Subsystem Service Model: Service-Oriented Architecture for Mainframe Environments

Create agility, efficiencies, and costs savings for mainframe architecture

Author: Jeremy S. Hamilton
Sr. Software Consultant, BMC
Table of Contents

1 EXECUTIVE SUMMARY

2 BACKGROUND
   The Service Concept
   Traditional Mainframe Environments

3 SUBSYSTEM SERVICE MODEL (SSM)
   BMC Subsystem Optimizer for zEnterprise

ADVANTAGE 1 – COST SAVINGS
   Traditional Mainframe Environment MLC
   Subsystem Service Model Mainframe Environment MLC

4 ADVANTAGE 2 – ENHANCED AVAILABILITY AND EASE OF MAINTENANCE
   Availability
   Maintenance

5 SAFETY

6 CONCLUSION
Executive Summary

Service-oriented architecture (SOA) has emerged as a popular standard to “modernize” legacy mainframe applications into today’s infrastructures. However, mainframes are often viewed as only a “black box” service provider to the loosely coupled distributed applications. As a result, the efficiencies of SOA are not realized on the mainframe itself.

The subsystem service model (SSM), enabled by BMC Subsystem Optimizer for zEnterprise®, extends the concept and benefits of SOA into the mainframe infrastructure. SSM separates service-consuming transaction managers (IBM MQSeries®, IBM CICS®, IBM IMS®) and the service providing database managers (IBM DB2®, IMS) on to respective LPARs.

The result has many benefits, including:
- Dramatic cost savings
- Enhanced availability
- Ease of maintenance
- Environmental simplification

This white paper discusses how the success of SOA in the distributed world is now possible on the mainframe with SSM.
**BACKGROUND**

In its simplest form, SOA is an architectural approach to designing applications that decouples service consumers from service providers. In distributed computing, service providers are housed in disparate systems that are specifically set up to gain cost and performance advantages. Mainframes are often viewed as only a “black box” service provider to the loosely coupled, distributed applications. While SOA has done a great job exposing mainframe applications as business services through interfacing and wrapping, the flexibility, agility, and efficiencies of SOA have not been realized on the mainframe itself.

**The Service Concept**

The “service concept” is key to understanding how SOA and the subsystem service model (SSM) provide value to mainframe shops. A “service” simply offers a capability by which the needs of a service consumer are satisfied by a service provider. One or more service providers, hosted together or separately, can be used to complete the consumer’s request.

**Traditional Mainframe Environments**

Unlike in the distributed world, the mainframe service consumer and service provider are traditionally co-located on the same LPAR. The service consumers are the subsystem transaction handlers, such as MQ, CICS, and/or IMS, which handle transactions from the outside world; and service providers are the subsystem database managers DB2 or IMS, which satisfy the data requests made by the transaction managers.
**SUBSYSTEM SERVICE MODEL (SSM)**

The subsystem service model extends the SOA concept into the mainframe with the separation of the transaction manager subsystem (TM) from the database manager subsystem (DB). The example below applies the SSM concept to our example mainframe environment. Transactions coming in from the outside world process normally through CICS A or CICS B located on LPAR C. This separation can be done using standard IBM-supplied APIs and code changes, or by using BMC Subsystem Optimizer for zEnterprise.

**BMC Subsystem Optimizer for zEnterprise**

Subsystem Optimizer can intercept the database calls from CICS, and use XCF communication to route calls over to a remote LPAR where a corresponding DB2 is running. DB2 then fulfills the request, and Subsystem Optimizer routes the response back to CICS, completing the transaction. No application changes are required for the data request routing, so the end user (internal or external) is unaware of the change.

The example mainframe environment without Subsystem Optimizer showed that CICS and DB2 were running on separate LPARs A and B. IBM’s sub-capacity pricing dictates that subsystems be charged based on the aggregate peak consumption for all the LPARs that a subsystem is running on. That means that even if the TM and/or DB are utilizing very few MSUs, they will each be charged for LPAR peak consumption. With SSM, the monthly license charge (MLC) for the subsystems will be charged at closer to utilization-based pricing.

Both LPAR A and LPAR B can now be optimized for hosting DB2, and LPAR C can be optimized for CICS. A new LPAR is not always required. As long as there is separation of the TM from the DB, the benefits can still be realized.

**ADVANTAGE 1 – COST SAVINGS**

Understand and manage complex MLC bills.
Watch the video: https://youtu.be/qPIKPP6dewo
Example mainframe environment for CICS MSU utilization

Example cost savings analysis for CICS isolation

Traditional Mainframe Environment MLC
The graphic below shows that the combined peak for our example environment is 144 MSUs, which means both CICS and DB2 each would be charged at 144 MSUs. CICS was only utilizing a maximum of 40 MSUs, which means an overpayment of 104 MSUs for CICS MLC. The DB2 charge is increased by the number of CICS MSUs located in the peak for its respective LPARs.

Subsystem Service Model Mainframe Environment MLC
Applying the SSM to the example environment isolates CICS to an individual LPAR, and DB2 to two respective LPARs. As a result, each of the subsystems will be charged only for the MSUs utilized on each of those LPARs. Isolating CICS produces dramatic cost savings, as well as savings across the entire software stack.

ADVANTAGE 2 – ENHANCED AVAILABILITY AND EASE OF MAINTENANCE
On both mainframe and distributed platforms, increased demand and reduction of budget have made everyday IT management issues, such as availability and maintenance, hard to accomplish. Constant upgrades, overloads, hardware and software issues, system crashes, and other common hitches make some downtime a part of doing business. The dispersed and modular aspects of SOA, and now SSM, allow businesses to boost their infrastructure’s stability and redundancy plans.
**Availability**

Despite years of striving for 100% “uptime,” downtime is inevitable as subsystems and LPARs are routinely brought down for maintenance. Unless robust plans are in place, the transactions coming from the outside world are delayed until the subsystem or LPAR is brought back up. With SSM, all of the DB2s are service providers to CICS and those requests can be routed to a different DB2 that is currently up and running.

For example, if a primary DB2 1 subsystem is brought down for maintenance or LPAR A is being initial program loaded (IPL’d) with the SSM, the secondary location, DB2 2 on LPAR B, can be designated to complete the transaction.

Subsystem Optimizer for zEnterprise enables this availability enhancement, and provides the ability to map out the primary, secondary, and tertiary locations for transactions to complete their database calls. These locations can be designated based on specific or generic DB2 SSIDs, DB2 groups, and even across LPARs and CPCs that are within the same sysplex environment. This gives your application a broad array of service providers to satisfy the service consumer’s requests.

**Maintenance**

Having fewer subsystems running on each LPAR narrows the impact of outages and subsystem changes. Maintenance operations and plans can be created with specific subsystems and environments in mind. Reducing LPAR complexity inherently makes changes easier since they are isolated individual subsystem environments.

**SAFETY**

A typical question asked regarding the SSM-enabling solution Subsystem Optimizer for zEnterprise is “what happens if the redirection fails, or Subsystem Optimizer itself goes down?” The answer is quite simple: you already do it. In a standard environment, if a CICS data request to DB2 cannot be completed for some reason, CICS and DB2 have clean-up routines that they go through to ensure data/transaction integrity. Subsystem Optimizer only acts as the intermediary between CICS and DB2, so should so should Subsystem Optimizer or DB2 fail, then transactions are handled the exact same way they would be natively. This built-in functionality reduces the failure exposure to the application level. For absolute worst-case scenario planning, a local copy of the remote subsystem could be kept so that it could be brought up fairly quickly in the event of an emergency. When a local version of a subsystem is detected, Subsystem Optimizer will disengage, allowing the transaction to complete using the native service for DB2.
CONCLUSION

Much like how a service-oriented architecture was a shift in thinking for distributed software applications and environments, a subsystem service model with Subsystem Optimizer changes how mainframe infrastructures can be arranged and managed. The decoupling of the service consumer (TM) from the service provider (DB) on the same LPAR is a dramatically different approach that opens the door to a variety of environmental configurations. No application changes are required for this separation, so the end user is unaffected by the change. Now the environment can be designed around workload placement, service-level agreements (SLAs), optimal MSU utilization, and throughput.

SSM with Subsystem Optimizer is not an “all or nothing” solution, so you have the power to choose the right mixture and configuration of subsystems for your environment. Cost savings and increased flexibility are not the only benefits. The SSM model can translate into suggestions for process and business model improvements, benefiting the overall health of your mainframe environment. Discover the benefits of SOA in the distributed world and let Subsystem Optimizer bring those benefits to the mainframe with SSM.

FOR MORE INFORMATION

To learn more about Subsystem Optimizer for zEnterprise, please visit bmc.com/subzero or contact your BMC sales representative.