



## Integrating ISPF Applications in RDz

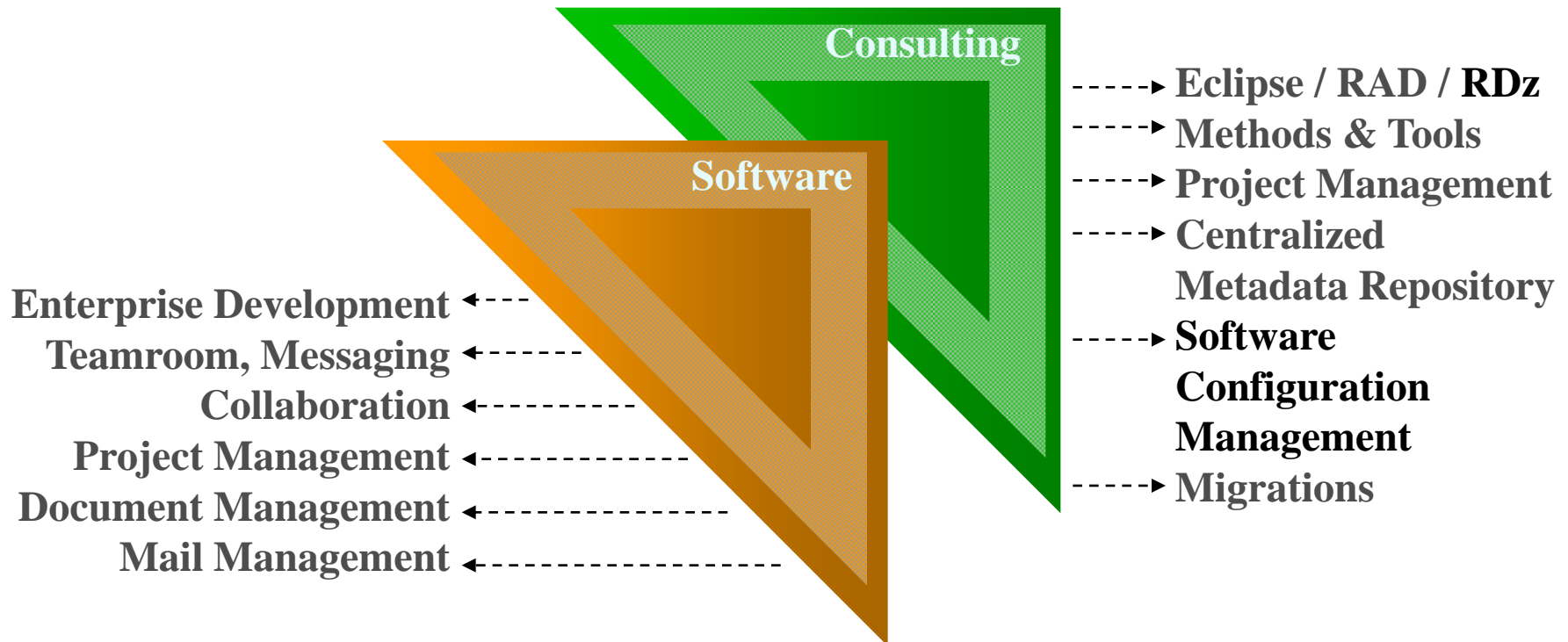
October 27, 2009

SoforTe GmbH

- Udo Partsch
- Joachim Kaltenbach

- Introduction SoforTe
- Problem Statement
- Solutions
  - Menu Manager
  - RDz plug-in development
  - HATS
  - CARMA
- Evaluation

## Your Specialist for the Software Development Process



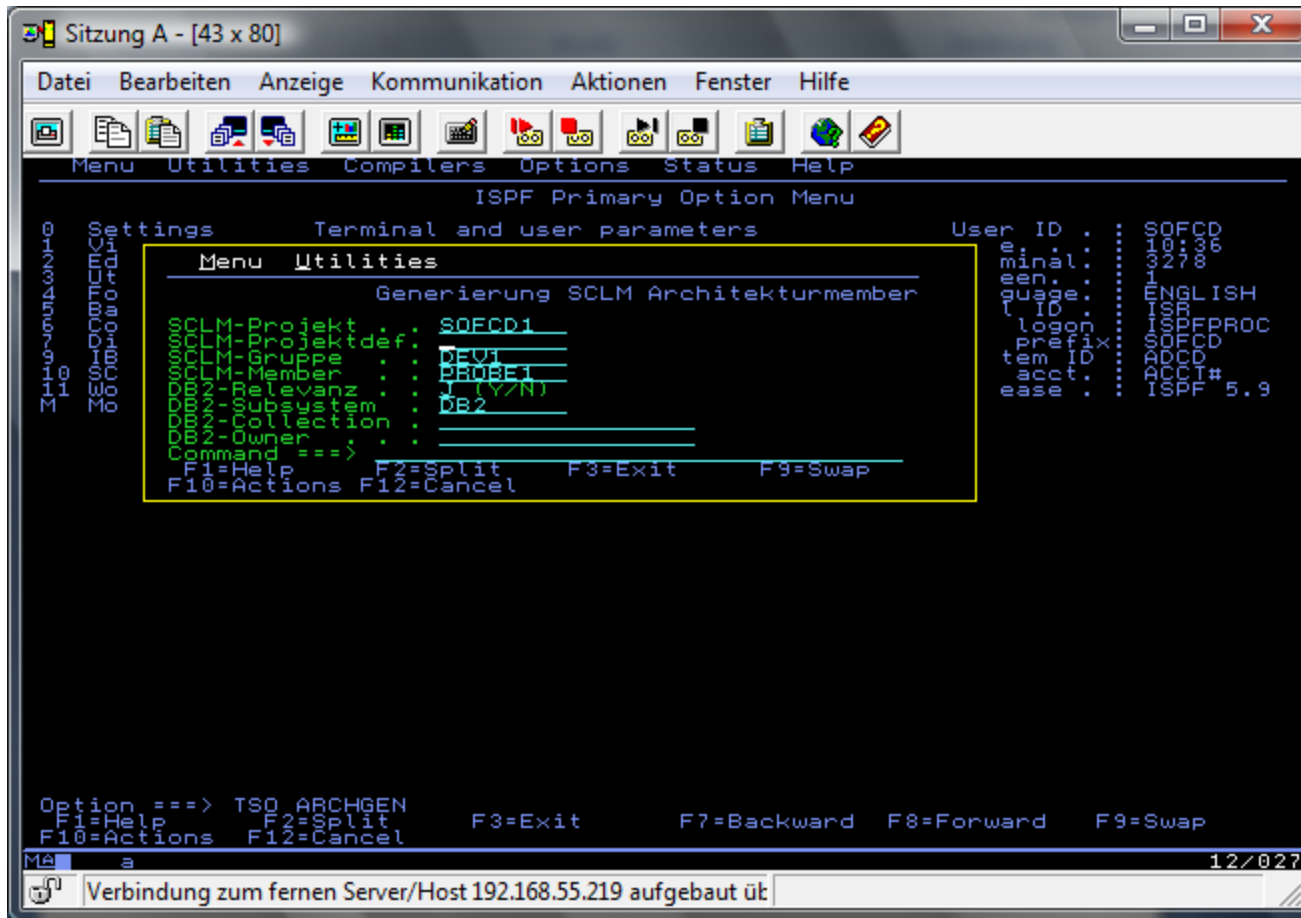
## Initial Situation

- Missing integration of customer applications and functions in RDz
- Continued need for the 3270 emulator

## Motivation

- Standard look and feel
  - Fast access to functions
  - Higher productivity
  - User acceptance

- Sample: Generating an SCLM architecture definition
- How to integrate this ISPF dialogue in RDz?



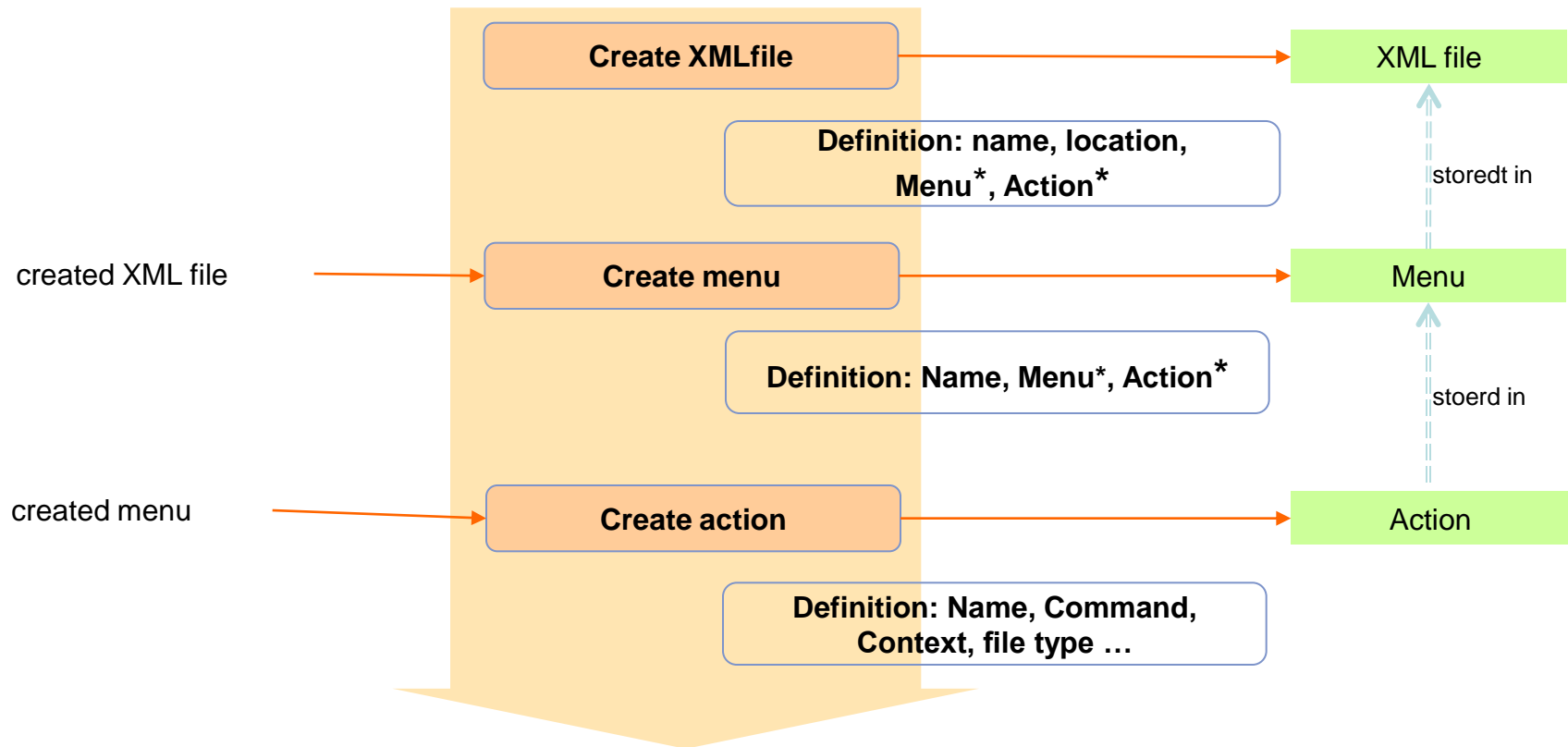
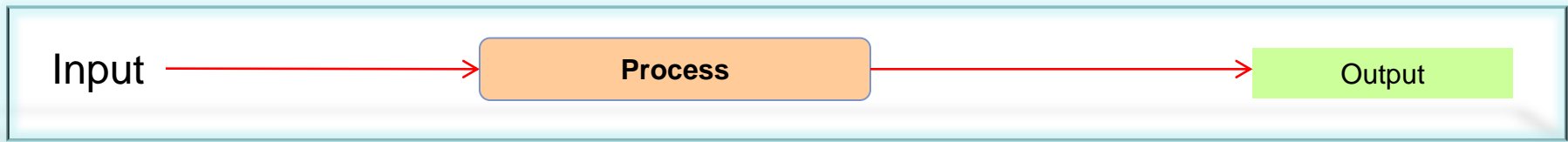


**SOFORTE**  
SOLUTIONS FOR TEAMS

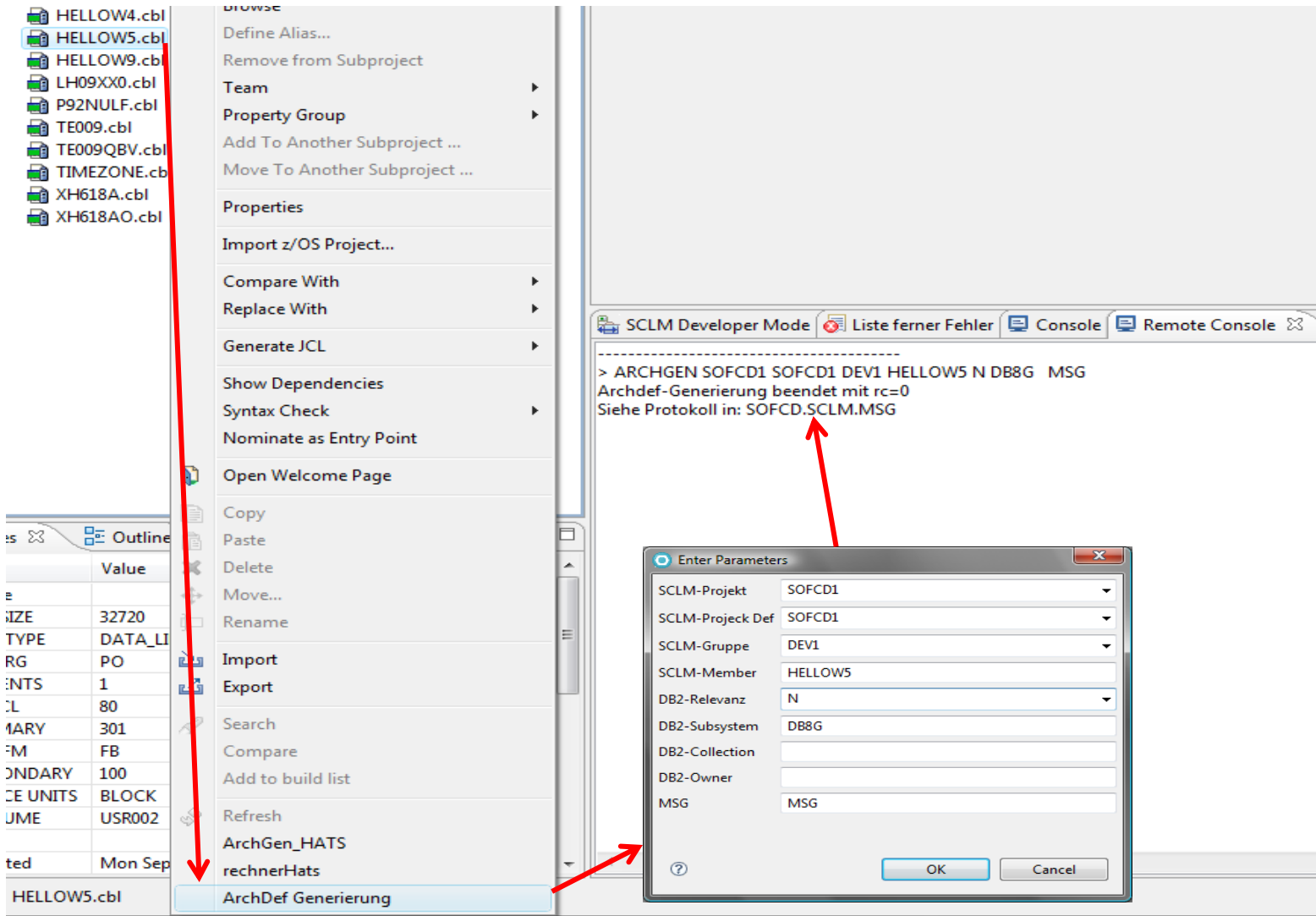
# Menu Manager

- What is the Menu Manager?
  - a graphical Interface to:
    - Extend the context menu
    - Create Pop-up Menus
  - available for the:
    - z/OS project view
    - LPEX Editor
  
- RDz configuration, stored as an XML file
  
- The created XML file can be imported into or distributed within RDz
  
- Actions are configured and connected to file types (e.g. CBL)

# Menu Manager: Procedure



# Menu Manager Sample



The screenshot displays the SCLM Developer Mode interface. On the left, a file tree lists several CBL files, with HELLOW5.cbl selected. A context menu is open over HELLOW5.cbl, showing options like 'Define Alias...', 'Remove from Subproject', 'Team', 'Property Group', 'Add To Another Subproject ...', 'Move To Another Subproject ...', 'Properties', 'Import z/OS Project...', 'Compare With', 'Replace With', 'Generate JCL', 'Show Dependencies', 'Syntax Check', 'Nominate as Entry Point', 'Open Welcome Page', 'Copy', 'Paste', 'Delete', 'Move...', 'Rename', 'Import', 'Export', 'Search', 'Compare', 'Add to build list', 'Refresh', 'ArchGen\_HATS', 'rechnerHats', and 'ArchDef Generierung'. The 'ArchDef Generierung' option is highlighted. Below the file tree is an 'Outline' table with columns 'Name' and 'Value'. The console window on the right shows the command '> ARCHGEN SOFCDD1 SOFCDD1 DEV1 HELLOW5 N DB8G MSG' and the output 'Archdef-Generierung beendet mit rc=0' and 'Siehe Protokoll in: SOFCD.SCLM.MSG'. An 'Enter Parameters' dialog box is open in the foreground, with fields for SCLM-Projekt (SOFCDD1), SCLM-Projekt Def (SOFCDD1), SCLM-Gruppe (DEV1), SCLM-Member (HELLOW5), DB2-Relevanz (N), DB2-Subsystem (DB8G), DB2-Collection, DB2-Owner, and MSG (MSG). Red arrows indicate the flow from the context menu to the dialog box and then to the console output.

Name	Value
SIZE	32720
TYPE	DATA_LI
RG	PO
INTS	1
IL	80
INARY	301
IM	FB
NDARY	100
CE UNITS	BLOCK
UME	USR002
ted	Mon Sep
HELLOW5.cbl	

## Advantages:

- Easy to create
- No Java skills necessary

## Disadvantages:

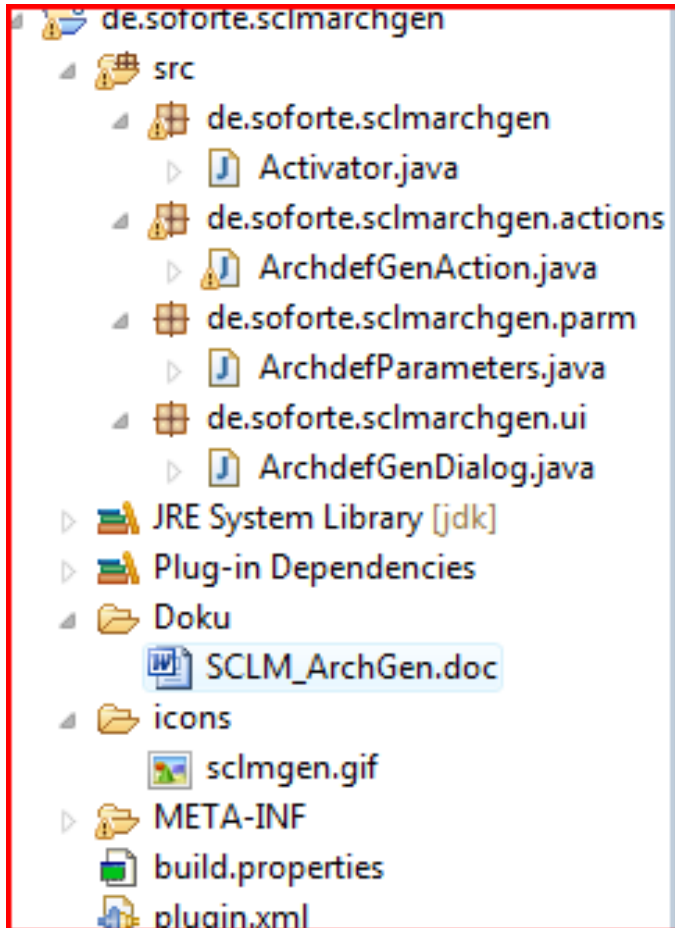
- Only supported using the context menu of the z/OS projects view or the LPEX editor
- Limited functions
  - No checking of input fields
  - No formatting of the output
- The ISPF dialogue application has to be maintained
  - Parameter passing
  - ISPF panel cannot be called up



**SOFORTE**  
SOLUTIONS FOR TEAMS

# RDz Plug-In Development using the Resource API

- **Plug-in development**
  - Create an action
  - GUI
  - Calling the ISPF function
  - Error handling
  - Output formatting
  
- **RDz resource API**
  - Java programming interface
  - Host communication
  - Calling TSO/ISPF functions
  
- **Plug-in has to be installed in RDz**



```
ZOSSystemImage si =  
    SystemSelector.getInstance().getConnectedSystem();
```

```
setZOSUserID(si.getUserID());
```

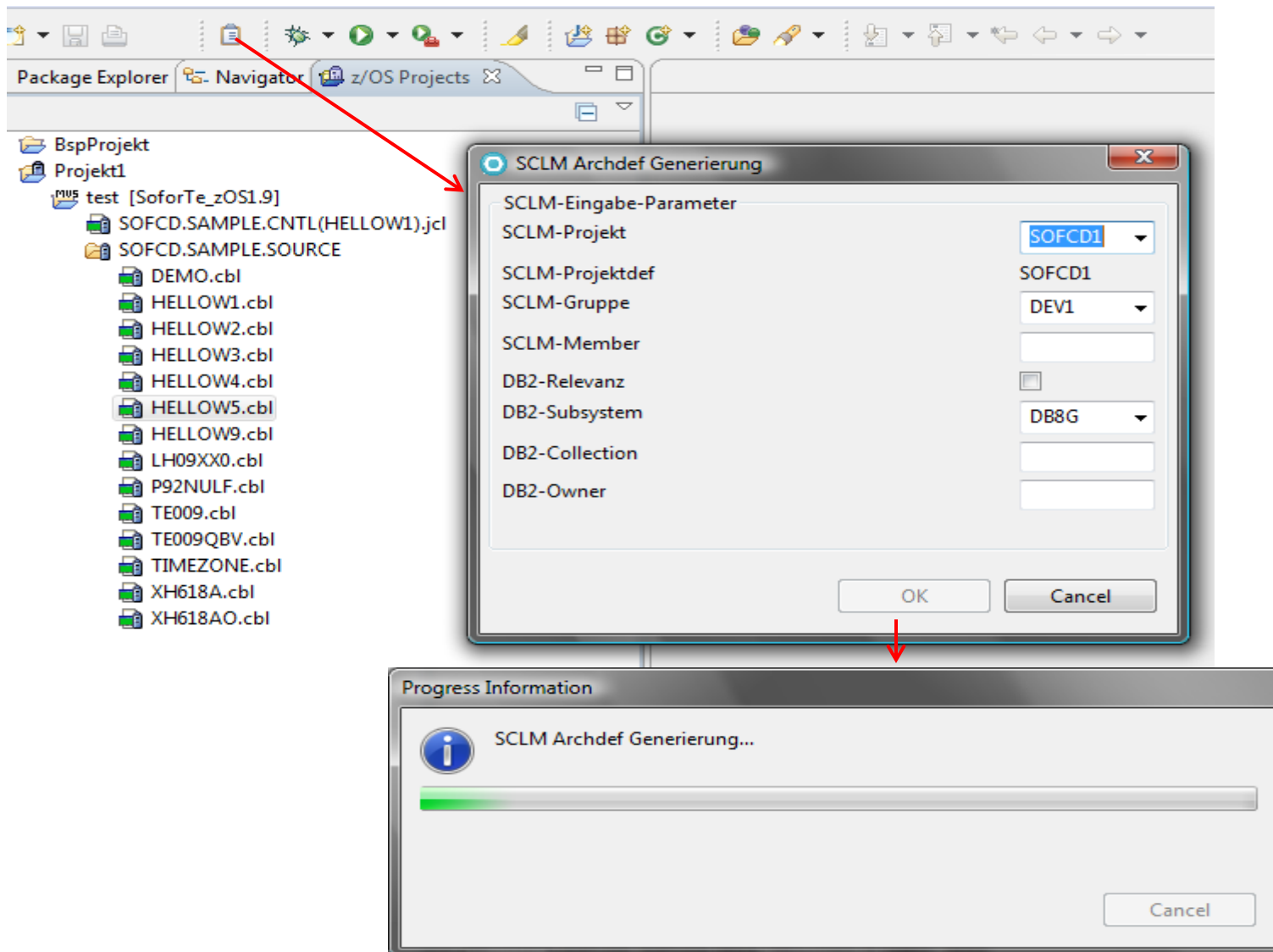
```
String pCommand = "ARCHGEN " + sclmProj + " "  
    + firstQual + " " + sclmGroup + " " + progName  
    + " " + db2Flag + " " + db2Sys+ " "+ owner + " "  
    + collID+ " " + msgDD;
```

```
_lastCommand = si.getCommandObject();
```

```
_lastCommand.setCommandString(pCommand);
```

```
_lastCommand.run();
```

# Sample



## **Advantages:**

- Flexibility
- User friendly
- Standard Eclipse plug-in installation process

## **Disadvantages:**

- Highly skilled plug-in developers necessary
- High programming effort
- The ISPF dialogue application has to be maintained
  - Parameter passing
  - ISPF panel cannot be called up



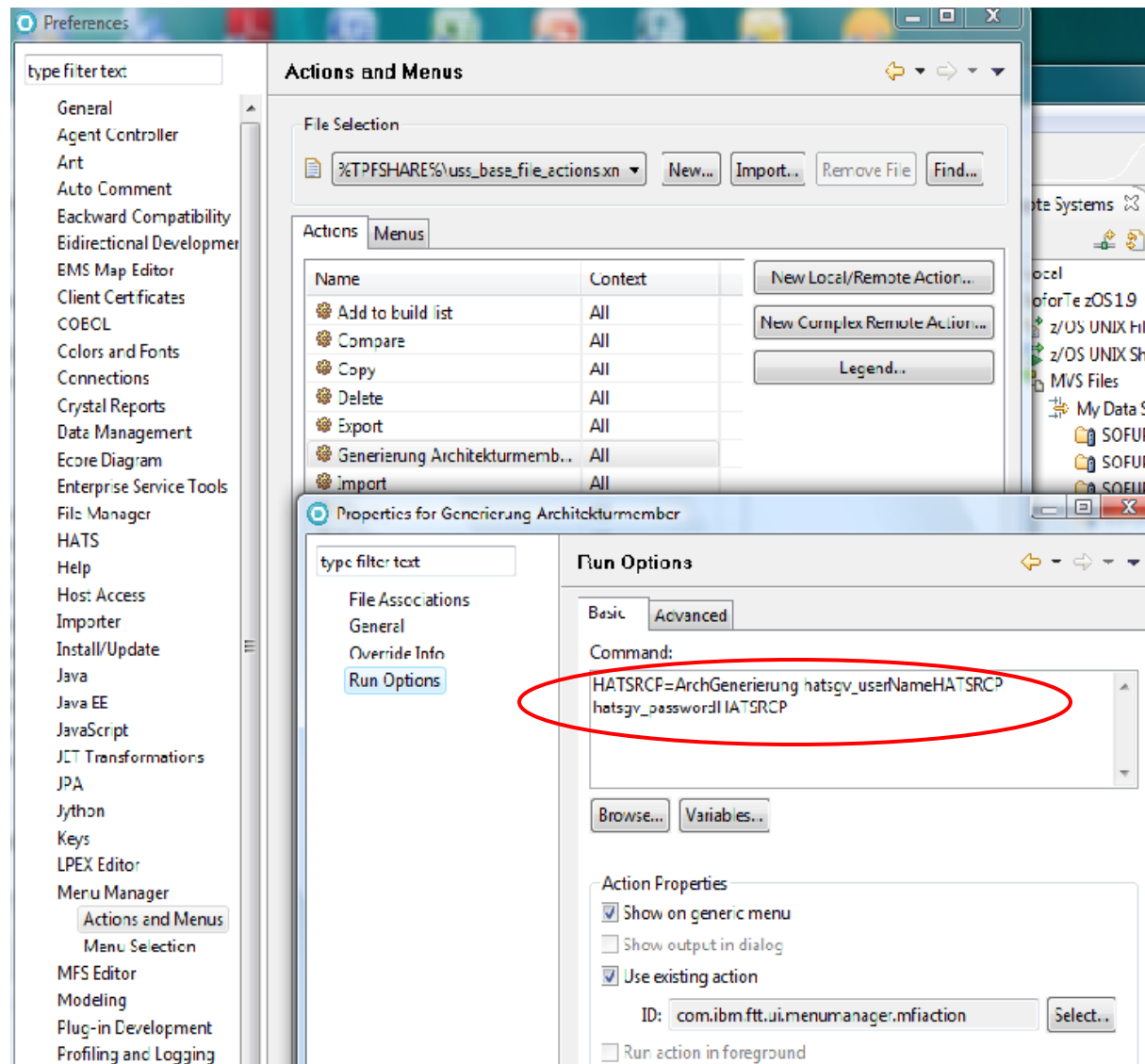
**SOFORTE**  
SOLUTIONS FOR TEAMS

# HATS RCP

- HATS
  - Host Application Transformation Services
- GUI-fying of any 3270 application (IMS, CICS, ISPF, ...)
- Integration of 3270 applications in
  - Web applications
  - **RCP (Rich Client Platform)**
    - E.g.: Eclipse Look and Feel of an ISPF application
  - Possibility to integrate a mainframe application with other desktop applications

- Software pre-requisites for the development of a HATS/RDz application
  - RDz 7.5.1 with RAD 7.5.2
  - HATS 7.5.0.1
  
- Generating an RCP application from an ISPF application using HATS
  - Configure screen identifiers
  - Recording macros
    - Logon -, logoff- procedure , ...
    - Menu navigation
  - Optimizing the look and feel of the generated GUI:
    - Integrate GUI elements
    - Hiding fields
    - Add new fields / buttons ....
  
- Integration of HATS applications with RDz:
  - Using the RDz Menu Manager to integrate a HATS application in the context menu
  - Using the HATS API with plug-in development
  
- Special features
  - Reuse RDz logon user and password to connect to the HATS session
  - Possibility to call any RDz function (e.g. LPEX editor) in a HATS application (JAVA development).

# Configuring a HATS Application in RDz Menu Manager



The image shows two overlapping windows from the RDz Menu Manager. The top window is titled "Preferences" and is on the "Actions and Menus" tab. It shows a "File Selection" field with the path `%TPFSHARE%\uss_base_file_actions.xn` and buttons for "New...", "Import...", "Remove File", and "Find...". Below this is a table of actions:

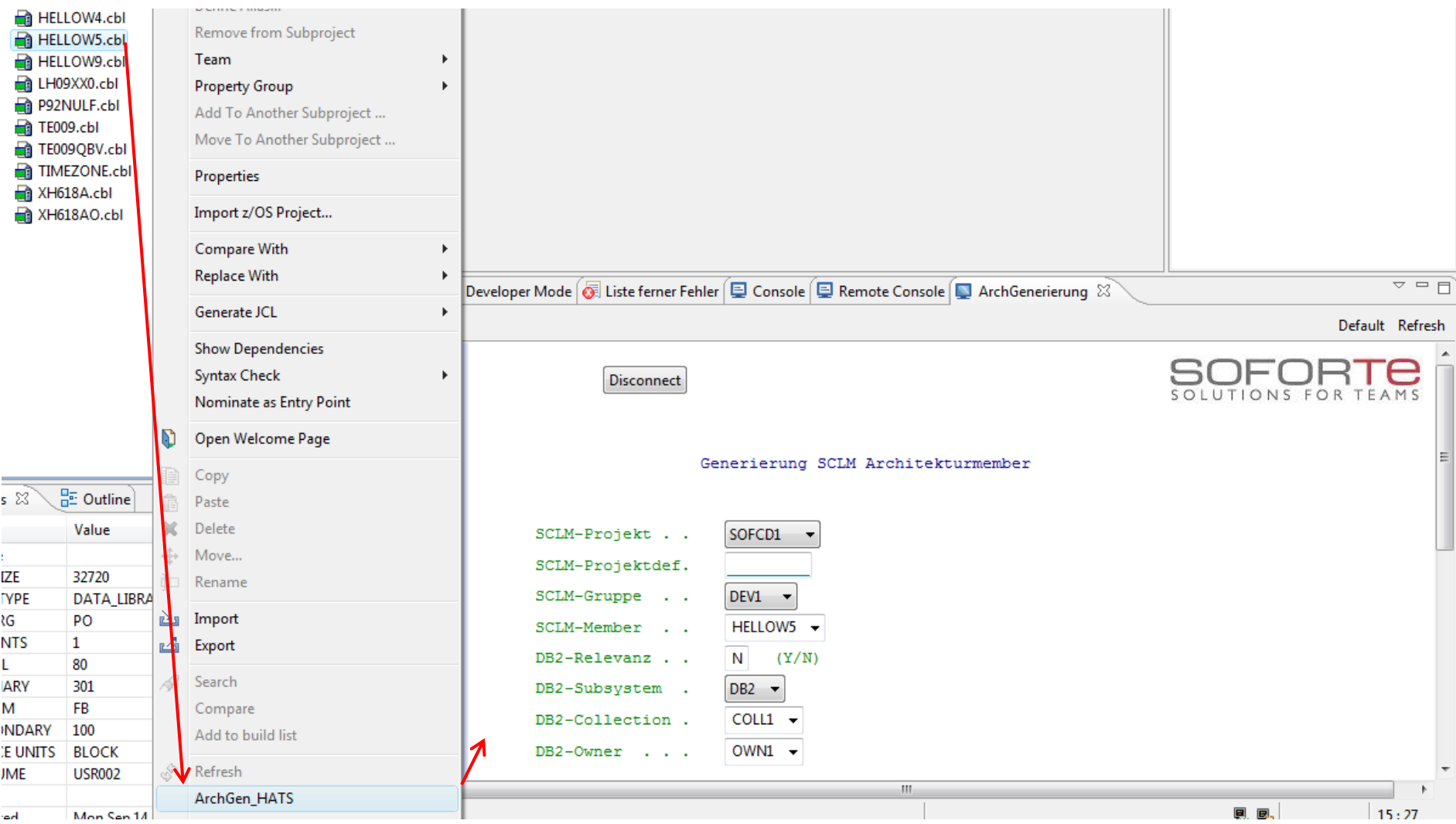
Name	Context
Add to build list	All
Compare	All
Copy	All
Delete	All
Export	All
Generierung Architekturmemb..	All
Import	All

The bottom window is titled "Properties for Generierung Architekturmembcr" and is on the "Run Options" tab. The "Command:" field contains the following text, which is circled in red:

```
HATSRCP=ArchGenerierung hatsgv_userNameHATSRCP  
hatsgv_passwordHATSRCP
```

Below the command field are "Browse..." and "Variables..." buttons. The "Action Properties" section includes checkboxes for "Show on generic menu" (checked), "Show output in dialog" (unchecked), and "Use existing action" (checked). The "ID:" field contains `com.ibm.ftt.ui.menumanager.mfiaction`.

# HATS Sample



The screenshot displays the SoforTe IDE interface. On the left, a file explorer shows a list of CBL files, with 'HELLOW5.cbl' selected. A context menu is open over this file, listing various actions such as 'Remove from Subproject', 'Team', 'Property Group', 'Add To Another Subproject ...', 'Move To Another Subproject ...', 'Properties', 'Import z/OS Project...', 'Compare With', 'Replace With', 'Generate JCL', 'Show Dependencies', 'Syntax Check', 'Nominate as Entry Point', 'Open Welcome Page', 'Copy', 'Paste', 'Delete', 'Move...', 'Rename', 'Import', 'Export', 'Search', 'Compare', 'Add to build list', 'Refresh', and 'ArchGen\_HATS'. The 'ArchGen\_HATS' option is highlighted with a red arrow. Below the file list is an 'Outline' table with columns for 'Value' and 'Type'. The main workspace shows a configuration window titled 'Generierung SCLM Architekturmember' with a 'Disconnect' button and the SoforTe logo. The configuration parameters are as follows:

Parameter	Value
SCLM-Projekt	SOFC D1
SCLM-Projektdef.	
SCLM-Gruppe	DEV1
SCLM-Member	HELLOW5
DB2-Relevanz	N (Y/N)
DB2-Subsystem	DB2
DB2-Collection	COLL1
DB2-Owner	OWN1

## Advantages:

- No customizing of the ISPF application necessary
- Get Eclipse look and feel without programming
- Information integration
  - Merging the TUI of different applications in one GUI
- Standard Eclipse plug- in installation process

## Disadvantages:

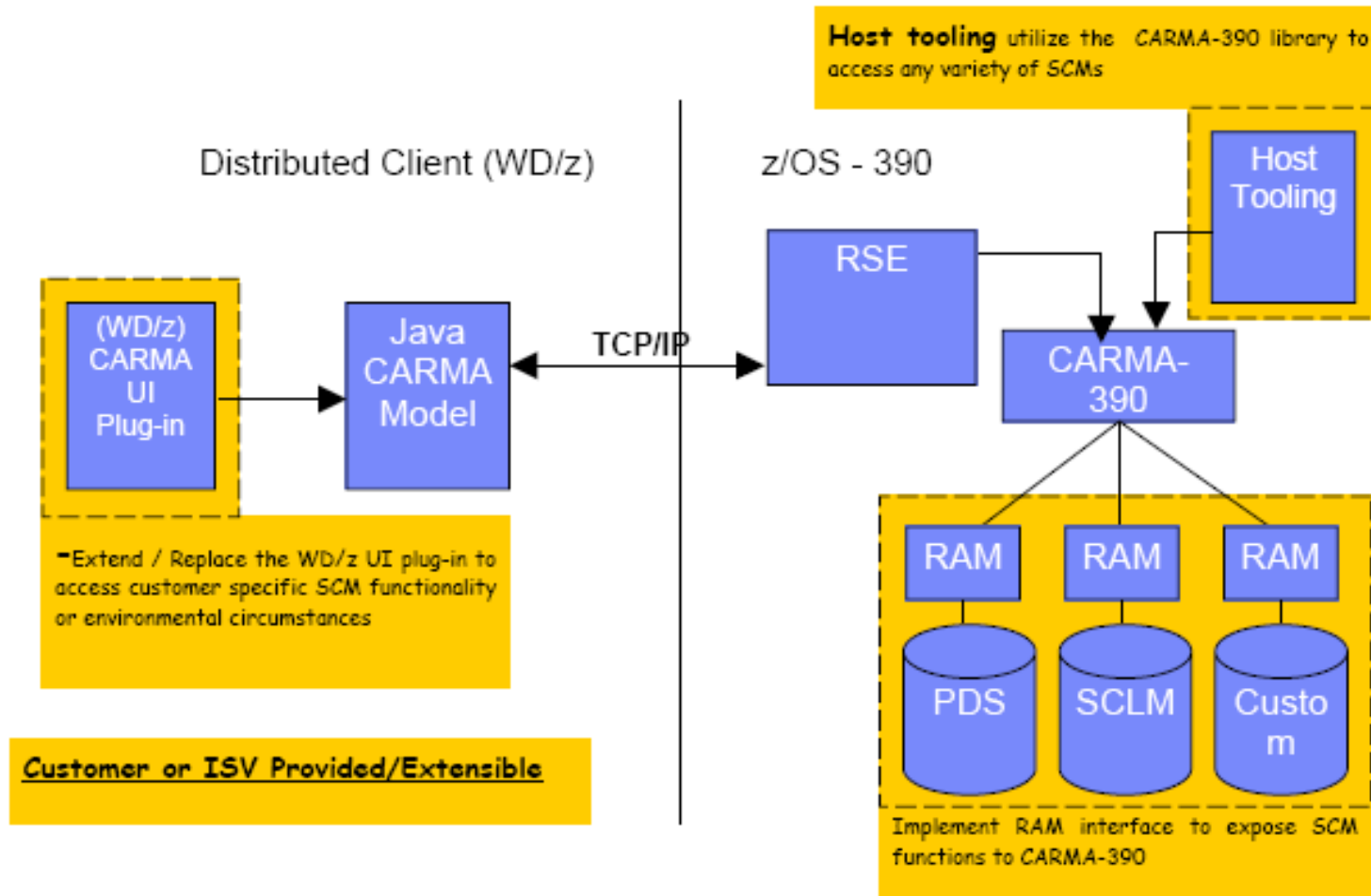
- Only one HATS ISPF application can be active at one time (an application always starts with TSO logon)
- To start a HATS ISPF application the user must be logged off from TSO
- Higher test and maintenance effort after changing the ISPF application



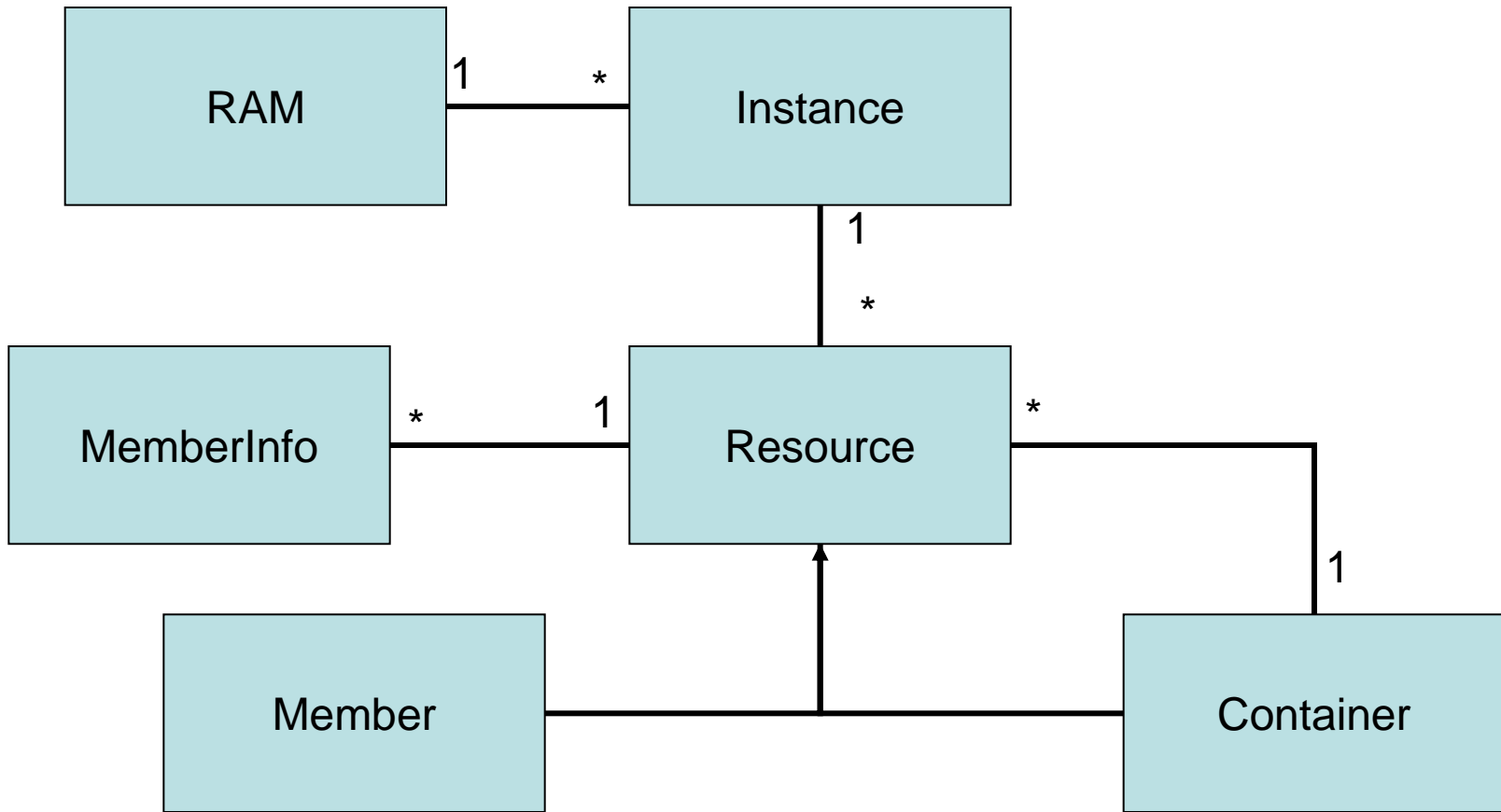
# CARMA

- CARMA
  - Common Access Repository Manager  
SCM framework as part of RDz
- Generic interface to integrate any z/OS SCM system in RDz
- CARMA is a customizable framework.
- The interface already supports the most important SCM functions
- User defined functions can be added.

# CARMA Technical Architecture



# CARMA Model



- **CARMA Server**  
an ISPF Task per user started by RDz/RSE
  - No batch job necessary
  - ISPF allocations customizable
  - Reuse existing mainframe tools (REXX etc.)
  
- **Generic CARMA client in RDz (CARMA view)**
  - All RAM definitions and functions are dynamically available using the RDz CARMA view
  - CARMA can be used without any JAVA plug-in development (reusing mainframe skill)
  
- **CARMA Java API**
  - Using Java plug-in development to create a user defined client
  - Extending / maintaining the generic CARMA client
  - Flexibility to integrate any defined CARMA function in RDz (menu actions, context menu, ...)

## ➤ RAM

- RAM implementation in C, but ...
  - Calling REXX for each defined CARMA function
  - Reuse existing SCM functions
- 2 VSAM files
  - Define functions, function parameters, ...

## ➤ CARMA client

- Extending the generic CARMA client if necessary

# CARMA Sample – Calling up the REXX Master Procedure for each function in the C-program (RAM)

---

```
.....
int performAction(int actionID, char instanceID[256],
                  char memberID[256], void** params,
                  void*** customReturn, char error[256]) {
.....
switch (actionID) {
.....
case ARCHG_ACTION:
.....
    memcpy(fkt, "ArchG", 5);
    sprintf(buf, "%i %.24s %.256s %.256s %c %.8s %.8s %.8s",
            &rexxRes1, fkt, instanceID, memberID, db2Opt,
            db2Sys, db2Owner, db2CollId);

    rexxRes = execREXXProcedure("SCLMCAST", 1, buf);

    memcpy(rexxRes, rexxRes1, 300);
.....
    retc = atoi(rexxRes);
    memcpy(error, rexxRes+10, 256);

    break;      /* end of case ARCHG_ACTION */
}
.....
```

# CARMA Sample – Calling the ARCHGEN Function in the Master REXX Procedure SCLMCAST

---

.....

**When fkt = 'ARCHG' Then Do**

```
Parse Var rest instanceid memberid db2opt db2sys ,  
         db2owner db2coll
```

```
db2coll = Strip(db2coll)
```

```
Call ISPF '0 SELECT CMD(SCLMCAAG 'instanceid' 'memberid' ',  
db2opt' 'db2sys' 'db2owner' 'db2coll' 'resadr')
```

End

.....

# CARMA Sample – REXX Procedure SCLMCAAG to call the reusable ARCHGEN Function

---

```
.....
Arg instanceID memberID db2opt db2sys db2owner collid
  resadr

instanceID = Strip(instanceID)
memberID = Strip(memberID)
user = USERID()
pref = Substr(memberID,1,3)

.....
grp = Substr(instanceID,4)
lp_strich = Lastpos('-',memberID)
type = Substr(memberID,4,lp_strich-4)
mem = Strip(Substr(memberID,lp_strich+1))
ADDRESS ISPEXEC "SELECT CMD(ARCHGEN "sclmproj" ",
  "sclmproj" "grp, mem" "db2opt" "db2sys" "db2owner,
  " "collid")"

Call Exit retc 0 errmsg
.....
```



## **Advantages:**

- Reusable mainframe skills (REXX, ISPF, ...)
- If using only the generic client:
  - No plug-in development and no Java skills necessary
  - No client installation necessary
- Flexibility when integrating mainframe SCM systems

## **Disadvantages:**

- Only applicable for integrating SCM systems
- The ISPF dialogue application has to be maintained
  - Parameter passing
  - ISPF panel cannot be called up
- Some standard RDz functions are not available if working with the CARMA view
  - No standard error feedback

# Evaluation

---

	<b>Effort</b>	<b>Flexibility</b>	<b>Performance</b>	<b>Techn. Skills</b>
Menu Manager	low	low	good	low
Plug-in development	very high	very high	good	high (normally not available)
HATS RCP	medium	medium	very good	Medium (normally not available)
CARMA	high	high	good	high (mostly available)

## 4 alternatives integrating ISPF tools in RDz:

- **Menu Manager**
  - Integrating non complex tools
  - Working with the RDz project view
  
- **RDz plug-in development**
  - High flexibility to integrate tools
  - High effort, high skill
  
- **HATS**
  - Integrating complex mainframe tools, e.g.
    - ISPF tool is not maintainable
    - No source available
  - Cannot use two HATS applications for TSO/ISPF at the same time
  
- **CARMA**
  - Integrating a mainframe SCM tool
  - Mainframe configuration and programming

---

THANK YOU

