Christoph Wildensee

Aufruf nicht remotefähiger Funktionsbausteine per remotefähigem „Tunnel-Baustein“ in SAP

1. Einleitung

Die Methodik des „Remote Function Call“ (RFC), im ursächlichen Sinn der systemübergreifende Aufruf von funktionalen/aufgabenorientierten Codebausteinen mit entsprechenden Übergabestrukturen, bildet eine der Grundlagen des modularen Programmierens in SAP. So wird eine Vielzahl solcher Bausteine für unterschiedliche Aufgaben zur Verfügung gestellt. Verschiedene kritische Aspekte wurden in den letzten Jahren mit revisorischem Blick bereits hinreichend dargelegt. Eine weitere Facette bildet die Möglichkeit, unter bestimmten Bedingungen – entsprechend starke Berechtigungsobjektausprägungen in einer Rolle des aufrufenden Benutzers vorausgesetzt – nicht remotefähige Funktionsbausteine über das Hilfskonstrukt eines bestehenden remotefähigen Bausteins per RFC in einem entfernten System aufzurufen und damit externen Programmen interne Funktionen zur Verfügung zu stellen.

2. Grundlagen und Vorgehen

Der Aufruf von remotefähigen Funktionsbausteinen und die Absicherung per Berechtigungsobjekt **S_RFC** sind bekannt. Nicht selten werden aus diesen Funktionsbausteinen wiederum Bausteine aufgerufen, die oftmals aus der identischen Funktionsgruppe, nicht selten aber auch aus differierenden Funktionsgruppen stammen. Der aufrufende Nutzer des SAP-Systems mit externen Mitteln (anderes SAP-System oder auch autonomes Programm) benötigt in der Berechtigungsbereitstellung mindestens im Objekt S_RFC die Aktivität (ACTVT) = 16 (Ausführen), als Angabe zum Typ des aufrufbaren RFC-Objektes (RFC_TYPE) = FUGR und die Nennungen der aufzurufenden RFC-Objekte (RFC_NAME), was den Funktionsgruppen entspricht, die für die Bausteinnutzung benötigt werden (Funktionsbausteine: Tabelle TFDIR; Funktionsgruppen zu den Bausteinen: Tabelle ENLFDIR; Parameter von Funktionsbausteinen: Tabelle FUPARAREF).

Die Tunnelfunktion kann am Baustein /SDF/GEN_FUNCS_FUNC_CALL der Funktionsgruppe /SDF/GEN_FUNCS erläutert und nachvollzogen werden, der als remotefähiger Baustein aus dem Bereich „Solution Manager General Functions“ mit der Beschreibung „Externer Aufruf eines Funktionsbausteins“ auch externen Zugriffen zur Verfügung steht.

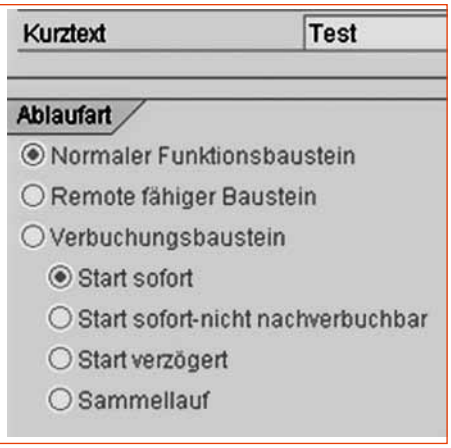
Der Aufbau des Bausteins sieht wie folgt aus:

```
FUNCTION      /SDF/GEN_FUNCS_FUNC_CALL.
        IMPORTING
        VALUE(FUNCNAME) LIKE ENLFDIR-FUNCNAME
        EXPORTING
        VALUE(P_FAILED) LIKE SY-SUBRC
        TABLES
        PT_IMPORTING STRUCTURE /SDF/RSIMP OPTIONAL
        PT_EXPORTING STRUCTURE /SDF/RSEXP OPTIONAL
        EXCEPTIONS
        WRONG_FUNCTION_CALL
        SYNTAX_ERROR
        NO_AUTHORITY
```

Über den Importparameter wird ein Funktionsbaustein mitgegeben, der im Zielsystem ausgeführt wird.

Um zunächst kontrolliert einschätzen zu können, wie dieser Baustein arbeitet, wird ein neuer FuBa Z_TESTNONRFC angelegt, der einen Datensatz in eine Tabelle im Kundennamensraum schreibt.

```
FUNCTION Z_TESTNONRFC.
*" Lokale Schnittstelle
*' IMPORTING
*'          VALUE(TPLNR) TYPE TPLNR
*' EXPORTING
*'          VALUE(ZTPLNR) TYPE TPLNR
*'_____________________________________________
_________-
TABLES: Zxxxxxxxxx.
ZTPLNR = TPLNR.
Zxxxxxxxxx -ZFWTPLNR = ZTPLNR.
INSERT Zxxxxxxxxx.
  ENDFUNCTION.
```



Dieser mitzugebende Baustein ist nicht remotefähig, um die Tunnelwirkung zu demonstrieren. Der Bausteinname Z_TESTNONRFC wird über den Importparameter FUNCNAME mitgegeben. Über den TABLES-Bereich PT_IMPORTING erfolgt die Mitgabe der Importparameter, die der aufzurufende Baustein benötigt. So wird beispielsweise der Wert 1200 in das Feld ZFWTPLNR der Tabelle Zxxxxxxxxx (Testtabelle) im RFC-Zielsystem geschrieben.

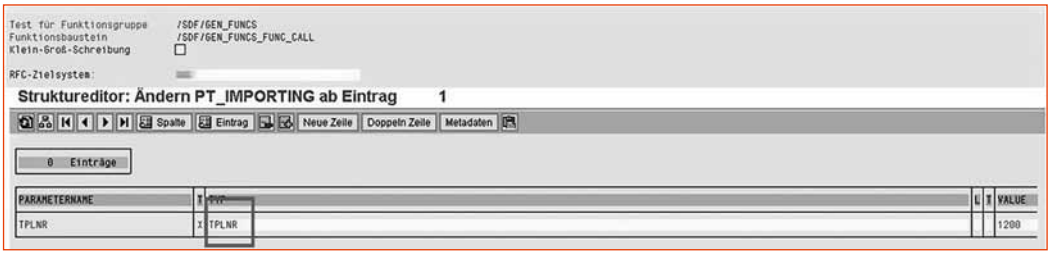


Abb. 1: Aufruf des Funktionsbausteins /SDF/GEN_FUNCS_FUNC_CALL mit Importwerten für PT_IMPORTING.

Die Überprüfung der Testtabelle vor (oben) und nach (unten) der Manipulation zeigt, dass der Aufruf erfolgreich war – der Beispielwert 1200 wurde in die Tabelle geschrieben.

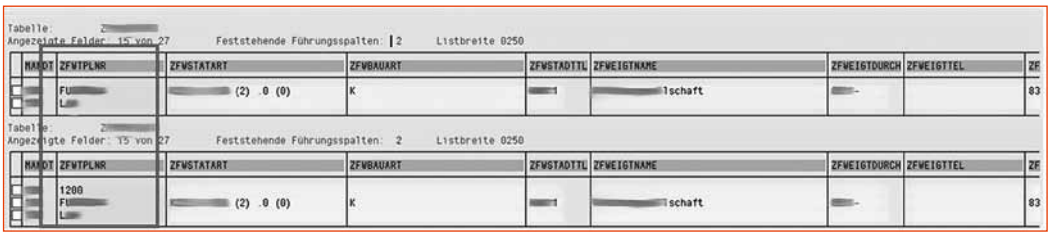


Abb. 2: Content der Testtabelle vor und nach INSERT.

Zu beachten ist, dass der zu rufende Baustein vom Tunnelbaustein zusammengesetzt wird, d. h., es werden alle benötigten Informationen zum gerufenen Baustein aus den Objektsteuerungstabellen herausgelesen und dann zusammengesetzt, bevor die Füllung der Parameter erfolgt; daher kann man das zugrundeliegende Konstrukt auch als Tunnelbau oder „Tunneling“ bezeichnen. Hierbei kann nicht jeder Baustein passend übergeben werden, denn der Vorfüllung der Importschnittstelle PT_IMPORTING (und auch der Rückgabe über PT_EXPORTING) sind Grenzen gesetzt, doch lassen sich hierüber einige kritisch-manipulierende Bausteine aufrufen, von denen bisher angenommen wurde, dass – ob durch SAP bereitgestellt oder entwickelt im Kundennamensraum – die fehlende Remotefähigkeit für einen gewissen Verwendungsschutz bzw. eine Begrenzung auf die Entwicklerebene mit analogen Transport- und Abnahmewegen sorgt. Im Zuge dessen können Restriktionen als Kopie im Kundennamensraum auch aufgehoben werden. PT_EXPORTING muss wiederum bei einer Datenmanipulation nicht vorgegeben werden, da das Ergebnis nicht zwingend als erfolgreiche Aktion aus dem Baustein heraus zurückgegeben und dann im Aufrufprogramm aufgefangen bzw. als Rückgabebehandlung verarbeitet werden muss, denn es reicht, den Erfolg über die Einsicht der manipulierten Tabelle(n) über z. B. Transaktionen wie SE16, SE16N etc. zu überprüfen.

3. Restriktion

Wie bereits erwähnt, kann nicht jeder Baustein über diese Methode aufgerufen werden. Die wesentliche Einschränkung wirkt sich über das Importing-Übergabeformat aus. Das Parameterformat des aufzurufenden Bausteins (IMPORTING … VALUE(…) TYPE…) darf nicht vom Typ C (einfaches char; TYPE C) sein, da ansonsten der Aufrufparameter abgeschnitten wird (z. B. in dem kritischen Funktionsbaustein DB_EXECUTE_SQL der Importparameter STMT für ein SQL-Statement vom Bezugstyp C). Bei der beschriebenen Nutzung des Tunnelbausteins – nachzuvollziehen im Debugging – entsteht in der Wirkung des Zusammensetzens der Bausteinkomponenten, dass ein Parameter vom Typ C einstellig ist (Speicherallokation) und daher die Value-Übergabe – da erst folgend integriert bzw. zugewiesen – abgeschnitten wird. Eine Anpassung im Kundennamensraum ist natürlich möglich (Bezugstypänderung).

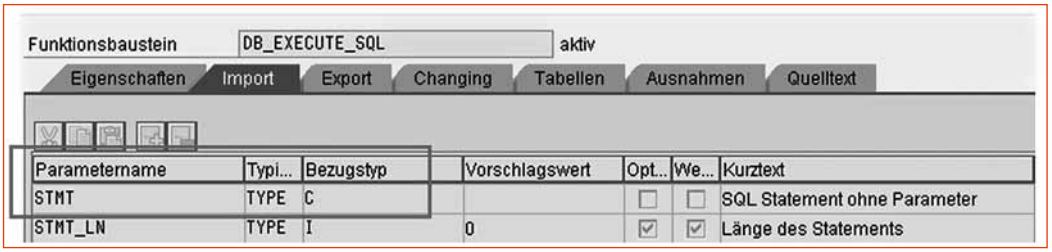


Abb. 3: Importparameter STMT des Bausteins DB_EXECUTE_SQL.

Als Ergebnis bleibt festzuhalten, dass der Zugriff auf nicht remotefähige Bausteine über den Bausteinaufruf /SDF/GEN_FUNCS_FUNC_CALL mit entsprechender Übergabestruktur sowohl mit SAP-eigenen als auch fremden Mitteln möglich ist und zu entsprechenden Datenmanipulationen führen kann.

4. Verifizierung

Der Funktionsnachweis anhand eines SAP-eigenen Standardbausteins soll zeigen, dass die praktische Relevanz hoch ist. Der **nicht remotefähige Baustein PRGN_ADD_USER_ASSIGNMENT** der Funktionsgruppe PRGN_EXCHANGE (Schnittstelle zum HR und Benutzerstamm; Benutzer einer Rolle zuweisen) wird über den Tunnelbaustein aufgerufen. Dabei wird an den offiziell definierten Genehmigungsverfahren vorbei beispielhaft einem existierenden Modulbetreuerstammsatz durch ihn selbst erfolgreich in einem anderen System eine bestehende Rolle (als Beispiel die Revisionsrolle, jedoch beliebig aus der Tabelle AGR_DEFINE wählbar) zugewiesen. Seine Transaktions- und Berechtigungsobjektzuweisungen sind dort zwar hinsichtlich des Fachmoduls stark ausgeprägt, jedoch fehlen die analogen Transaktionsberechtigungen für eine solche Aktion.

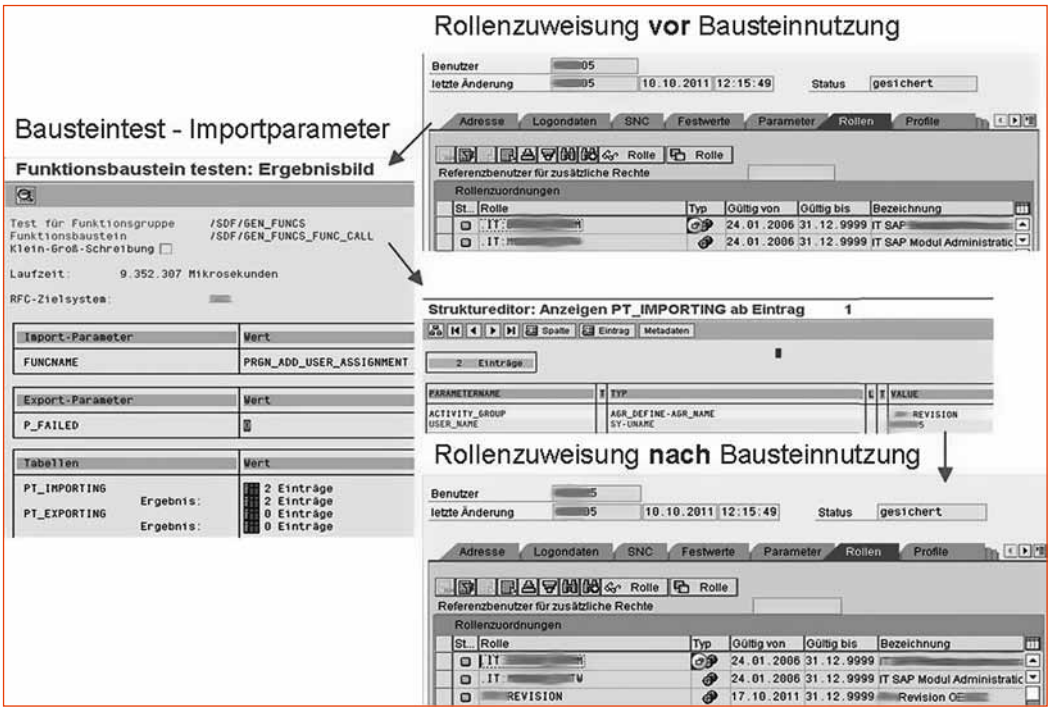


Abb. 4: Aufruf von PRGN_ADD_USER_ASSIGNMENT über den Tunnelbaustein mit Rollenzuweisung.

5. Fazit

Das Vorgehen zeigt, dass SAP entsprechend brisante Funktionen regelmäßig selbst implementiert. **Zu diesen existieren folgend nur unzureichend Sicherheitseinschätzungen und dokumentierte „Cross-Module“-Auswirkungsanalysen auf andere Funktionalitäten**, die es Kunden erleichtern würden, bewusst mit derartigen Risiken umzugehen. Immer dort, wo eine Tür – und sei es zweckgebunden wie diese Implementierung – mit den bekannten RFC-Unzulänglichkeiten geöffnet wird, kann diese auch für zweckfremde Aktivitäten bei häufig üblichen, stark ausgeprägten Objektzuweisungen in den Rollen genutzt werden. Weiterhin können auch im Kundennamensraum remotefähige und erweiterte / angepasste Funktionen implementiert sein / werden, die Tore öffnen.

Kurz gefasst ergibt sich folgendes Bild:

- Bei Vorliegen stark ausgeprägter Berechtigungsobjekte in den Rollen – speziell im Basisobjektbereich (S*) – ist das Fehlen von Transaktionsberechtigungen häufig irrelevant. **Zuweisungen und Einschränkungen auf Objektebene sind maßgeblich / essenziell.**
- Bereits die Testmöglichkeit von Funktionsbausteinen ist heikel, denn hierüber können als ausgesprochen gefährlich eingestufte Bausteine aufgerufen werden. Beispielhaft sei der (allerdings nicht remotefähige [und ohne Berechtigungsprüfung laufende]) Bau-

stein DB_EXECUTE_SQL (Gruppe SDBI_DYNSQL) erwähnt, der ein parameterloses SQL-Statement ausführt (Aufruf von ‚C_DB_EXECUTE‘). Dieser Aufruf muss „ohne Anzeige eines Resultats“ nur mit Ergebnis in einer Tabelle ablaufen, also kein SELECT, sondern z. B. ein „DELETE FROM <Tabelle>“. **Hierüber ist ein Destabilisieren des Systems möglich.** Andere Bausteine manipulieren ebenfalls durch den Aufruf von ähnlichen System-Kernel-Calls, so dass solche Bausteine bzw. Bausteingruppen Restriktionen über das Berechtigungskonzept erfahren sollten.

- Sofern eine Tunnel-Funktion (Programm oder Baustein) Verwendung findet, sind auch die Zuweisung weiterer Rechte (mit Protokollierung) und die Ausführung nicht remotefähiger Codes (teilweise unprotokolliert) möglich, so dass Beschränkungen kaum gelten und Nutzungen nur unter erschwerten Bedingungen nachvollzogen werden können.

Die Frage steht, wenn bestimmte, nicht remotefähige Funktionsbausteine über diesen Weg auch externen Zugriffen ausgesetzt sein können, durchaus berechtigt im Raum: Kann bei einer solchen Berechtigungskonzeptaufweichung von hinreichend realisierter Basissicherheit gesprochen werden? Dass im Kundennamensraum „risikobehaftete“ Funktionen – auch mit Manipulation von SAP-Tabellen/SAP-Daten – etabliert werden, kann kaum verhindert werden (ein abgestimmtes Qualitätssicherungs- und Abnahmeverfahren ist obligatorisch); dass SAP-Objekte jedoch als Vorlage dienen können, weitgehend an bestehenden bzw. organisatorisch festgelegten Sicherheitsmechanismen vorbeilaufende Funktionalitäten her- und bereitzustellen, ist unbefriedigend.

Diese kurze Facettenbetrachtung belegt, dass RFC ein „neuralgischer Punkt“ bleibt und dass regelmäßig Funktionsanalysen, -einschränkungen in den Rollen und Risikobetrachtungen dringend geboten sind und erweitert werden müssen um unproblematisch zu implementierenden Tunnel-Code.

Literatur

Wiegenstein, Andreas: BlackHat Briefings 2011, Barcelona, virtualforge.com, https://media.blackhat.com/bh-eu-11/Andreas_Wiegenstein/BlackHat_EU_2011_Wiegenstein_The_ABAP_Underverse-WP.pdf.

Anhang

Funktionsbaustein	DB_EXECUTE_SQL
Funktionsgruppe	SDBI_DYNSQL
Beschreibung	Ausführen eines parameterlosen SQL Statements
Importparameter	STMT (Type C)–SQL Statement ohne Parameter STMT_LN (Type I)–Länge des Statements (optional)
Quellcode	FUNCTION DB_EXECUTE_SQL . *“—————————————————————— *“*“Lokale Schnittstelle: *“ IMPORTING *“ REFERENCE(STMT) TYPE C *“ VALUE(STMT_LN) TYPE I DEFAULT 0 *“ EXPORTING *“ REFERENCE(SQL_CODE) TYPE I *“ REFERENCE(SQL_MSG) TYPE C *“ REFERENCE(ROW_NUM) TYPE I

Quellcode

```
*"       EXCEPTIONS
*"            SQL_ERROR
*"            NOT_FOUND_OR_DUPLICATE
*"            INTERNAL_ERROR
*"_____________________________________________________
 DATA LN TYPE I.

IF STMT_LN <= 0.
    LN = STRLEN( STMT ).
ELSE.
    LN = STMT_LN.
ENDIF.
CALL ‚C_DB_EXECUTE‘
         ID ‚STATLEN‘ FIELD LN
         ID ‚STATTXT‘ FIELD STMT
         ID ‚SQLERR‘ FIELD SQL_CODE
         ID ‚ERRTXT‘ FIELD SQL_MSG
         ID ‚ROWNUM‘ FIELD ROW_NUM.
CASE SY-SUBRC.
    WHEN 0.
    WHEN 1.
      RAISE SQL_ERROR.
    WHEN 4.
      RAISE NOT_FOUND_OR_DUPLICATE.
    WHEN OTHERS.
       RAISE INTERNAL_ERROR.
     ENDCASE.
    ENDFUNCTION.
```

Dipl.-Betriebswirt Christoph Wildensee, CISM, CRISC, ist seit 1993 als IV-Revisor und zusätzlich seit 2008 als Datenschutzbeauftragter bei den Stadtwerken Hannover AG tätig.