

Interfacebau (fast) ohne Anwendungsanpassungen

Dr. Martin Rosenbauer, Stefan Jüngling
16. November 2009

Interfacebau ohne Anwendungsanpassungen

1

Merkwürdiger Titel? 2 Begriffe:

1) Interfacebau?

Immer mehr **Bedarf für Schnittstellen**

- Trend: Outsourcing grundlegender Konzern-Bereiche, IT-Bereiche: werden eigene GmbHs oder sogar verkauft
- Services externe Dienstleister halten Einzug
... Beispiel: Hewlett-Packert **MPS** mit eigenem Incident-Management-System

2) „ohne Anwendungsanpassungen“

Immer **mehr Fertiganwendungen**

- Upgrade-Fähigkeit

Vorstellung →...

MTI Management of Technology and Information Dr. Martin Rosenbauer
Workflow-Projekte - ARS-Programmierung

- Projektleitung für Remedy-Projekte
- Beratung bei Workflow-Konzeption
- Umsetzung von Workflows
- zeitlich befristete Verstärkung von Remedy-Teams
- Kompetenz in Remedy-Schnittstellen, ARSPerl, Oracle, SQL und UNIX, sowie Business Objects (BO) und HP Loadrunner für Performance-Analysen

N-Tuition
Business Solutions AG

↓
Management Consulting for IT

↓
BMC Remedy Lösungen

↓
Produkte

- Über 10 Jahre Erfahrung bei der Umsetzung von IT Service Management Prozessen in BMC Remedy
- 25 fest angestellte, BMC und ITIL geschulte Consultants mit umfangreichen Erfahrungen in nationalen als auch in internationalen Projekten.
- BMC Tech Alliance Partner



ARSXLink®
Remedy Integration Suite

<linke Folie>

Sie haben Glück, dass ich dieses Jahr nicht alleine vortrage...

Stefan Jüngling (n-Tuition)...

Nicht genannt ist hier

Boris Noack (E.ON IS GmbH)

Würde hier eigentlich als Dritter auf die Bühne gehören, konnte dieses Mal aber nicht vortragen....

Überblick → ...

Übersicht

- Teil I: Einführung in Remedy-Transaktionen und das Interface-Problem
- Teil II: Interfacebau ohne Anwendungsanpassungen
 - Individuelle Schnittstellen
 - „Single Message End Point“
 - ESB (Enterprise Service Bus) Produkte

Interfacebau ohne Anwendungsanpassungen

3

Nach so viel Einführung zunächst die Übersicht ☺

Zwei Teile: Teil I: kurze Einführung, notwendig für Teil 2

<Folie>

→Zunächst kurz: Welche Kommunikationsmöglichkeiten gibt es für den Remedy Server, Wie baut man Schnittstellen?

Überblick über **grundsätzliche** Möglichkeiten →...

Schnittstellen-Architekturen

- Trigger-Signal-Übermittlung („**wann** wird übertragen?“)
 - Asynchron: Zeitgesteuert / seltenes Polling
 - Synchron aus (Remedy-)Transaktion / von außen über API-Aufruf
 - Minütliches Polling
 - \$PROCESS\$-Aufrufe
(z.B. über die Remedy-Clients: runmacro oder DataImport)
 - Alerts (nur nach außen)
 - Plugins / Web Services (bidirektional)
- Daten-Kanal („**wie** wird übertragen“)
 - Datenbank-Anbindungen / DB-Links
 - runmacro / API-Programmanbindungen (C/Perl/Java/.Net API)
 - Plugin-Anbindungen (Filter-API, Vendor-Forms)
 - Web-Services - Anbindungen

Interfacebau ohne Anwendungsanpassungen

4

Hier: Nur serverseitige Schnittstellen: Es gibt auch clientseitige Anbindungen: ODBC / OLE / DDE / Data Visualization Field

<Folie>

Trennung von Datenkanals und Triggersignal möglich (z.B. von außen in DB-Tabelle per SQL, dann runmacro zum Triggern...)

„**runmacro**“ immer wieder unbekannt...

- einfach und mächtig und ohne API-Programmierung
- Zugriff von außen auf Remedy-Server
- Nachteil: Passwortverwaltung...

Ziel von Teil 1: Verständnis von Problem und Lösungsmöglichkeiten

WIE bauen wir eine Remedy-Schnittstelle ?

→ Theoretisch einfachste Lösung – „**einstufig**“ – hat Nachteile → ...

=====

... Sehr viel Auswahl

... gewünscht wird häufig eine „**synchrone**“ Datenübertragung.

→ Dabei technische Probleme mit Remedy,

→ in beide Richtungen !

Interface-Problem

Einstufige Verarbeitung



Nachteile

- Fremdsystem down, Netzproblem

→ Error: Incident nicht abspeicherbar

- Feldlänge überschritten, Pflichtfeld fehlt

→ Error: In Remedy wird nichts gespeichert

Interfacebau ohne Anwendungsanpassungen

5

Web-Service nur Beispiel! Könnte auch z.B. set-Fields mit \$RPROCESS\$ sein, oder Filter-API o.ä.

<Folie>

<click>

Problem:

Im Fehlerfall: Transaktionsabbruch ohne Datensatzspeicherung

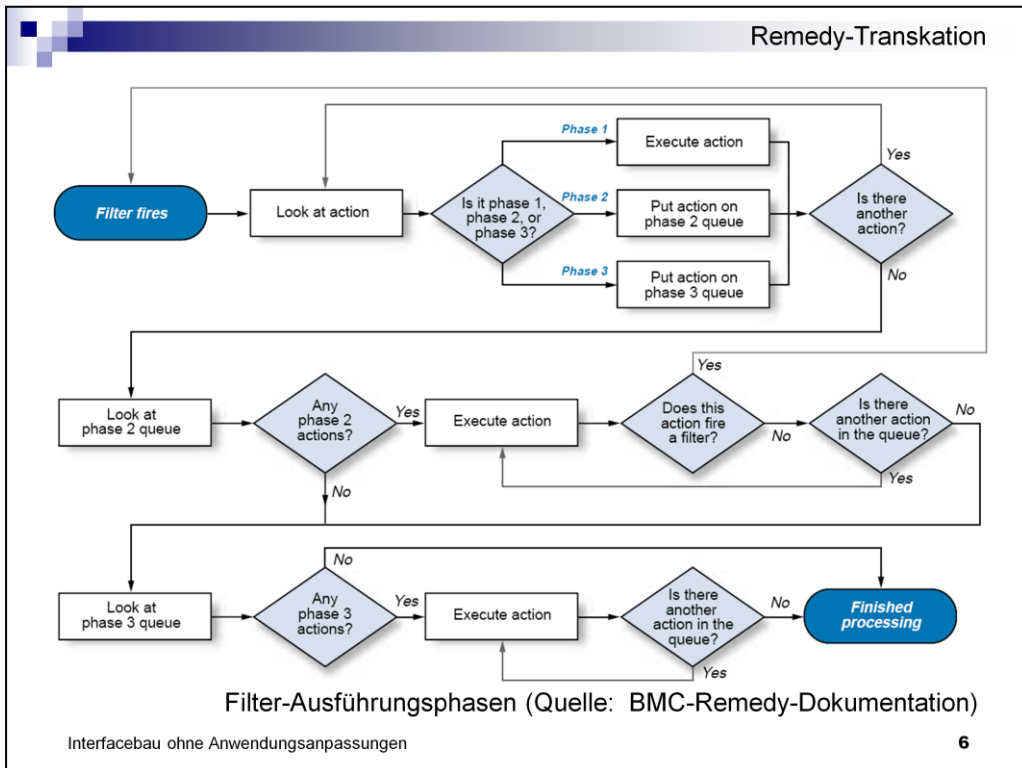
→ Die Schnittstelle legt das System lahm.

Umgekehrt:

<click>

In Remedy kann auf missglückte „Update-Versuche von außen“ nicht reagiert werden, da diese Versuche keine Spuren hinterlassen.

Warum ist das eigentlich so? Dokumentation von BMC-Remedy → ...



Sie sehen die Ausführungsphasen von Filter-Aktionen - viele interessante Details, wären ein eigener Vortrag...

Hier nur kurz: Sortierung nach Art der Aktionen (set-fields = Phase 1, Push-Fields/SQL = Phase 2, Notification/\$PROCESS\$ = Phase 3)

→ Einordnung in Queues

<Folie>

Warum erzähle ich das?

Strategie des Servers **unmkehrbare Aktionen möglichst spät!**

Wichtig: Bei Fehler bis Phase 2: **Alle Datenbankvorgänge werden vollständig rückgängig gemacht.**

Dann Error

Dann Rollback (Löcher aller neu angelegten verknüpften Datensätze)

Dann Kein Abspeichern unseres Incidents !

→ Remedy wurde ursprünglich als Service Desk Tool entwickelt, nicht für Schnittstellen !

Wie gehen wir damit um? Was soll unser System im Fehlerfall tun? → ...

=====

Parsen aller Filter, dabei:

Sortierung nach Art der Aktionen (set-fields = Phase 1, Push-Fields/SQL = Phase 2, Notification/\$PROCESS\$ = Phase 3)

→ Einordnung in Queues

→ Push-Fields haben Zielform, dort ebenfalls Filter-Parse-Vorgang, Sortierung, Queues → Phase-2-Queue davon wird zuerst ausgeführt (Z.B. das Anlegen eines Datensatzes),

dann Fortsetzung der Phase 2 des ursprünglichen Filters

`! zieht Aktionen vor nach Phase 1

Damit das geht (bei Push-Fields-Aktionen):

- „Merken“ der IDs frischer Datensätze (Create-Aktionen, löscherbar)
- Aufschieben von „Modify“-Aktionen nach Phase 2 (unumkehrbar)

Server Down-Zeit in Texas um Mitternacht: Bei uns kann um 7:00 Morgens niemand arbeiten

Ziele bei Verbindungsausfall

- Funktion der Anwendung
 - Abspeichern von Incidents IMMER möglich
- Transaktionssicherheit
 - Nachholen der Schnittstellenaktion, sobald Zielsystem wieder erreichbar

Interfacebau ohne Anwendungsanpassungen

7

<Folie>

... kleinere Probleme/Überschneidungen, wenn auf beiden Seiten gleichzeitig gearbeitet wird ...

Bsp:

Unser Ticket hat sich erledigt, wir schließen es (Schnittstelle ist down).

Ticket ist geschlossen, 2 Stunden später treffen Lösungsmeldungen vom Partner ein (Schnittstelle geht wieder).

Was machen wir mit der nachträglichen Lösung?????

... Diese 2 Ziele – wie erreichbar mit einem Remedy Server, der bei Fehlern alles zurückrollt?

→ wie lösen wir das Schnittstellenproblem? Übersicht → ...

Lösungsmöglichkeiten

- Einstufige Verarbeitung
 - Error Handler
- Zweistufige Verarbeitung, zweite Stufe dann ...
 - zeitgesteuert, z.B. über Eskalationen
 - über \$PROCESS\$ - Aufrufe von API-Programmen
 - über Filter-API (z.B. REMEDY.ARF.CAI)

Interfacebau ohne Anwendungsanpassungen

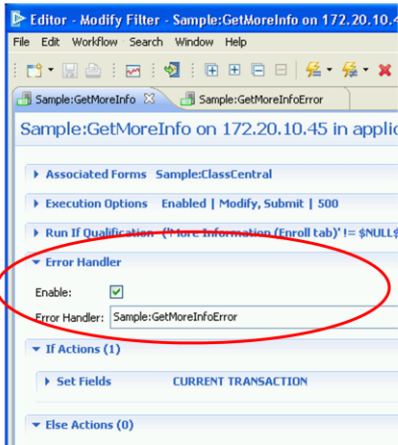
8

<Folie>

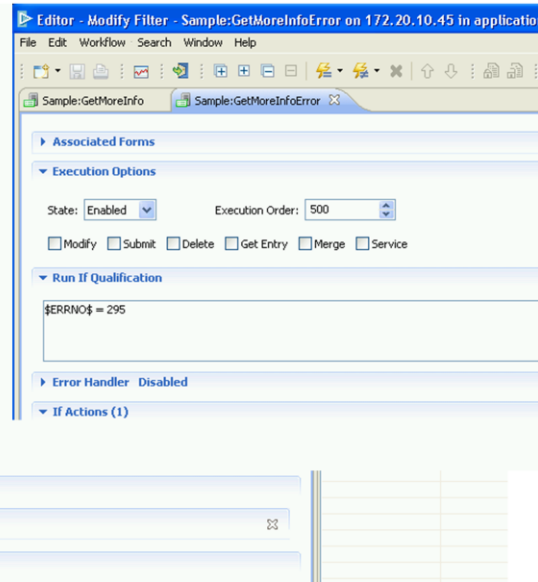
... zunächst der **Error-Handler** → ...

Error Handler

Seit Version 7.1



Interfacebau ohne Anwendungsanpassungen



9

Ziel: Funktion der Anwendung auch im Fehlerfall.

→ Abfangen von Fehlern möglich, **kein Rollback!**

Beispiel → ...

=====

Kurzeinführung:

Eine Filteraktion in einem Filter mit zugeordnetem Error-Handler-Filter wird
 ... normal ausgeführt, wenn kein Fehler auftritt
 ... durch Aktionen im Error-Handler-Filter „ersetzt“, wenn ein Fehler auftritt.
 Die Aktion mit dem Fehler wird dann nicht ausgeführt.

→ Incident ist abspeicherbar, die Schnittstellenaktion muss dann jedoch später nachgeholt werden.

Einstufige Verarbeitung Generell: → Problem mit Transaktions-Sicherheit;

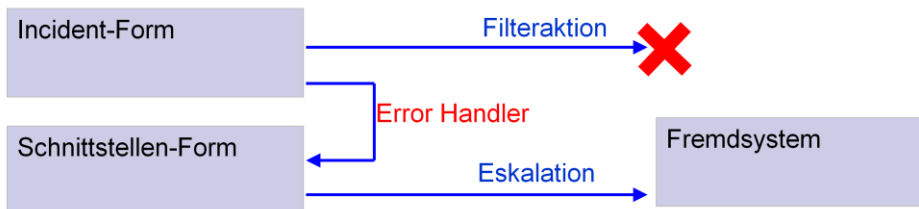
Mit Error-Handler aber lösbar, z.B. wie folgt:

Error Handler

Normalfall



Fehlerfall



Interfacebau ohne Anwendungsanpassungen

10

... Im Fehlerfall: Zweistufige Verarbeitung

... geht also, aber erst seit AR Server Version 7.1.

Gegenrichtung ist in ähnlicher Weise machbar:
Schreiben von außen direkt auf Incident-Form.

Bei Fehler: Error-Handler:

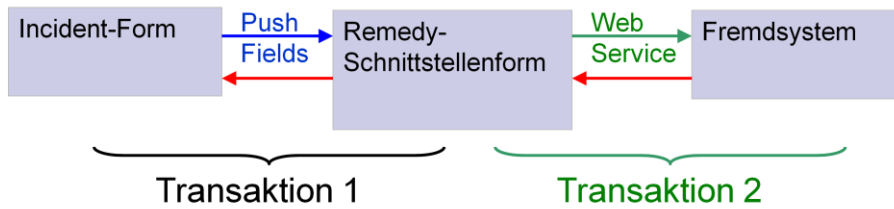
Ersatzweises Schreiben auf eine Schnittstellenform und z.B. Auslösen einer Benachrichtigung...

Alternative:

Klassische Architektur

... geht **immer** über eine Schnittstellenform...

Zweistufige Verarbeitung: Entkopplung



Startmöglichkeiten für Transaktion 2:

- Termingebunden über
 - Eskalation
 - Zeitsteuerung von außen (cron, Windows-Scheduler etc.)
- „Sofort-Start“ über
 - Häufiges Polling per Eskalation (spezielle Queues seit 7.1)
 - \$PROCESS\$-Aufruf z.B. von „runmacro“

Interfacebau ohne Anwendungsanpassungen

11

Klassische Lösung

Ziel

- Erfolgreiche Erste Stufe, d.h. Schreiben von Datensatz **auch im Fehlerfall der 2. Stufe**
- Sofortiges Schreiben (ohne Eskalation/ohne Scheduling etc.)

Start der Stufe-2-Aktion **NICHT** „On Submit/Modify“ ...

<click>

Sondern:

\$PROCESS\$-Aktion (ohne Return-Parameter, „fire and forget“!),

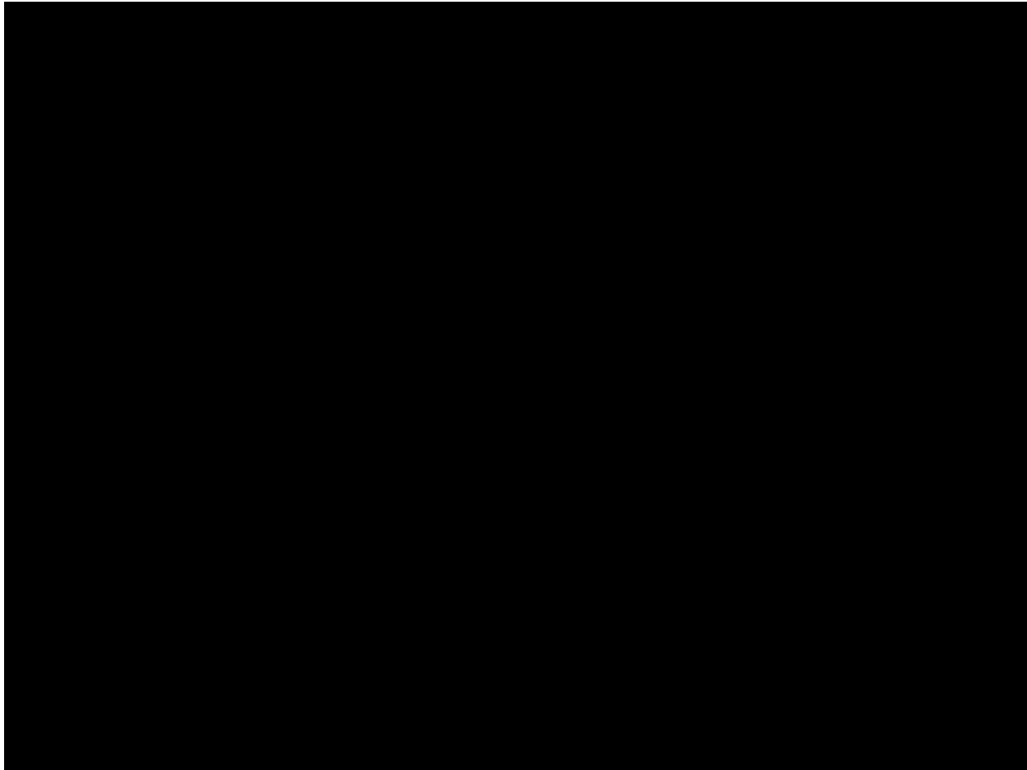
NICHT: \$RPROCESS\$ innerhalb von set fields-Aktion

\$PROCESS\$-Aktion wird „On Submit“ gestartet, also innerhalb der Transaktion.

Meldet sich „von außen“ als „2. Benutzer“ an und führt die Aktion zu Ende

<click>

In der Gegenrichtung gilt dasselbe.



Problem gelöst ? → „eigentlich ja“ (schon mehrere Projekte so verwirklicht, siehe auch mein Vortrag von 2007).

Ist das die beste Architektur? ... auch heute noch?

- Langsamer Prozess-Aufruf („fork“ etc.)
- Hinterlegen von Login-Daten/Kennwörtern im Aufruf
- Heute geht es auch besser!

Vorbild Remedy-ITSM-Suite: Interne Interfaces

- Ebenfalls enkoppelt:
- Beispiel Requester Konsole: Request wird erstellt, auch wenn Incident scheitert.
- Incident wird dennoch scheinbar synchron erstellt – aber wie?

→ Nähere Untersuchung – Wie macht Remedy das?

- Eskalation? → zu langsam, nur 1x/Minute
- \$PROCESS\$-Aufruf? Externes Programm? runmacro? → finden wir nicht !
- Antworten im Serverlog:

Finden dort: Aktionen/Einträge von Benutzer „Remedy Application Service“ –
Wer ist das ?

- Am Ende finden wir dann nach langer Suche einen filter-API-Aufruf
→ Ergänzung der Liste →...

Entkopplung zweier Transaktionen



Startmöglichkeiten für Transaktion 2:

- „Verzögert“ über
 - Eskalation
 - Zeitsteuerung von außen (cron, Windows-Scheduler etc.)
- „Sofort-Start“ über
 - Häufiges Polling per Eskalation (spezielle Queue seit 7.1)
 - \$PROCESS\$-Aufruf z.B. von „runmacro“
 - **Remedy Filter-API**
 - Alert-Mechanismus (Remedy Orchestrator)

Interfacebau ohne Anwendungsanpassungen

13

Liste unten ergänzt !

Details → ...

Filter-API

- REMEDY.ARF.CAI
 - Voraussetzungen
 - ITSM-Suite mit
 - Form CAI:EventParam
 - Form CAI:Event
 - caieventcmd.dll bzw. libcaieventcmd.so

Interfacebau ohne Anwendungsanpassungen

Set Fields

Data Source: FILTER API ▼

Server Name: 172.20.10.45 ▼

Plugin Name: REMEDY.ARF.CAI ...

Position	Value	Type
1	\$EventGUID\$	Field
2	\$SourceKeyword\$	Field
3	\$TargetServer\$	Field
4	\$EventEntryID\$	Field
5	\$AOIGUID\$	Field
6	\$Event\$	Field
7	\$EventType\$	Field
8	\$AppInterfaceForm\$	Field
9	\$ExternalPassword\$	Field
10	\$ExecutionType\$	Field
11	\$AccessMode\$	Field

Field	Value
ReturnMessageCode	\$3\$
ReturnCode	\$1\$
ReturnMessage	\$2\$

ITSM-Suite: Filter auf CAI:Events mit Namen CAI....
 ruft auf: **Filter-API-Plugin** „REMEDY.ARF.CAI“

Übergibt verschiedene Parameter, darunter

- GUID
- Name der Ziel-Form für die Daten

Wie werden Parameter übergeben? → ...

=====

Remedy-Dokumentation zu diesem Filter-Plugin sehr dürftig eigentlich ist es „undokumentiert“...

Zweck des Plugins: Daten aus einer Form lesen und Eintrag in beliebiger anderer Form mit diesen Daten erstellen.

Es gibt 11 Übergabeparameter und 3 Return-Parameter

Die meisten Übergabeparameter brauchen nur Konstanten!

Wichtig sind:

<click>

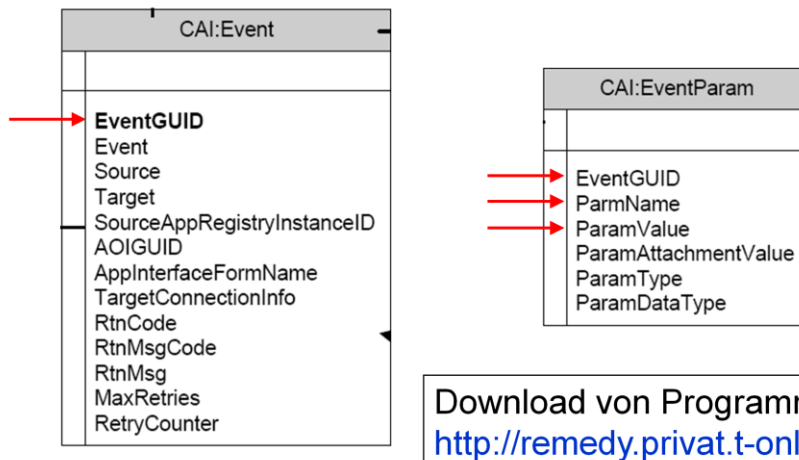
Die mit den roten Pfeilen markierten Parameter

- Zielform (AppInterfaceForm)
- Schlüssel für die Parameter-Datensätze (GUID)
- ID des Quell- Datensatzes (EntryID)

Prinzipiell auch möglich: Über Web-Services Daten in Forms auf anderen Servern übergeben (hier nicht Thema).

Wie werden Parameter übergeben? → ...

Datenstruktur für Filter-Plugin REMEDY.ARF.CAI



Download von Programmbeispiel:
<http://remedy.privat.t-online.de>

Interfacebau ohne Anwendungsanpassungen

15

Übergabe-Daten werden *vorher* als „Parameter“ in die Form CAI:EventParam geschrieben.

... Genügt in 7.0, ...wenn man alle Parameter selbst wieder löscht.

Die gesamte Datenstruktur jedoch

<click>

... setzt noch einen Eintrag in der Form CAI:Event voraus.

- Fehlt der Eintrag in Server-Version 7.5, dann wird 30 Mal im Abstand von 10 Sekunden versucht, diesen Eintrag zu finden um ihn dann zu löschen !!!
- Filter-Plugin möchte wieder aufräumen!

Weitere Details gehen hier zu weit ...

<click>

Download

•**Beispielprogramm aus 1 Quell- und 1 Zielform**

•**Leicht zu verstehen (wenige Filter)**

•**auf jedem Remedy-Server mit ITSM-Suite installierbar**

•**getestet mit Oracle-Datenbank mit Remedy 7.0 und 7.5**

•**Einfach und komplett wieder deinstallierbar.**

Zuletzt noch ein kurzes Schema zur Funktion → ...

=====

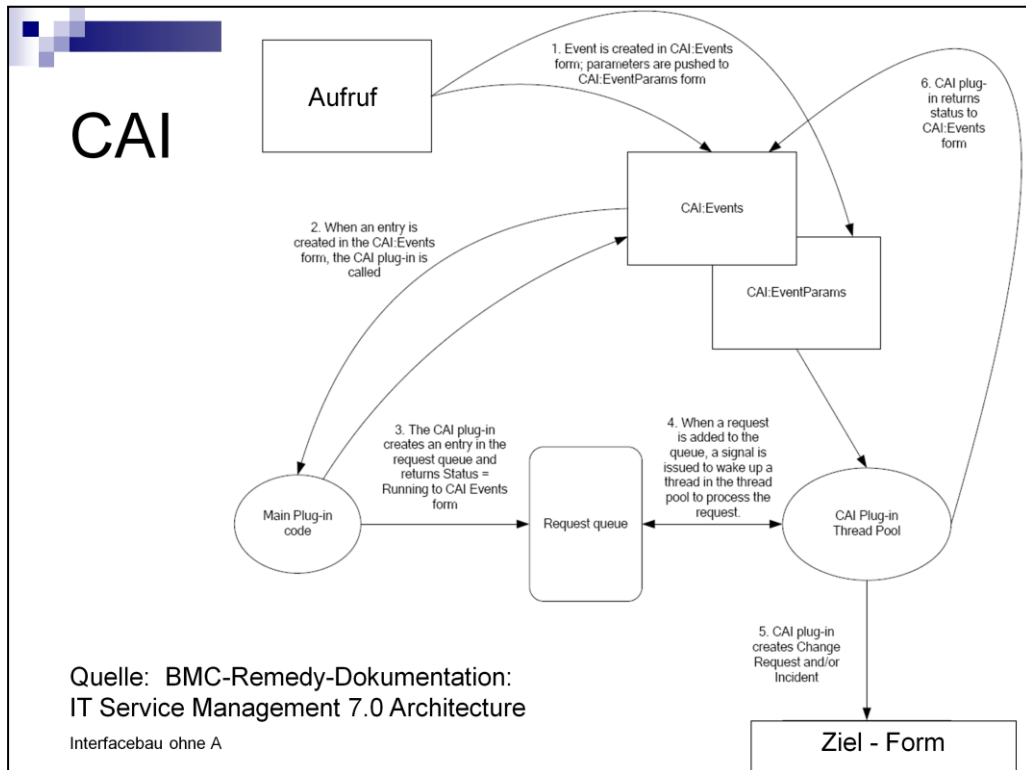
In Remedy-Version 7.0:

CAI:Event-Eintrag komplett verzichtbar

In Version 7.5 notwendig und auch in 7.0 besser:

1 Pseudo-Eintrag in CAI:Events erzeugen, ohne dass Filter laufen, also z.B. mit SQL-Mitteln

... damit danach vom Plugin alle Einträge automatisch gelöscht werden !!!! ☺



<Folie>

Das war die Methode „klein aber fein“ die „**kleine Lösung**“, und das Ende von Teil 1

Jetzt folgt → ...

Teil II: Interfacebau (fast) ohne Anwendungsanpassung

■ Übersicht

- Schattenform-Technik
- Remedy „Command Automation Interface“ (CAI)
- Single Message Endpoint: ARSXLink
- Externe Schnittstellensoftware
 - ARSXLink + Magic iBolt
 - ARS Power Connect
 - Remedy BMC Atrium Orchestrator

Interfacebau ohne Anwendungsanpassungen

17

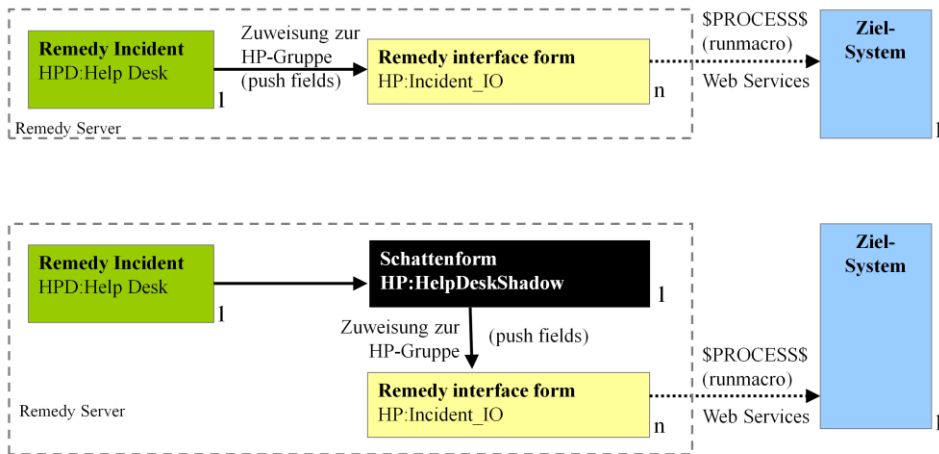
-Schattenform-Technik: Komplette Eigenprogrammierung ohne Customizing **innerhalb** der ITSM-Suite

-CAI auch manchmal „**Common Application** Interface“, arbeitet nur innerhalb Remedy

Start mit Beispiel zur Schattenform-Technik:

→ ...

Schattenform-Technik



Keine Schnittstellfelder und keine Schnittstellenlogik in Incident-Form

Interfacebau ohne Anwendungsanpassungen

18

Zweistufige Technik – kennen wir schon.

Probleme:

- Konfigurations-Form mit Konfiguration der HP-Gruppe → HP-Gruppe wird gebraucht für Filter in Incident-Form
- Schnittstellenlogik braucht Felder im Incident (z.B. Nummer des letzten übertragenen Worklog-Eintrags) → Felder auf Incident-Form werden benötigt (sowohl persistente als auch temporäre) → Beides wäre „Customizing“.

→ Wo packen wir diese Felder hin, wenn nicht auf die Incident-Form?

<click>

... auf die Schattenform (schwarz). Dort Entscheidung, ob Aktion an Interface-Form geht oder nicht.

Wichtig dabei:

<click>

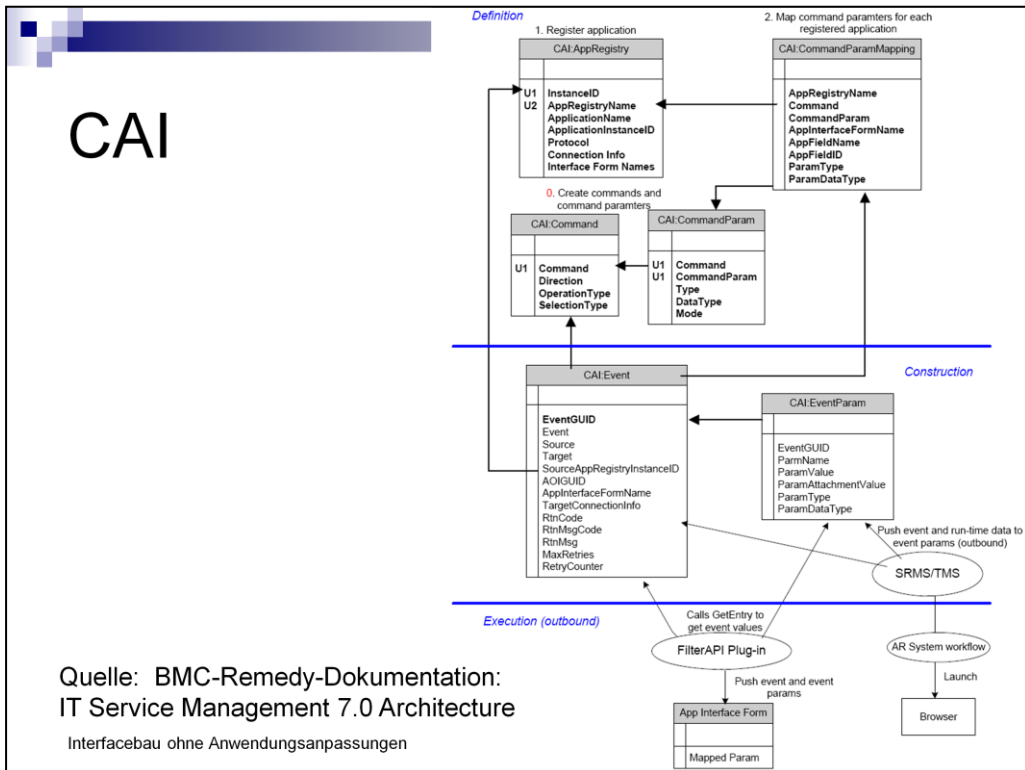
Keine Schnittstellenlogik auf Incident-Form – **kein Customizing**

Anmerkung:

- Nur 1 Filter auf Remedy Incident-Form für Trigger-Signal, mit abweichender Benennung (Name beginnt z.B. mit HP:).

Bisher alles „zu Fuß“ progammiert. Funktioniert hervorragend.

... Geht es auch mit weniger oder ganz ohne Programmierung? Evtl. mit dem Remedy CAI? → ...



Command Automation Interface / Common Application Interface

→ Für Schnittstellen (nur) **zwischen Remedy-Applikationen**

<Folie> ... starten mit CAI:Events in der Mitte

Wo Definition: Übertragung von welchem Original-Feld aus welcher Original-Form → welches neue Feld in welcher neuen Form?

... CAI:CommandParamMapping → ...

=====

Konfigurationsarbeit:

Definition

- Applikation
- Kommando („Schnittstellename“)
- Kommando-Parameter
- Parameter-Mapping

Schnittstellenaktion „Construction“

- Parameter hinterlegen
- Event auslösen (durch Abspeichern)

Ausführung

-Parameter werden in die Ziel-Form übertragen

Command Automation Interface

App Registry Name	Command	Name	App Interface Form	Field ID	Field Name
BMC Remedy Incident Management	SRM_OUT_PUSH_SR_INFO	ADI GUID	HPD:IncidentInterface	301572000	SRMSADIGuid
BMC Remedy Incident Management	SRM_OUT_PUSH_SR_INFO	SRMS_APP_REGISTRY_INSTANCE_ID	HPD:IncidentInterface	301571900	SRMS Registry Instance ID
BMC Remedy Incident Management	SRM_OUT_PUSH_SR_INFO	APP_REQUEST_GUID	HPD:IncidentInterface	179	Instancelid
BMC Remedy Incident Management	SRM_OUT_PUSH_SR_INFO	SRID_LABEL	HPD:IncidentInterface	301572100	SRID

Für jede Schnittstelle:

Anpassungen im SRMS notwendig:

- Filter Guide („Plugin“)
- 1 Aufruf-Filter für den Guide
- 1 Filter / Parameter
- weitere Filter...

→ Andere Lösung gesucht !

Filter Guides - (arsdev)

Filtering Options

Name	Primary Form	Ac
SRM:AOI:CreateEvent_PushSRInfo_CreateParams_Loop	SRM:AppInstanceBridge	No
SRM:AOI:CreateEvent_UpdateAppRequestWorklog_CreateParams_Loop	SRM:AppInstanceBridge	No
SRM:AOI:GetEvent_AppRequestInfo_GetParams_Loop	SRM:AppInstanceBridge	No

Object Count: 336

SRM:AOI:CreateEvent_PushSRInfo_CreateParams_Loop on arsdev

Associated Forms: SRM:AppInstanceBridge

Display: "This loops through the parameter definit..."

Filters and Labels

- SRM:AOI:CreateEvent_GenericParam_AppRequestGuid'
- SRM:AOI:CreateEvent_GenericParam_AOIGuid'
- SRM:AOI:CreateEvent_GenericParam_SRIDLabel'
- SRM:AOI:CreateEvent_GenericParam_SRMSAppRegistryInstanceID'

Interfacebau ohne Anwendungsanpassungen

Konfigurator-Herz schlägt höher!!

Angabe von Zielform und Zielfeld für jeden Parameter möglich

Luxuriöser Mechanismus auf Form SRM:AppInstanceBridge

Frage: Wo ist Definition des Ausgangs-Felds (Source Felds) zu jedem Parameter ???

<click>

<click>

<Rest der Folie>

Evtl. ist ganz anderes Konstrukt, eine andere Alternative nötig...

Dazu werden wird Herr Jüngling ein wenig weiter ausholen und uns überlegen, was man tun kann, wenn man nicht nur eine, sondern gleich eine große Zahl an Schnittstellen an unsere ITSM-Suite „anschließen“ möchte und dabei der Überblick nicht verloren gehen soll...

→ ...

=====

In Form CAI:CommandParamMapping:

Felder SRMSField, SRMSFieldID

(Fields nicht in View) – nicht verwendet ;-(

Stattdessen **<click>**

... Filter-Guide

<Click>

... geht mit CAI leider nicht ohne „Anwendungsanpassung“ im SRMS !!!

<Click>

→ ...

EIN AUTONOMES SYSTEM



Interfacebau ohne Anwendungsanpassungen

21

-Vorstellung Burg

- In sich geschlossenes, funktionierendes, ökonomisches System
- Verschiedene Einheiten, Bäcker (Backhaus), Wirt (Taverne), Schmied (Schmiede), König (Administration)
- Einheiten sind untereinander verbunden und arbeiten miteinander (Straßen)

-Analogie zu ARS herstellen (ARS Struktur einblenden)

-ARS

- In sich geschlossenes, funktionierendes IT-System
- Verschiedene Einheiten (Module), Incident... Core Komponenten (Dictionary)
- Module sind untereinander verbunden und arbeiten miteinander (Workflow)

INTEGRATION IN DIE UMWELT



Interfacebau ohne Anwendungsanpassungen

22

-Verbindung zur Außenwelt

-Burg ist nicht alleine in der Welt, der Bedarf oder sogar die Notwendigkeit entsteht, sich mit seiner Umwelt auszutauschen (Stichwort Handel)

-Rote Markierungen einblenden

-Wirt benötigt Genussmittel, Schmiede möchte Waren verkaufen und Rohstoffe bekommen, Bäcker benötigt Rohstoffe

-Pfeile einblenden

-Irgendwann gibt es so viele Seiteneingänge, dass der Selbstschutz der Burg nur noch mit erheblichen Aufwand, wenn überhaupt, gewahrt werden kann

INTEGRATION IN DIE UNTERNEHMENSWELT



Interfacebau ohne Anwendungsanpassungen

23

- Verbindung zu weiteren Applikationen, Integration in die Unternehmenswelt
 - System-Management Tools sollten Incidents erstellen können (Automatisierung des Prozesses)
 - Schnittstelle Incident einblenden*
 - Purchasing Applikation benötigt Budgetpläne und aktualisiert Asset Register
 - Schnittstelle Purchasing einblenden*
 - Change sendet Daten an einen Reportingserver
 - Schnittstelle Problem einblenden*

KONTROLLIERTE INTEGRATION IN DIE UMWELT



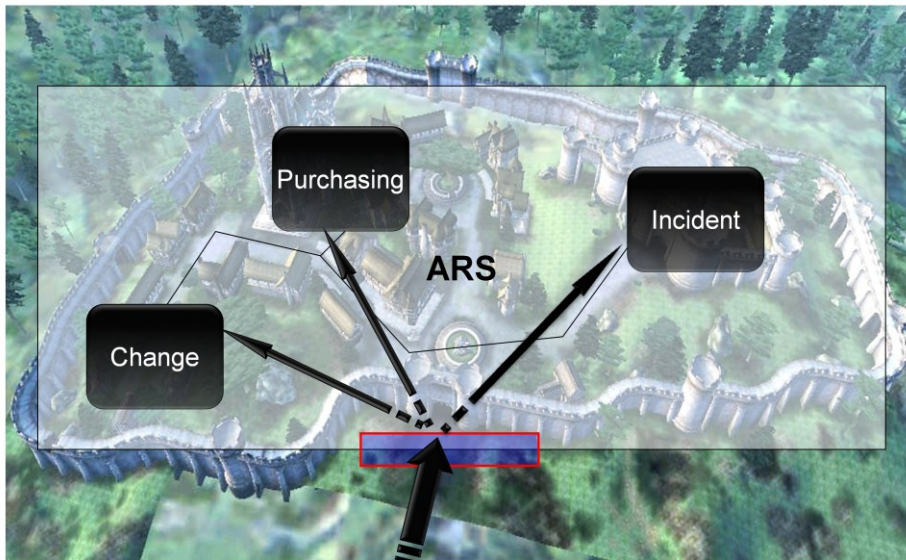
Interfacebau ohne Anwendungsanpassungen

24

-Burgtor

- Einblenden der Tormarkierung
- Einziges, überwachter Eingang
- Einblenden der Institutionen
- Der eingehende Warenverkehr wird überwacht
- Einblenden der Pfeile
- Die Administration weiß wer, wann, wohin will

KONTROLLIERTE INTEGRATION IN DIE UNTERNEHMENSWELT



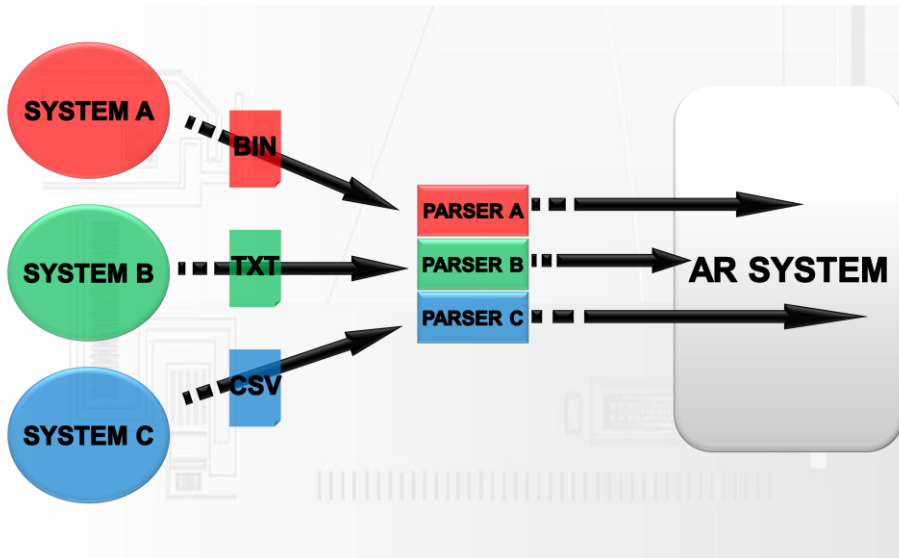
Interfacebau ohne Anwendungsanpassungen

25

-Remedy

- Einblenden der Schnittstelle
- Ein einziges, überwachtes und protokolliertes Interface
- Einblenden der Pfeile
- Erweiterung durch Konfiguration der Schnittstelle in einer Remedy Applikation
- Enabler hierfür, XML und ein Leistungsfähiger XML parser

REALISIERUNG OHNE XML



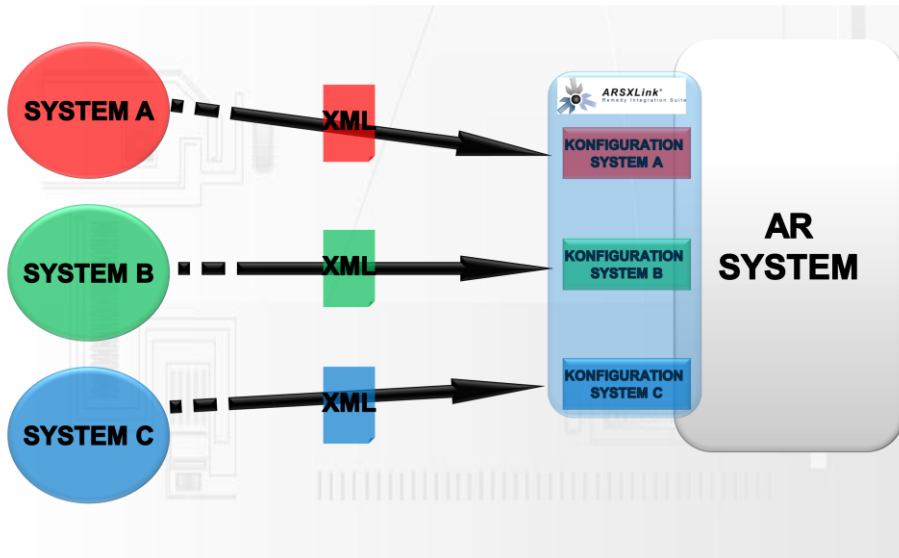
Interfacebau ohne Anwendungsanpassungen

26

-Realisierung SME ohne bzw. vor XML

- Einblenden der Datei A, B, C Erklärung des Parsers (kurze Anekdote HL7-Interface)
- Vorgelagerter Parser
- Angepasst für jede neue Schnittstelle bzw. für jeden Dateityp (Programmieraufwand) und jede Änderung

REALISIERUNG MIT XML



Interfacebau ohne Anwendungsanpassungen

27

-Realisierung mit XML

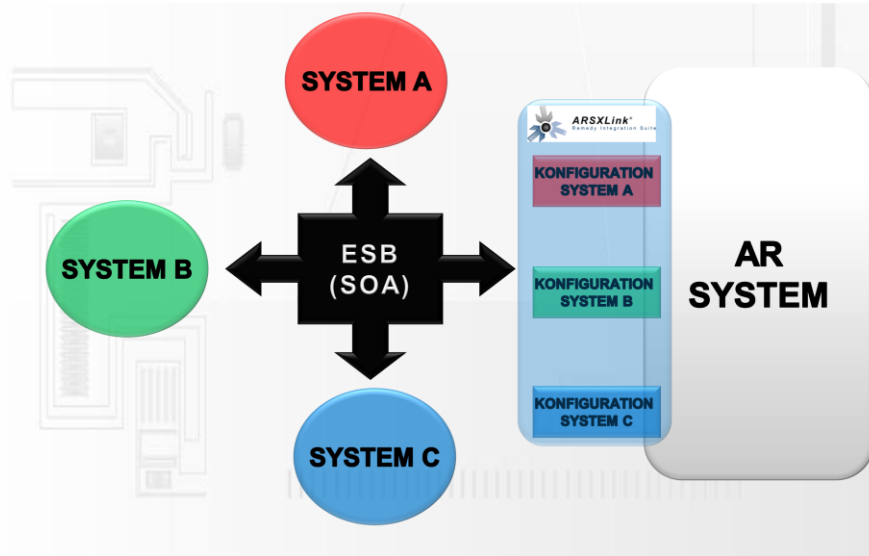
-Einblenden ARSXMLink

-Parser ist in der Lage, beliebig strukturierte XML Dateien zu verarbeiten

-Einblenden der Dateien A, B, C

-Die Daten werden einfach per ARS Applikation zu bestimmten Formularen gemappt (kein Programmieraufwand)

REALISIERUNG MIT XML UND ESB



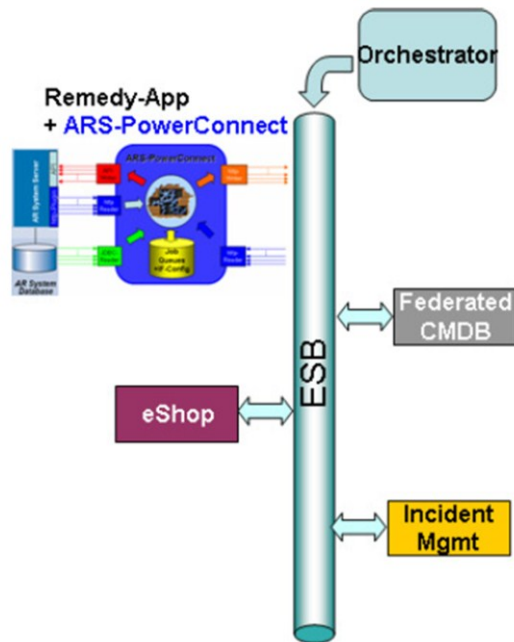
Interfacebau ohne Anwendungsanpassungen

28

-Realisierung mit XML und ESB

- Mit einer Single-Message-Endpoint Architektur ist eine Anbindung an einen Service-Bus problemlos möglich
- Beispiele ESB: kommerzielle Produkte wie :**iBolt**, WebSpere, Tibco, WebMethods, Biztalk, oder Open Source: Open ESB, Jboss ESB, Apache Synapse

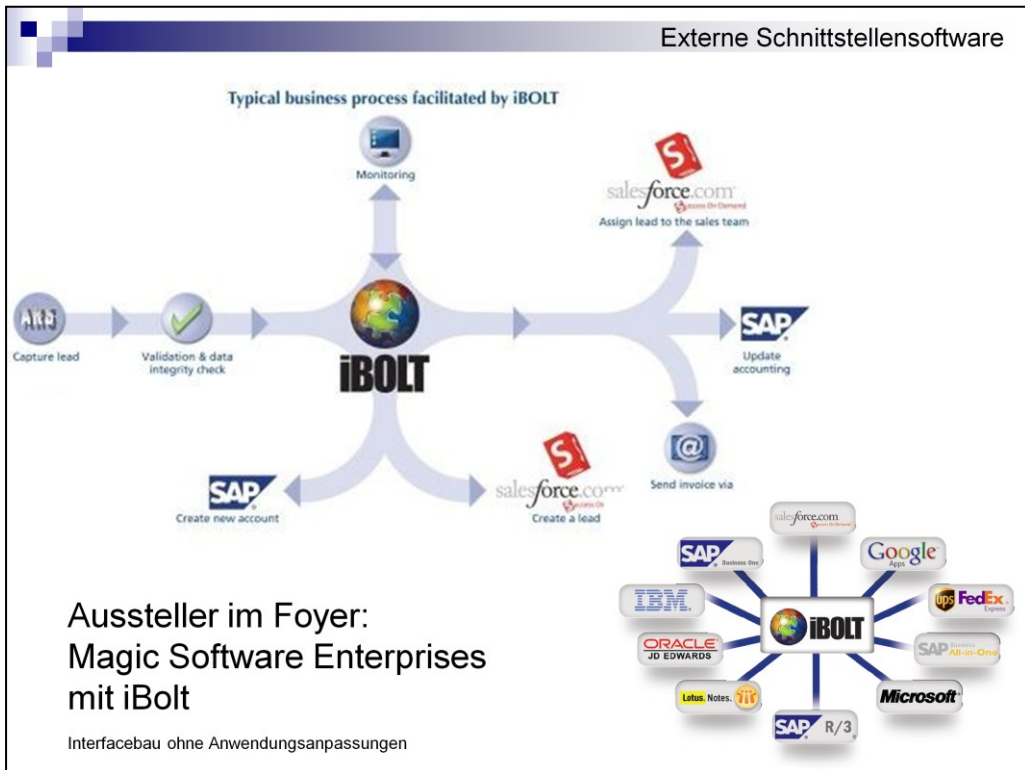
Anderer ESB Vortrag
(16:00 Uhr Track 3):



ARS Power Connect
Roland Rossmüller
(Infosys GmbH)

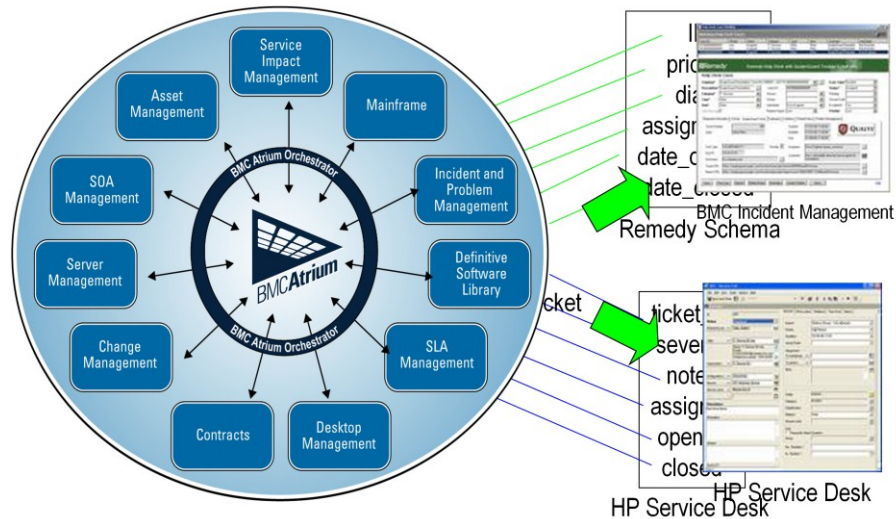
Interfacebau ohne Anwendungsanpassungen

29



... 1 Schnittstelle zum ARS
Dort mit ARSXLINK

BMC Atrium Orchestrator – morgen 10:30 Uhr



Interfacebau ohne Anwendungsanpassungen

31

Ebenfalls als Vortrag auf dieser RUG: Morgen früh: Dr. O. Franke (BMC)

Wichtig: Data-Mapping hier komplett konfigurierbar (OHNE Remedy-Anwendungsanpassung, d.h. ohne Programmierung!).

Zusammenfassung

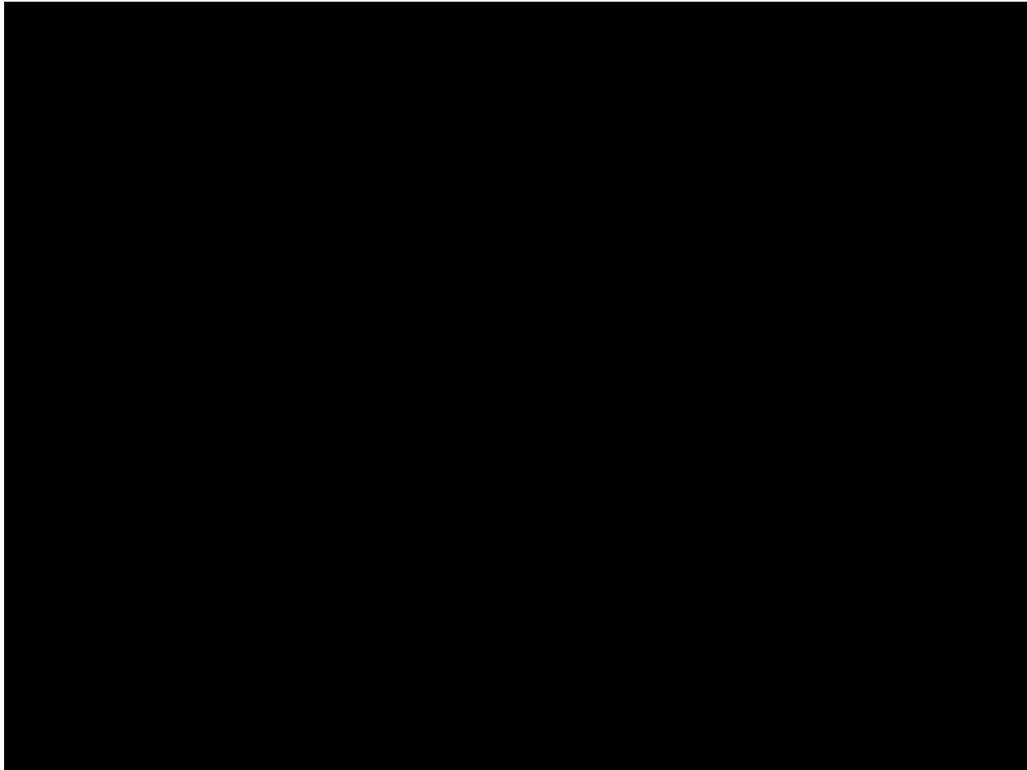
- Remedy Schnittstellen-“Problem“ – Lösung
 - Zweistufige Transaktions-Verarbeitung, 2. Stufe über
 - Eskalationen (asynchron) ...**langsam**
 - \$PROCESS\$ - Aufrufe mit runmacro (ohne ITSM-Suite) ...**o.k.**
 - Filter-API REMEDY.ARF.CAI („kleine Lösung“, mit ITSM-Suite) ...**elegant**
- Interfacebau ohne Anpassungen
 - Schattenform-Technik ...**geht immer, einfach**
 - CAI (Command Automation Interface) ...**SRMS-Anpassung notwendig**
 - Single Message End-Point-Architekturen (z.B. ARSXlink) ...**elegant**
 - Externe Software mit ESB-Architekturen (z.B. iBolt, ARSPowerConnect, Orchestrator) ...**mächtig, gut, z.T. teuer**

FRAGEN ?



Vielen Dank !

Dr. Martin Rosenbauer, Stefan Jüngling
16. November 2009



Problem gelöst ? → „eigentlich ja“ (schon mehrere Projekte so verwirklicht, siehe auch mein Vortrag von 2007).

Ist das die Beste Architektur?

- Langsamer Prozess-Aufruf („fork“ etc.)
- Hinterlegen von Login-Daten/Kennwörtern im Aufruf
- Heute geht es auch besser!

Vorbild Remedy-ITSM-Suite: Interne Interfaces

- Ebenfalls enkoppelt:
- Beispiel Requester Konsole: Request wird erstellt, auch wenn Incident scheitert.
- Incident wird dennoch scheinbar synchron erstellt – aber wie?

→ Nähere Untersuchung – Wie macht Remedy das?

- Eskalation? → zu langsam, nur 1x/Minute
- \$PROCESS\$-Aufruf? Externes Programm? runmacro? → finden wir nicht !
- Antworten im Serverlog:

Finden dort: Aktionen/Einträge von Benutzer „Remedy Application Service“ –
Wer ist das ?

- Am Ende finden wir dann nach langer Suche einen filter-API-Aufruf
→ Ergänzung der Liste →...