

TITLE: Strategic options for extending CICS to an SOA

HOST: Hi, and welcome to the "Did you say Mainframe?" podcast series. This is where we regularly interview IBM technical experts who can help you to understand important IBM mainframe hardware and software issues. I'm your host Sherrie Abshire.

Today we're going to talk about the strategic options available to you for extending IBM CICS Transaction Server for z/OS to a Service Oriented Architecture (SOA).

Our guest today is Mark Cocker, IBM CICS Technical Strategist from IBM Hursley Laboratory in the UK. Mark, it's great to have you here.

SME: I'm delighted to be here.

HOST: Before we begin, I'd like to mention to our listeners that there is a recorded teleconference expanding on the subject of our podcast today. I'll be giving out more information at the end of this podcast.

HOST: Why do IBM customers have a need to connect to CICS from an SOA environment?

SME: CICS applications and the information they manage are critical to many businesses today. In essence those CICS applications are the business logic that implement the desired rules, policies, and procedures, and manage the access and updating of databases. Together CICS and System z continue to provide the most robust and efficient platform to meet businesses' most demanding needs. So IBM customers are looking to efficiently turn their CICS applications into services and easily use them from other components in their SOA environment.

HOST: What are the strategic options for connecting to CICS?

SME: To achieve the best reuse of CICS programs, and to support multiple access options, you should architect clear and concise business logic interfaces of the right granularity for the solution. CICS and other tools help transform and aggregate your programs to provide the right level of granularity for your service requesters.

CICS has always provided a choice of integration options - the strategic methods today are Web services, the CICS Transaction Gateway, and WebSphere MQ.

Web services are supported directly in CICS Transaction Server V3. CICS provides the tools and facilities for CICS applications to be a Web service provider and/or a Web service requester. For example the CICS Web services assistant can create the Web services description (WSDL) from your COBOL, C, C++, or PL/I program interface definitions. The WSDL can then be placed into a registry and

2008 WebSphere System z Podcasts - 'Did you say Mainframe?' repository for easy access by all development teams who can independently create their components that invoke a service in CICS. Web services are popular because they are based on open standards and as such there is broad support across many vendors, tools and environments.

The second method is the CICS Transaction Gateway - it primarily provides the APIs and infrastructure for you to call CICS application from Java environments such as WebSphere Application Server, but it also supports .NET and other environments. The CICS Transaction Gateway provides a solution where little if anything needs to be changed in CICS or your applications. It is often used when high Quality of Service and synchronous invocation are needed. It is particularly fast and efficient when the Java application and CICS are in the same System z environment.

Thirdly, WebSphere MQ is a high quality messaging backbone which provides reliable messaging coupled with transactional integrity, which supports composition of complex applications and parallel processing. It enables you to connect any applications to any other applications on any computing platform, so it can be used as a standard way to deliver inter-process communications between your applications running on CICS, IMS, IBM WebSphere Application Server for z/OS, Batch or TSO.

WebSphere MQ is often used when assured delivery, loose coupling, and/or asynchronous invocation are needed. WebSphere MQ is also particularly efficient in handling large data sizes. WebSphere MQ can be used as a Web services transport either using SOAP over JMS standard, or by sending SOAP messages in raw MQ messages. WebSphere MQ applications can be described in a standard way like other Web service assets using WSDL and IRI's and stored in a registry and repository for reuse.

HOST: How do you select which one to use?

SME: Before making decisions on whether to use Web services, the CICS Transaction Gateway, or WebSphere MQ, all the business consideration and technical requirements should be compared with the capabilities of each access option to select the most appropriate.

Architects, client developers, CICS application developers, and CICS system programmers will find it easy to setup and use these integration options.

HOST: Can you tell us where our customers can find more details on the technical differences of these connectivity methods?

SME: The key differences are in the areas of security, transactional scope, tools and APIs, synchronicity, and the level of coupling between the requester and provider. The detail of these differences can be found in the white papers and IBM Redbooks in the CICS library on ibm.com/cics. The

2008 WebSphere System z Podcasts - 'Did you say Mainframe?' key publications are 'Options for integrating CICS applications in an SOA' white paper and 'Architecting Access to CICS within an SOA' redbook.

HOST: Are there example scenarios which use a mixture of these connection methods?

SME: A mixture of these methods is often used by our customers to fully meet their different application requirements and demanding workloads. You can find some recent examples in the teleconference which you mentioned at the start.

HOST: Mark, that was really interesting.

SME: Thanks for giving me the opportunity to talk about SOA access to CICS.

HOST: Well, that wraps up this podcast discussion. To find out more about the Strategic options for extending IBM CICS Transaction Server for z/OS to a Service Oriented Architecture teleconference I mentioned earlier, please go to the description for this podcast at:
<http://www.ibm.com/software/os/systemz/podcasts/websphereonz/>

Join us next time as we talk about another important mainframe topic. For now, this is Sherrie Abshire saying "Thanks for listening".