

Mobile Computing

There is little question that mobile computing is playing a significant role in today's I/T landscape. From banking applications to airline flight status apps to insurance claim apps, mobile devices are in wide-spread use. In fact, it is often seen as a competitive advantage to have mobile functionality as part of your I/T solution.

So we start with that assertion: *the influence of mobile computing is having a profound effect on the way we view and design I/T architectures.*

Mobile Computing and System z

System z and z/OS have a long history of providing critical business value with systems such as CICS and IMS. Those systems, often backed by DB2, are repositories for vast amounts of business transactions and data.

It is no surprise that mobile computing requirements seek to access the transactions and data on System z. The question is how to connect the mobile environment to the core System z environment in a way that is consistent and manageable.

The Challenge

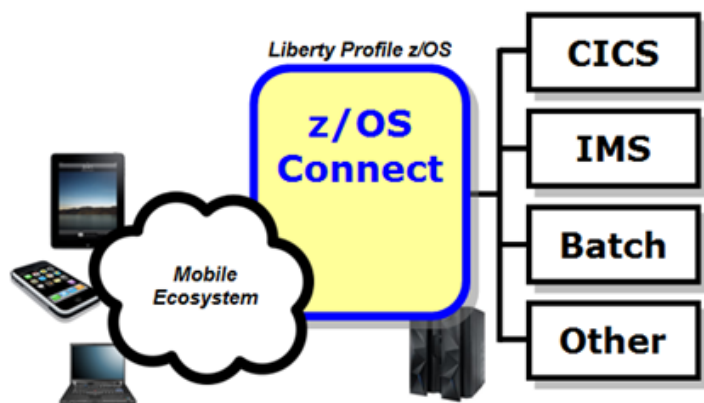
Mobile devices tend to make use of standard protocols and standard data format packaging, while some z/OS systems do not accept those formats (or if they do, it requires configuring the support).

A better solution is a single interface on z/OS that accepts standard protocols and can interact with multiple backend programs and systems. Further, this interface would do data conversion as needed, provide authorization services to allow or deny users based on roles, and track activity details using SMF recording.

Such a solution would provide a common and consistent interface for mobile access.

Introducing z/OS Connect

z/OS Connect is software function written to run inside Liberty Profile for z/OS and serves as an enabler of connectivity between the mobile environment and backend z/OS systems:



The following is a summary list of the features of z/OS Connect:

- There is no additional cost to use z/OS Connect, and it is packaged with WebSphere Application Server, CICS and IMS software products.

- Built on Liberty Profile for z/OS, which means it is lightweight and dynamic. It can run as a z/OS Started Task, which means it can run within your current operational procedures and routines.
- Provides a RESTful API and accepts JSON data payloads.
- Configurable so you control what backend programs or applications are exposed and accessible.
- Provides a discovery function so application developers may query for a list of configured services as well as query for details on the services.
- Capable of performing data conversion from JSON to the data format required by the backend configured service.
- Optional authorization checking using SAF to allow or deny users access to z/OS Connect services.
- Optional activity recording using SMF to track requests by date and time, bytes sent and received, and response time.

z/OS Connect becomes the solution that addresses the challenges we cited earlier.

Built on Liberty Profile

Liberty Profile is IBM's fast, dynamic and composable application server runtime. Liberty Profile for z/OS requires relatively little memory, and it starts very quickly. It has a simple configuration model consisting of one required file: `server.xml`.

Liberty Profile for z/OS can be run as a Started Task, and can be monitored and controlled using common procedures and common system automation routines.

z/OS Connect is written to run inside an instance of Liberty Profile for z/OS. You enable and configure z/OS Connect by providing updates to the `server.xml` file. Liberty Profile is dynamic, so a change to `server.xml` is detected and dynamically enabled.

Multiple instances of z/OS Connect may be configured for availability. The configuration model of Liberty Profile allows for sharing of common configuration elements between instances, so a single configuration change can automatically propagate to multiple instances of Liberty Profile and z/OS Connect.

RESTful Services and JSON Data

REST (Representational State Transfer) is a simple stateless architecture that is based on the HTTP Uniform Resource Indicator (URI) that is sent in. For example, a URI of `/getCustomer?custno=1234` is understood by the receiving server to run a customer query on customer number 1234.

JSON (JavaScript Object Notation) is a way to encapsulate data in name/value pairs and exchange that data back and forth on the HTTP flows between devices.

z/OS Connect is designed to accept REST URIs and JSON data. That makes it compatible with a wide variety of devices and systems in the world today. Developers of mobile applications do not need to think about the details of CICS or IMS. They need only understand the RESTful APIs and the JSON data requirements for the service exposed by z/OS Connect.

Discovery Function

z/OS Connect comes with a discovery function that allows developers to query for the services that have been configured for an instance of z/OS Connect. One REST call to z/OS Connect will return a list of all configured services, and another REST call will return the details of a given service.

Developers do not need to know CICS or IMS to do this; they only need to know the REST API for the service and the JSON requirements. The discovery function of z/OS Connect provides this.

Not Just Mobile Devices, Cloud Systems Too

z/OS Connect is useful for more than just mobile devices. Any function capable of using RESTful APIs and JSON data can use z/OS Connect.

An emerging trend involves systems that are part of a dynamically provisioned cloud system. An example of this is IBM's Bluemix, which is an implementation of IBM's Open Cloud Architecture, leveraging Cloud Foundry to enable developers to rapidly build, deploy, and manage their cloud applications¹.

IBM Bluemix is not required to use z/OS Connect; rather, z/OS Connect enables systems in IBM Bluemix to more easily access z/OS transactions and data. The standards-based RESTful and JSON design of z/OS Connect makes this possible.

Configured Services

Configured services is a phrase used to indicate a backend program or application that has been defined in z/OS Connect to be accessible. This involves updating the Liberty Profile configuration XML in which z/OS Connect is operating.

Key: Only those backend programs, applications or data you configure are exposed. You control what gets exposed as an accessible service. All others are inaccessible. Further, with the security interceptor (discussed below), you may control access to z/OS Connect and services based on the identity of the user.

Data Conversion

We mentioned earlier that z/OS Connect takes in JSON as its data format from mobile devices and systems. However, backend programs such as those in CICS and IMS will most likely not understand JSON. To address that, z/OS Connect has a data conversion function that will map JSON to the data format requirement of the backend configured service.

To accomplish this z/OS Connect requires an understanding of how to perform the data conversion. You provide this in the form of a bind file, which describes the relationship between the JSON and the backend data. These bind files are produced by a supplied utility, and is based on the same function provided by CICS for its web services support.

In the z/OS Connect configuration you indicate the directory where the bind files are located for a given configured service. When a request comes in z/OS Connect converts the data; when a response is produced z/OS Connect converts it back to JSON.

'Interceptors' for Security and Audit

Interceptors is a term used to indicate code that gets control before and after processing a request. You choose whether to configure interceptors or not based on your needs.

We'll look at two interceptors here:

- **Security** – The identity of the calling user can be checked against defined *roles* to determine if they have authority to use z/OS Connect or the configured service. On z/OS the security interface is SAF, which is backed by a security product such as IBM's RACF.
- **Audit** – Information about the request can be logged for later analysis. The SMF function on z/OS is used for activity recording. Information about each request is logged, including the date and time, the bytes sent and received, the response time seen, and the identity of the user.

In addition, it is possible to write your own custom interceptor code and have it called as part of the processing.

z/OS Connect and IMS, CICS

For connectivity to IMS, z/OS Connect can use a provided IMS JCA resource adapter and connect through IMS Connect. This is provided with the IMS Mobile Feature Pack.

For CICS, you have two options: use the WebSphere Liberty Optimized Adapters connector to access the CICS region, or operate z/OS Connect *inside* the CICS region and make use of the JCICS interface to access CICS transactions and data.

IBM Worklight and API Management

IBM Worklight serves as the mobile enterprise application platform (MEAP) in a mobile-enabled architecture. It provides application management, security and operational governance for mobile applications. That is *not* the role of z/OS Connect.

z/OS Connect augments the role of IBM Worklight by providing a standards-based interface to backend z/OS programs and applications with data conversion services. That allows IBM Worklight to focus on mobile application management and lets z/OS Connect do the connectivity handling.

This is part of IBM's API Management strategy. API Management allows you to create, manage, and monitor an on-premise environment from which you can create, promote, consume, and track APIs.

Summary

z/OS Connect is designed to enable better and more manageable connectivity between mobile systems and your backend z/OS programs, applications and systems. It provides a consistent front-end interface for mobile systems using REST and JSON, and it shields backend systems from having to understand those protocols and formats.

WebSphere Application Server z/OS	IBM US Software Announcement 214-204
CICS	IBM US Software Announcement 214-107
IMS	IBM US Software Announcement 214-220

More Details

For more, contact Dave Sudlik, Product Manager - WebSphere Application Foundation, at dsudlik@us.ibm.com

End of Document

¹ See: ibm.com/software/ebusiness/jstart/bluemix/