

ABAP in der Cloud

— Erfahrungsbericht Custom Code Migration
der Lösung „EDI exakt“ nach SAP BTP, ABAP Environment

Matthias Jäger | 07. Dezember 2022 | ABAPConf 2022

matthias.jaeger@erp-developer.com | Freelance SAP ABAP Development Expert & Architect

Agenda

**1.
Ausgangs-
situation**

**2.
Lösungs-
strategie**

**3.
Design-
Taktiken**

**4.
Geschäfts-
logik
verschalen**

**5.
UI & Multi-
tenancy**

**6.
Fazit**

1. Ausgangssituation

..., Projektziele und Herausforderungen

Ausgangssituation

- > „EDI exakt“ - ABAP-basierte EDI-Messaging Software der All for One Group
- > Ca. 180 produktive onPrem-Installationen
- > Lauffähig ab 7.02 Basis-Release, kompatibel mit SAP ERP und SAP S/4 HANA
- > Benötigt lediglich „nackten“ NetWeaver Application Server ABAP



Nachricht: 4849

Notiz speichern | Abbrechen | Kopieren | Fortsetzen | Neu starten | Log anzeigen | Abschließen

Kopfdaten Erweiterte Daten Eigenschaften

Status: Warte auf IDoc-Verarbeitung

Sender / Werk: IRIS 60
Sender fix:

Sender Nachrichtenformat: XML RC_INVOIC 01
Sender Nachrichtenformat fix:

Acknowledgement Nachrichtenformat:

Queue-ID:
PNB: 60

Nachrichtennotiz:

Schritte Referenzen

	Status	Status-ID	Text	Prozessketten-ID	Position	Buchungsdatum	Buchungszeit	Nachricht anzeigen
046	grün	Nachricht als manuelle Kopie angelegt		000000	17.11.2022	15:40:40	60	
300	grün	PNB gefunden	SYS_GLOBAL	000030	17.11.2022	15:40:40		
400	grün	PNB-Prozesskette angestossen	SYS_GLOBAL	000040	17.11.2022	15:40:41		
600	grün	Mapping durchgeführt	EDI_IN_IRIS_XML_PDF	000020	17.11.2022	15:40:41	60	
650	grün	Referenzen aus Nachricht ermittelt	EDI_IN_IRIS_XML_PDF	000030	17.11.2022	15:40:41		
800	grün	Nachricht an Adapter übergeben	EDI_IN_IRIS_XML_PDF	000040	17.11.2022	15:41:45	60	
802	orange	Warte auf IDoc-Verarbeitung	EDI_IN_IRIS_XML_PDF	000040	17.11.2022	15:41:45		

Ausgangssituation — Projektziele

> Wartbarkeit / Änderbarkeit

- > Code-Redundanzen oder semantisch gleiches Coding auf ein Minimum reduzieren

> Auslieferbarkeit / Portabilität

- > Regelmäßige Belieferung der onPrem-Installationen mit Updates während der Custom Code Migration
- > In verschiedenen Umgebungen lauffähiges Produkt mit leicht angepasstem Funktionsumfang zur Verfügung stellen

> SaaS-fähige Lösung, Multitenancy-Fähigkeit von Steampunk nutzen

> Modernes SAP Fiori UI

Ausgangssituation — Code-Basis

>ca. 3000 Entwicklungsobjekte

Bereits vorhanden:

- >Saubere Trennung zwischen Fachlichkeit und Technik, bspw. durch geeignete Paketierung
- >API-Layer für verschiedene Fachkomponenten (=Geschäftslogik)
- >Umfangreiche automatisierte Test-Suite

Technische Rahmenbedingungen

- >Cloud bereinigtes ABAP
- >Nicht alle Objekttypen unterstützt
- >SAP Public APIs - Nur freigegebene Entwicklungsobjekte verwendbar
- >Neue Anwendungsinfrastruktur, bspw. IAM, Anwendungsjobs, ...

Entwicklungsumgebung

- >Backend-Entwicklung ausschließlich mit ABAP Development Tools
- >Keine explizite SAP Gateway Programmierung möglich bzw. nötig

2. Lösungsstrategie

... zur Custom Code Migration

Generelle Lösungsstrategie

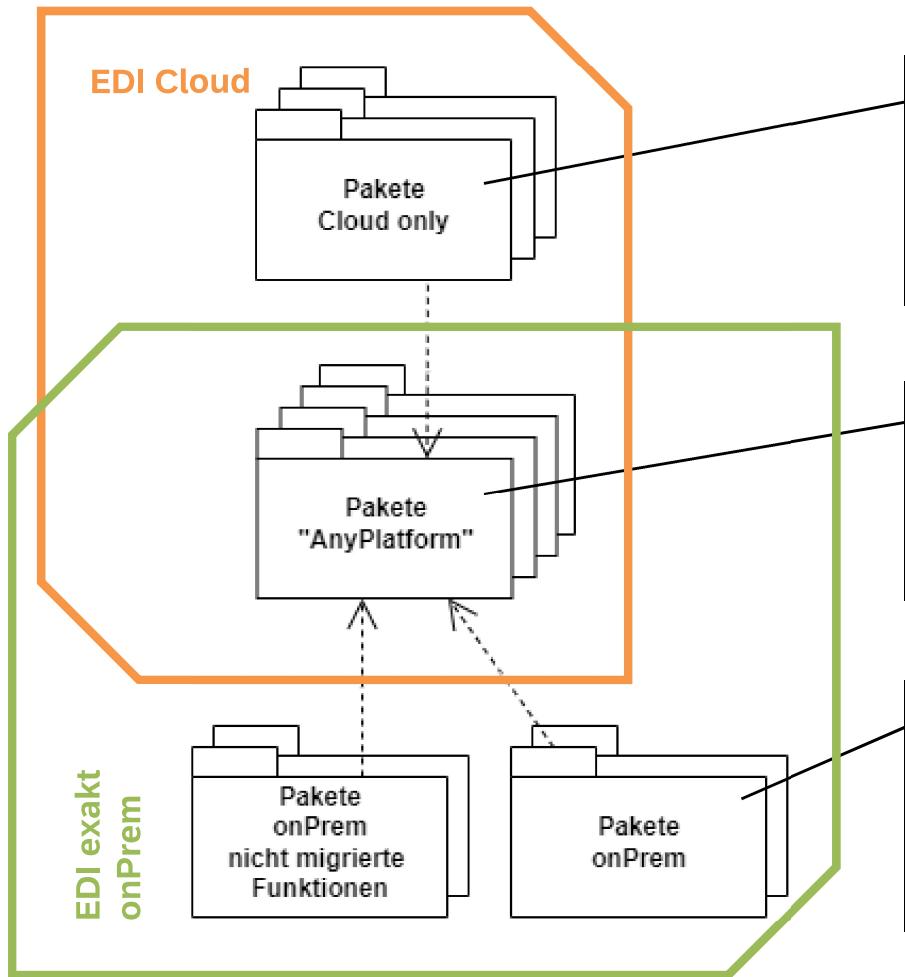
Grundsatzentscheidungen:

- >Gemeinsame Code-Linie oder vollständige Abkoppelung für Cloud?
- >Welche Funktionalitäten übernehmen?

Entscheidung: Eine Code-Linie für Kern-Bestandteil der Anwendung

- >Trennung von onPrem- und Cloud-Bestandteilen in verschiedene Pakete
- >Bestimmte onPrem-Pakete entfallen ganz (bspw. UI-Pakete)
- >Groß angelegtes Refactoring / Modularisierung der Codebasis **nicht** notwendig

Generelle Lösungsstrategie - Zielbild



Je Fachkomponente ein Cloud only Paket

- > RAP-Objekte
- > IAM-Objekte
- > Plattformspezifische Implementierungen
- > Separate UI-Pakete je Fachkomponente für Fiori UIs

Kompatibler Kern von EDI exakt / EDI Cloud

- > Fachkomponenten
- > Großteil der ursprünglichen onPrem-Pakete
- > Lauffähig unter NW ABAP, ERP, S4, BTP

Je Fachkomponente ein onPrem-Paket

- > Im Laufe der Migration entstanden
- > onPrem-Bestandteile herausgelöst, damit Kern kompatibel lauffähig ist
- > Plattformspezifische Implementierungen

Vorgehensweise

abapGit Import in die Cloud (*)

- > Entwicklungsobjekte sind inaktiv
- >Aktivierung fehlerhafter Entwicklungsobjekte („Force activation“)
- >Prüfung via ABAP Test Cockpit (ATC) / Problems View
- >Befunde onPrem beheben + periodischer Import in die Cloud

Aktivierbare Entwicklungsobjekte

- > Fachkomponenten mit RAP verschalen und in Anwendungsinfrastruktur (IAM, UI, etc.) einfassen

(*) Nicht unterstützte Objekttypen oder Bestandteile bei abapGit-Import automatisch rausgefiltert, bspw. Ausführbare Programme, gen. Tabellenpflegedialoge, Suchhilfen, ...

3. Gewählte Design-Taktiken

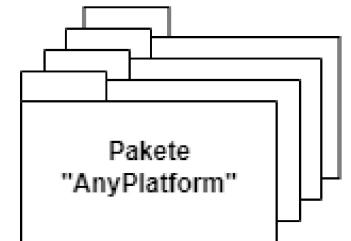
Übersicht Design-Taktiken

(A)
Kompatible Änderung

(B)
**Plattformunterschiede
auslagern**

(C)
Paketverschiebung

(D)
**Cloud-Only
Implementierung**



(A) - Kompatible Änderung

- >Befund kann „AnyPlatform“-like maximal portabel onPrem gelöst werden
- >Hatte oberste Priorität :-)

Beispiele:

- >Eigenes Datenelement für Boolean-Datentyp [`/JA41/APP_BAS_BOOLEAN`](#)
- >Struktur nicht freigegeben —> Als lokalen Datentyp via **TYPES** deklarieren, falls Abhängigkeiten dies zulassen
- >Verwendung nicht freigegebener Domäne im Datenelement —> Eingebauten DDic-Typ im Datenelement festlegen, falls sinnvoll; Bezug zur Domäne entfernen

(B) - Plattformunterschiede auslagern

- > API-Unterschiede in verschiedene implementierende Klassen auslagern —> Verwendung der Public APIs unter Steampunk
- > Diese Design-Taktik häufig verwendet

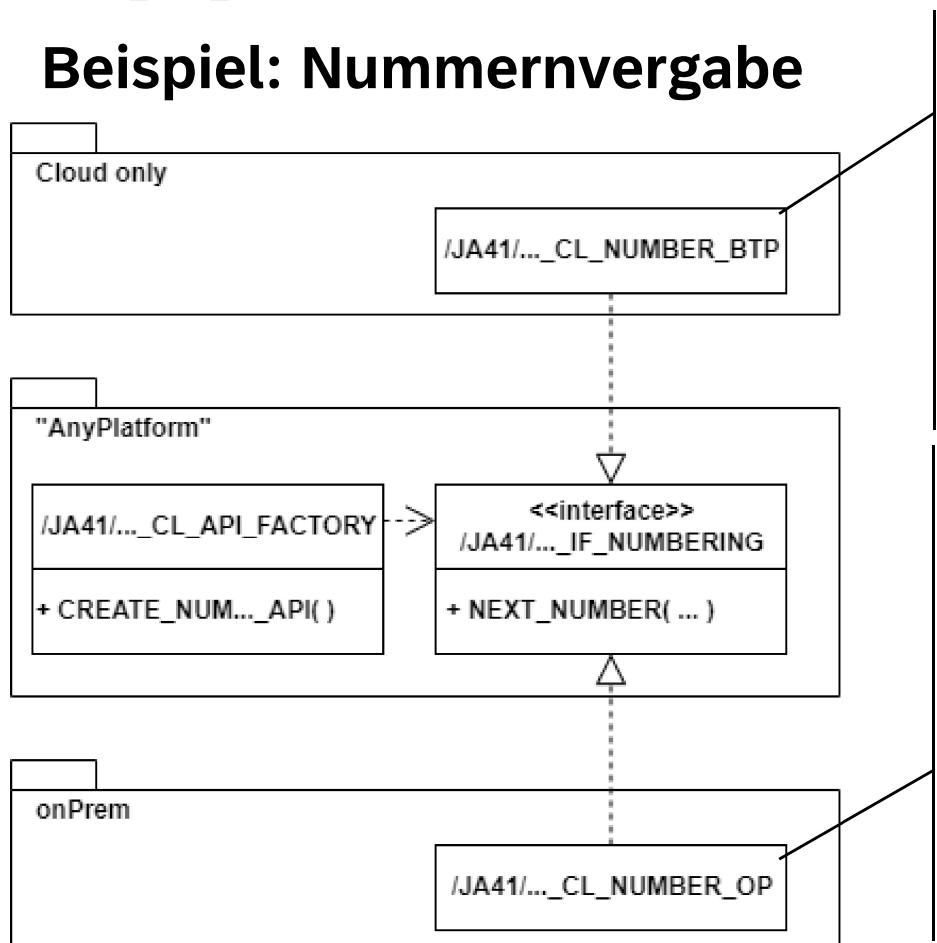
Beispiele:

- > Zugriff auf `sy-datum` via `CL_ABAP_CONTEXT_INFO` (*)
- > Ermittlung RFC-Destination für Remote-Aufruf
- > Utility-Funktionen, etwa String-Konvertierung
- > API XML-Verarbeitung

(*) Zugriff auf `sy-datum` mittlerweile mit Warnung erlaubt

(B) - Plattformunterschiede auslagern

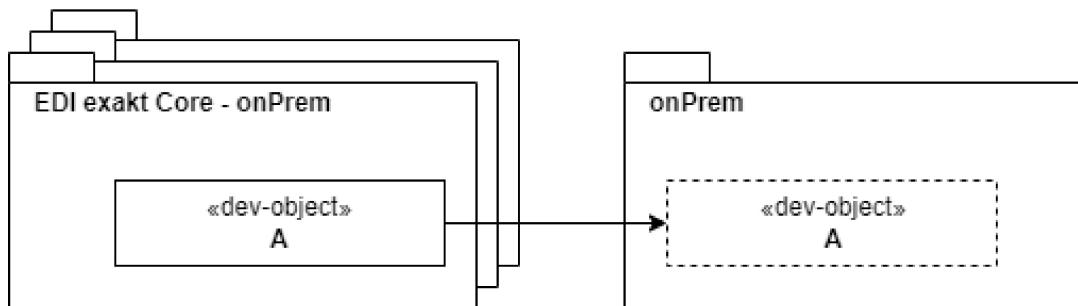
Beispiel: Nummernvergabe



```
METHOD /ja41/...if_numbering~next_number.  
" Public API von Steampunk verwenden  
cl_numberrange_runtime=>number_get()  
EXPORTING  
    nr_range_nr = num_range  
    object      = num_object  
IMPORTING  
    number      = next_number ).  
ENDMETHOD.
```

```
METHOD /ja41/...if_numbering~next_number.  
" Klassischen FB verwenden  
CALL FUNCTION 'NUMBER_GET_NEXT'  
EXPORTING  
    nr_range_nr          = nr_range_nr  
    object              = object  
IMPORTING  
    number              = next_number  
EXCEPTIONS  
    ...  
ENDMETHOD.
```

(C) - Paketverschiebung



- > Einzelne Entwicklungsobjekte, deren Befunden nicht behoben werden und nicht in die Cloud sollen —> in separate onPrem-Pakete verschieben
- > Kann im Vorfeld einer Migration erfolgen
- > Von abapGit gefilterte Entwicklungsobjekte verbleiben im Ursprungspaket

(D) – Cloud-only Implementierung

- > Eigene Paketstruktur hierfür
- > Zusammenhang mit produktivem Deployment beachten (d.h. Abbildung als Software Component mit Clone ins Dev-System)
- > Darunter Anlage der jeweiligen Pakete
- > Verwendet fachlichen Kern („AnyPlatform“)
- > Inhalte im Wesentlichen:
 - > Artefakte aus RAP: CDS-Entitäten, Behavior Definition+Impl., Annotationen
 - > SAP Fiori UIs
 - > IAM-Objekte
 - > Plattformspezifische Implementierungen aus (B)

OpenSQL

ABAP for Cloud Development —> im Projekt unproblematisch, aber ...

...nur neue OpenSQL-Syntax erlaubt —> Zwischenzeitlicher Show-Stopper?

Situation entschärft durch

- >**ABAP Test Cockpit Massenänderungen**
- >hauptsächliche Arbeit an Cloud-Only Komponenten zu diesem Zeitpunkt
- >D.h. Refactoring / Modularisierung OpenSQL ist nicht erfolgt

OpenSQL

The screenshot shows the SAP IDEAS interface with a 'Quick Fix' dialog open over a list of errors and warnings.

IDEAS Navigation Bar: Templates, Bookmarks, Feed Reader, Search, abapGit Staging, Dictionary, A4D_DE, ABAP_CLOUD_READINESS, Letzter Prüflauf.

Description: Findings: 1 Errors, 18 Warnings

- Errors (1 Errors):**
 - Syntaxfehler bei eingeschränktem Sprachumfang (Open-SQL)
- Warnings (18 Warnings):**
 - Usage of deprecated API.
 - Syntaxwarnung bei eingeschränktem Sprachumfang (Forms)
 - Usage of deprecated API.
 - Syntaxwarnung bei eingeschränktem Sprachumfang (Forms)
 - Syntaxwarnung bei eingeschränktem Sprachumfang (Forms)
 - Syntaxwarnung bei eingeschränktem Sprachumfang (SY-Felder)
 - Syntaxwarnung bei eingeschränktem Sprachumfang (SY-Felder)
 - Syntaxwarnung bei eingeschränktem Sprachumfang (SY-Felder)

Quick Fix Dialog:

Select the fix for 'Syntaxfehler bei eingeschränktem Sprachumfang (Open-SQL): ABAP-Sprachversion (Syntax)'.

Select a fix: **OLD SQL durch NEW SQL ersetzen.**

Problems:

Resource	Location	Actions
/JA41/GNS_CON_API_MLC_LIST (Function Module)	line 25	<input checked="" type="checkbox"/> Select All <input type="checkbox"/> Deselect All

Buttons: ? Cancel Finish

4. Verschalen von Geschäftslogik

... mit dem ABAP RESTful Application Programming Model

Wahl des RAP-Szenarios

Legacy-Code in RAP integrieren?

Das RAP-BO muss immer **unmanaged** sein, oder?!

Unser Vorgehen

- >Analyse unserer API
- >Welche Geschäftsobjekte mit welchen API-Operationen sind vorhanden?

Analyse der APIs und Geschäftsobjekte

- > Je Geschäftsobjekt eine Funktionsgruppe mit CRUD-Operationen (BAPI-Style-Granularität mit OO-Implementierung dahinter)
 - + Spezifische Operationen je Geschäftsobjekt
 - + Validierungs-Operationen als API-Bestandteil teilweise vorhanden
 - > Stateless implementiert, somit an dieser Stelle **kein** Transaktionspuffer-Konzept
 - > CUD-Operationen führen direkt DB-Operationen durch
- Begleitende Überlegung: Wie passt das ins RAP-Transaktionsmodell?**

Lesende Zugriffe

- > **Vollständige Modellierung** via Core Data Services als Basis eines RAP-BOs
 - > Neue API via CDS, Berechtigungsprüfung via CDS-Zugriffskontrolle
- > Für alle Geschäftsobjekte
- > Kein Reuse der Legacy-API
- > Vertretbar, da: Wenig komplexe lesende Zugriffe mit sehr geringer Änderungshäufigkeit

- > Es gab jedoch zwei Ausnahmen, eine davon ist ...

Lesende Zugriffe - Spezialfall

... Berechnungen zur Laufzeit für Monitoring der Fachkomponenten, d.h. ohne Persistenz

Lösung: Custom CDS Entities

- >Custom CDS Entity für Monitoring-Berechnungen
 - > vollständig **unmanaged**
 - > Reuse der Legacy-API für lesende Zugriffe

Unmanaged Query für Monitoring

```
METHOD if_rap_query_provider~select.  
  DATA lt_nodes TYPE /ja41/gns_mon_tree_tt.  
  TRY.  
    " Legacy-API aufrufen, für Ermittlung der Monitoring-Information  
    CALL FUNCTION '/JA41/GNS_MON_API_TREE_READ'  
      EXPORTING  
        language = sy-langu  
      IMPORTING  
        nodes      = lt_nodes.  
    " Konvertierung von Legacy-Datentyp nach Custom CDS Entity  
    lt_data = VALUE #( FOR n IN lt_nodes  
      ( Id = n-id  
        Title = n-title ... ) ).  
  " ... Unmanaged Query Kontrakt implementieren: Filter auswerten, sortieren, etc.  
  CATCH /ja41/cx_gns ... ENDTRY.  
ENDMETHOD.
```

Schreibende Zugriffe

>Customizing mit CUD-Operationen

> **managed** Scenario gewählt

> Einfache Ablösung der generierten Pflegedialoge

>Stammdaten mit CUD-Operationen

> **managed with unmanaged save** Scenario gewählt

> **Interaktionsphase von RAP-Runtime übernommen**

> Validierungs-API der Legacy-API per **validation** integriert

> Legacy-API war stateless implementiert und führt direkt DB-Operationen durch
—> Aufruf im **Speicher-Handler** in **save_modified()**

Schreibende Zugriffe

>Customizing mit CUD-Operationen

> **managed** Scenario gewählt

> Einfache Ablösung der generierten Pflegedialoge

>Stammdaten mit CUD-Operationen

> **managed with unmanaged save** Scenario gewählt

> **Interaktionsphase von RAP-Runtime übernommen**

> Validierungs-API der Legacy-API per **validation** integriert

> Legacy-API war stateless implementiert und führt direkt DB-Operationen durch
—> Aufruf im **Speicher-Handler** in **save_modified()**

Schreibende Zugriffe - Spezialfall

Bewegungsdatum „EDI-Nachricht“

Verarbeitungslogik komplex und passt nicht ins RAP-Transaktionsmodell

- > Nach jedem Verarbeitungsschritt der EDI-Nachricht erfolgt ein expliziter Commit Work oder Rollback
- > Workaround: Legacy-API in parallelen Task via `CL_ABAP_PARALLEL` gewrappt ...
 - > ... Legacy-API kümmert sich weiterhin um Locking und Transaktionssteuerung

5. UI & Multitenancy

EDI Cloud: UI

Integration ins Fiori Launchpad von Steampunk via Spaces & Pages

Mini-Erweiterung: Custom Action zur Anlage einer EDI-Nachricht via File-Upload

Fiori Elements List Report für EDI-Nachricht

Nachrichten-ID	Status	Angelegt am	Sender	Sender Werk	EDI-Standard - Send.	Nachrichtentyp Send.	Version - Sen...
1000000006	✓ OK	28.01.2022, 13:54:39	CUSTOMER		EDIFACT	DELFOR	D97A
1000000005	✓ OK	28.01.2022, 12:22:45	CUSTOMER		EDIFACT	DELFOR	D97A
1000000004	✓ OK	26.01.2022, 17:44:46	CUSTOMER		EDIFACT	DELFOR	D97A
1000000003	✓ OK	26.01.2022, 17:41:18	CUSTOMER		EDIFACT	DELFOR	D97A
1000000002	✓ OK	03.12.2021, 11:37:57	CUSTOMER		EDIFACT	DELFOR	D97A
1000000001	✓ OK	02.12.2021, 16:18:10	CUSTOMER		EDIFACT	DELFOR	D97A
21	✓ OK	02.12.2021, 16:03:17	CUSTOMER		EDIFACT	DELFOR	D97A
20	⚠ WAIT	02.12.2021, 15:32:40	CUSTOMER		EDIFACT	DELFOR	D97A
19	⚠ WAIT	02.12.2021, 14:47:26	CUSTOMER		EDIFACT	DELFOR	D97A
18	✓ OK	02.12.2021, 14:40:28	CUSTOMER		EDIFACT	DELFOR	D97A
17	⚠ WAIT	02.12.2021, 13:12:39	CUSTOMER		EDIFACT	DELFOR	D97A

Multitenancy und SaaS-Fähigkeit

- > Multitenancy Development Guideline von SAP beachten (*)
 - > Keine Befunde —> Anwendung ohne weiteres Multitenancy geeignet
- > Eigene AppRouter-Komponente (node.js) notwendig

Provisionierung von Kunden-Tenants

- > Aktuell noch manueller Aufwand auf Ebene SAP BTP
 - > Spaces & Pages müssen manuell angelegt werden
 - > Einrichtung von EDI Cloud auf Kunden-Tenant mit manuellem Aufwand verbunden
- > Workaround: Customizing Provisionierung für Produktiv-Tenant notwendig !
- > Warum? —> DB-Inhalte per Business Configuration derzeit nur nach Mandant 100 importierbar

(*) Siehe SAP Help Portal —> SAP Business Technology Platform

6. Fazit

... zum Custom Code Migration Projekt

Fazit

Migrationsaufwand nicht unterschätzen

—> Custom Code Migration = Mehr als nur Quellcode anpassen ...

Eine Menge an Themen ...

- >Entwicklungsumgebung + Einrichtung (bspw. Git-Repos)
- >SAP BTP Account-Modell
- >RAP
- >Neue Anwendungsinfrastruktur (IAM, Konnektivität, Logging, etc.),
nicht nur neue APIs

- >Workarounds via Steampunk kreativ möglich

Fazit II

- > Tenant-Provisionierung
 - > Optimierungspotential vorhanden
- > Eine Code-Linie als Strategie gut für den Übergang !
 - > Mit Einschränkungen im Laufe der Zeit —> Angepasster Ansatz gewählt
 - > Cloud-First Ansatz hinsichtlich neuer Funktionen nun gewählt
- > RAP: **managed** Scenario für mehr Fälle nutzbar, als zunächst angenommen

Thank you! + About...

Matthias Jäger

- > Freelance SAP ABAP Development Expert & Architect | SAP-PRESS Autor
- > matthias.jaeger@erp-developer.com

Andreas Klose, All for One Group SE

- > Senior Manager Solution Development
- > andreas.klose@all-for-one.com

