

# CICS-program som Stored Procedure

Lennart Henäng, Folksam  
Björn Lanzén, Folksam

## Agenda

- Review of our Architecture
- CICS contaminated modules
- A solution – or rather two solutions
- Considerations
- Performance characteristics

## Crossing the Chasm Between Java and COBOL



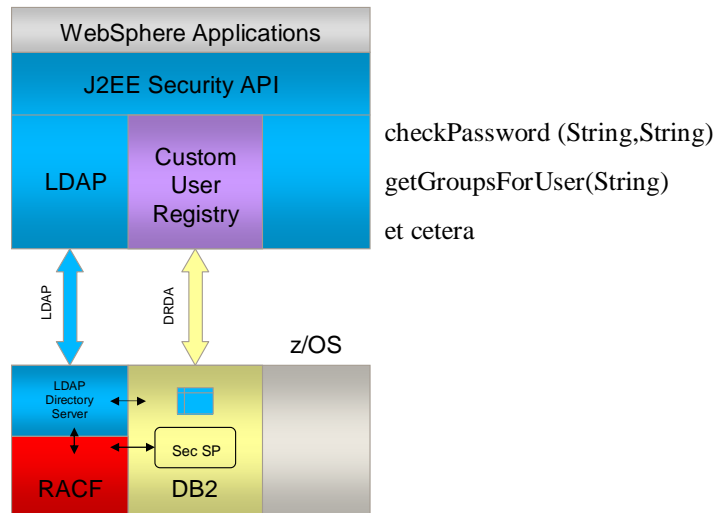
Folksam®

3

Internally within Folksam we illustrate the problem to get development in Java to go hand in hand in with our development in COBOL with a bridge = middleware. We use to say that there is a chasm between Java and COBOL when it comes to understanding each others environments and programming paradigms.

In that respect, Stored Procedures comes in as a way to cross that chasm.

## Authentication and Access Control



Folsam®

4

To accomplish authentication of individual users as well as access control, our strategic solution is to use the standardized J2EE Security API from within our WebSphere applications.

Furthermore, our long term solution for a common directory is to use a centralized LDAP directory. The LDAP Directory Server component of the z/OS Security Server is believed to be the product for such an implementation.

Since the implementation of such a centralized LDAP directory is not in place at Folsam, we needed to find a short term solution. The WebSphere server allows us to provide an own Custom User Registry that has to respond to a certain set of method calls to do authentication as well as access control. Important examples of the 18 defined methods are

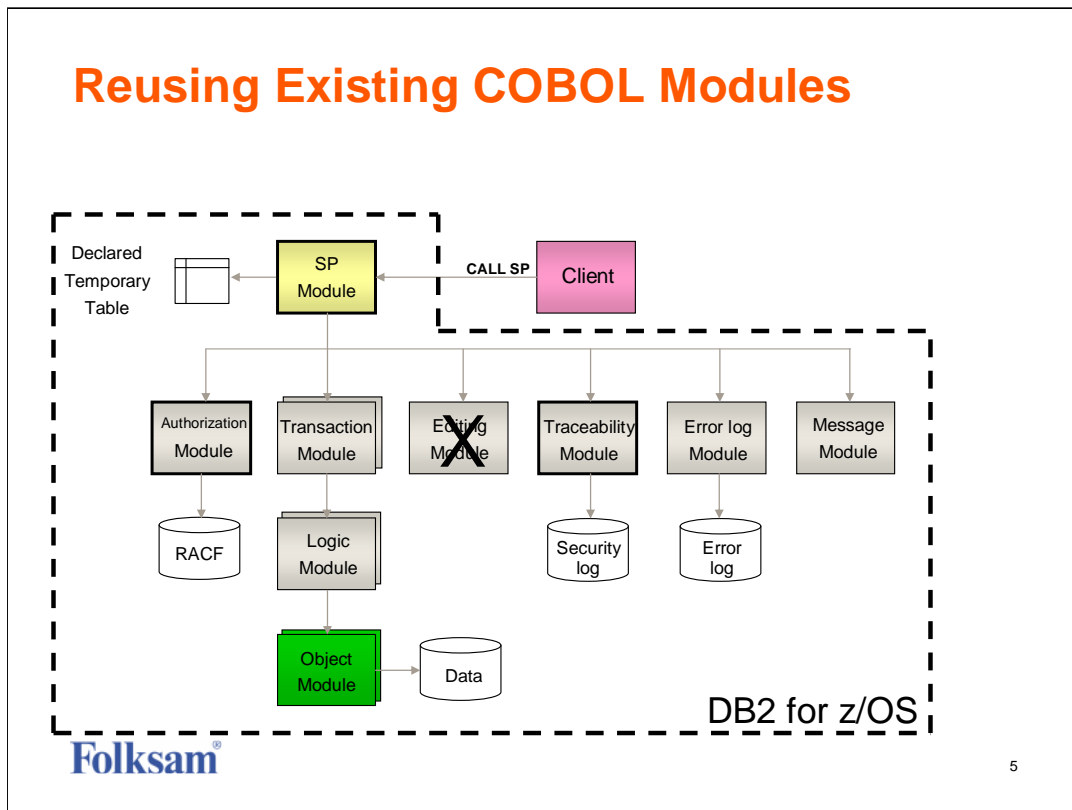
- checkPassword - that checks that the combination of specified userid and password is correct
- getGroupsForUser - that lists the groups that a specified userid belongs to

The authorization checking is based on users belonging to groups (as opposed to having access to defined resources)

Since all security information needed is within RACF, Folsam needed to develop a Custom User Registry communicating with RACF as a short term solution. To access RACF, Folsam developed a number of Stored Procedures written in assembler that corresponds to the different methods mentioned above.

An important consideration is that WebSphere currently only allows one single Custom User Registry per server. If you, for example, need different authentication methods for different applications, you have to develop a single Custom User Registry that can detect which application wants a security check to be accomplished and then, in your own code, call the corresponding back-end security server.

## Reusing Existing COBOL Modules



What about the reality? Changing from our home grown middleware to Stored Procedures has to be as inexpensive as possible. Most of the already developed business logic should therefore be reused. The highly structured application architecture that we have developed makes it possible for us to reuse much of what is already in place.

This foil shows a view of which types of modules that are typically involved in a transaction.

With Stored Procedures and their structured interface, an Interface Module has to be developed. The Interface Module is the module that is defined in the Stored Procedure definition on the EXTERNAL NAME parameter. It receives control from DB2 and is responsible to handle all the input and output parameters. It will also be responsible to call the business logic. In our case a so called Transaction Module which in turn calls Logic Modules which in turn calls Object Modules (which are the ones that operates on DB2 data).

There are also other services that is called by the Interface Module. There are Editing Modules, Error Logging Modules, and Message Modules. For the sake of our Stored Procedure implementation, we also needed to develop a new Traceability Module that will provide security logging (for traceability) on an application level. Furthermore, there were still requirements to do authorization checking within the back-end. We developed a Stored Procedure called checkAuthority to ask RACF for the authorization level of a certain user on a certain resource class and resource name (READ, UPDATE, CONTROL, ALTER). The idea behind using a Stored Procedure is that it can be called from wherever you can issue a SQL-statement - not only from a COBOL program running in back-end.

Since STAYRESIDENT YES only applies to our interface module we have chosen to use STAYRESIDENT NO and use VLF to speed up the loading of modules.

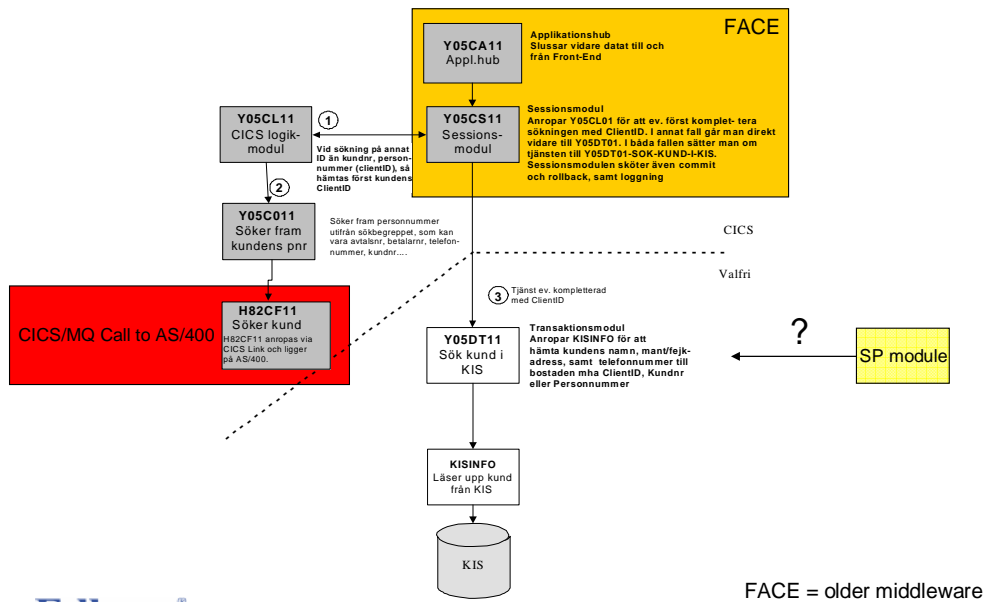
So far, so good! But what happens when the packages related to the different modules are in various collections? You will get an SQLCODE -805 DB2 telling you that it cannot find the required package...

## CICS Contaminated Modules

- Some applications don't follow the architecture
- There might be reasons why they contain CICS calls
  - Legacy
  - Communication with other systems
  - Using CICS functionality
- We call these modules "CICS Contaminated"
- Need a solution
  - rewrite or execute CICS programs in some other way

# An Example

Tjänst: SÖK KUND I KIS



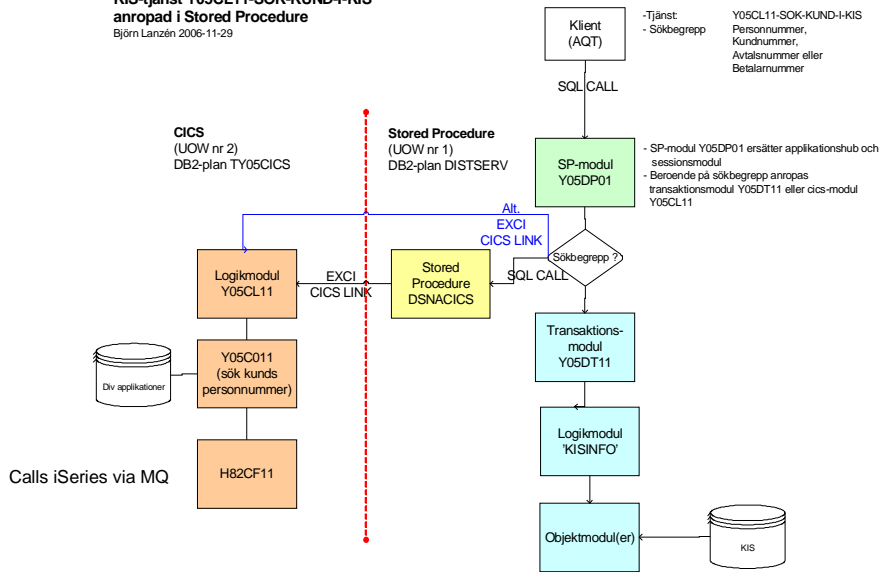
Folksam®

7

Y05CL11 is used to call CBR, SPAR as well as our company insurance application on iSeries

# A Solution – or Two

**KIS-tjänst Y05CL11-SOK-KUND-I-KIS**  
**anropad i Stored Procedure**  
 Björn Lanzén 2006-11-29



Folksam®

8

EXEC CICS LINK is also called Distributed Program Link (DPL)

## Calling DSNACICS (from Y05DP01)

```
*****  
*** CALL SP DSNACICS  
*****
```

```
EXEC SQL  
CALL SYSPROC.DSNACICS  
  (: PARM-LEVEL :IND-PARM-LEVEL ,  
   : PGM-NAME   :IND-PGM-NAME ,  
   : CICS-APPLID :IND-CICS-APPLID ,  
   : CICS-LEVEL  :IND-CICS-LEVEL ,  
   : CONNECT-TYPE :IND-CONNECT-TYPE ,  
   : NETNAME     :IND-NETNAME ,  
   : MIRROR-TRANS :IND-MIRROR-TRANS ,  
   : COMM-AREA   :IND-COMM-AREA ,  
   : COMM-LEN    :IND-COMM-LEN ,  
   : SYNC-OPTS   :IND-SYNC-OPTS ,  
   : RET-CODE    :IND-RET-CODE ,  
   : MSG-AREA    :IND-MSG-AREA)  
END-EXEC.
```

Folksam®

9

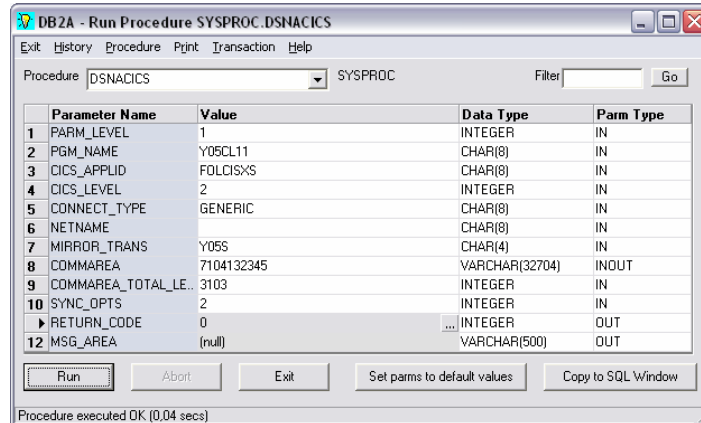
\*\*\*\*\*  
\* PARMS FOR CALL TO DSNACICS

```
*****  
01 DSNACICS-PARMS.  
  02 PARM-LEVEL      PIC S9(9) COMP.  
  02 PGM-NAME        PIC X(8).  
  02 CICS-APPLID     PIC X(8).  
  02 CICS-LEVEL      PIC S9(9) COMP.  
  02 CONNECT-TYPE   PIC X(8).  
  02 NETNAME         PIC X(8).  
  02 MIRROR-TRANS   PIC X(4).  
  02 COMM-AREA       PIC X(31030).  
  02 COMM-LEN        PIC S9(9) COMP.  
  02 SYNC-OPTS       PIC S9(9) COMP.  
  02 RET-CODE        PIC S9(9) COMP.  
  02 MSG-AREA        PIC X(500).
```

\*\*\*\*\*  
\* INDICATOR VARIABLES FOR CALL TO DSNACICS

```
*****  
01 DSNACICS-PARMI.  
  02 IND-PARM-LEVEL  PIC S9(4) COMP.  
  02 IND-PGM-NAME    PIC S9(4) COMP.  
  02 IND-CICS-APPLID PIC S9(4) COMP.  
  02 IND-CICS-LEVEL  PIC S9(4) COMP.  
  02 IND-CONNECT-TYPE PIC S9(4) COMP.  
  02 IND-NETNAME     PIC S9(4) COMP.  
  02 IND-MIRROR-TRANS PIC S9(4) COMP.  
  02 IND-COMM-AREA   PIC S9(4) COMP.  
  02 IND-COMM-LEN    PIC S9(4) COMP.  
  02 IND-SYNC-OPTS   PIC S9(4) COMP.  
  02 IND-RET-CODE    PIC S9(4) COMP.  
  02 IND-MSG-AREA    PIC S9(4) COMP.
```

## Calling DSNACICS from AQT



Folksam®

10

PARM\_LEVEL is the level of the parameter list ALWAYS 1  
PGM\_NAME is the program to be called by the mirror transaction  
CICS\_APPLID is the applid of CICS  
CICS\_LEVEL is the level of CICS. CICS TS 1.3 or higher is 2  
CONNECT\_TYPE is the connection type. GENERIC or SPECIFIC  
NETNAME is the name of specific connection (of SPECIFIC)  
MIRROR\_TRANS is the name of the mirror transaction (default CSMI)  
COMMAREA is commarea contents  
COMMAREA\_TOTAL\_LEN is total length of commarea  
SYNC\_OPTS is syncpoint options if CICS\_LEVEL is 2  
1: CICS is not allowed to perform a syncpoint (2PC requires RRS)  
2: CICS performs syncpoints (creating more than one UoW)  
RETURN\_CODE is the return code from DSNACICS. 0 or 12  
MSG\_AREA is messages from DSNACICS + CICS + DSNACICX

## DSNACICS Parameters

- PARM\_LEVEL is the level of the parameter list ALWAYS 1
- PGM\_NAME is the program to be called by the mirror transaction
- CICS\_APPLID is the applid of CICS
- CICS\_LEVEL is the level of CICS. 2 means CICS TS 1.3 or higher
- CONNECT\_TYPE is the connection type. GENERIC or SPECIFIC
- NETNAME is the name of specific connection (of SPECIFIC)
- MIRROR\_TRANS is the name of the mirror transaction (default CSMI)
- COMMAREA is commarea contents
- COMMAREA\_TOTAL\_LEN is total length of commarea
- SYNC\_OPTS is syncpoint options (if CICS\_LEVEL is 2)
  - 1: CICS is not allowed to perform a syncpoint, i.e. 2PC (requires RRS)
  - 2: CICS performs syncpoints (creating more than one UoW)
- RETURN\_CODE is the return code from DSNACICS. 0 or 12
- MSG\_AREA is messages from DSNACICS + CICS + DSNACICX

## Calling CICS Directly (from Y05DP01)

```
*****  
*** CALL CICS PROGRAM VIA EXCI (DPL)  
*****
```

```
EXEC CICS LINK PROGRAM('Y05CL11')  
      APPLID(W-TARGET-SYSTEM)  
      TRANSID('Y05S')  
      COMMAREA(W-COMM-AREA)  
      LENGTH(W-COMM-LEN)  
      RETCODE(EXCI-EXEC-RETURN-CODE)  
      SYNCONRETURN  
END-EXEC
```

```
01 W-TARGET-SYSTEM PIC X(8).  
01 W-COMM-AREA PIC X(31030).  
01 W-COMM-LEN PIC S9(4) COMP.  
01 EXCI-EXEC-RETURN-CODE.  
    02 EXEC-RESP PIC 9(8) COMP.  
    02 EXEC-RESP2 PIC 9(8) COMP.  
    02 EXEC-ABCODE PIC X(4).  
    02 EXEC-MSG-LEN PIC 9(8) COMP.  
    02 EXEC-MSG-PTR POINTER.
```

Folksam®

12

W-COMM-LEN is S9(4) for EXCI and S9(9) for DSNACICS!

```
*****  
* PARMs FOR EXCI LINK  
*****  
01 EXCI-EXEC-RETURN-CODE.  
    02 EXEC-RESP PIC 9(8) COMP.  
    02 EXEC-RESP2 PIC 9(8) COMP.  
    02 EXEC-ABCODE PIC X(4).  
    02 EXEC-MSG-LEN PIC 9(8) COMP.  
    02 EXEC-MSG-PTR POINTER.  
01 W-CICS-LINK.  
    05 W-TARGET-PROGRAM PIC X(8) VALUE 'Y05CL11'.  
    05 W-TARGET-SYSTEM PIC X(8).  
    05 W-TARGET-TRANSID PIC X(4) VALUE 'Y05S'.  
01 W-COMM-AREA PIC X(31030).  
01 W-COMM-LEN PIC S9(4) COMP.  
*****  
* PARMs FOR EXCI LINK slut  
*****
```

## Considerations

- Two-phase Commit requires CICS to be registered to RRS
- EXCI requires CICS translation (integrated or SA)
- prefix.SDFHEXCI has to be STEPLIB'ed in WLM AS
  - and APF authorized
    - SETPROG APF,ADD,DSNAME=prefix.SDFHEXCI,SMS
- Pass on Authid of Stored Procedure Caller
  - SECURITY USER on CREATE PROCEDURE
- Length in DSNACICS is a fullword
  - as opposed to EXCI where it is a halfword
- DSNACICS always calls user exit routine DSNACICX
  - can be used to change parameters sent to CICS
  - prefix.SDSNSAMP(DSNASCIO) contains sample
- COMMAREA can be changed to FOR BIT DATA



13

EXCI requires CICS translation ("precompile") either a translator integrated in a compiler or a stand alone CICS translator to be run before the compiler.

The redbook recommend that a separate WLM AE is used for EXCI.

SDFHEXCI also have to be included in SYSLIB DD at compile. DFHXCSTB is a common stub, designed for inclusion in programs written in all the supported languages.

SECURITY **USER** or DB2 or DEFINER on CREATE PROCEDURE SYSIBM.SYSROUTINES.EXTERNAL\_SECURITY in the catalog describes the defined way to handle external security.

The DSNACICX exit routine can provide default values for all the parameters or change parameters before they are sent to CICS to simplify the parameter list

DSNASCIO is a sample written in COBOL

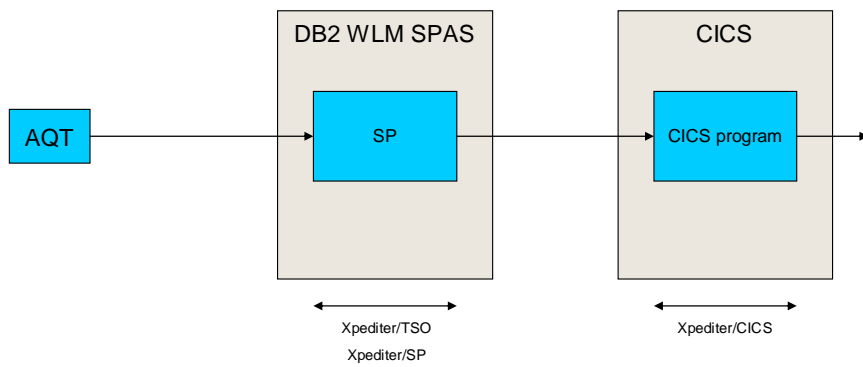
DSNASCIX is source of default exit written in Assembler

## Performance

	<b>DSNACICS</b>	<b>EXCI</b>
<b>SP</b>	(0.2 s)	(0.2 s)
<b>SP + CICS</b>	0.5 s	0.35 s
<b>SP + CICS + MQ + iSeries</b>	0.8 s	0.6 s

- Time measured by AQT
- 0.2 seconds is baseline not using DSNACICS or EXCI
  - different functionality in the three cases
- No surprises when comparing DSNACICS and EXCI

## Debugging



- There is no way to debug everything in one pass
  - Debug either Stored Procedure or CICS program

## Summary

	+	-
<b>DSNACICS</b>	Keep SPs clean (only SQL)	Architecture
	Useful when called from non- z/OS application	Performance
		Additional administration
		Problem determination
<b>EXCI</b>	Architecture	CICS translation
	Performance	APF authorized CICS library

**Folksam**<sup>®</sup>

16

According to the redbook DB2-people prefer DSNACICS while CICS-people prefer EXCI.

When running DSNACICS without SDFHEXCI being APF authorized, you will get an error message from DSNACICS. I.e., SDFHEXCI must be authorized.

However, when concatenating SDFHEXCI with unauthorized libraries in our ordinary WLM SPAS JCL, we can run the applications without any errors. To our knowledge, concatenating an authorized library with an unauthorized library will result in all libraries being viewed as unauthorized in the job. Strange...