

Integrate CICS Applications as Web Services

A HostBridge™ White Paper



Toll-free: (866) 965-2427
Email: info@hostbridge.com

Copyright Notice

Copyright © 2002 by HostBridge Technology. All rights reserved. No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise, without prior written permission. You have limited permission to make hardcopy or other reproductions of any machine-readable documentation for your own use, provided that each such reproduction shall carry this copyright notice. No other rights under copyright are granted without prior written permission. The document is not intended for production and is furnished "as is" without warranty of any kind. All warranties on this document are hereby disclaimed including the warranties of merchantability and fitness for a particular purpose.

Revision date: 4/18/2002

Trademarks

HostBridge is a trademark of HostBridge Technology.
IBM is a trademark of International Business Machines.

Integrate CICS Applications as Web Services

Web services promise to lower the costs of integration and help legacy systems retain their value. This white paper introduces you to the basics of web services, how they reduce costs, and how you can use HostBridge to help your organization integrate CICS applications as web services.

Web Services in a Nutshell

- Programs, not documents
- Remote programs invoked through a loosely-coupled messaging system
- Application interface based on Internet standards (XML and HTTP)
- Meant for application-to-application communication
- A family of technologies and specifications (SOAP, WSDL, and UDDI)
- Platform-independent
- Broad vendor support

Web services are platform-independent interfaces that allow communication with other applications using standards-based Internet technologies, such as HTTP and XML. They provide an opportunity for organizations to reduce the costs and complexities of application integration inside the firewall and open up new possibilities for legacy applications to participate in eBusiness.

Standards-based and Platform-independent

While applications have been able to communicate using Internet technologies for years, only recently have standards evolved to allow any web-based application to talk to any other web-based application. Web services depend upon three key open standards to allow them to communicate regardless of the hardware they run on, operating system they run under, or programming language used to produce them.

Table 1. Standards for web services

Standard	Description
SOAP	The Simple Object Access Protocol specifies a format for messages passed between web services.
WSDL	The Web Services Description Language describes web services so that other web services know how to access them, what to send as input, and what to expect as output.
UDDI	The Universal Description, Discovery, and Integration standard is a searchable registry that allows web services to publish WSDL documents so other web services can find them by name or category. UDDI also provides specification information concerning the input data formats, security models, protocols they use, and response data formats.

Emerging standards for web services include:

- IBM's WSFL and Microsoft's WS-Routing for handling complex business process workflow.
- OASIS' SAML and Microsoft's WS-Security for handling authentication and message integrity.
- XAML and OASIS' BTP for handling transactions that span multiple enterprises and long lasting transactions to ensure that the outcomes of the transactions are reliable.

Underlying all of this is the use XML as the message format and a standard protocol such as HTTP as the transport.

"G3,500 firms have valuable business logic embedded in a hodgepodge of legacy systems - from CICS to customized provisioning apps. Adding Web services interfaces to the mix makes those elements available for widespread reuse."

Laura Koetzle
Forrester Research, Inc.

Standards-based components diminish the costs and skills required to integrate applications within an organization or between partners. Because there is widespread vendor support for web services, interoperability among vendor solutions will improve. However, the greatest benefits of standards-based web services for developers and administrators are reduced complexity and increased flexibility of integration architectures.

Reducing Interface Complexity and Increasing eBusiness Flexibility

“With a new web services coating, existing systems, from CICS to client/server, will participate in new business scenarios. And with that participation codified in standard interfaces, the apps themselves can be modified or swapped out without bringing operations to a halt.”

Ted Schadler
Forrester Research, Inc.

The number of application interfaces and data formats that developers must learn and support affects the complexity and flexibility of integration. The more interfaces and formats the greater the costs to implement and maintain an integration framework. Web services provide a common interface to applications and XML provides a common data format — together, they simplify integration and lower integration costs.

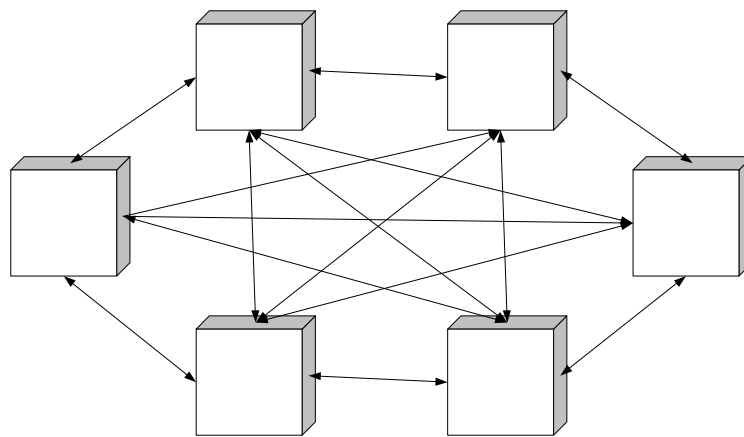


Figure 1. Traditional EAI frameworks (6 applications, 6 APIs, and 6 data formats with 30 data transformations required to synchronize data) . Source: META Group

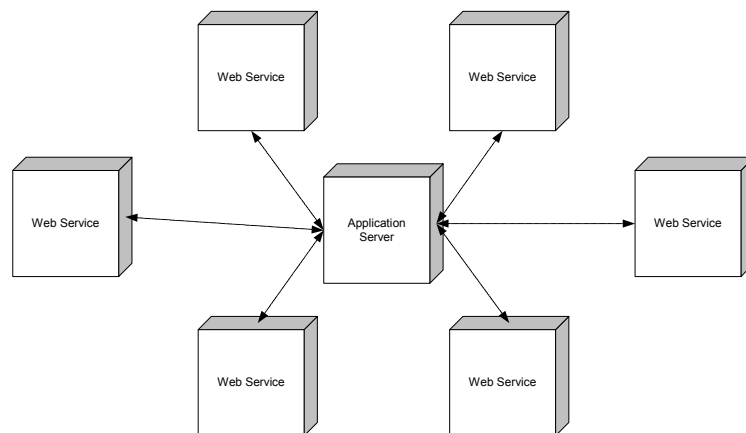


Figure 2. Web services integration frameworks (6 web services, 1 API, and 1 data format with 12 data transformations required to synchronize data)

Application Interfaces

Traditionally, developers create application programming interfaces (APIs) for each application in the integration project to allow inter-application communication.¹ Because the APIs for each application are different, companies integrating large numbers of applications develop complex integration frameworks that require specialized development skills and lead to increased integration costs. Increased complexity also makes integration architectures less flexible, so that changes in the network and business processes (following M&A, partnership changes, etc.) are slow and cumbersome.

Reducing the number of APIs reduces complexity and increases flexibility. (See Figures 1 and 2.) Web services based on Internet technologies are not specific to any platform and provide a single API for any application to use. They are loosely-coupled and can be invoked directly as traditional APIs or requests can be sent to a queuing system using where transactions can occur at specified dates or times.² Web services also provide greater flexibility when designing integration architectures. Because web services use a common interface, changes to the network or business processes do not affect the ability of individual applications to communicate with each other.

Common Data Formats

Another source of complexity is the proliferation of data formats used by each application. Reducing the number of data formats reduces the number of data transformations that take place as information passes through an integration framework. Figures 1 and 2 show how a common data format based on XML can reduce the complexity in an integration framework.

HostBridge and the Web Services Model

Figure 3 below shows the basic model for accessing CICS applications as web services through HostBridge.

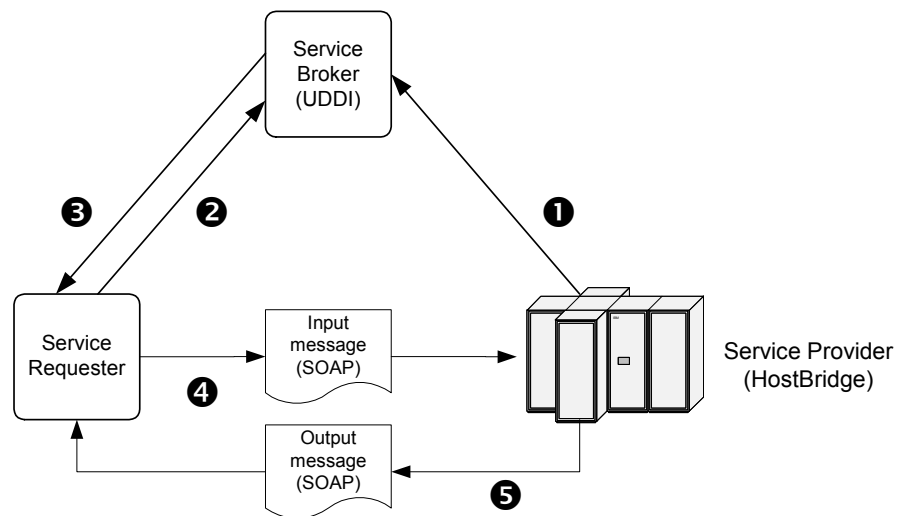


Figure 3. HostBridge and the web services model

¹ These interfaces — even interfaces built upon standardized interfaces such as CORBA, J2EE, or Microsoft's DCOM — have platform-specific and vendor-specific components that keep them from being truly technology-neutral.

² Integration using traditional APIs is tightly-coupled in that each application accepts input and output using unique data formats and only recognizes direct invocation that expects to receive an immediate response.

The diagram in Figure 3 shows three web services: a Provider (HostBridge) that provides the web service, a Requester that uses a web service, and a Broker that finds Providers for Requesters. The following steps are required to find and use a web service. (Subsequent requests do not require steps 1-3.)

1. HostBridge uploads a WSDL specification to publish its web service with a Broker.
2. The Requester queries the Broker for a web service by name or category.
3. The Broker selects a Provider and returns the Provider information to the Requester.
4. The Requester uses the information from the Broker to format and send a SOAP message to the HostBridge.
5. HostBridge returns a SOAP message to the Requester with the CICS data enclosed.

Enabling CICS Applications as Web Services

Because HostBridge XML-enables the CICS transactions, they can be accessed as web services using the same methods used to integrate native web-based applications. The steps required allowing remote applications to invoke CICS transactions as web services using HostBridge are simple and straightforward.

1. XML-enable your CICS transactions with HostBridge.³
2. Set up the CICS transactions with HostBridge that you want to define as a web service.
3. Create a WSDL file to describe your web service.
4. Publish your WSDL file to an internal or external UDDI server so other web services can find your CICS web service.
5. Access the web service using SOAP, HTTP GET, SMTP, or any other standard protocol.

Conclusion

The problem with traditional integration is the proliferation of point-to-point data conversions that must change as new applications are integrated or data formats change. The problem gets worse when you add business partners into the integration mix. Web services reduce the costs and complexities of performing data conversions internally and externally between partners. By using standards-based technologies and widely available skill sets, web services allow companies to develop flexible integration solutions that can change as needed.

HostBridge is the perfect companion to enable CICS applications as web services. By using the same industry standard technologies as web services, HostBridge makes it possible for applications to transparently invoke CICS transactions within a web services architecture and receive the resulting data as well-formed XML. For organizations that want to retain the value of their CICS applications, HostBridge and web services offer an ideal solution.

Contact us to learn more about using HostBridge to turn your CICS applications into web services or to request a free 30-day evaluation.

Contact Information

For more information on HostBridge or to inquire about our free 30-day trial, please contact us using the information below.

Toll-free:
1.866.XML.CICS (965.2427)

International:
1.405.533.2900

Email:
info@hostbridge.com

³ See white paper entitled *XML-Enabling CICS Applications for e-Business* to see how you can use HostBridge in your organization.