

Fallstudie

Spezifikation und Entwurf eines grossen verteilten Systems

Bernhard Rytz

bernhard.rytz <bei> sbb.ch

Bernhard Rytz 2006 / 01

Überblick

Ziele

Anhand Fallstudie...

- Einblick in Besonderheiten bei der Anforderungsanalyse grosser Projekte
- Verständnis für Rolle der Software-Architektur für ein grosses, verteiltes System

Fragen jederzeit !

Inhalt

- Einstieg Fallstudie
- Anforderungsanalyse
 - viele Bedürfnisträger
 - Vorgehen und Mittel
 - Beurteilung
- Architektur & Design
 - Was ist Architektur ?
 - Architektur-Varianten
 - Architektur-Sichten
 - Beurteilung
- Unternehmensarchitektur
- Zusammenfassung

Einstieg Fallstudie

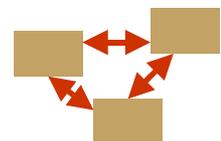
Bernhard Rytz 2006 / 01

Ausgangslage (1)

Ziele

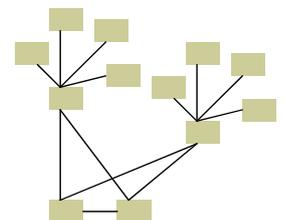
- grosser Bankdienstleister will stärker kundenorientierte Sicht und Verbesserung der Abwicklung in der Kundenberatung
- grosse, existierende Systeme müssen dazu besser integriert werden

```
Name : .....  
Vorname : .....  
Adresse: .....
```



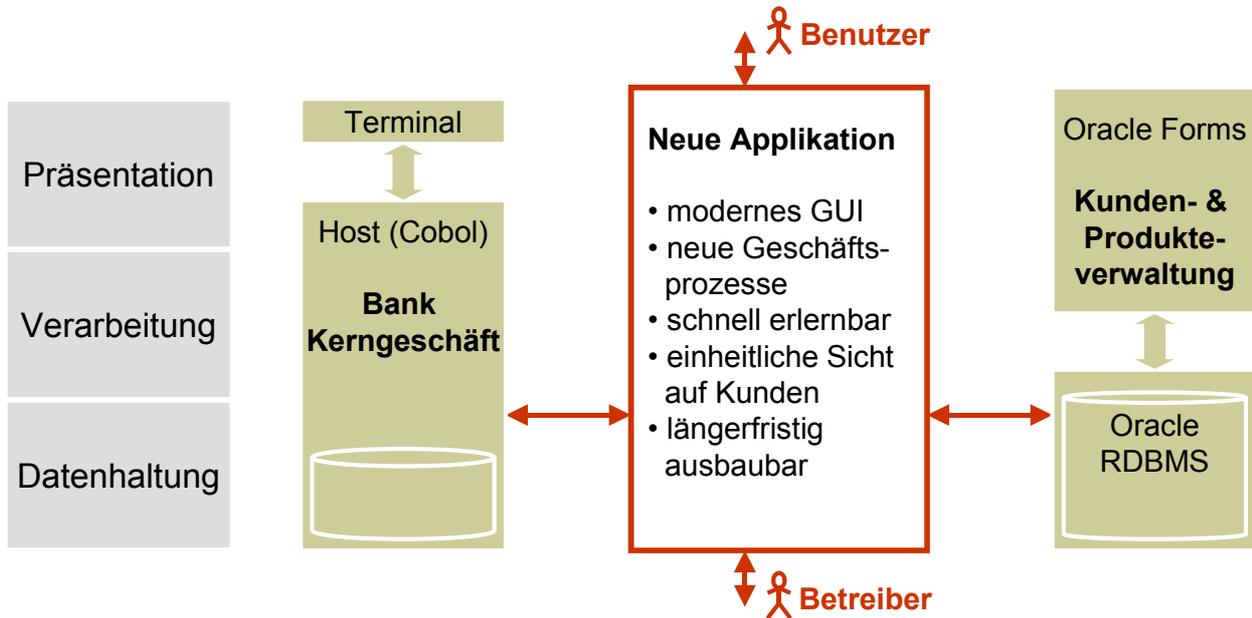
Besonderes

- Mehrere 1000 Benutzer arbeiten gleichzeitig
- In einem zentralen Rechenzentrum wird Betrieb vieler Banken abgewickelt (Mandantenfähigkeit)



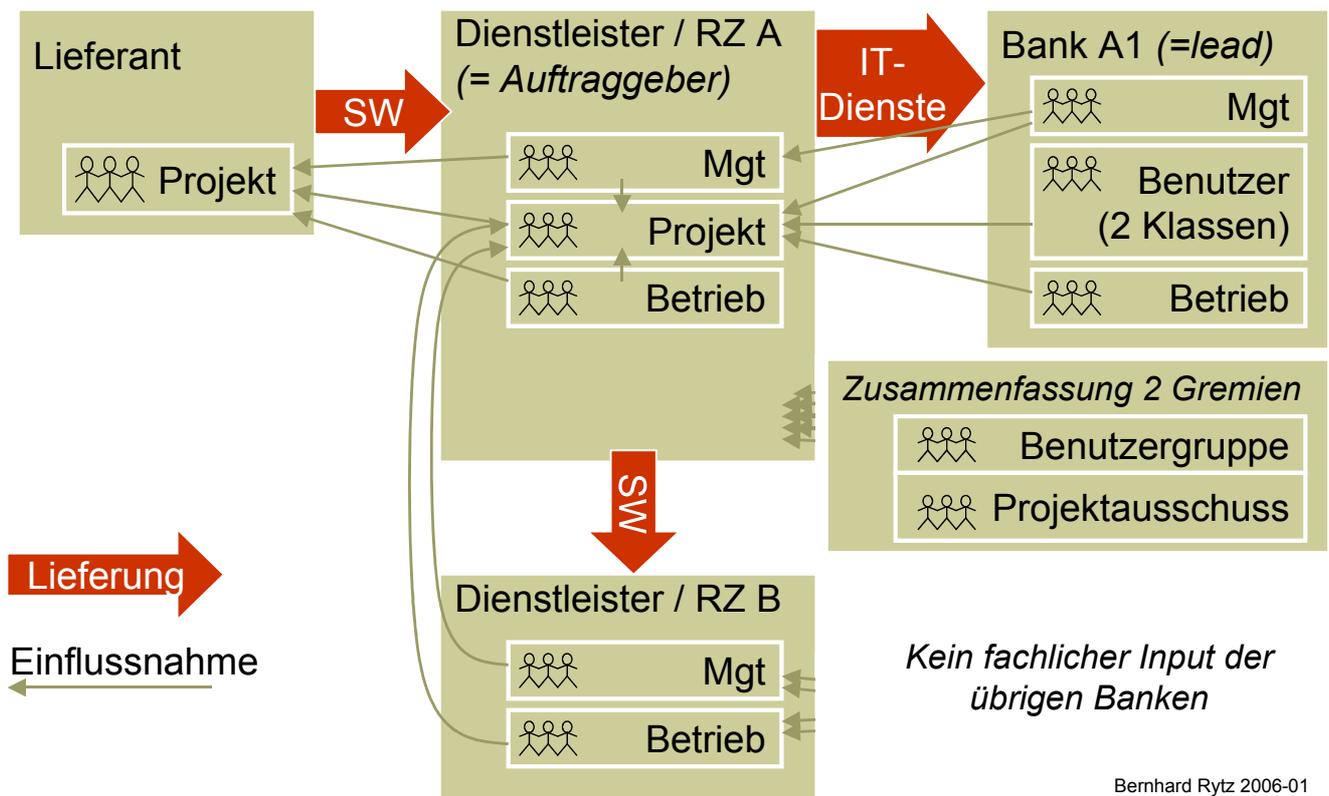
Ausgangslage (2) - Systemkontext

Eine zentrale Aufgabe ist die Integration mit den existierenden ("legacy") Systemen.



Anforderungsanalyse

Wer ist betroffen ?



11

Bernhard Rytz 2006-01

Vorgehen/Mittel für Anforderungen

- Vereinfachung der komplexen Situation (vorhergehende Grafik)
 - eine "Leadbank" verantwortlich für fachliche Aspekte
 - "Benutzergruppe": Repräsentanten aller Banken prüfen Anforderungen
 - "Projektausschuss" : Repräsentanten des Managements für Entscheide
 - Zweites Rechenzentrum (RZ B) liefert nur betriebliche Anforderungen
- Kommunikation mit den Benutzer(-vertretern)
 - Visions-Dokument
 - GUI Prototyping (throw-away), usability tests
 - regelmässige Besprechung der Anforderungs-Dokumente
- Anforderungsdokument (Word+Visio, vorgegebene Struktur)
 - Use cases (v.a. detaillierte Abläufe in Textform),
 - natürlichsprachliche Anforderungen, ergänzt mit Tabellen, Navigationsdiagrammen und (selten) Klassen-, Sequenz- oder Zustandsdiagrammen
 - nicht-funktionale Anforderungen (Mengen und Häufigkeiten, Sicherheit, Integrität, Performance, Verfügbarkeit, Testbarkeit, ...)

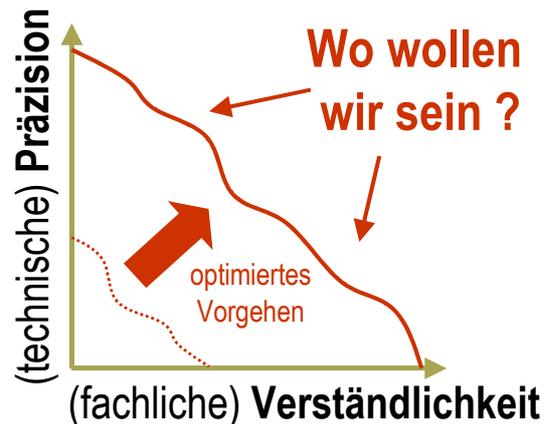
12

Bernhard Rytz 2006-01

Risikomanagement

⇒ Gewichtung Anforderungsanalyse im Projekteffort

- Vorgehen muss auf Projekt zugeschnitten sein
 - Risikotoleranz
 - Fachliche Vorkenntnisse Entwickler
 - Technische Vorkenntnisse Kunde
 - Komplexität der Materie
 - Änderungshäufigkeit
 - erwartete Lebensdauer Applikation
- Adäquate Mitte zwischen Präzision und Verständlichkeit suchen
- Anforderungen sind Mittel zum Zweck



Bernhard Rytz 2006-01

13

Beurteilung Anforderungsprozess (1)

- Was hat gut funktioniert ?
 - Fachliche Bedürfnisse genügend richtig und vollständig erkannt
 - Neues GUI wird als grosser Fortschritt empfunden
 - Effizienzsteigerung erreicht
- Verbesserungswürdig...
 - Wenig Verständnis der Benutzer für OO Formalismen
 - Performance-Anforderungen nicht genügend erhoben
 - Anforderungen spezifiziert ohne Machbarkeit zu berücksichtigen (Performance)
 - Schnittstellen externer Systeme nicht präzise genug spezifiziert → Probleme bei Datenqualität, Performance und Integrität

14

Bernhard Rytz 2006-01

Beurteilung Anforderungsprozess (2)

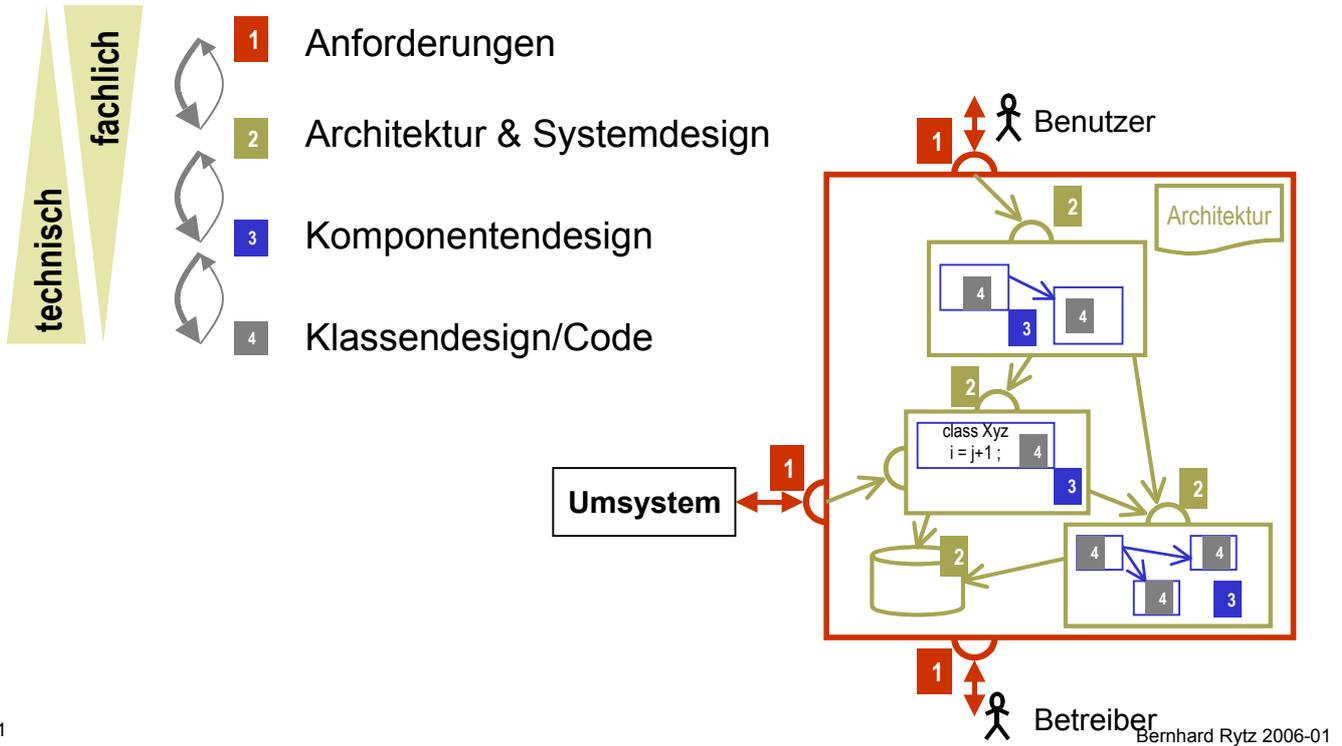
- Dauernde Herausforderung
 - Anforderungsspezifikation genug **verständlich** für Auftraggeber und genug **präzise** für Entwickler ?

➔ Diagramme mit Text kombinieren

- Gedankliche **Trennung** Anforderungen und Design

Vorgehen Architektur & Design

Überblick Ablauf



21

Was ist Architektur ?

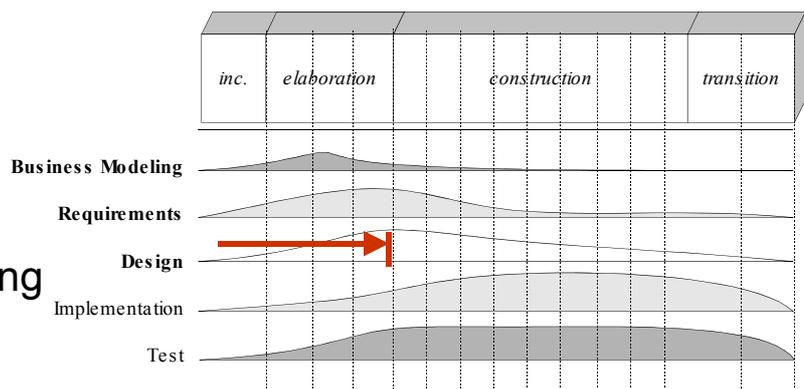
Architektur =

Richtlinien und Empfehlungen betreffend

- Strukturierungs- und Kommunikationsprinzipien ("patterns")
- konkrete Grobstruktur des Systems (logisch und physisch)
- Behandlung transversaler nicht-funktionaler Aspekte wie Sicherheit, Performance, Datenintegrität, ...

Was sollte in der Architektur sein ?

- Designpunkte, für welche eine falsche oder fehlende systemweite Regelung besonders weh tut
- projekt-spezifisch....

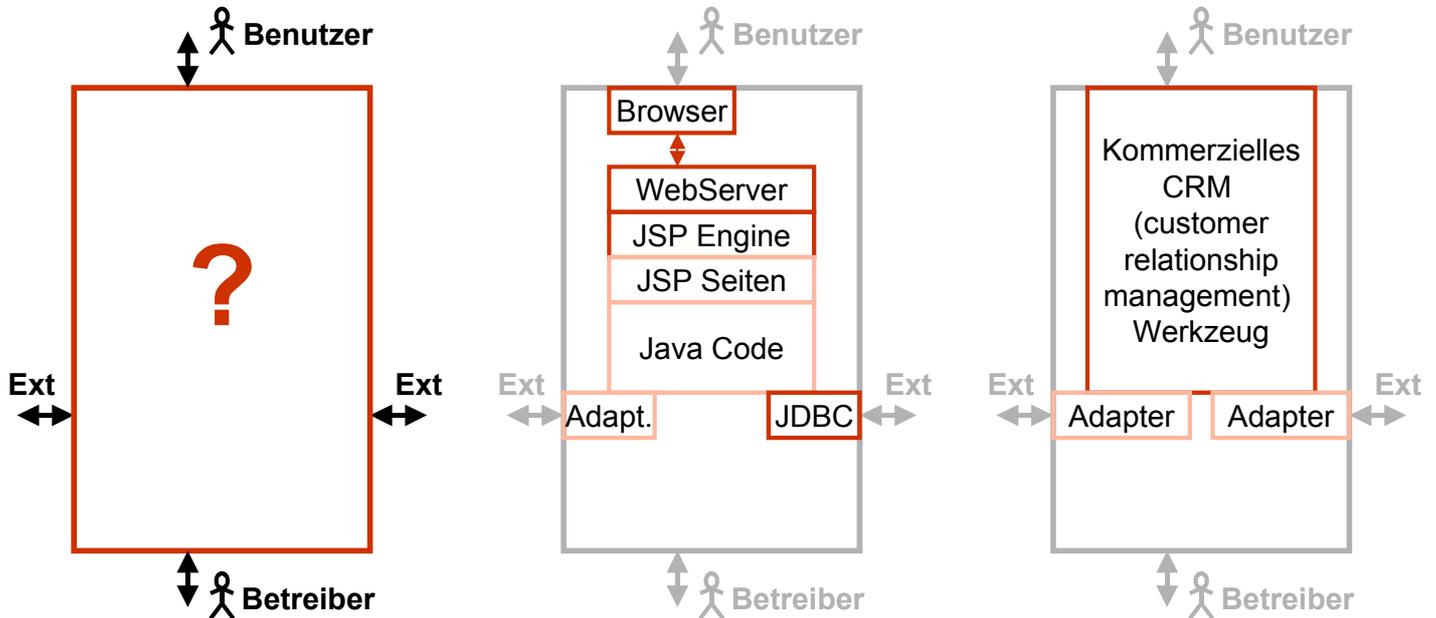


22

Bernhard Rytz 2006-01

Architektur-Alternativen

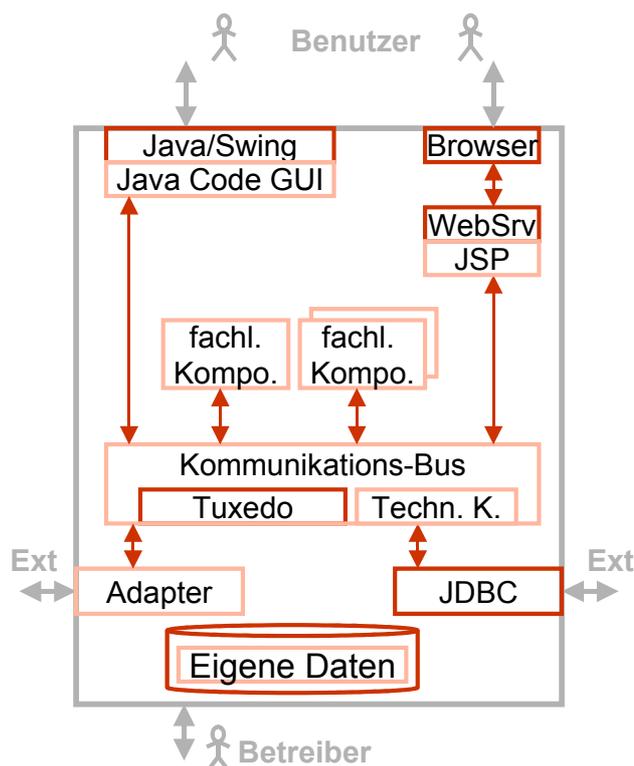
Es gibt immer mehr als eine Möglichkeit....



23

Bernhard Rytz 2006-01

Gewählte Architektur Prinzipien



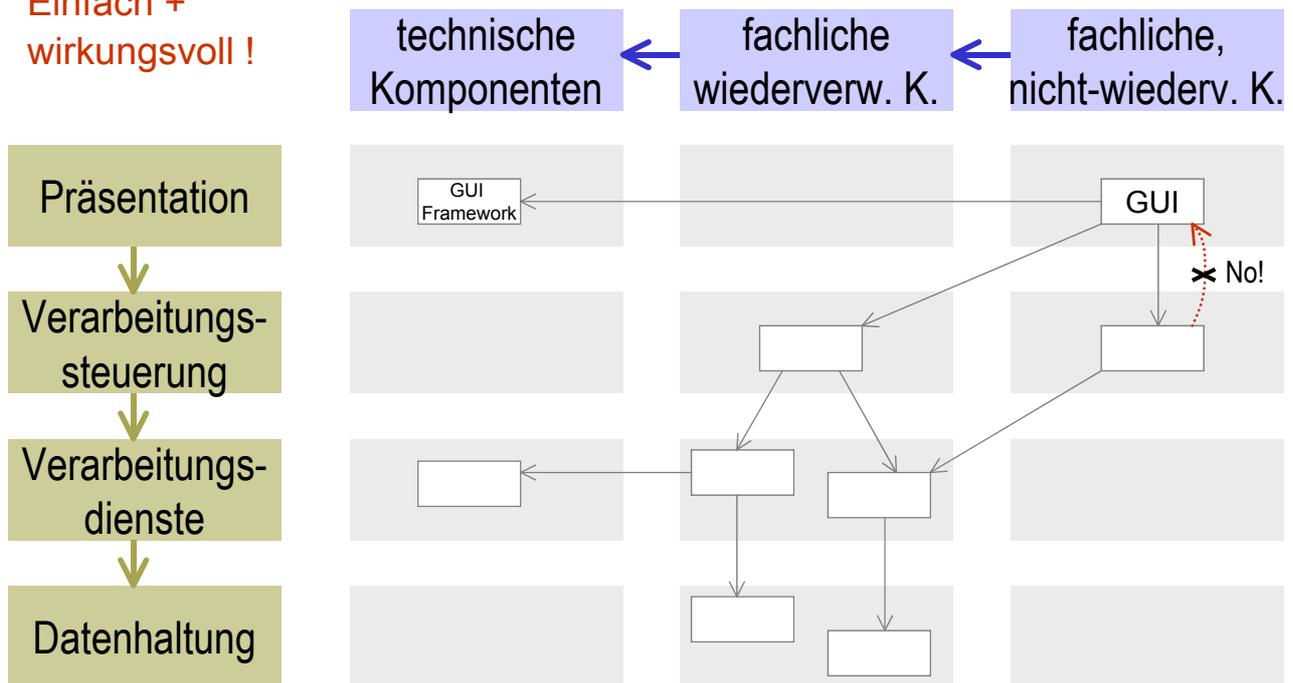
24

Bernhard Rytz 2006-01

- Multi-Channel Frontend
 - Swing GUI
 - Browser (als Versuch)
- Aufteilung fachliche Funktionalität in wiederverwendbare Komponenten
- Kommunikation der Komponenten untereinander und mit Frontend über Kommunikations-Bus (Tuxedo-basierend)
- Kommunikation mit Host über speziellen "Connector"
- Kommunikation mit Oracle-Applikation über JDBC
- eigene Datenhaltung zur "Synchronisation" der beiden externen Systeme

Strukturierungshilfe Schichten-Modelle ("layering")

Einfach +
wirkungsvoll !



25

Bernhard Rytz 2006-01

Dokumentationshilfe: Sichten ("views")

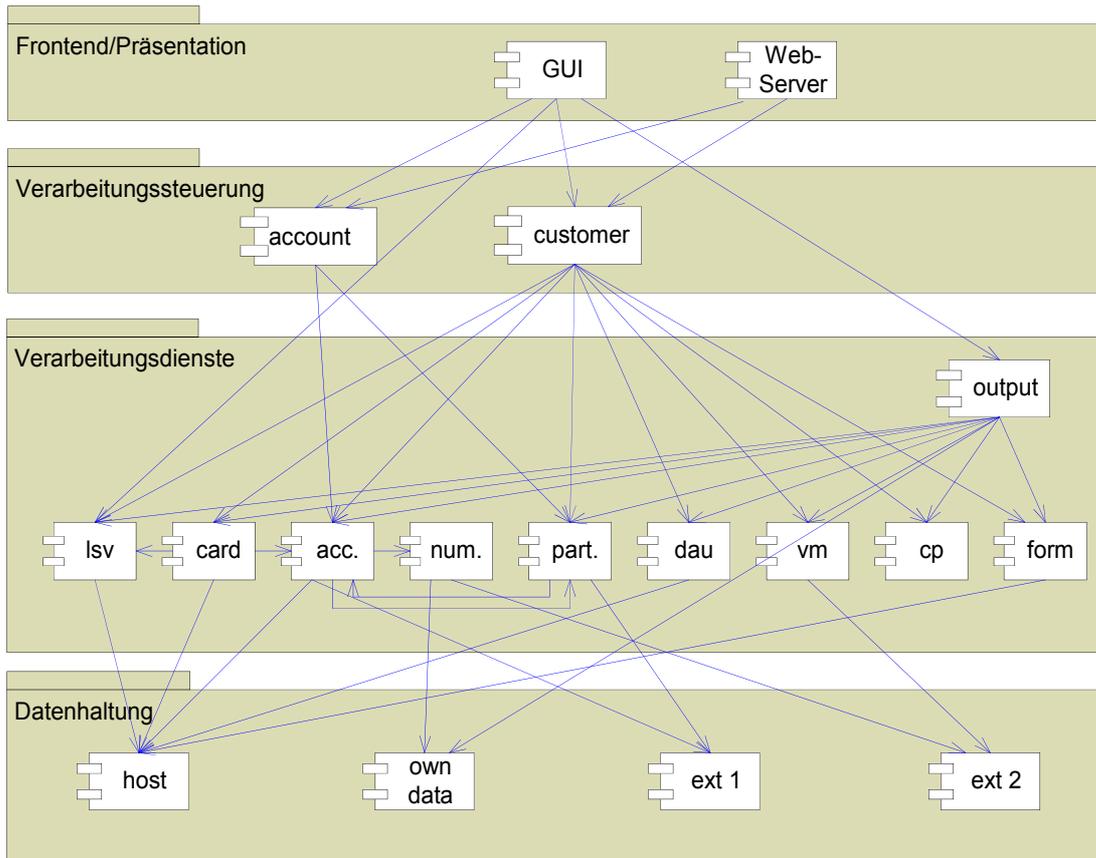
Unser Modell für Architektur-Sichten

- **Hardware view:** everything you can touch: all the machines (including operating systems) plus the networks
- **Runtime view:** the software parts, which you or the operator can "see" while the system is running: operating system processes, DB schemas, components such as EJB's, .NET assemblies
- **Logical view:** how the software is decomposed to help us develop it efficiently. Decomposition into layers, classes, packages etc.
- **Data view:** business entities, DB schema
- **Code view:** how the source files (configuration items) are organized, how we get from source files to executable code.

26

Bernhard Rytz 2006-01

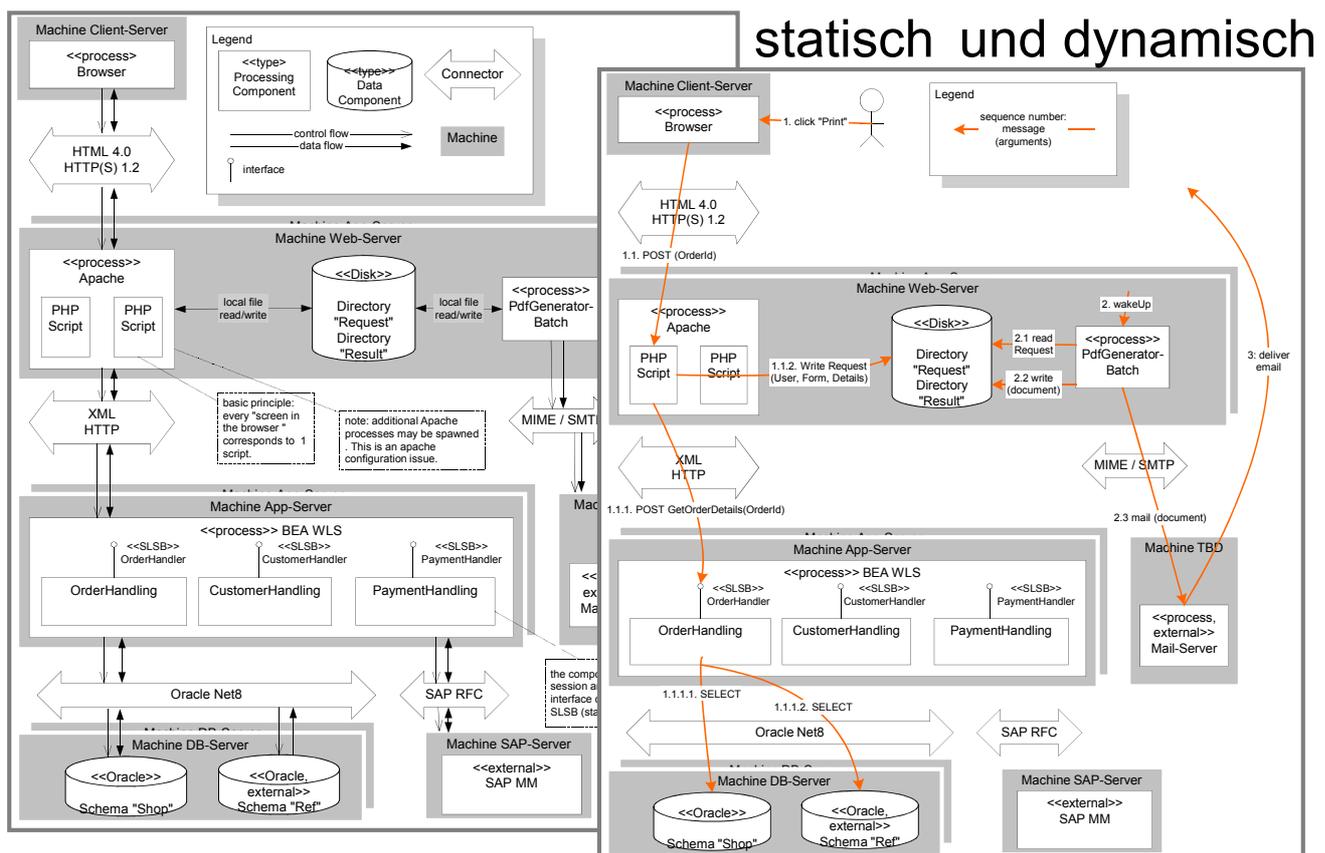
Architektur: Konzeptionelle Sicht (logical view)



27

d Rytz 2006-01

Architektur: Laufzeitsicht (runtime view)



28

Beurteilung Architektur

- Was hat gut funktioniert ?
 - Komponentenmodell
 - Wiederverwendung in neuen Projekten schon Tatsache
 - Skalierbarkeit gut

- Wo gab es anfänglich Probleme ?
 - Konfigurationsmanagement (neue Abhängigkeiten mit Umsystemen)
 - Antwortzeiten (Performance-Bugs, zu "naives" Design)
 - Betriebsaufnahme (ein verteiltes System wird anders betrieben als ein zentraler Host)

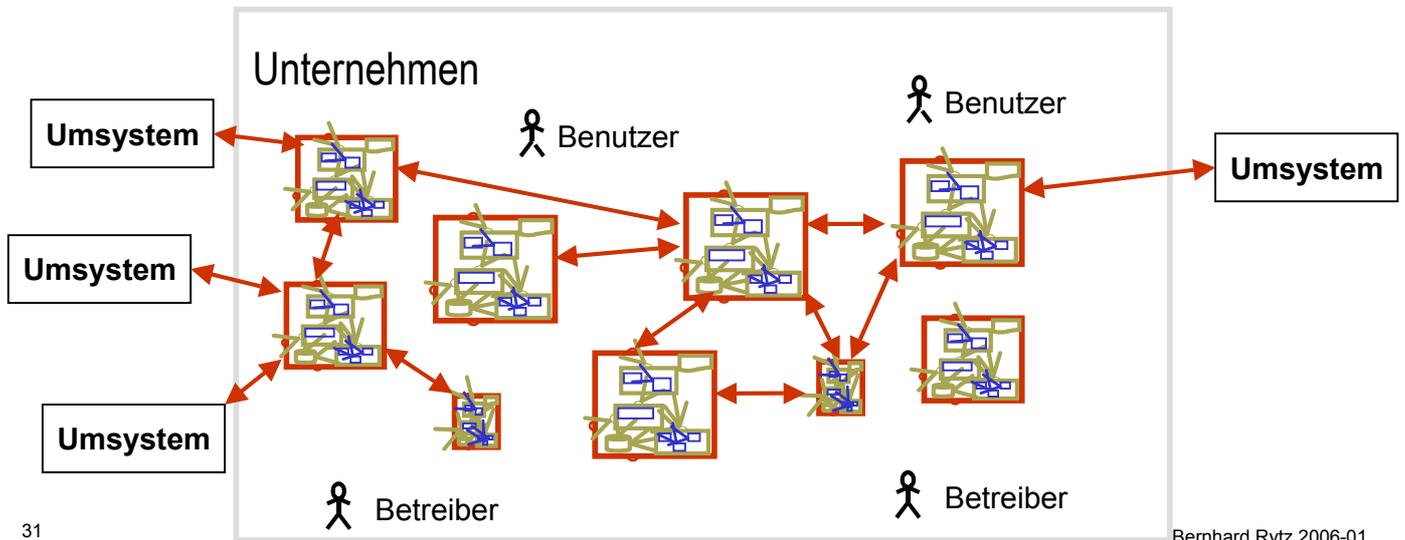
- Dauernde Herausforderung
 - "so einfach wie möglich, so kompliziert wie nötig"

Unternehmensarchitektur

Eine Applikation kommt selten allein...

Die Gesamtheit der Applikationen eines Unternehmens bildet eine „Superapplikation“.

- Welches sind die Anforderungen der „Superapplikation“?
- Wie sieht die Architektur aus ?

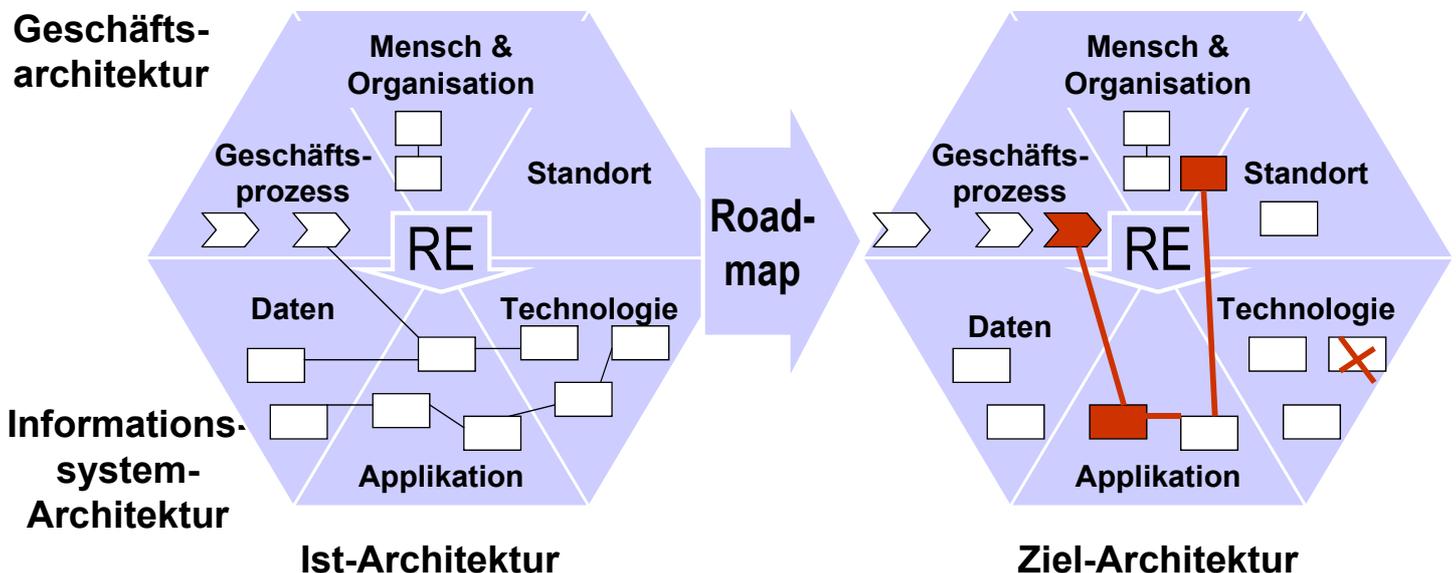


31

Bernhard Rytz 2006-01

Steuerung durch Unternehmensarchitektur (Enterprise Architecture EA)

Erfassung der Ist-Architektur, Definition Ziel-Architektur,
Umsetzung Roadmap



32

Bernhard Rytz 2006-01

Abschluss

Bernhard Rytz 2006 / 01

Zusammenfassung

- Ohne "genügende" Anforderungen sind Enttäuschungen vorprogrammiert.
 - Alle potentiellen Interessenten (stakeholder) kennen, dazu gehören auch die Betreiber
 - Welches Problem versucht das System zu lösen? In welchem Kontext?
 - Richtiger Kompromiss zwischen Präzision und Verständlichkeit (und Aufwand) für Anforderungen - Stakeholder und Entwickler müssen sich treffen
- Die globale Architektur eines verteilten Systems muss (vor allem) die nicht-funktionalen Aspekte berücksichtigen! Verschiedene Architektursichten mit Bezug zu den Anforderungen helfen.

Danke für Ihre Aufmerksamkeit

Bernhard Rytz 2006 / 01

Literatur + URLs

- Applying UML and Patterns, C. Larman, 2002
- *Applied Software Architecture, C. Hofmeister et al., 2000
- Component Software, C. Szyperski, 1998
- Enterprise SOA, Dirk Krafzig, 2005
- *Pattern-orientierte Software-Architektur, F. Buschmann et al., 2000
- Patterns of Enterprise Application Architecture, Martin Fowler, 2003
- Software Architecture in Practice, L. Bass et al., 1998
- Software Project Survival Guide, McConnell, 1998
- Software Requirements, K. Wiegers, 1999
- UML Components, J. Cheeseman et al., 2001

- www.volere.co.uk
- www.theserverside.com
- www.ibm.com/developerworks und [.../redbooks](#)