



Service-Enabling the Mainframe:

*Creating reusable business services
from existing mainframe assets*

IONA, IONA Technologies, the IONA logo, Orbix, Orbacus, Artix, Orchestrator, Mobile Orchestrator, Enterprise Integrator, and Adaptive Runtime Technology are trademarks or registered trademarks of IONA Technologies PLC and/or its subsidiaries. Java and J2EE are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. CORBA is a trademark or registered trademark of the Object Management Group, Inc. in the United States and other countries. All other trademarks that appear herein are the property of their respective owners.

While the information in this publication is believed to be accurate, IONA Technologies PLC makes no warranty of any kind to this material including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. IONA Technologies PLC shall not be liable for errors contained herein, or for incidental or consequential damages in connection with the furnishing, performance or use of this material.

COPYRIGHT NOTICE

No part of this publication may be reproduced, stored in a retrieval system or transmitted, in any form or by any means, photocopying, recording or otherwise, without prior written consent of IONA Technologies PLC. No third-party intellectual property right liability is assumed with respect to the use of the information contained herein. IONA Technologies PLC assumes no responsibility for errors or omissions contained in this white paper. This publication and features described herein are subject to change without notice.

Copyright © 1999-2004 IONA Technologies PLC. All rights reserved.

All products or services mentioned in this white paper are covered by the trademarks, service marks, or product names as designated by the companies that market those products.

Table of Contents

1	Introduction	1
2	The Need for a New Approach to Mainframe Integration	2
2.1	Current Mainframe Integration Approaches	2
2.2	Service-Enabling the Mainframe	3
2.3	The Benefits of Service-Oriented Mainframe Integration	5
3	IONA Solutions for the Mainframe	7
3.1	Artix Mainframe Transformer—A Mainframe-Hosted Adapter	7
3.2	Artix Encompass—A Service-Oriented Gateway	9
4	Conclusion	11

1 Introduction

Mainframe environments are undergoing a phase of quiet resurgence. Despite the turbulent and often disruptive shifts in enterprise computing trends between centralized and distributed computing over the past decade, mainframes have maintained their core IT position within companies. They have done this by providing the highest qualities of service, continuous availability, transaction integrity and comprehensive security.

Companies are understandably reluctant to abandon systems in which they have made significant hardware, software and human investment. Mainframes have established themselves by performing certain functions—large-volume business transactions in particular—better than any other system. Many large corporations have built their business processes around the mainframe and consider the mainframe integral to their organizational culture.

Client environments have also undergone dramatic change over the past decade, from character terminals to high-end bit-mapped terminals with higher resolutions and better image definitions. Changes in the client environment have been welcomed by the industry, and have led to the proliferation of more visually appealing client applications. The rise of the Internet has also had a major impact on client environments, and has resulted in the creation of applications that are more customer self-service focused. Mainframe business processes, however, continue to be the IT foundation on which companies build their businesses.

Industry trends and technology diversity make it difficult to realize returns on the investments made in mainframe systems. Over the past decade, IONA has been a leader in service-oriented mainframe integration products that enable companies to seamlessly integrate mainframe assets with non-mainframe-based applications in a standards-compliant manner. Orbix Mainframe has been a very successful offering for the past decade, and has been deployed in many large mission-critical mainframe environments around the world.

These large-scale customer deployments have delivered massive returns on investment to banking and finance customers (such as Credit Suisse, Winterthur, and Zurich Insurance), and to the leading US telecom companies.

IONA's Artix family of service-oriented integration products builds on this expertise. It offers both mainframe native and off-host solutions to extend CICS and IMS application assets, written in COBOL and PL/I, to other applications within the enterprise in a secure, reliable and non-invasive manner.

This paper discusses the value of service-oriented integration in mainframe environments. It outlines how IONA's Artix family of products facilitates the creation of service-oriented architectures (SOAs) where mainframes are the key building blocks in the process.

IONA provides service-oriented mainframe solutions to some of the largest companies in the world, processing tens of millions of transactions every day. This paper outlines why service-oriented integration makes sense for extending host-based processes to the rest of the enterprise.

2 The Need for a New Approach to Mainframe Integration

Companies are finding themselves prisoners of the same technology that originally promised to lower the overall cost of maintenance and create highly reusable IT assets. Integration issues have tended to be addressed at departmental levels with tactical, ad hoc and proprietary approaches. Old and aging architectures, non-scalable and restrictive technologies, and the uneven adoption of standards by vendors has led to ever-increasing costs and not enough return on investment (ROI).

The resulting inflexible architectures hinder cost-saving initiatives such as customer self-service portals and middle-tier server consolidation. To make matters worse, many integration vendors add thin support for new technologies, such as Web services, and advocate a highly invasive “rip and replace” approach to implementing them. This increases the risks and costs associated with migrating to these new technologies.

The rip and replace approach is fundamentally at odds with the concepts of service-oriented integration—one that exploits open standards to create technology independent and reusable interfaces to high-level business processes. Service-oriented integration allows IT applications to more closely approximate actual business processes without building in technology dependencies that later prove hard to change. The result is a lean, agile and customer-focused IT architecture.

Service-oriented architectures (SOAs) are based on *well-defined interfaces* that hide service implementation details. These interfaces form contracts that clients can use for programmatic access to mainframe assets. Meanwhile service implementations can change without disruption. The result is system-level components that can be quickly reconfigured to react to new requirements—the essence of business agility.

2.1 Current Mainframe Integration Approaches

Compared with the continuously evolving ecosystem of enterprise applications, mainframe environments have been islands of stability. However, mainframe-based applications have not been immune to the disruptive changes in the enterprise computing fabric.

Mainframe applications are based on centralized computing. They are the primary hosting environments surrounded by a collection of thin display-centric clients tightly coupled to the applications. Mainframe applications are typically coarse-grained, program/task driven, monolithic and built using older languages such as COBOL and PL/I.

Non-mainframe applications based on distributed computing take the opposite approach. These applications tend to be expressed as a loose collection of independent business components that can be hosted on diverse platforms and co-operate to perform business functions. Application components tend to be more fine grained and are viewed more as building blocks in an overall process. In addition, there is typically a sharp separation between business logic and presentation logic. Business logic components tend to be agnostic of client environments and can interoperate with both rich clients as well as Internet-based thin clients (browsers).

The disparity in application models leads to several problems when integrating mainframe applications with applications built using distributed computing models such as CORBA, J2EE, and so on. Questions arise about the integration approach, location, deployment model and administration. In most cases, these have been addressed by proprietary approaches and non-standard and expensive message/data-centric integration solutions.

Message-centric integration solutions further drive mainframe applications and business components towards stove-piped behavior, resulting in application islands or silos that display rigid operational behavior. It also creates operational dependencies among applications, which results in a resistance to change and leads to issues cascading through applications that depend on specific message formats and types.

This results in unanticipated and recurring maintenance costs and orchestration/integration offerings. IT budgets are directed towards a brittle integration layer that is not designed to be agile, adaptable and service-contract based.

2.2 Service-Enabling the Mainframe

Without a service-level contract to standardize the relationship between on-host and off-host business components, there is no formality to their integration. This is akin to entering a business agreement without specifying the terms of that agreement; simply relying on the professional ethics and discipline of all involved.

The lack of a standard service-level agreement or contract between mainframe applications and non-mainframe applications requires that they informally “agree” on the message or data exchanged between applications. This leads to tightly coupled sets of applications that are hard to maintain when data/message changes are mandated to address new business requirements. Every application that uses the data/message must be updated to understand the new format.

The solution to this maintenance quagmire is to define an interface that is technology independent and unchanging. Service-oriented integration requires a service-level contract to specify the handshake between off-host and mainframe application assets. Even if the service *implementation* changes, the service *contract* does not. This means clients can continue to invoke services without disruption as long as they continue to adhere to the terms of the service contract.

With service-oriented integration:

- Policies, standards and integration choices are declared and enforced by *service contracts*.
- Enterprises renovate existing IT assets, consolidate expensive middle tiers and channel IT budgets towards innovation rather than maintenance.
- Best-of-breed applications provide service-based interfaces so they can be introduced in a phased and non-disruptive manner.
- Applications can be quickly switched from tactical to more strategic, cost-efficient or future-focused technology choices.

Service contracts are a fundamental requirement in implementing a Service-Oriented Architecture (SOA).

A service-oriented approach creates a solid integration foundation based on an enterprise service contract model. With SOAs, applications and application components are encapsulated as services. Each service declaratively exposes a service contract that defines the valid operations, payload, access rules, security, qualities of service requirements, and a choice of protocol/transport required to communicate with the specific operations exposed by the service.

At its fullest expression, a service-oriented approach to integration exploits the reusable assets created by a SOA—assets that can be non-invasively renovated by simply modifying the terms of the service contract. Service contracts make mainframe programs and business components independent of languages, operating systems and networks. Service interfaces equate to business processes whose implementation can be migrated, rationalized and consolidated across platforms, protocols and specification islands by making the appropriate changes to the service contract.

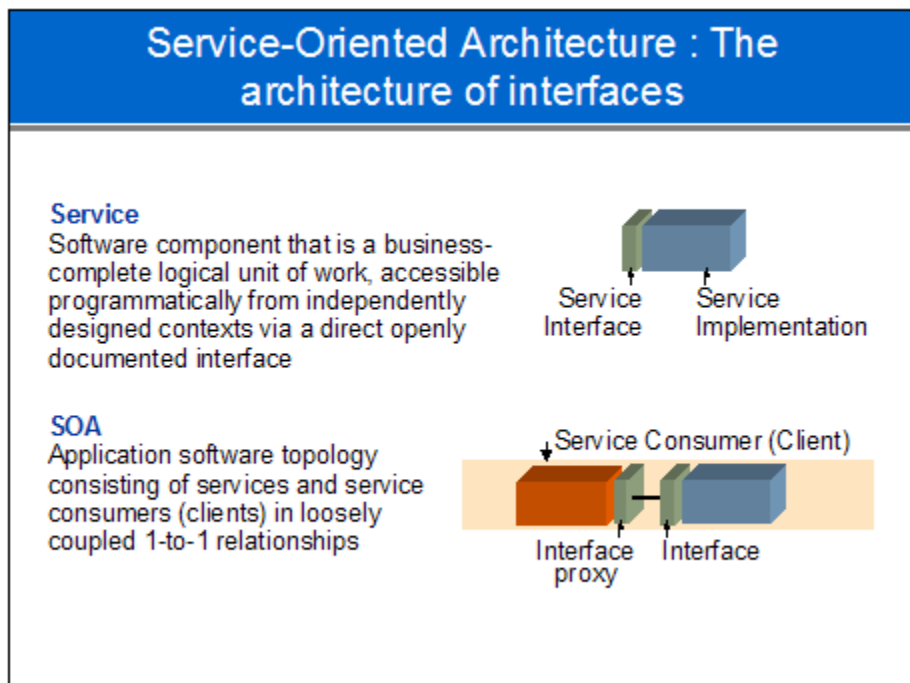


Figure 1: Service-Oriented Architecture

SOAs can show significant cost savings over time in terms of hardware, software and labor by simplifying application renovation, facilitating server consolidation and reducing tightly coupled applications. In addition, SOAs can future-proof a company's IT investment by enabling the independent evolution of mainframe assets (without impacting the rest of the company). This further drives vendor, specification and technology neutrality—everything works together with less emphasis on the specifics of how the work gets done.

“Service-oriented architecture (SOA) is described as the next-generation all-encompassing software architecture for the realtime enterprise.”

*Massimo Pezzini,
Gartner, 2003*

2.3 The Benefits of Service-Oriented Mainframe Integration

IONA enables service-oriented integration for host-based assets with products that securely extend IMS and CICS transactions to the enterprise, without disrupting production applications. IONA provides two configurations to conform to the preferences of IT management.

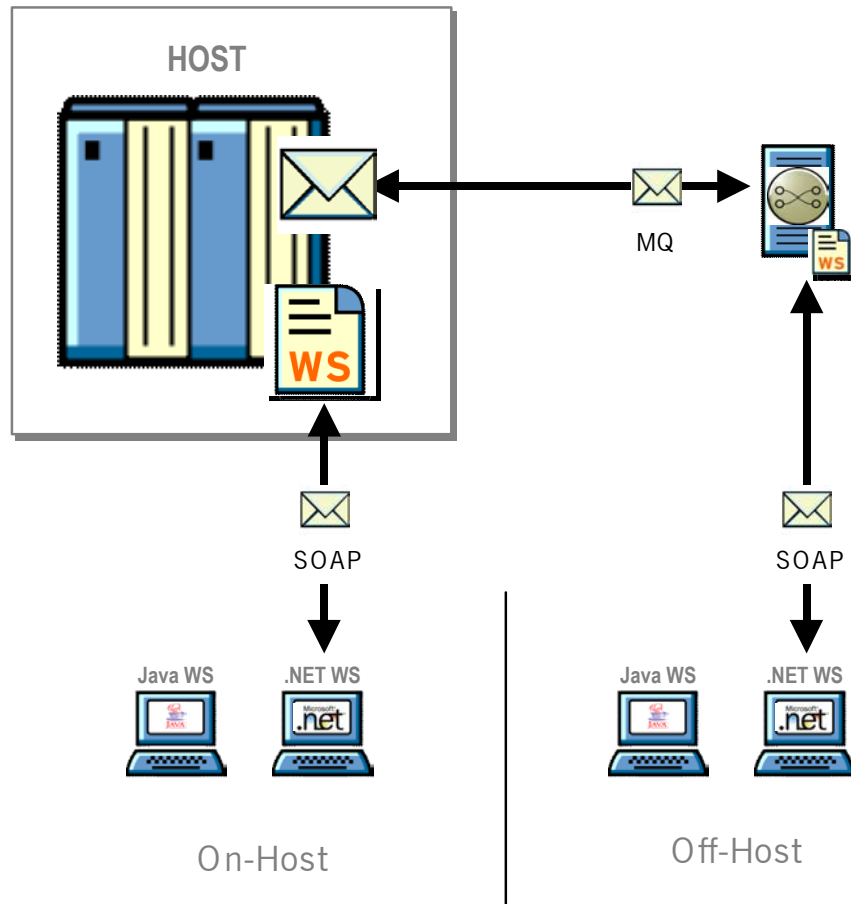


Figure 2: Two Solutions for Service-Oriented Mainframe Integration

Artix Mainframe runs directly on the host, while Artix Encompass runs off-host. Both provide the service-oriented integration capabilities that enable companies to extract the following critical business value from their existing mainframe assets and investments:

- **Better return on investment (ROI)**—The creation of robust service contracts ensures a better ROI for existing IT infrastructures. No changes need to be made to the applications, and the service can be enabled non-invasively. The service contract becomes a template for renovation, consolidation and innovation.

- **Code mobility and location transparency**—Because location transparency is a property of SOAs, code mobility is easily accomplished. The ability to look up and dynamically bind to a service allows service requestor and service consumer applications to be location independent. This gives enterprises the flexibility to migrate services to different machines, or to move a service to an external provider (business process outsourcing).
- **Localization of application issues and problems**—Service contracts have well-defined, published interfaces. This allows issues and problems associated with applications to be easily isolated and localized. Applications can be independently tested and brought down for scheduled maintenance, without causing a problem to cascade through the enterprise.
- **Support for multiple client types**—Because a service contract defines the entry point for access to a service, multiple client types (.NET, CORBA, J2EE, and mainframe clients) can invoke on the service using their native protocols and get information in the format that they understand.
- **More code, component and service reuse**—Artix Encompass and Artix Mainframe both facilitate code, component and service reuse through service enablement of legacy infrastructures. By abstracting the idiosyncrasies associated with language, platforms and networks to a service contract, application assets can be relocated, refactored and reused. Competitive differentiators based on the quality of application assets can be selectively preserved, versioned and maintained.
- **Better scalability**—One of the requirements of a SOA is location transparency. To achieve location transparency, applications look up services in a directory and bind to them dynamically at runtime. This feature promotes scalability because requests can be load balanced and forwarded to multiple service instances based on rules specified in the service contract and applied without the knowledge of the service client.
- **Higher availability**—Also because of location transparency, multiple servers can have multiple instances of a service running on them. If a network segment or a machine goes down, the Artix runtime dispatcher can reroute and send requests to another service location without the client's knowledge.

3 IONA Solutions for the Mainframe

3.1 Artix Mainframe Transformer—A Mainframe-Hosted Adapter

Artix Mainframe Transformer is a mainframe-based product that facilitates non-invasive service enablement of mainframe applications. Artix Mainframe Transformer consists of two primary components: The Artix Mainframe Transformer Service and the Artix Mainframe Designer.

The Artix Mainframe Transformer Service is a task running on-host that manages service contracts, service invocation and transaction initiation. The Artix Mainframe Designer is a Windows-based GUI tool that reads copybooks named in the linkage sections of programs to create service contract definitions expressed as Web Service Definition Language (WSDL). These service contracts contain the details of how non-mainframe-based applications can communicate with service-enabled IMS or CICS transactions.

The Artix Mainframe Designer generates information used by the Artix Mainframe Transformer Service at runtime to map copybook data structures to and from the equivalent SOAP XML-based message format for Web services. Simply transferring the runtime information to the host makes it available to the Artix Mainframe Transformer Service and thereby deploys the new service.

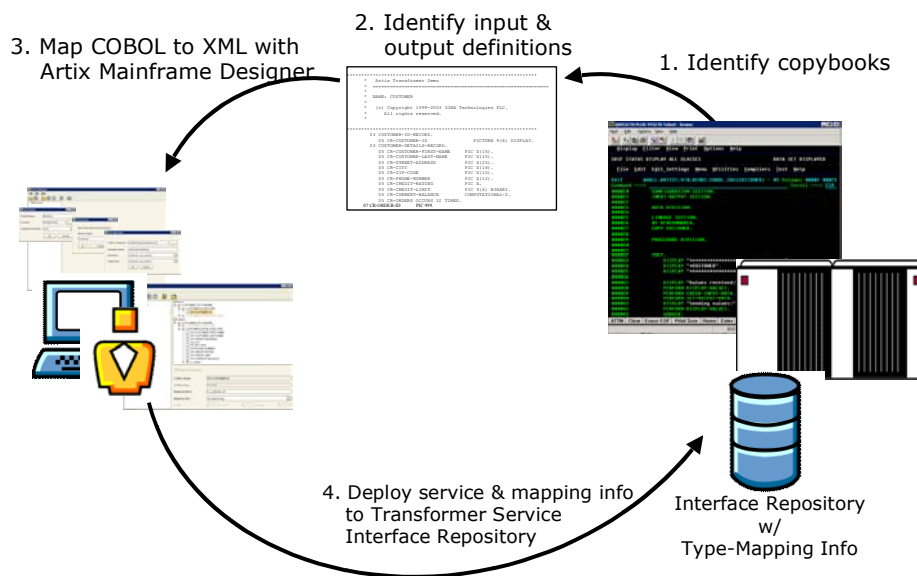


Figure 3: Service Development and Deployment

Once the service is defined and deployed on the host, the Artix Mainframe Transformer Service manages it. Client application developers can use tools such as Microsoft's .NET Framework, BEA's WebLogic Workshop, or IONA's Artix Encompass Designer to request the service contract (expressed as WSDL) from the Artix Mainframe Transformer Service. These tools can use the WSDL service definition to quickly create client applications that invoke the service. When the service is invoked, the Artix Mainframe Transformer translates the incoming XML payload into the original copybook data structure format, then starts the appropriate IMS or CICS transaction. When the transaction is complete, control is transferred back to the Artix Mainframe Transformer Service, which translates the outbound copybook data structure to the SOAP XML message format expected by the service requestor.

The Artix Mainframe Transformer runs on the mainframe in its own address space. This enables the process to be monitored and administered like any other mainframe process.

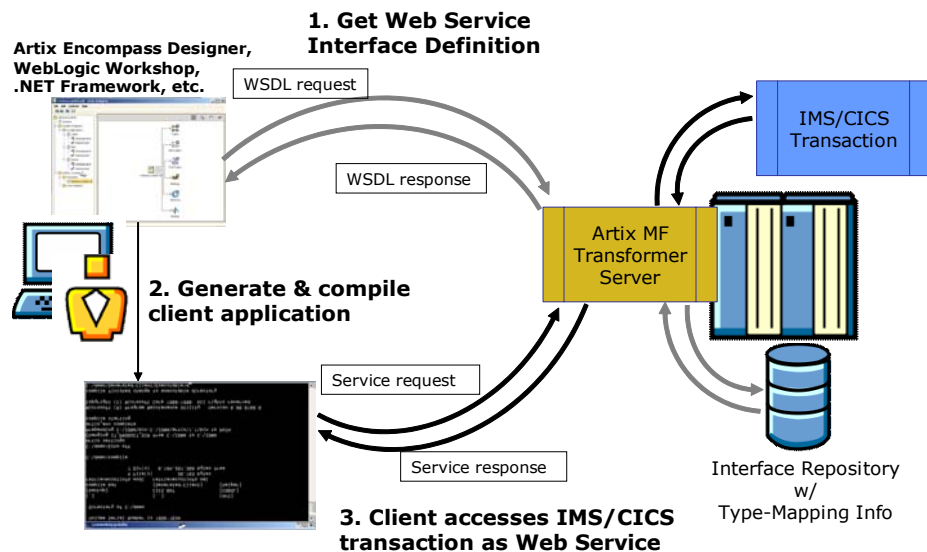


Figure 4: Client Development Process Using the Deployed Service

The service requestor and the Artix Mainframe Transformer Service exchange messages in the form of a standard SOAP payload. The Artix Mainframe Transformer Service is built on top of IONA's pluggable Adaptive Runtime Technology (ART) architecture. This modular architecture allows the dynamic selection of a wide range of protocols and transports (HTTP, MQ, and so on) that can be specified within the service definition. This means an existing service implementation can be changed, or new protocols and message formats can be added, without affecting existing clients or host transactions.

The Artix Mainframe Transformer Service is highly optimized to reduce CPU cycles on a per transaction basis. This enables the product to scale to the needs of extremely demanding mainframe environments.

What is especially empowering about Artix Mainframe Transformer is that service requestors can be a CORBA, J2EE, .NET or packaged application. All they have to do is obtain the WSDL service contract and invoke the Web service operation resident on the mainframe under the purview of the Artix Mainframe Transformer Service.

The benefits of the Artix Mainframe Transformer are substantial. The service contract is an asset that can be versioned, maintained and managed. In addition, the approach is non-invasive, service-oriented and enables a wide range of clients developed using different standards to communicate with mainframe-based assets written in COBOL and PL/I.

3.2 Artix Encompass—A Service-Oriented Gateway

Artix Encompass is an off-host service-oriented integration solution that enables mainframes to become integral components of the enterprise integration fabric. Artix Encompass enables organizations to quickly and easily create high-performance links between applications, including mainframe-based applications, without rewriting or re-architecting those systems.

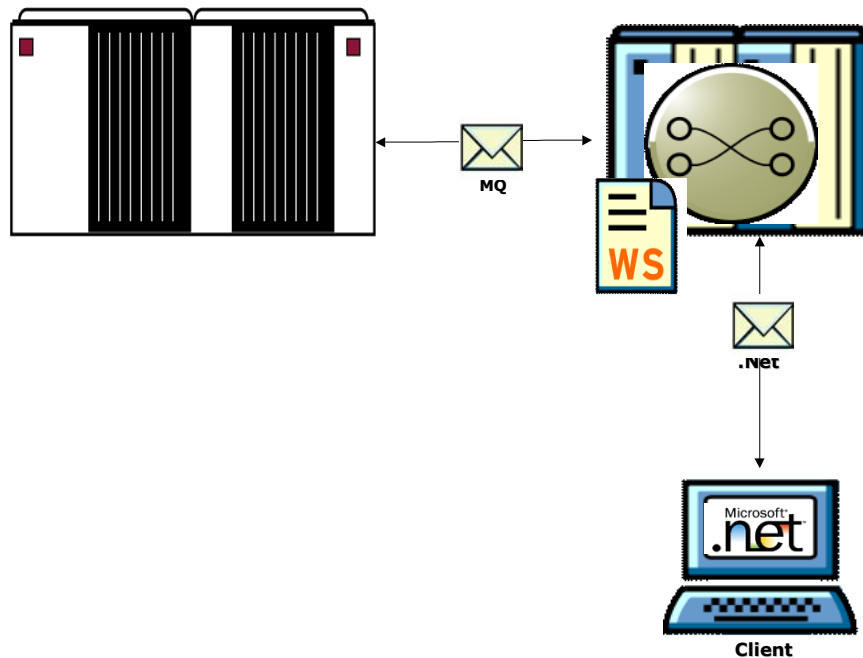


Figure 5: Artix Encompass Service-Enables WebSphere MQ Gateways

Artix Encompass service-enables existing mainframe gateways based on IBM's WebSphere MQ by encapsulating the MQ transport as a set of services.

The Artix Encompass Designer is a Windows-based GUI tool that can read COBOL copybooks that define existing MQ message formats. Application clients need not be aware that they are communicating with the mainframe under existing mainframe connectivity rules and invoke on services as if they were components or objects providing well-defined service interfaces. Artix Encompass directly translates incoming service requests to native MQ fixed-format payloads. It then directs the payload to the outbound queues, where mainframe-based applications collect incoming messages. Outbound messages from mainframe applications are correlated by the Artix Encompass runtime element based on the correlation id, and are directed to the appropriate service requestor waiting for a response.

Artix Encompass is a non-invasive, high-performance middleware service-oriented integration product designed to enable the following enterprise architectural needs:

- Multi-middleware request brokering. Artix Encompass translates between TXN processing (TUX), component middleware (J2EE, .NET), and messaging transports (MQ, TIBCO, JMS, HTTP/S).
- Simplified platform connectivity. Artix Encompass supports a wide range of programming interfaces and platforms, including .NET, J2EE, CORBA, Tuxedo, TIBCO and WebSphere MQ. It also provides support for popular programming languages, including Java, C++, PL/I and COBOL.
- Enterprise Web services deployment and management.
- Non-invasive middleware integration. Because it supports entrenched programming interfaces (C++, COBOL, PL/I, and proprietary APIs) and the ability to integrate with existing runtime environments without introducing new architectural elements (for example, using a transaction manager that is already there), Artix Encompass can be non-invasively deployed into an enterprise.

Artix Encompass is intended for companies looking for corporate LAN-based application integration.

Artix Encompass makes it easy for organizations to rapidly integrate and adapt business systems, and create a single view of existing and new IT elements. Fundamental to the product is a service-oriented view of application development and middleware interaction, even for systems that were not developed with service-oriented integration in mind (such as WebSphere MQ gateways and mainframe applications written using older programming languages such as COBOL and PL/I).

Built on IONA's pluggable ART micro-kernel technology, Artix Encompass can seamlessly broker and route data between WebSphere MQ and other Artix Encompass Service Access Points (SAPs), creating a standards-based enterprise communication fabric. The product supports de facto standards such as WebSphere MQ and Tuxedo, as well as formal standards like J2EE, .NET, CORBA, SOAP, and WSDL.

4 Conclusion

The Artix family of products is designed to facilitate the creation of SOAs where applications can interoperate, invoke and consume services over an open standards-based integration fabric. Artix works within the operational parameters of mainframe IT organizations and uses an incremental, phased approach to service enablement. Mainframe IT organizations do not need to change their existing technology choices or operational guidelines and procedures to benefit from this solutions offering.

Artix is designed to allow companies to fully participate in this services-centric and Internet-friendly vision of integration. It removes impeding, dissonant elements at all layers of the enterprise computing fabric, while continuing to harness the technology assets that play a role in the organization's past and current success.

The Artix Mainframe native offerings let companies repurpose their host business logic without message/data-centric application integration. This allows mainframes to participate as powerful and agile stakeholders in a service-oriented integration fabric involving multiple platforms, standards and technologies. It facilitates the creation of an enterprise-wide service contract model, which is a prerequisite to the creation of SOAs. It uniquely minimizes deployment risks by transparently manages integration issues such as security and data conversion.

By adopting this SOA-enabling framework, mainframe heavy enterprises will reap the benefits of a standards-agnostic environment. They can make significant cost savings and gain maintenance benefits. It enables companies to create a strategic, high-speed and standards-based Web services integration backbone that can be highly leveraged by a variety of applications built using different standards and packages in different time periods.

The Artix product family enables controlled Web services access to CICS and IMS transactions. It features native as well as off-host product sets and can facilitate non-invasive renovation of production host applications. Artix Mainframe Transformer and Artix Encompass are high performance implementations, with performance metrics that compare very favorably with the needs and expectations of mainframe-resident applications.

Artix makes the vision of SOAs a reality, now, through its proven mainframe solutions and service-oriented integration products.

Contact Details

IONA Technologies PLC
The IONA Building
Shelbourne Road
Dublin 4
Ireland
Phone:+353 1 637 2000
Fax:+353 1 637 2888

IONA Technologies Inc.
200 West St
Waltham, MA 02451
USA
Phone:+1 781 902 8000
Fax:+1 781 902 8001

IONA Technologies Japan Ltd
Akasaka Sanchome Bldg 7/F
3-21-16 Akasaka
Minato-ku, Tokyo
Japan 107-0052
Phone:+813 3560 5611
Fax:+813 3560 5612

Support:support@iona.com
Training:training@iona.com
Orbix Sales:sales@iona.com
IONA's FTP siteftp.iona.com

World Wide Web: www.iona.com