

Hauptüberschrift

Name des Autors
Firma
Standort

Schlüsselworte

Bitte schreiben Sie hier Ihre Schlüsselwörter.

Als klassischer Business-toBusiness-Informationsanbieter hat der Fachverlag HUSS-MEDIEN einen großen Kundenstamm und eine Vielzahl an verschiedenen Produkten im Portfolio. Zur Verwaltung der Kundendaten und Abonnements wurde vor vielen Jahren eine interne Anwendung mit dem Oracle Application Development Framework (ADF) 10g implementiert.

22. Januar 2014 12:12:49 [Dokumentation](#)
Angemeldet als [Praktikant](#) [Support](#)

Stammdaten | Produktschlüssel/Merkmale | Branche79 | Branche2008 | Sonstiges | Werbung | OA | EE

Suche | Aktivität neu anlegen | meine Aktivitäten | Selektion | Tools | Logout

speichern | Adresse neu anlegen | abbrechen | Änderungshistorie | ExportEE | als Etikett speichern

Kundennummer: 16543328

HUSS-MEDIEN GmbH
Am Friedrichshain 22
10407 Berlin
LKZ: D

Tel: 030/42151-0
Fax: 030/42151-273
E-Mail: huss.medien@hussberlin.de
B08: 46.49.4/58.11.0/58.14.0/82.3/
B79: 76

ADR ZAHL FREMD DB
ABO ANZ WV OA

- Stammdaten
- Ansprechpartner
- Bank
- Dublettenadresse
- Abonnement
- Anzeigen
- Warenvertrieb
- Kundenaktivitäten
- Adressbeziehungen
- Dokumente
- Debitorendatensätze

Kundenstammdaten

Letzte Änderung vom 22.01.2014

Kundenart: Anrede: Titel:

Name1:
Name2:
Name3:

Telefon: bevorzugt:
Mobil: Telefon
Fax: Mobil
E-Mail: Telefax
Web: eMail Brief

Sonstiges

UStID:
Steuernummer:
Geburtsdag:

Status

Adress-Status:
Zahl-Status:
Mahnsperre:

speichern

Notizen

Adresse: 0/1000
Anzeigen: Abonnement:
Warenvertrieb: Debitoren:

Stammdaten | Produktschlüssel/Merkmale | Branche79 | Branche2008 | Sonstiges | Werbung | OA | EE

Sich in ADF einzuarbeiten stellt für viele Entwickler eine große Hürde dar und beansprucht oft einen hohen Zeitaufwand. Dieser steigt enorm wenn keine Unterstützung durch einen erfahrenen ADF Profi gegeben ist. Dennoch stellten sich die damaligen Mitarbeiter dieser Herausforderung allein. Dabei mussten Sie sich nicht nur mit den Konzepten der ADF Entwicklung vertraut machen, sondern waren zugleich unerfahren im Umgang mit Java als Programmiersprache. Zuvor hatten die Entwickler lange Zeit mit Oracle Forms und prozeduraler Programmierung in PL/SQL gearbeitet. Nun musste man sich mit ADF/Java und

Objektorientierter Programmierung auseinandersetzen.

Am Ende entstand eine stabile und lauffähige Anwendung mit der das Unternehmen viele Jahre zufrieden arbeiten konnte. Dennoch wurden bei der Entwicklung diverse Konzepte wie Schichtentrennung im Model-View-Controller (MVC) Pattern missachtet. Diese und weitere "eigenwillige" Implementierungen, führten immer weiter zu einem Berg an technischen Schulden mit dem man sich irgendwann befassen müsste.

Mit der Zeit entwickelte sich ADF weiter und bietet heute eine wesentlich größere Vielfalt an Komponenten mit einer ansprechenden Oberfläche. Beim Vergleich zwischen der etwas angestaubten 10g Applikation und dem aktuellen ADF 12c wird auf den ersten Blick klar, dass die Oberfläche der aktuellen Anwendung nicht mehr ganz zeitgemäß ist. Zudem ist das beheben der Fehler zeitaufwändig und Weiterentwicklungen stoßen immer häufiger an die Grenzen der 10g Version.

Darum wurde beschlossen die Software auf einen aktuellen Stand zu bringen und am liebsten gleich bis ADF 12c zu migrieren.

In einem ersten Anlauf, wurde die ADF 10g Version auf die nächst höhere ADF 11g Version migriert. Nach einigen Anpassungen an den Klassenpfaden und den Profilen verblieben noch immer über 200 Errors die man nun, nach und nach sichtete. Dabei wurde schnell klar, das in vielen Bereichen der Anwendungen, keine saubere Schichtentrennung vorgenommen wurde. Hinzu kommt, dass bei einer Migration von ADF 10g zu 11g das Front-End von UX zu Trinidad umgewandelt wird. Das mag an manchen Stellen ganz OK sein. Grundsätzlich kann man sich jedoch eher darauf einstellen, das komplette Front-End neu entwickeln zu müssen.

Nun ging es darum, wie weiter mit der Migration umgegangen werden sollte. Die Errors beheben und die Migration durchziehen, hätte einen hohen Aufwand gehabt und letztendlich wäre dabei ein Großteil der technischen Schulden bestehen geblieben. Eine direkte Neuimplementierung ist jedoch auch keine Alternative. Zwar wäre der Entwicklungsaufwand aufgrund der neuen Komponenten und Wizards in ADF durchaus überschaubar gewesen. Dennoch würde die Neuentwicklung einen langen Zeitraum in Anspruch nehmen. Innerhalb dieser Zeit wären Änderungen und neue Features in der alten Anwendung liegen geblieben.

Die letzte Alternative war eine Neuimplementierung im Parallelbetrieb. Dabei könnten die technischen Schulden abgebaut werden und Bereiche die neue Anforderungen haben, werden als erstes in ADF 12c neu geschrieben. Auch der direkte Sprung von ADF 10g zu 12c ist so möglich.

Die Idee klang vielversprechend genug um die Anforderungen zu definieren und einen POC zu bauen. Dieser zeigte schnell das es möglich ist die Anwendungen parallel zu betreiben und via Java Script beide mit der selben Kunden-ID zu versorgen.

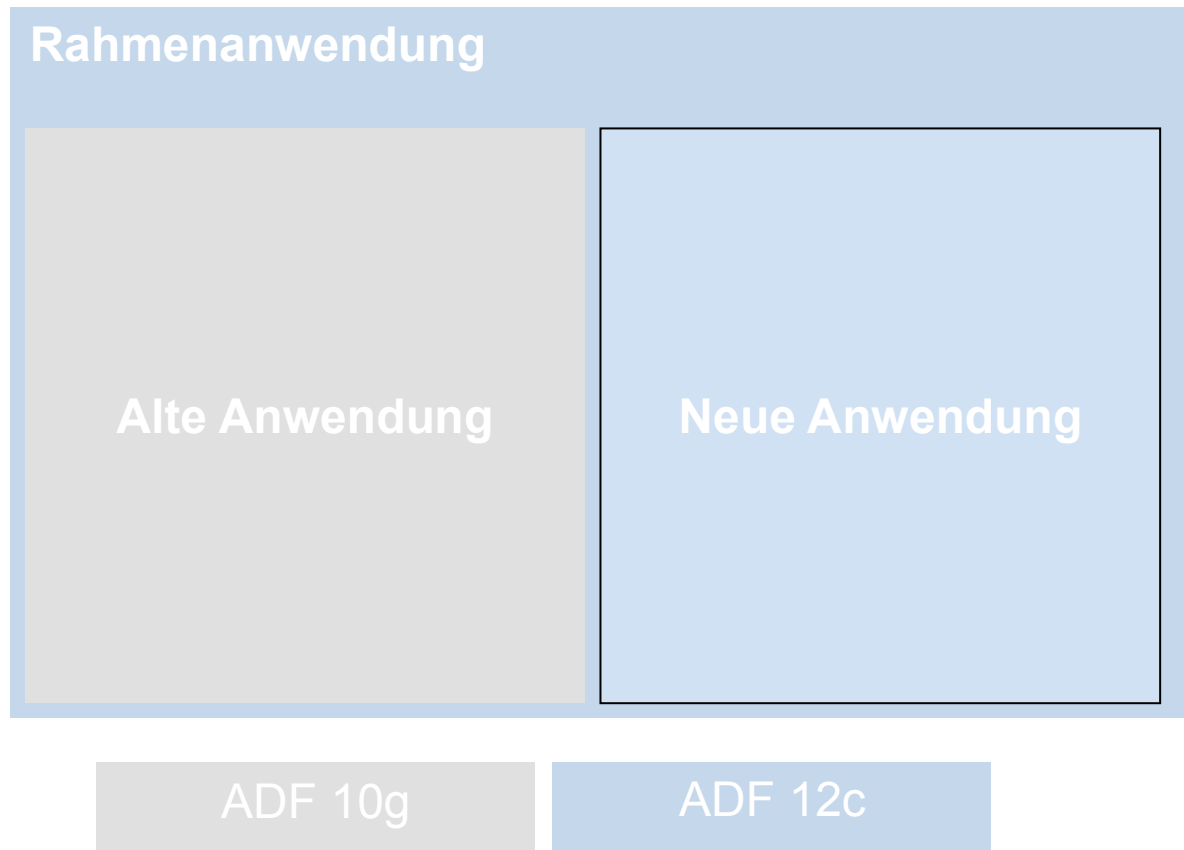


Abb. 1: Konzept

Zum aktuellen Zeitpunkt ist die alte 10g Anwendung bereits innerhalb eines iFrames in der ADF 12c Anwendung integriert. Nun kann mit dem partiellen Neubau einzelner Module begonnen werden.