



# Hauptseminar “Wartung von Softwaresystemen”

Legacy Migrationsstrategien

Christoph Erdle

13. Dezember 2005



1. Einführung und Definitionen
2. Migrationsstrategien
3. Migration bei verschiedenen Systemstrukturen
4. Datenmigration
5. Beispiel: Domänen-Migration Windows
6. Zusammenfassung



# 1. Einführung und Definition

- Legacy System: abzulösendes Altsystem, wird Anforderungen nicht mehr gerecht
- Oft verwendet im Umfeld betrieblicher Informationssysteme (ERP, CRM, ...)
- Kennzeichen: hohe Instandhaltungskosten, wartungsintensiv, anpassungsresistent
- Migration: wesentliche Systemteile werden von einem in ein anderes System überführt



# 1. Wann migrieren?



- Finanzielle Lage
- Bei Ersatz durch existierendes Produkt:
  - Reifegrad
  - Erfahrungswerte
- Bei Individuallösung
  - Einbindung der Mitarbeiter



# 1. Warum Migration?

- Objektive Bewertungverfahren nötig: z.B. “SRAH”
- Betrachten technischer, wirtschaftlicher und Managementaspekte
- Beispielhafte Fragen:
  - Aktuelles System noch für (Geschäfts-)Prozesse der Firma geeignet?
  - System noch wartbar?
  - Support von Hersteller noch gegeben?



# 1. Migration – wirtschaftliche Aspekte

- Ersatzsystem bereits verfügbar oder Individuallösung entwickeln
- Finanzielle Belastbarkeit des Unternehmens
- Migration durch externe Firma oder eigenes Personal
- Kosten für Schulungen, laufende Verträge, ...



## 2. Migrationsstrategien - Anforderungen

- ununterbrochener, sicherer, zuverlässiger Betrieb
- so viele Änderungen wie vernünftig
- so wenige Änderungen wie möglich
- alten Code ändern, dass er Migration vereinfacht
- möglichst große Flexibilität für spätere Anpassungen integrieren
- mögliche negative Auswirkungen minimieren
- Nutzen moderner Technologien und Funktionen maximieren



## 2. Migrationsstrategien - “Chicken Little” (1)

Orientierung an 11 kleinen Schritten” zur Migration  
(iteratives Vorgehen, komponentenweise):

- 1) Analyse des Altsystems
- 2) Zerlegen des Altsystems
- 3) Benutzeroberfläche des Zielsystems entwickeln
- 4) Entwickeln der Zielanwendung
- 5) Entwickeln des Datenbankbackends



## 2. Migrationsstrategien - “Chicken Little” (2)

- 6) Installation der Zielumgebung
- 7) Entwicklung und Installation von Gateways
- 8) Migration der Datenbank
- 9) Migration der Altanwendung
- 10) Migration der Benutzeroberfläche
- 11) Umschalten der Komponenten vom Altsystem auf das Zielsystem



## 2. Migrationsstrategien - “Chicken Little” (3)

Vorteile:

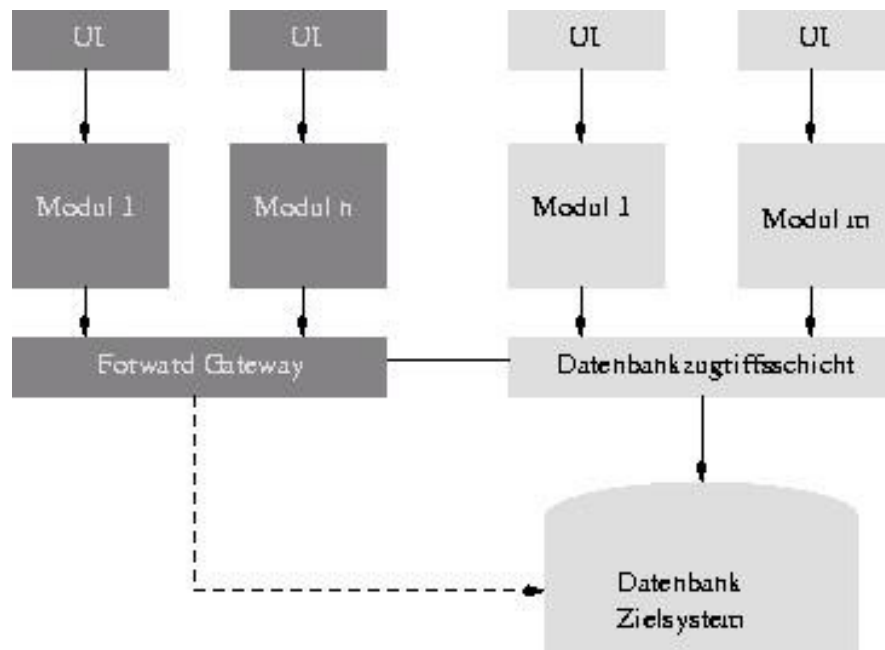
- Iteratives Vorgehen  $\Rightarrow$  geringes Projektrisiko
- Fehler werden schnell gefunden, da Prüfung nach jeder Komponente möglich

Nachteile:

- Gateways komplex



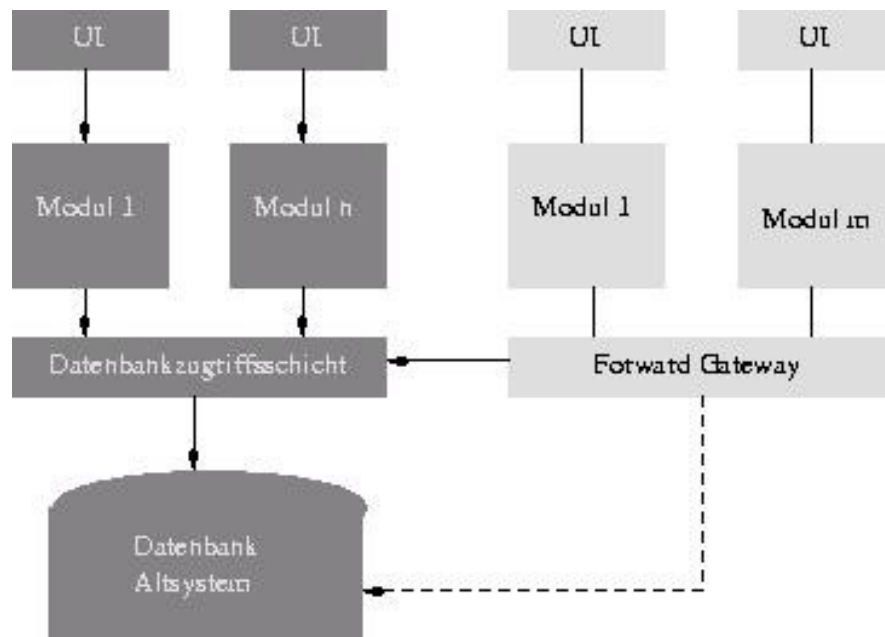
## 2. Migrationsstrategien - “Chicken Little” (4)



### “Database First”

- Erst Migration der Datenbank, dann der Komponenten
- Anwendung passt immer zur Datenbank
- Nur für Systeme mit definierter Datenbank-schnittstelle

## 2. Migrationsstrategien - “Chicken Little” (5)



### “Database Last”

- Schrittweise Migration der Anwendungen des Altsystems
- Zugriff des neuen Systems über Gateway auf alte Datenbasis
- Nur für Systeme mit definierter Datenbank-schnittstelle



## 2. Migrationsstrategien - “Cold Turkey”/”Big Bang” (1)

“Big Bang”: Entwicklung unabhängig vom Altsystem,  
finales Umschalten des kompletten Systems

Probleme bei diesem Ansatz:

- Anpassungen des Altsystems müssen auch in bereits fertige Teile eingepflegt werden
- Hohes Projektrisiko: bereits kleinste Fehler lassen ganzes Projekt scheitern



## 2. Migrationsstrategien - “Cold Turkey”/“Big Bang” (2)

- Oft undokumentierte Abhängigkeiten zwischen Altsystem und anderen Systemen
- Dauer der Migration der Daten  $\Rightarrow$  Downtime
- Management solch großer Projekte schwierig
- Roadmap kann kaum eingehalten werden



## 2. Migrationsstrategien - “Cold Turkey”/“Big Bang” (3)

Vorteile:

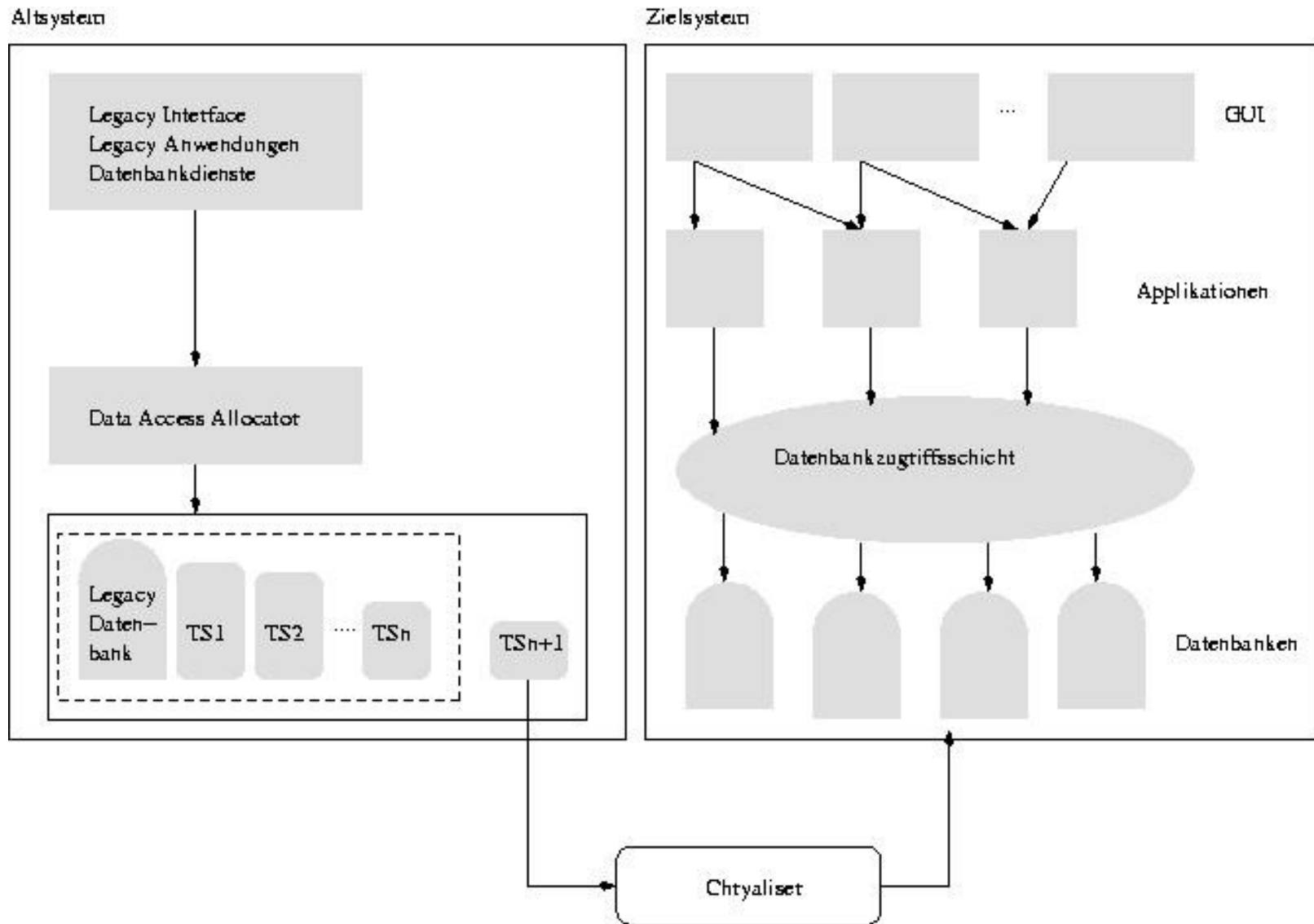
- Erfolg nach Inbetriebnahme sofort sichtbar

Nachteile:

- Im Fehlerfall schlägt komplette Migration fehl
- Nicht leicht erkennbar, wo Fehler liegen
- Kosten: Kosten Entwicklung + Pflege Altsystem bis Ablösung



# 2. Migrationsstrategien - “Butterfly” (1)





## 2. Migrationsstrategien - “Butterfly” (2)

Annahme: Keine Kooperation zwischen Alt- und Neusystem während Migration!

- Altdatenbank schreibgeschützt
- Alle Abfragen laufen über DAA
- Änderungen werden in Temporärspeicher geschrieben
- Migration des Datenbestandes in Altdatenbank
- Anschliessend sukzessives Abarbeiten der Temporärspeicher



## 2. Migrationsstrategien - “Butterfly” (3)

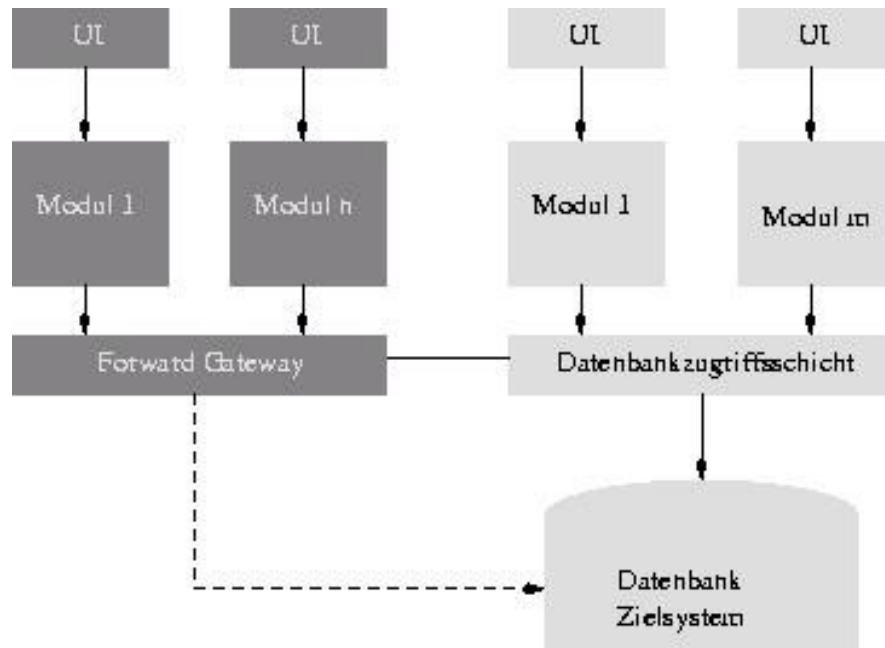
Vorteile:

- Keine Gateways nötig!
- Methode basiert auf Migration der Datenbank
- Altsystem bleibt betriebsbereit, keine Ausfallzeiten
- Abbruch der Migration bis zum Cut-Over möglich, Recovery über Temporärspeicher

Nachteile:

- Data Access Allocator komplex
- Annahme nicht realistisch

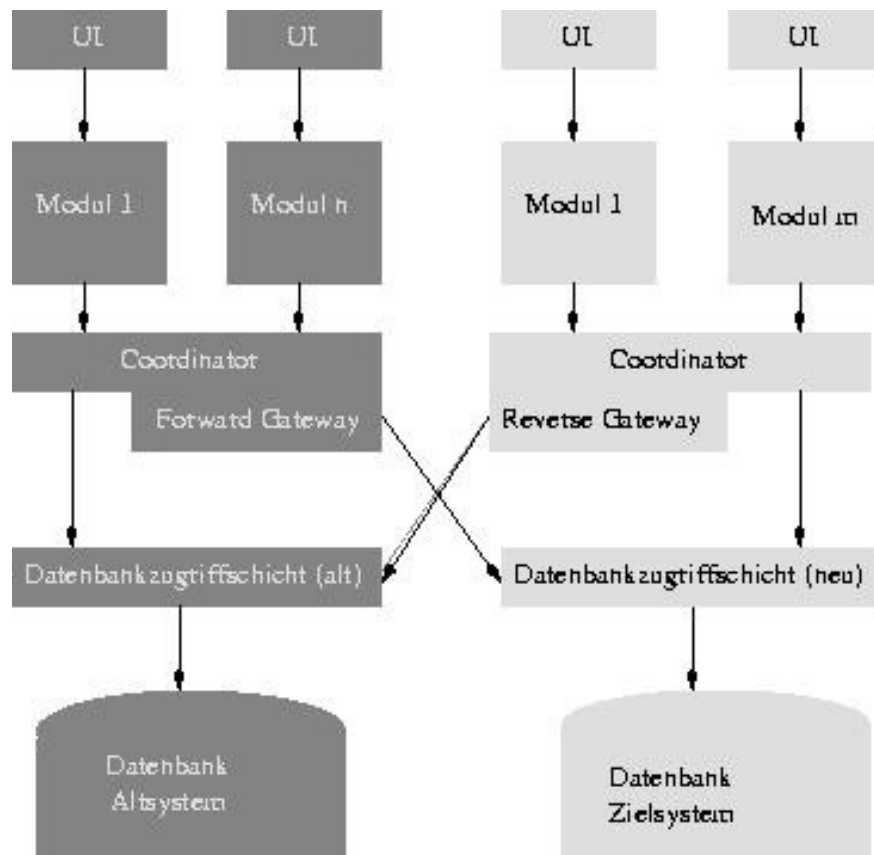
# 3. Vollständig zerlegbare Systeme



- Definierte Schnittstellen zwischen GUI, Applikation, DB
- Beispiel Forward-Migration: Ersetzen der Datenbank, Zugriff für Altsystem über Datenbankgateway
- Analog: Backward-Migration

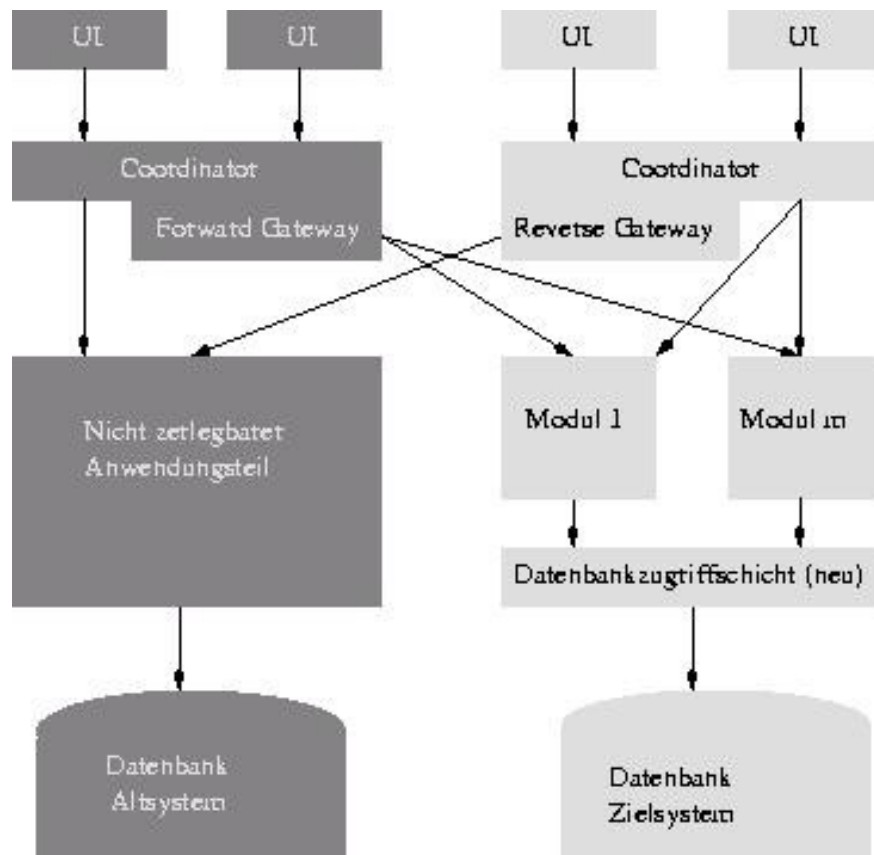


# 3. Vollständig zerlegbare Systeme - Allg. Migration



- Verschmelzen von Forward- und Backward-Migration
- Routing über Coordinator
- Kein Ausfall des Benutzerbetriebs
- Sinnvoll für unternehmenskritische Anwendungen

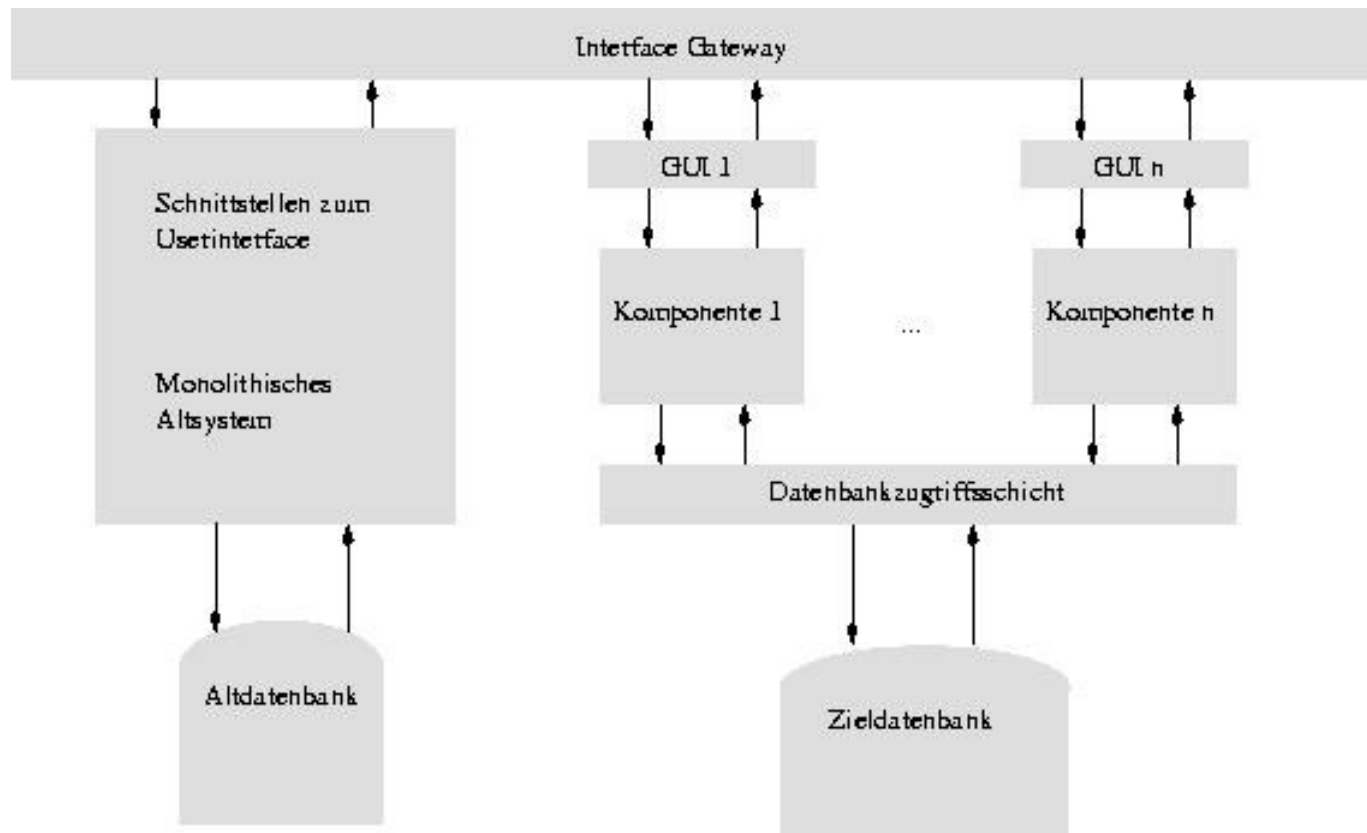
# 3. Teilweise zerlegbare Systeme



- Ebene der Gateways zwischen UI und Anwendung
- Routing über Coordinator
- Durch monolithischen Anwendungskern Coordinator komplex



# 3. Monolithische Systeme (1)





# 3. Monolithische Systeme (2)



- Schnittpunkt der Systeme ist nur das Userinterface
- Implementierung eines Gateways
- “Mapping table” zum Abrufen der Daten nötig



# 4. Datenmigration

- Migrationsstrategien meist nur Codebasiert
- Auch Daten müssen migriert werden
- Beispiele
  - Änderung des Titels einer Person  
Altsystem: Freitext  
Neusystem: Auswahl
  - Beispiel 2: Umstellung Postleitzahlen (~1990)  
Postleitzahl Teil der Kundennummer
- Folgen: ständige Datentransformation nötig,  
rechenintensiv



# 5. Beispiel: Windows NT - Windows 2000/2003

## Gründe für eine Migration

- Funktion
- Funktionalität
- Sicherheit
- Support
- Beispiele: fehlende Treiberunterstützung, Spezialsoftware



# 5. Beispiel: Probleme



- nicht mehr unterstützte Hardware, Inkompatibilitäten
- Nicht mehr unterstützte Software
- Sicherheitsvorkehrungen: z.B. SQL Server 2000
- ...



# 5. Beispiel: Ablauf (1)

- Analyse der Funktion anhand Dokumentation, Erstellung Pflichtenheft
- Test auf Funktion des Altsystems nach Dokumentation
- Backup des Altsystems
- Einrichten eines neuen Rechners als Windows-2000/2003-Domain-Controller, Test
- Windows-NT-Migration-Wizard starten



# 5. Beispiel: Ablauf (2)

- Kopieren der User-Homes und Profiles, anschliessender Test
- Einrichten eines weiteren Windows-2000/2003-Domain-Controllers
- Einrichten der übrigen geforderten Eigenschaften: Dienstkonfiguration (DNS, DHCP, IIS, Radius, VPN, ...) sowie Softwareinstallation



# 5. Beispiel: Zusammenfassung



- Windowsmigration ist “Cold Turkey”
- Funktion des Systems erst nach Abschluss der Migration testbar
- Fehleranfällig durch Fehl-/Nicht-Standard-Konfiguration während des Prozesses



# 6. Zusammenfassung

- Keine globale Empfehlung möglich
- Geringe Änderungen am Datenschema  $\Rightarrow$  "Chicken Little"
- Größere Schemaänderungen  $\Rightarrow$  "Cold Turkey"
- Entscheidung immer an jeweilige Situation angepasst



Vielen Dank für die Aufmerksamkeit