

## Z-Merge: Echte Server-to-Server Kopplung

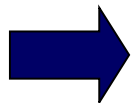
Sebastian Kummer; [s.kummer@zarafa.com](mailto:s.kummer@zarafa.com)



## Z-Merge – Warum Server koppeln?

*Mehrere Datenbanken und Anwendungen enthalten gleiche Daten*

- **Zarafa:** E-Mail, Kalender, Kontakte, Aufgaben
- **CRM-Systems:** Kundenkontakte, archivierte E-Mail, Kalender (Termine, Besprechungen, Anrufe), Aufgaben
- **Projektmanagement:** Aufgaben, Besprechungen



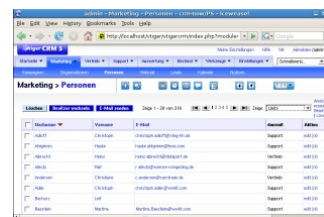
*Kunden und Anwender benutzen Funktionalitäten auf allen Systemen und versuchen (?) Daten auf gleichem Stand zu halten*

## Wie funktioniert es heute?

### *Outlook als gemeinsame Grundlage*

- Outlook ist auf allen Client-Rechnern installiert und ist stets verfügbar
- Anwendungs-Plugin wird in Outlook installiert
- Synchronisation wird vom Plugin durch Benutzerinteraktion durchgeführt
- Benutzer Know-How muss vorhanden sein
- Probleme bei serverbasierten Profilen (Outlook auf mehreren Geräten für einen Nutzer installiert)
- Keine Funktionalität wenn Rechner/Outlook ausgeschaltet ist

### Outlook + Outlook-Plugin

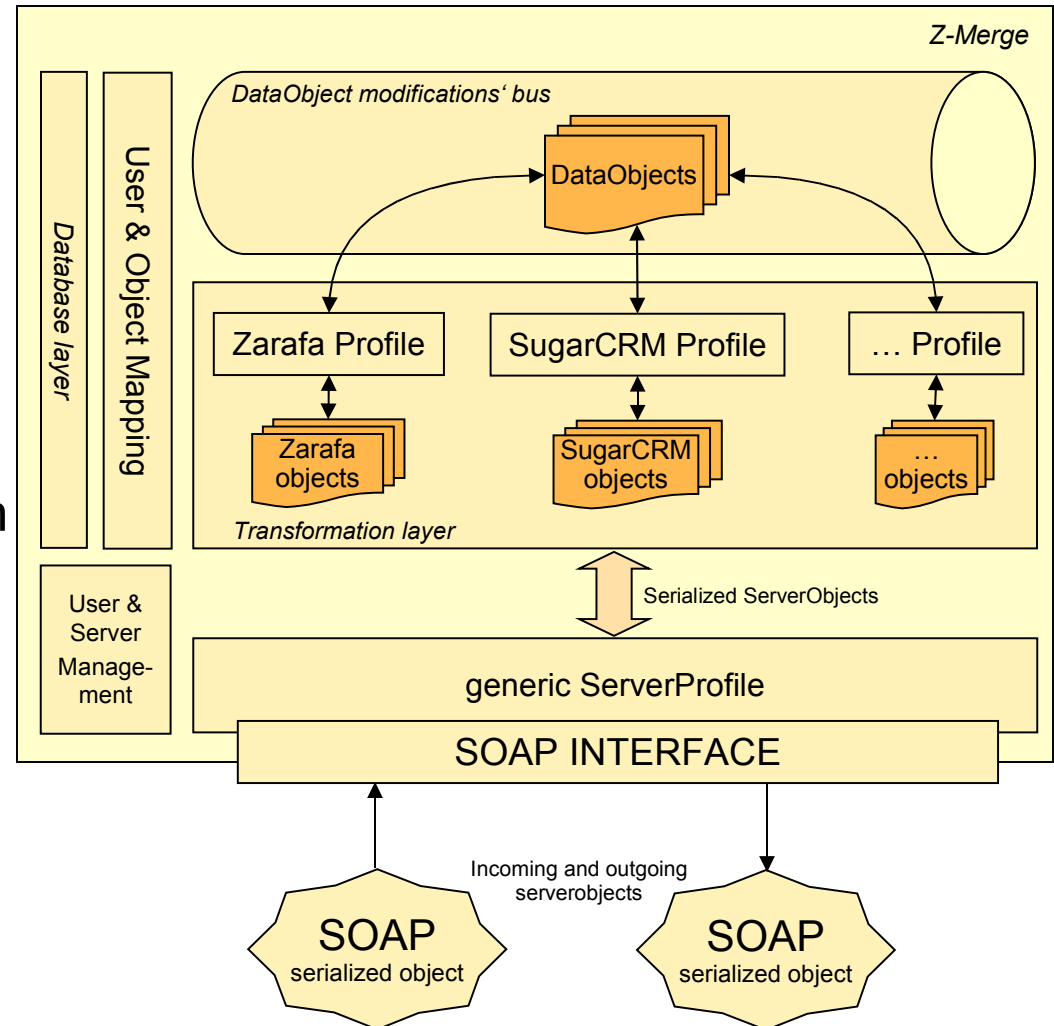


## Warum Z-Merge?

- Verlässlicher Datenabgleich **ohne** Dubletten zu erzeugen
- Unabhängig von Benutzer-Know-How und Benutzung
- Kopplung ohne Benutzerinteraktion
- Möglichkeit auch mehr als zwei Server gleichzeitig abzugleichen
- Verlässliche Funktion auch mit grossen Datenbanken
- Möglichkeit auch private Objekte (nach gewissen Regeln) abzugleichen
  - *z.B. werden dem Benutzer zugewiesene Aufgaben automatisch in seinen Store übertragen (und sind somit automatisch mobil verfügbar)*
- Bekannte Instabilitäten mit “Konnektor-Anwendungen zu Outlook” vermeiden

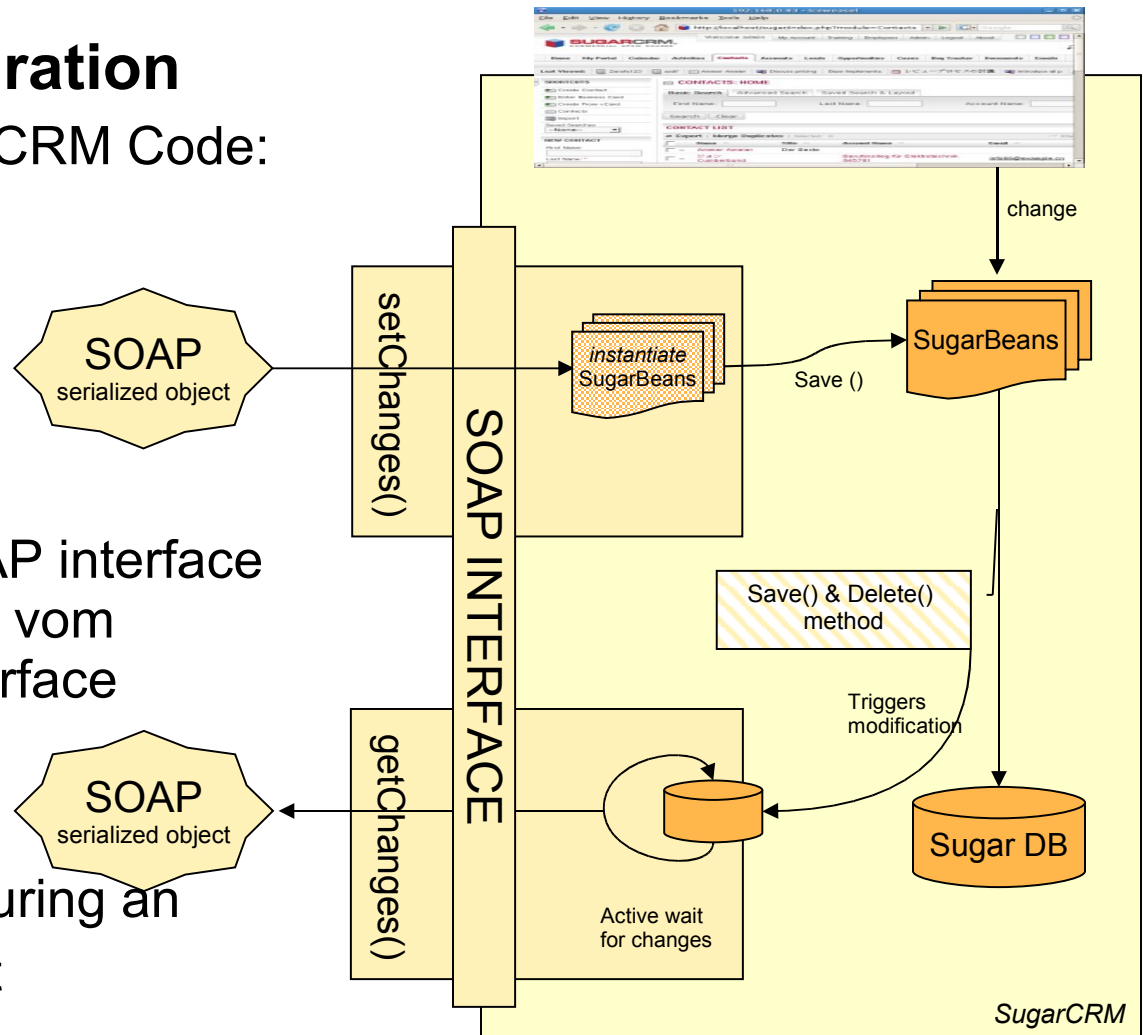
## Wie macht's Z-Merge ?

- Monitoriert alle angeschlossenen Server
- Sendet und empfängt Objekte von Drittanwendungen
- Bi-direktionale transformation von Objeten von und zu Z-Merge
- DatenObjekte haben gemeinsame user/gruppen und Objektdefinitionen
- ServerAnbindung unterstützt Gruppen zu Benutzer Mapping



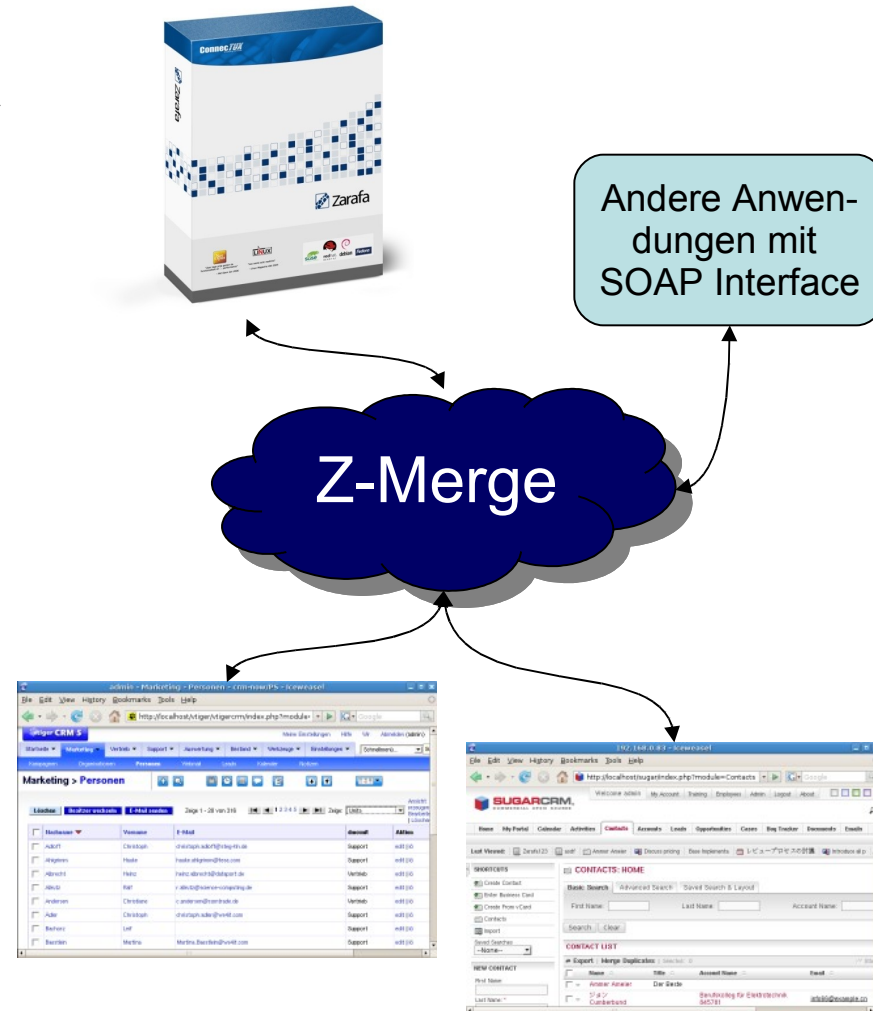
## Nicht-invasive Integration

- Änderungen im SugarCRM Code: **8 loc** im *SugarBean*
- Keine Datentransformation notwendig
- Sehr leichtgewichtige Implementation
- Eigenes Z-Merge SOAP interface und damit unabhängig vom SugarCRM SOAP interface
- Änderungen werden nach max. 0,25s durch das Event-Capturing an Z-Merge weitergeleitet



## Server-to-Server Synchronisation ist Besser

- Outlook-Konnektor nicht länger nötig
- Synchronisation ohne Benutzerinteraktion
- Änderungen werden fast in **echtzeit** an alle angeschlossenen Server propagiert (ca. 1 sec)
- Alle Anwendungen arbeiten mit gleichen Daten
- Kompatible mit allen Systemen welche eine SOAP-Schnittstelle bieten



## Mehr über Z-Merge?

- Propagieren von Änderungen
  - Wenn ein Server offline ist, gehen Änderungen nicht verloren, sondern werden nachträglich eingebunden sowie der Server wieder online ist
  - Server Verfügbarkeit wird durch einen “watch-dog” Mechanismus sichergestellt
- User/Group Mapping kann durch LDAP-Anbindung implementiert werden

