, and track work item progress.

> "Building an agile development team requires a multiplatform approach, and Sodifrance uses Rational Developer for System z and Rational Team Concert for System z to help application teams synchronize their efforts and improve collaboration. Rational on System z offers a powerful and valuable combination for any company that wants to boost its development team's productivity."
>
> — Hugh Smith, Project Manager, Sodifrance

IBM is following its own advice in developing CICS and IMS. The product team developing CICS Transaction Server (TS) and the associated family of tools and connectors recognized that its product development cycle could be made more agile, incorporating client feedback into beta releases more rapidly and effectively. The team sought an environment for the delivery and service of future releases of CICS TS that would enable all stakeholders to collaborate using a single tool. To that end, they created a highly configurable, end-to-end, integrated development environment using both RTC and RDz.

"Rational Team Concert brings a huge amount of added value to managing projects," says Simon Rachman, delivery manager, IBM CICS TS. "Its power, when used well, is that it enables a complete view of project workloads and progress, all with less effort."

This new, agile approach has helped the team increase transparency, increase efficiency, improve responsiveness to change and improve software quality. The reporting tools have reduced the amount of preparation required for status meetings by 75%, and improved the efficiency of status meetings, reducing meeting times by one-third.

4.4 Business application modernization

In the description of maturity levels provided in 4.1, "Introduction to maturity levels" on page 27, we explain how evolutionary steps are able to change a monolithic application to benefit from a service-oriented architecture.

In this section, we describe steps that can add innovative technology to existing applications to extend, transform, and grow them. Note the following suggestions:

- Extend what exists by using new programming languages and new services as appropriate. Do not rewrite existing business logic unless you must. Existing business logic on mainframes performs well, represents the business, and is tested.
- Modernize the perception of the core business applications. New "buzz words" include business rules, business events, and master data. See how these can be integrated with existing applications.
- Grow utilization by providing new use cases. As an example, an existing pricing module used in batch could be extended for transactional scenarios.

Technologies are in place for mainframe extensibility, as illustrated in Figure 4-6 on page 39.

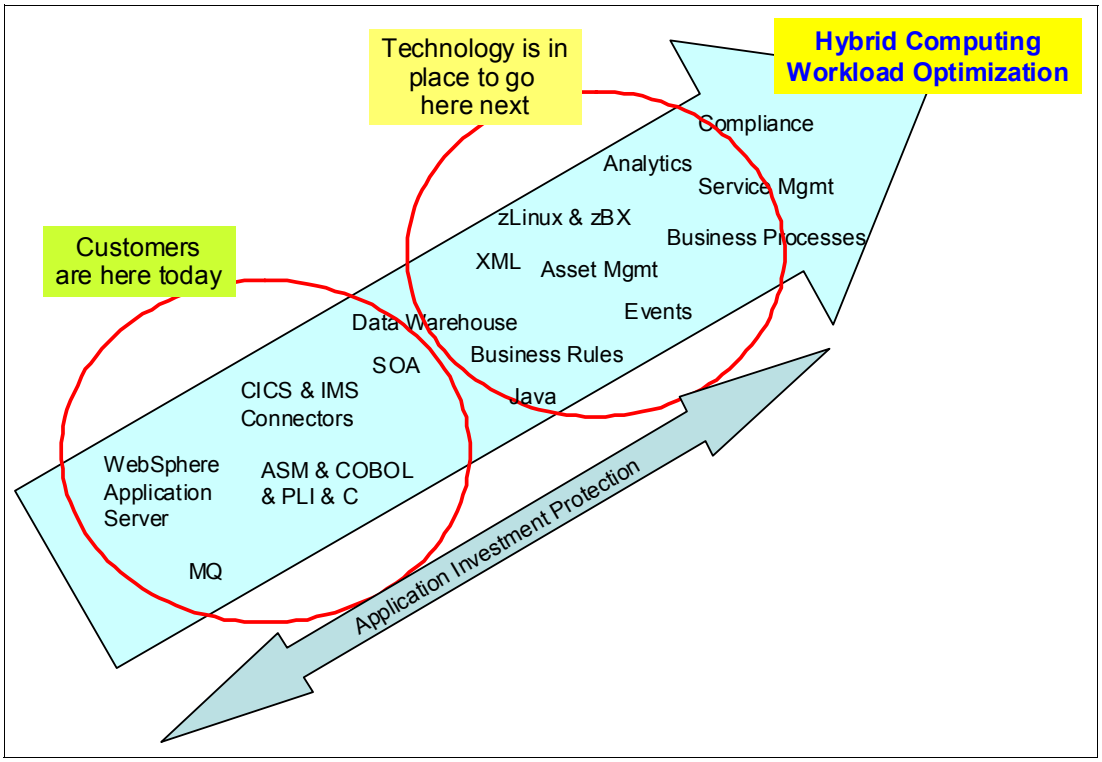


Figure 4-6 Technology is in place on the mainframe for business application modernization

4.4.1 Application restructuring

First we need to describe how a monolithic transactional application looks, and then explain how to transform it in a way to make it future-proof, and easier and less costly to maintain.

Figure 4-7 on page 40 shows a simplified view of a transactional application. The transactional aspects are managed by CICS or IMS. The `B` box represents the business logic layer. The `D` boxes represent the data access layer, where resources are managed by a z/OS resource manager such as DB2 or IMS.

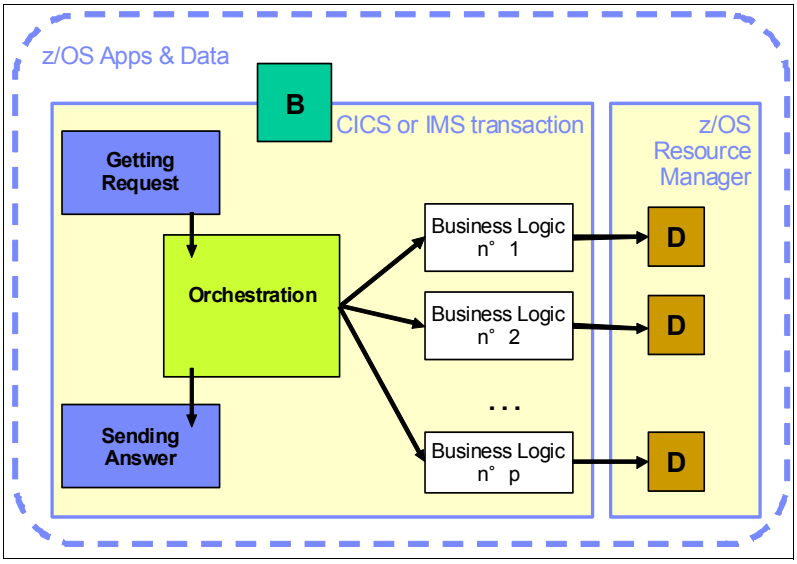


Figure 4-7 Logical view of a transaction application

These logical steps make up a transaction:

1. An input layer processes the data coming from a client. At the end of this process we have an input area filled with EBCDIC characters.
2. An orchestration layer analyzes the input area and implements a sequence of calls to business logic.
3. The business logic layer implements the service, directly calling resource managers to access and update data managed by DB2, VSAM, or IMS DB, or eventually send a message to MQ, taking care of delivery to the target resource manager.
4. An output layer builds the data to be sent back to the client and finishes the unit of work. All updates are committed using the two-phase commit protocol.

The CICS transaction scope can span processing in different CICS regions. For an IMS transaction, the transaction scope includes all layers executed in a single message processing region.

The following sections describe how to extend business logic by providing access to services outside of the CICS or IMS environment. The S box in Figure 4-8 on page 41 represents those services.

Performance degradation might occur when calling a service on a distributed platform versus running the service in the same CICS or IMS business logic layer. But it was the premise of SOA: allowing B2B implementation involving external business partners, external service providers, or external packaged applications.

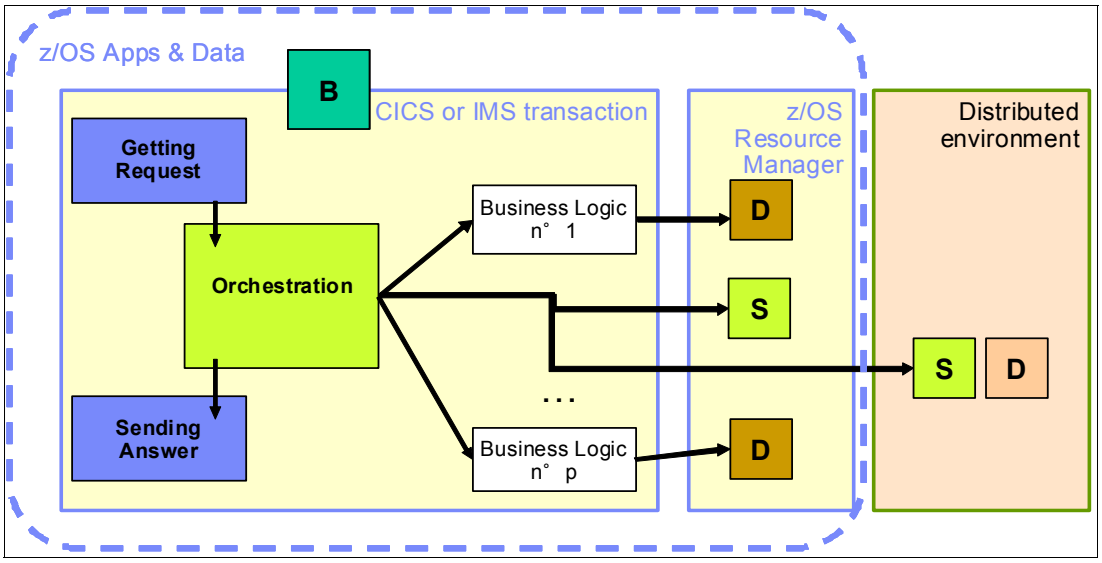


Figure 4-8 Access services outside of the transaction server environment

To perform this transformation you need extensive knowledge of your assets. IBM provides the following tools to help you gain this knowledge:

Rational Asset Analyzer This tool delivers up-to-date knowledge of application components, and quickly provides a comprehensive impact analysis in perspective of application modernization.

CICS Interdependency Analyzer This runtime tool identifies the sets of resources that are used by individual CICS transactions, and collects and analyzes data about transaction affinities.

IBM Tivoli® Asset Discovery for z/OS This runtime tool discovers running assets.

4.4.2 Call out to services running in WebSphere Application Server on z/OS

Service-oriented architecture is an architecture style centered around services, with standardized interfaces. It is a methodology of designing and running the components of an end-to-end business application in various IT infrastructures, depending on who is providing the services. SOA helps to provide agility by giving deployment options for each component.

WebSphere Application Server is the IBM cross-platform offering that provides an application server supporting the SOA and Web Services standards.

If you require optimized access to those services from your z/OS-based applications, you can privilege a direct integration inside the same z/OS LPAR. For that purpose, WebSphere Optimized Local Adapters (WOLA), a component of WebSphere Application Server for z/OS, provides an efficient cross-memory mechanism for calls both inbound and outbound to and from WebSphere Application Server for z/OS.

WOLA can communicate with external address spaces, which include CICS, IMS, batch programs and UNIX Systems Services programs. Because it avoids the overhead of other communication mechanisms, WOLA is capable of high-volume exchange of messages at low cost.

Applications may use directly the WOLA outbound APIs to call a service running in WebSphere Application Server for z/OS with three main benefits:

- Transactional integration of external services running in CICS or IMS
- Shorter path length, due to cross-memory communication and z/OS optimizations
- No network involvement

To obtain more technical information about WebSphere z/OS Optimized Local Adapters, visit this page:

http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101490

4.4.3 Integrating external business rules

One problem facing business architects is the lack of responsiveness to new and evolving business requirements, such as driving simple to complex day-to-day business decisions in real time. The new trend is to be able to change business rules more dynamically, based on some business event reception, to make improved business decisions.

In many environments today decision logic has been embedded within existing global business logic. Programming and updating decision logic was in the hands of the COBOL or PLI developer instead of being managed by people who have the business expertise. In the worst case, updating business rules had to be done as part of release-by-release application updates, involving development and test. This arrangement has some major drawbacks:

- Decisions are locked in processes and applications
- Programming skills are needed to create and modify decision logic
- Speed of business change is limited by IT bandwidth and cannot take place in a continuous mode
- Manual intervention increases costs and reduces client satisfaction

The norm these days, however, is being able to update business rules on a continuous basis, without the need for an “IT project”. A business rules management system (BRMS) addresses these drawbacks. When acquired by IBM, ILOG® was a leader in business rules management systems. IBM extended the BRMS capabilities to provide z/OS clients with the best options either on the z/OS platform or on distributed. With ILOG, decision services can be implemented in every step of business processes, including those running on the mainframe.

IBM Operational Decision Manager for z/OS, illustrated in Figure 4-9 on page 43, provides functionality for authoring, testing, analyzing, and deploying decision logic used by mainframe applications. It offers several deployment options for the rules generated for the z/OS platform. Rules can run as COBOL programs inside CICS or IMS transaction managers, batch programs, or as a Java service inside the Rules Execution Server (RES). The principle of using a BRMS, such as IBM ODM, allows the separation of decision logic from core application logic.

- When deploying rules as Java services in RES, you can call these as a read-only service from any CICS or IMS application with a simple API. Rules run outside the transactional or batch environment on a centralized JEE container.
- CICS also has the option to run the rules server inside a CICS region using a JVM server.
- For compatibility with previous releases, there is also support for the deployment of rules as COBOL subroutines, which are implemented in the traditional business logic container or batch job. This option is currently stabilized.

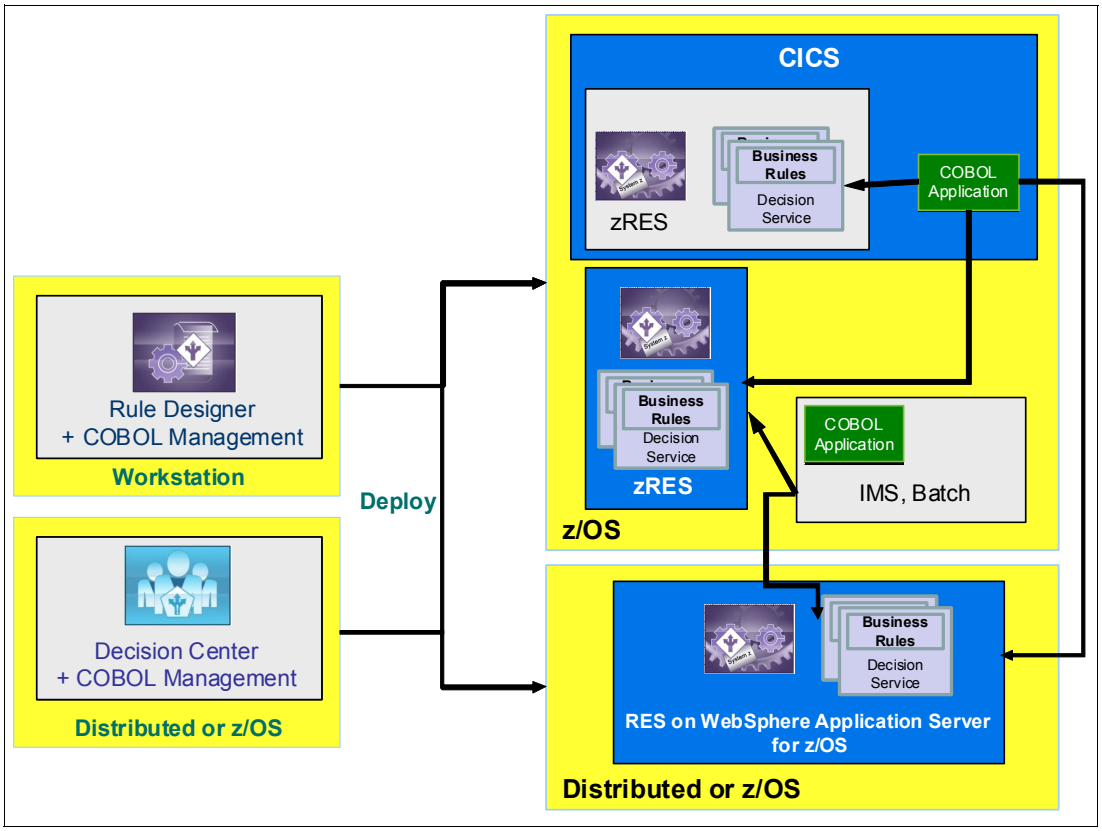


Figure 4-9 IBM Operational Decision Manager architecture

4.4.4 Incorporating direct access to master data management services

We could try to define master data as a single repository of critical reference information coming from different sources. Master data management is “the” answer to address the challenge of having a “single version of the truth.”

For many years, when business processes were supported by mainframe applications only, mainframe databases were viewed as this central repository of data. However, over time and because of the trends in IT previously discussed, data has become more distributed. Due to mergers and acquisitions, duplicate datastores began to increase. It is not uncommon for a large global corporation to keep customer and product information in multiple databases in different locations.

These developments all lead to a massive challenge in keeping data synchronized and avoiding the risk of using untrustworthy information. A new style of master data management solution is required, one that tunnels multiple sources of data into one view for all users and applications using that data.

IBM provides a leading edge solution for master data management, including data governance, change management, and access services. This solution has the following goals:

- Decouples master information from individual applications
- Becomes a central, application independent resource
- Simplifies ongoing integration tasks and new application development
- Ensures consistent master information across transactional and analytical systems

InfoSphere Master Data Management Server offers a high performance, high scalability foundation to access master data using several options (combining distributed with z/OS).

For example, when data is stored in both IMS DB and DB2, a combination of a master data management solution and a data warehouse can be used to achieve the following benefits:

- The MDM server holds the master data and enforces a unified set of business rules on the creation and management of the master data. The MDM system then feeds the clean dimensional data back to the data warehouse.
- Some master data stays in one of the mainframe applications where it is not possible to remove the data structures of master data.
- New application development is enhanced by using the master data management functions. For example, the “Create Customer Service” is already in the MDM system and can be reused by new applications, thereby speeding up development. See Figure 4-10.

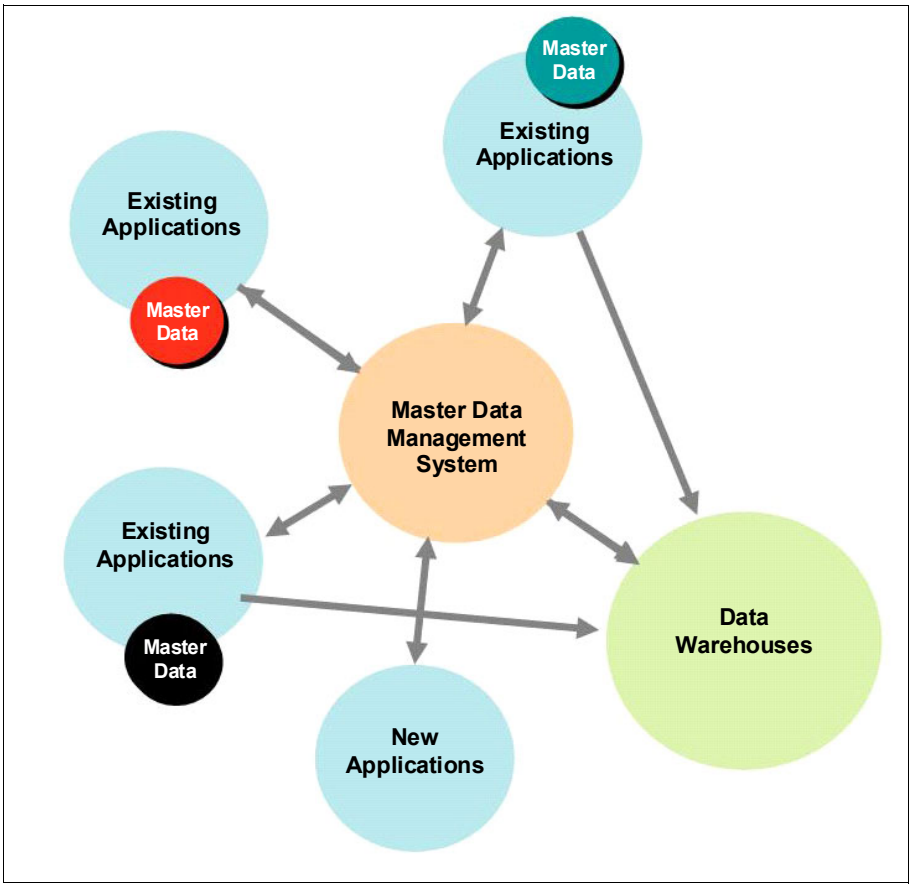


Figure 4-10 MDM function reuse

For clients using the IBM MDM Server, the MDM Query Connect server is designed to supplement the Central Transaction server by providing a native mainframe capability to address searches and inquiries specifically when data is residing in DB2 for z/OS.

A COBOL adapter enables COBOL programs to access the Master Data Management Server services through the MDM Server Central Transaction server (for update requests) and through the MDM Server “Query” Connect (for read-only requests).

As shown in Figure 4-11 on page 45, client transactional systems running in the mainframe environment send the request message to the request queue. The Central Transaction server

and MDM Query Connect server can share the same queue or have separate ones. If they share a queue, each transaction server must select the message (transaction) it supports using the JMS Message Selector mechanism.

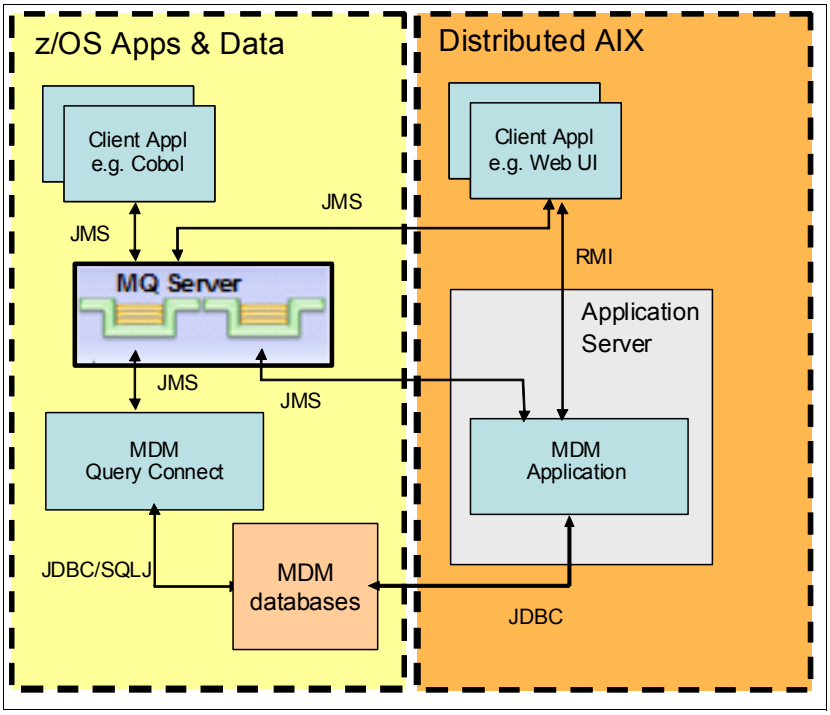


Figure 4-11 Client/server architecture

Now we examine how technology such as business rules and master data can be integrated in core business applications and data. Clients in the retail industry often rely on traditional applications for managing products. These applications integrate catalog capabilities and marketing capabilities such as pricing, promotions and eligibility conditions. This worked fine in the past, when the market was expecting no more than plain, unrelated products.

With new market conditions and more demanding clients, companies need to propose personalized product bundles. Figure 4-12 on page 46 shows a hybrid architecture, where simple products are still managed in the traditional applications while bundles are managed in new applications.

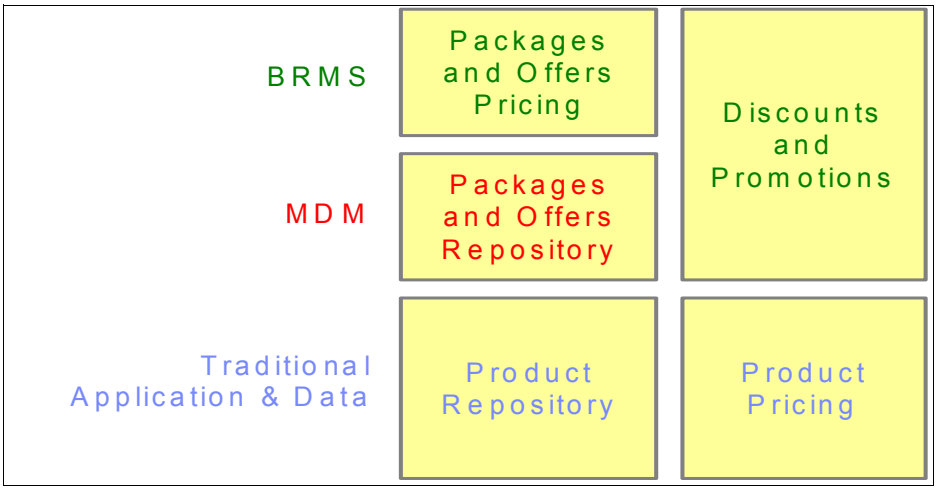


Figure 4-12 Hybrid architecture to manage old and new offerings

Bundles are defined in master data management software, but they are defined as assemblies of products managed and referenced in the traditional application. Pricing the bundles can be complex, and can depend on dimensions external to the catalog, such as the channel or the client segment. Therefore, the logic defining the price of each bundle and the eligibility conditions are managed in a business rule management system, and are de-correlated from the structure of the bundle, defined in the master data management software.

In many cases, the price of a bundle is calculated with the price of the products the bundle assembles, and uses the price list as it is managed in the traditional application.

4.4.5 Publishing business events

Business events occur all the time in the business world, but are not often analyzed. *Business rules* are conditions that, when analyzed, determine a result. In contrast, a *business event* refers to the occurrence of something, or the non-occurrence of something expected.

Although a business rule is called synchronously with a mandatory answer, an *event rule* is called asynchronously and might trigger an asynchronous answer. Events are generated and processed asynchronously in near real-time. The processing of an event is de-coupled from the computer operations that cause it to be emitted.

For many years, mainframe applications created events by building an event message in the application and using the MQ API to send it to any application capable of receiving MQ messages. It is now possible to emit events in formats suitable for consumption by WebSphere Business Events, WebSphere Business Monitor, and other consumers.

More recently CICS has provided an innovative solution for event processing. The CICS run time detects instances of events that are enabled and captures the events and payload without the need to make application code changes. When CICS captures events, it carries out specified filtering, enriches the event with information about the application context in which it occurred, formats the event, and routes it to the appropriate event consumer.

In IMS averments, an event message still has to be created inside the IMS application to be sent, either by MQ as before or through the SOAP protocol.

4.4.6 Accessing real-time scoring capabilities

Business growth is attained by attracting new customers and expanding the footprint with existing customers through a combination of cross-selling and up-selling new products and services. This growth must be balanced with the capability to reduce risk and deliver better customer service.

Maximizing growth while limiting exposure to risk requires insight into all customer interactions as they are happening so that today's interactions can be evaluated against historical behavioral patterns to determine the best course of action now. Today, many customers are exploring the possibility of scoring their data in real-time, but are unable to proceed due to the technical limitations of how the scoring of new transactional data must be handled.

Being dependent on a real-time web services call/response can result in much higher costs, not to mention the possible impact to service levels with network bandwidth limitations. By incorporating the scoring algorithms directly within the transactional application, you will have better insight into customer behavior at the point of interaction, at a price point you can afford.

A new real-time scoring technology embeds IBM SPSS Modeler scoring algorithms directly into IBM DB2 for z/OS transactional data, making the possibility of leveraging real-time scoring in your organization a viable option. The solution enables the real-time scoring of new transactional data as it is created. This results in decision-making with greater accuracy and efficiency, while ensuring that you can continue to deliver upon your service level agreements (SLAs).

Scores can be invoked by SQL statements and run in user-defined functions (UDFs) in DB2 for z/OS. Rather than extracting data from DB2 and sending it to a separate scoring engine on System z or another platform, data can be scored within the scope of a transaction accessing the DB2 primary data for faster and more accurate results. Lab measurements demonstrate the ability to drive 10,000 scoring transactions per second with 15 millisecond response times.

Moving computation to the OLTP data provides the following benefits:

- Reduces latency
- Minimizes data movement
- Enables more complex scores to be run within SLAs
- Reduces complexity
- Benefits from the high qualities of service of System z

4.4.7 Integrating local CPLEX mathematical algorithms

The standard in optimization software, technology, and solutions, CPLEX is now available on z/OS, enabling new mathematical optimization capabilities for mainframe applications.

CPLEX Optimizer for z/OS provides flexible, high-performance mathematical programming solvers for linear programming, mixed integer programming, quadratic programming, and quadratically constrained programming problems. CPLEX Optimizer for z/OS includes interfaces in the C and C++ programming languages.

This offering provides greater end-to-end performance, reliability, simplified architecture, and maintenance requirements as opposed to applications deployed in a distributed environment.

Finance, health care, and government clients can:

- Maintain the security and reliability of data for optimization applications within the mainframe environment.
- Reduce IT and software maintenance costs through simplified architecture.
- Gain greater end-to-end performance by running optimization models within z/OS directly from their core business applications.

4.5 End-to-end application infrastructure extensibility

This path is one that has been considered by many clients for years through diverse service-oriented architecture implementation. The focus has mainly been on reusing existing CICS and IMS programs as services in other applications. In this section we discuss how to perform the following tasks:

- Integrate CICS and IMS programs as services in a service-oriented architecture
- Augment the integration architecture with business process management
- Expand the multi-channel architecture and integrate the mobile channel
- Access mainframe data directly

4.5.1 Integrating CICS and IMS programs as services in an SOA

The components of an SOA implementation provide for communication, governance, data management, application management and non-functional services. Services are either consumed or provided by the software middleware such as CICS, IMS, WebSphere Application Server, and DB2. The method to find and call these services is provided by a bus or a direct connection that provides the communication mechanism between consumer and provider.

Coupling is the term used to describe the communication channel such that we can determine whether there is a close integration or distant integration. The terms commonly used are *loosely* coupled or *tightly* coupled. SOA exploits a loosely coupled environment to provide maximum flexibility and reuse. Tighter coupling is provided in point-to-point configurations, with both ends responsible for communication standardization.

The key benefit of loose coupling is agility. The service layer helps to isolate change in the providing or requesting applications. For example, the frequency of change in the business process, or the device and technology diversity used in the requesting application, might be more frequent than that in the core back-end systems that are used by the service provider.

In the following sections we provide brief explanations of the technologies that can come into play when integrating CICS and IMS programs as services. These technologies are:

- Enterprise Service Bus (ESB)
- Connectors for IMS
- Connectors for CICS

Enterprise Service Bus

At the core of the SOA reference architecture is the Enterprise Service Bus (ESB)[2]. This architectural construct delivers all the interconnectivity capabilities required to use and reuse services implemented across the entire architecture.

[2] Refer to *SOA Transition Scenarios*, SG24-7331

The ESB provides the following fundamental services:

- Transport services that provide the fundamental connection layer
- Event services that allow the system to respond to specific events that are part of a business process
- Mediation services, such as transformation and validation services, that allow loose coupling between interacting services in the system

The Enterprise Service Bus (ESB) is a “silent partner” in the SOA logical architecture. Its presence in the architecture is transparent to the services of your SOA application. However, the presence of an ESB is fundamental to simplifying the task of invoking services. It makes use of services wherever they are needed, independent of the details of locating those services and transporting service requests across the network to invoke those services wherever they reside within your enterprise. Figure 4-13 shows a simplified diagram of an ESB.

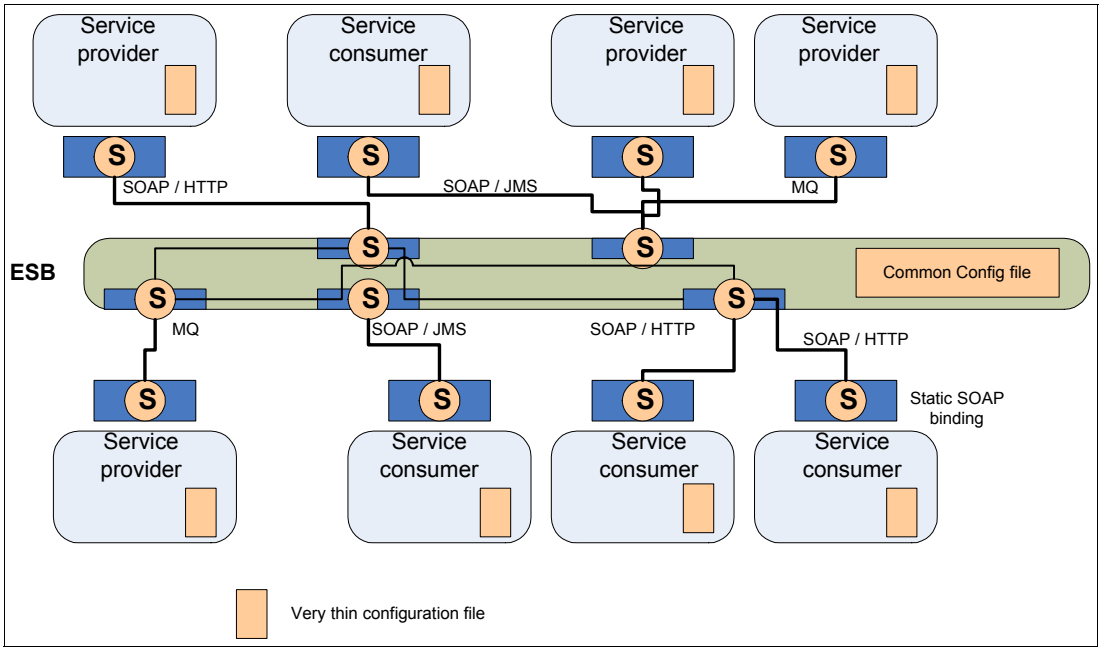


Figure 4-13 ESB connecting service providers and consumers

IBM ESB solutions are evolving based on changing market needs. The main efforts are focused today on WebSphere Message Broker and WebSphere DataPower. These both provide integration with CICS, IMS, WebSphere Application Server, and DB2 on z/OS.

Overview of integration technologies for IMS Transaction Manager

IMS Transaction Manager is architected to act as a participant in an enterprise SOA implementation, both from a client perspective and a provider perspective.

This is why IMS has provided the following capabilities to act as a server and as a client:

CALL_IN IMS transactions can be driven by business logic running in many different environments including Java EE application containers, web services container, DB2 stored procedures, Microsoft .NET server, and a variety of other vendor solutions or client-created applications.

CALL_OUT IMS transactions can call business logic outside the IMS environment and on any platform using many communication standards including SOAP, WebSphere MQ, and JCA.

For access to and from IMS transactions, IMS provides the IMS SOA Integration Suite, which is a collection of IMS middleware functions and tools that support both the IMS environment and the distributed client application environment. The most current versions of some IMS SOA Integration Suite tools and IMS SOAP Gateway are included as components of the IMS Enterprise Suite available at no-cost for unlimited installs (Product ID: 5655-T60, S&S 5655-T61). Refer to the following URL for more information:

http://www-01.ibm.com/software/data/ims/soa-integration-suite/

Figure 4-14 shows an overview of all integration mechanisms for IMS Transaction Manager in the context of an SOA architecture.

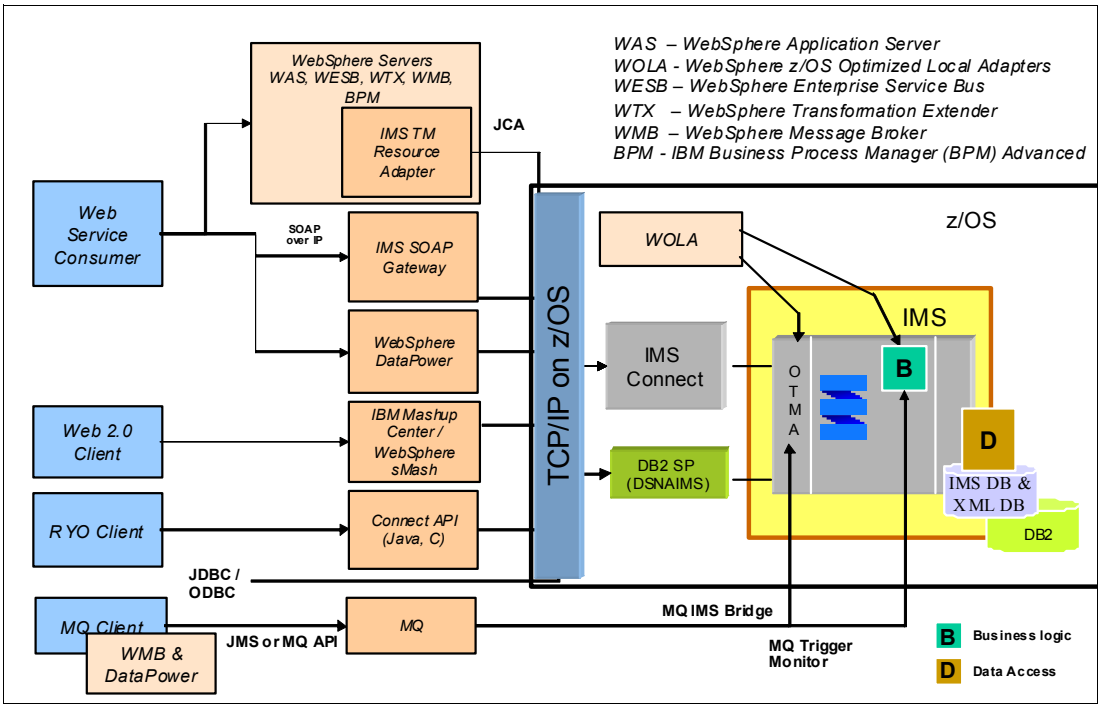


Figure 4-14 IMS integration overview

In summary:

- IMS TM Resource Adapter (ITRA), formerly known as the IMS Connector for Java, is a component of IMS that allows rapid development and deployment of WebSphere Application Server components (Java) that access IMS transactions over TCP/IP. Call_Out capability has been added r to allow an IMS transaction to call a WebSphere Application Server component through this JCA connector.
- IMS SOAP Gateway is a solution that integrates IMS into the distributed world using SOAP/XML-based connectivity. SOAP protocols can be used for call_in and call_out access, enabling remote servers to access IMS transactions and providing existing or new IMS applications with the ability to access remote services.
- WebSphere DataPower is an appliance used as ESB to access IMS transactions, using either WebSphere MQ or IMS Connect.
- IBM Web 2.0 can be used to access IMS transactions and data.

- Messaging-style access to IMS transactions is possible using WebSphere MQ through the MQ IMS Bridge or the MQ Trigger Monitor.

WebSphere Message Broker and its IMS nodes can be used to route requests to IMS based on either WebSphere MQ or IMS Connect.

Overview of integration technologies for CICS

CICS is designed to participate in an SOA architecture in different ways. Inbound to CICS, that is, using CICS as a service provider, can be implemented in two ways:

- With the service endpoint inside CICS and accessing this service endpoint directly using Web Services from any other application in the enterprise.

  Either WebSphere MQ or HTTP can be chosen as the transport layer for these Web Services.

- With the service endpoint outside CICS, for example in WebSphere Application Server or an ESB, and accessing the CICS program using any of the supported communication protocols or CICS connectors.

  The communication protocols are WebSphere MQ, TCP/IP, Telnet and HTTP. Connectors supported are IP interconnectivity (IPIC) and JEE Connector Architecture (JCA). JCA is the underlying connector for the CICS Transaction Gateway when used by a JEE client application and for the WebSphere Optimized Local Adapter (WOLA) on z/OS.

Outbound from CICS, that is, using CICS as a service requester, is supported as well by the CICS Web Services support.

Figure 4-15 on page 52 shows an overview of all possible integration mechanisms, for both inbound and outbound.

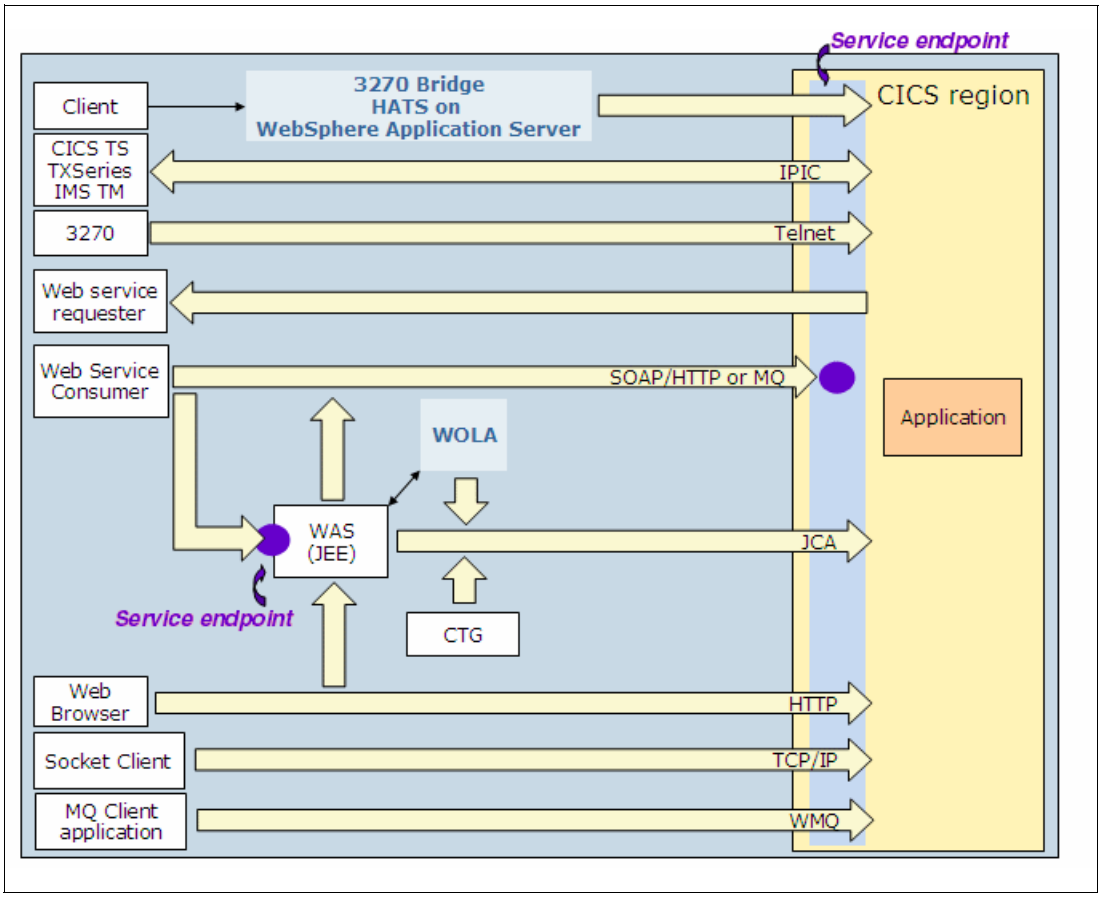


Figure 4-15 CICS integration overview

We now briefly discuss the integration technologies that are typically used in an SOA architecture

CICS Transaction Gateway

The *CICS Transaction Gateway* (CICS TG) is a set of client and server software components that allow a remote client application to invoke a program in a CICS region. The client application can be either a Java application or a non-Java application using either C, C++, workstation COBOL or.NET interfaces (depending on the platform used). With IBM CICS Transaction Gateway, you can use your CICS applications in comprehensive and sophisticated Java and Web services solutions hosted in a JEE server or Enterprise Service Bus.

CICS Web Services Support

Application programs running in CICS can participate in a heterogeneous Web Services environment as service requesters, service providers, or both. CICS Web Services Support is a function of CICS that enables both inbound and outbound Web Services.

Figure 4-16 on page 53 shows an outline of the Web Services Support in CICS.

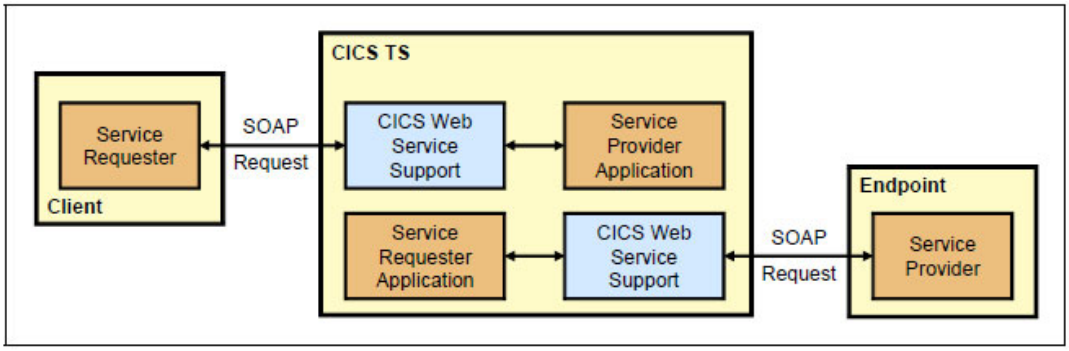


Figure 4-16 CICS Web Services Support

4.5.2 Augmenting the integration architecture with business process management

We discussed business process management in 1.4, “Business process management” on page 4. Here is a brief explanation of how a business process management (BPM) approach can be added to an existing CICS or IMS environment.

Business process management relies on available IT services. These IT services provide an interface to the application components providing certain functions required by the business process. Therefore it is useful to first make IT services available and then implement the business process using these services. As described in previous sections, there are excellent ways to make existing and new CICS and IMS programs available as easy-to-consume IT services, so these services can be directly incorporated in the business process.

4.5.3 Expanding the multichannel architecture and integrating the mobile channel

By leveraging the millions of lines of code in enterprise mainframe business applications and working processes, corporations can become even more competitive and strategic with the applications that they build. By moving from first-generation applications, through the web, and now into increasingly sophisticated mobile channels, an organization should be able to deliver even more services to both internal and external customers.

With the advent of web and REST services, mainframe transaction managers such as CICS, IMS, and WebSphere Application Server can be excellent places to host services and processes that mobile front-ends invoke. Thus, the transaction managers can provide both information and business value. Often these processes already exist, but now need to be linked into the new open infrastructures that are available to the mobile platform.

Integration between mobile devices and existing or new CICS and IMS programs can be achieved in different ways. Because many companies have already built a multichannel architecture, it makes sense to extend that architecture for mobile devices instead of developing a completely new channel architecture just for mobile or a separate channel for each variation of mobile device (iPhone, Android smartphone, Android tablet, or Windows phone).

Although a entire book could be written about integrating mobile with back-end CICS and IMS applications[3], any solution strategy can condense to the following requirements:

- The ability to serve a variation of mobile devices, using different window sizes, different operating systems, and thus different user interface protocols.

- Business logic and data access logic should only be written once and then reused by all mobile devices.
- Communication between the mobile device and the first entry point on the server must be secure when needed, with support for SSL and VPN, for example.
- Availability of adapters or connectors for integration with the existing application landscape, including CICS and IMS.

IBM Worklight

IBM Worklight provides a solution for developing, managing and running mobile applications. Applications can be developed using IBM Worklight Studio and deployed to IBM Worklight Server. Adapters are available to communicate between Worklight Server and back-end systems and databases. Generally, any back-end application that can handle Web Services or HTTP and any database that can handle SQL can be accessed through an adapter. A preferred practice could be to use an Enterprise Service Bus for integration between Worklight and the back-end application.

By implementing an architecture based on Worklight, the dynamic character of the mobile world is isolated from the secure and robust CICS and IMS back-ends, which is where the real business logic and data access keeps on running.

Figure 4-17 illustrates a Worklight server implementation in which data is retrieved from a back-end zEC12 mainframe.

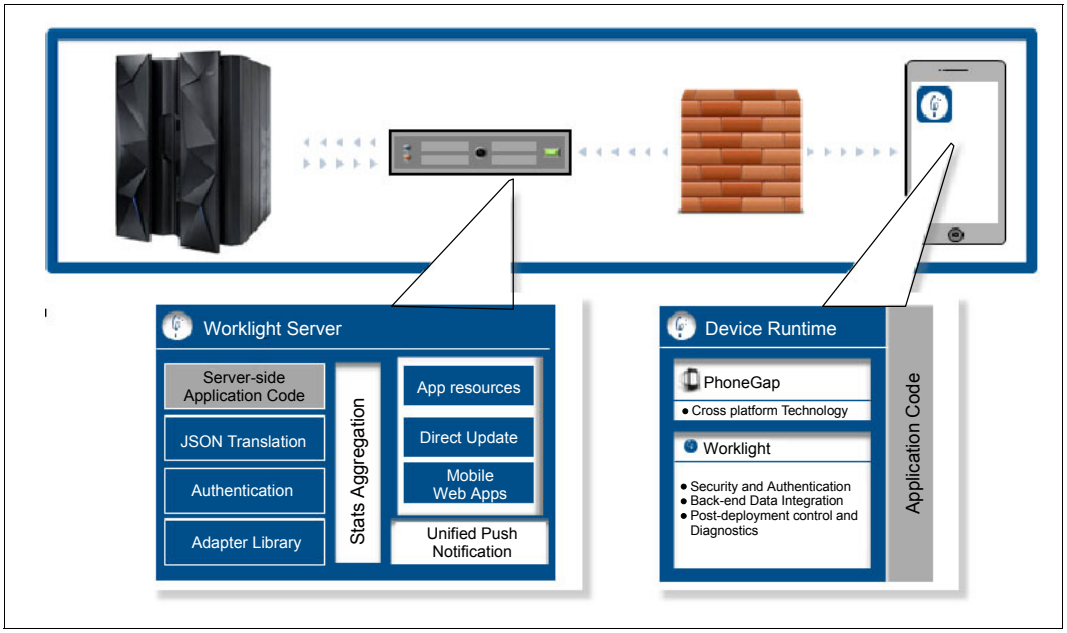


Figure 4-17 Worklight Server architecture

For more information about this topic, also see an IBM developerWorks® article by Leigh Williamson.[4]

[3] An IBM Redbooks publication is available on this topic, though technologies have evolved since then. See http://www.redbooks.ibm.com/abstracts/sg247836.html

[4] http://www.ibm.com/developerworks/rational/library/enterprise-teams-mobile-application-projects/index.html#author1

4.5.4 Accessing mainframe data directly

The universal standard in Java EE architecture to access databases is JDBC. This standard has been supported by relational datastores like DB2 UDB and DB2 for z/OS for years.

However, besides for relational databases, JDBC can also be used to access IMS databases. Access to IMS databases using “JDBC SQL calls” can be made from the following environments:

- Applications written in Java and running in IMS, CICS, or as DB2 stored procedures
- Applications running in WebSphere Application Server for z/OS
- Applications using the IMS Universal driver from any distributed environment that supports the open standard IBM Distributed Relational Database Architecture™ (DRDA)
- InfoSphere Classic Federation Server, which hides the specifics of z/OS mainframe data sources from the application developer

Beginning with IMS 11, IMS supports the Distributed Data Management (DDM) architecture of DRDA and offers IMS Universal drivers that include:

- IMS Universal DB resource adapter: a Java EE Connector Architecture (JCA) 1.5-compliant resource adapter
- IMS Universal JDBC driver: a Java Database Connectivity (JDBC) driver that implements the JDBC 3.0 API
- IMS Universal DL/I driver: a Java API for making calls with traditional DL/I programming semantics

You can now imagine new ways of integrating your IMS databases in your overall architecture.

5

Summary of reasons and approaches for extension, transformation, and growth

Three questions of importance to users of IBM and CICS on the System z mainframe are answered in this paper:

- Why transform existing IBM mainframe applications?
- What are the benefits of using IBM mainframe environments today?
- What are the key strategies for the transformation and extensibility of applications?

The chapters in this paper address those questions:

- Chapter 1. Business and IT trends and how they affect existing applications
- Chapter 2. Attributes of an agile IT architecture
- Chapter 3. The way today's mainframe applications are implemented
- Chapter 4. Evolution paths to achieve more value

5.1 Why transformation of mainframe applications is needed

Chapter 1, “Business and IT trends and how they affect existing applications” on page 1 and Chapter 2, “Attributes of an agile IT architecture” on page 11, explain why such transformation is necessary. The business needs driving change are risk, agility, cost, and functionality. In a poll of IT decision makers, these factors were key in driving application transformation:

- Business Intelligence and analytics
- Mobility solutions
- Business process management
- Risk management and compliance
- Driving down IT cost while gaining flexibility

The paper examines topics focused on whether the mainframe can handle twenty-first century computing, namely flexibility, integration, skills, and cost. The authors provided answers for each topic based on experiences with clients.

5.2 Benefits provided by IBM mainframes today

Chapter 3, “The way today’s mainframe applications are implemented” on page 15, highlights the qualities that have been part of the mainframe environment since the IBM S/360 was shipped, and which can be condensed into the phrase “qualities of service.” The z/OS operating system and mainframe applications provide the following benefits:

- z/OS-related benefits:
  - Advanced workload management
  - Centralized operations and integrated systems management
  - Non-disruptive change management
  - System integrity and security
  - Virtualization and scalability
- Application-related benefits:
  - Transaction processing
  - Interoperability with distributed systems
  - Colocation of application components
  - A single version of the truth

The System z mainframe has been providing high quality, high availability service for decades, which is why two-thirds of the business process worldwide runs on mainframes.

5.3 Strategies for transformation

Chapter 4, “Evolution paths to achieve more value” on page 25, discusses several key strategies available to bring your business applications into the twenty-first century. These strategies include ways to extend BPM, Business Rules, and Portal in WebSphere on z/OS or Linux on System z to meet new business requirements.

We first explain and detail the concept of *application maturity levels*. Then we describe four strategies you can use to extend and transform your applications to handle your current and future business needs:

- Application runtime optimization:
  - Running regular system health checks
  - Implementing agile development and test environments
  - Optimizing the batch window
  - Benefitting from a highly available infrastructure
- Application development foundation modernization:
  - Using the latest compilers
  - Using a single integrated development environment
  - Benefitting from an end-to-end view of development
- Business application modernization:
  - Application restructuring
  - Call out to services running in WebSphere Application Server on z/OS
  - Integrating external business rules
  - Incorporating direct access to master data management services
  - Publishing business events
  - Accessing real-time scoring capabilities
  - Integrating local CPLEX mathematical algorithms
- End-to-end application infrastructure modernization:
  - Integrating CICS and IMS programs as services in a service-oriented architecture
  - Augmenting the integration architecture with business process management
  - Expanding the multi-channel architecture and integrate the mobile channel
  - Accessing mainframe data directly

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this paper.

IBM Redbooks

The following IBM Redbooks publications provide additional information about the topic in this document. Note that some publications referenced in this list might be available in softcopy only.

- *Combining Business Process Management and Enterprise Architecture for Better Business Outcomes,* SG24-7947
- *Batch Modernization on z/OS,* SG24-7779
- *System z Mean Time to Recovery Best Practices*, SG24-7816
- *SOA Transition scenarios*, SG24-7331
- *System z on the Go: Accessing z/OS from Smartphones*, SG24-7836

You can search for, view, download or order these documents and other Redbooks, Redpapers, Web Docs, draft and additional materials, at the following website:

ibm.com/redbooks

Other publications

These publications are also relevant as further information sources:

- *IMS V12 Application Programming APIs*, SC19-3008
- *CICS Transaction Server for z/OS V4.2: What's New*, SC34-7192

Online resources

These websites are also relevant as further information sources:

- IMS SOA Integration Suite

  http://www-01.ibm.com/software/data/ims/soa-integration-suite/
- *Cloud computing with Linux* by Tim Jones

  http://www.ibm.com/developerworks/linux/library/l-cloud-computing
- A search of the IBM.com website for IMS modernization

  http://www.ibm.com/Search/?q=ims+modernization&v=17&en=utf&lang=en&cc=us
- A search of the IBM.com website for CICS modernization

  http://www.ibm.com/Search/?q=cics+modernization&co=us&lo=any&ibm-submit.x=9&ibm-submit.y=9&sn=&lang=en&cc=US&en=utf&hpp=

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Rethink Your Mainframe Applications
Reasons and Approaches for Extension, Transformation, and Growth

Redpaper™

Why to transform existing IBM mainframe applications

Benefits of IBM mainframe environments today

Strategies for transformation

Today there are new and exciting possibilities available to you for creating a robust IT landscape. Such possibilities include those that can move current IT assets into the twenty-first century, while supporting state-of-the-art new applications. With advancements in software, hardware and networks, old and new applications can be integrated into a seamless IT landscape.

Mobile devices are growing at exponential rates and will require access to data across the current and new application suites through new channels. Cloud computing is the new paradigm, featuring anything from SaaS to full server deployment. And although some environments are trying to virtualize and secure themselves, others such as IBM zEnterprise have been at the forefront even before cloud computing entered the scene.

This IBM Redpaper publication discusses how transformation and extensibility can let you keep core business logic in IBM IMS and IBM CICS, and extend BPM, Business Rules and Portal in IBM WebSphere on IBM z/OS or Linux on IBM System z to meet new business requirements. The audience for this paper includes mainframe architects and consultants.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

REDP-4938-00