

Rich Internet Applications

Leif Hartmann
INF-M3 - Seminar/Ringvorlesung - Wintersemester 2007/2008
07. Dezember 2007

Inhalt

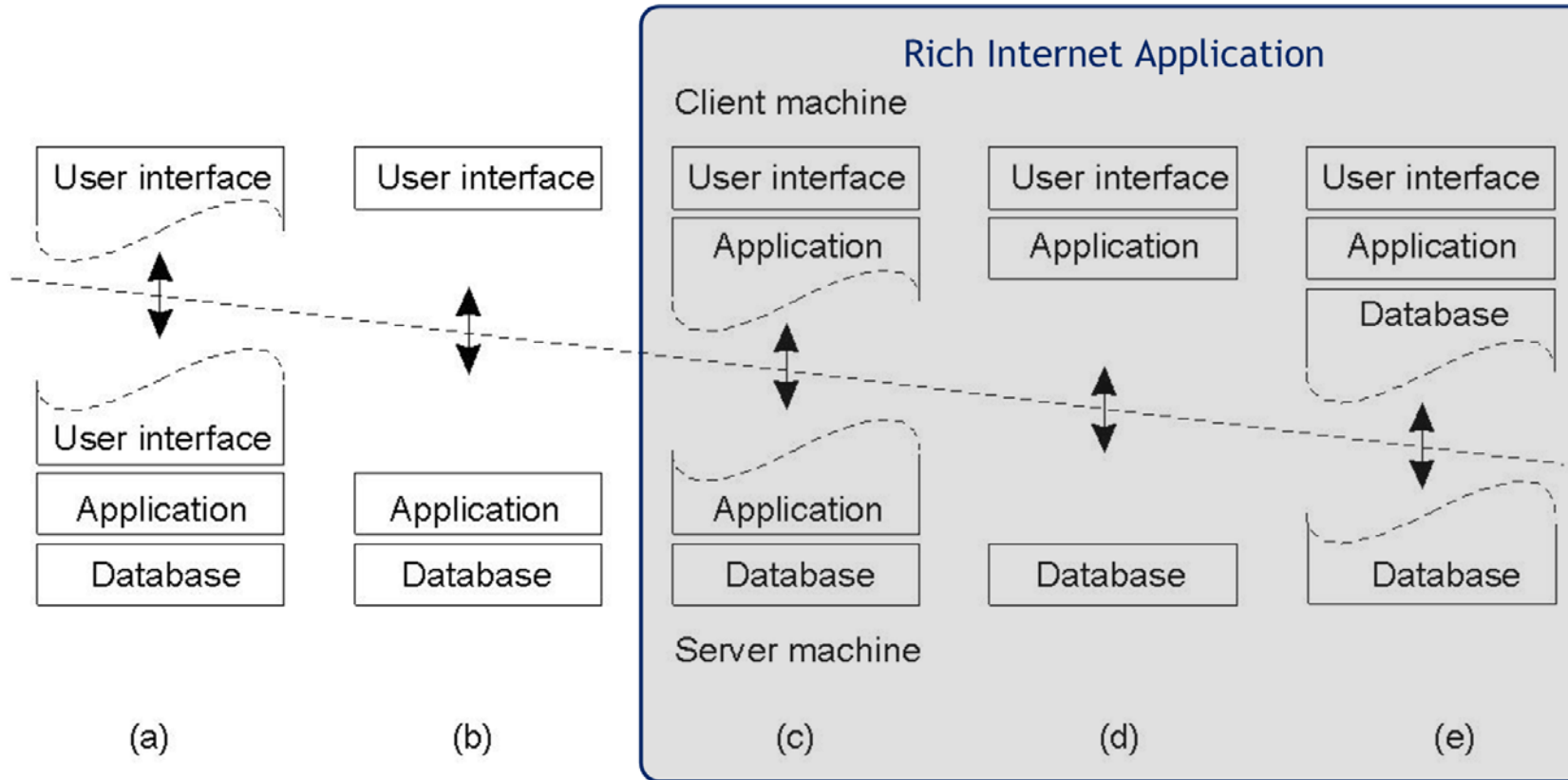
- ▶ Einleitung
 - ▶ Problemstellungen
 - ▶ Daten
 - ▶ Anwendungslogik
 - ▶ Präsentation
 - ▶ Kommunikation
 - ▶ Sicherheit
 - ▶ Testen
 - ▶ Anforderungen
 - ▶ Fazit / Risiken
-

Was sind Rich Internet Applications?

„Macromedia defines RIAs as combining the best user interface functionality of desktop software applications with the broad reach and low-cost deployment of Web applications and the best of interactive, multimedia communication. The end result: an application providing a more intuitive, responsive, and effective user experience.“ [3]



Kompetenzverteilung



Nach [9]

Vorteile ...

gegenüber klassischen Web-Anwendungen

- ▶ Flüssigere Bedienung
 - ▶ Schnellere Antwortzeiten
 - ▶ Erweiterte GUI
- ▶ Weniger Netzwerkverkehr
- ▶ Bekannte Benutzerschnittstelle
 - ▶ Angelehnt an Desktop-Anwendungen
- ▶ Mehr Daten und Logik auf Clientseite nutzbar
- ▶ Online und offline benutzbar

Vorteile ...

gegenüber Desktop-Anwendungen

- ▶ Geringer Administrationsaufwand
 - ▶ (Fast) kein Installationsaufwand auf Clientseite
- ▶ Plattformübergreifend
- ▶ Einfacher Zugriff

Problemstellungen

- ▶ Komplexität
- ▶ Kompatibilitätsprobleme
- ▶ Accessibility
- ▶ Modellierung
- ▶ Entwicklungstools (CASE-Tools, IDEs, Code-Generatoren)
- ▶ Testen

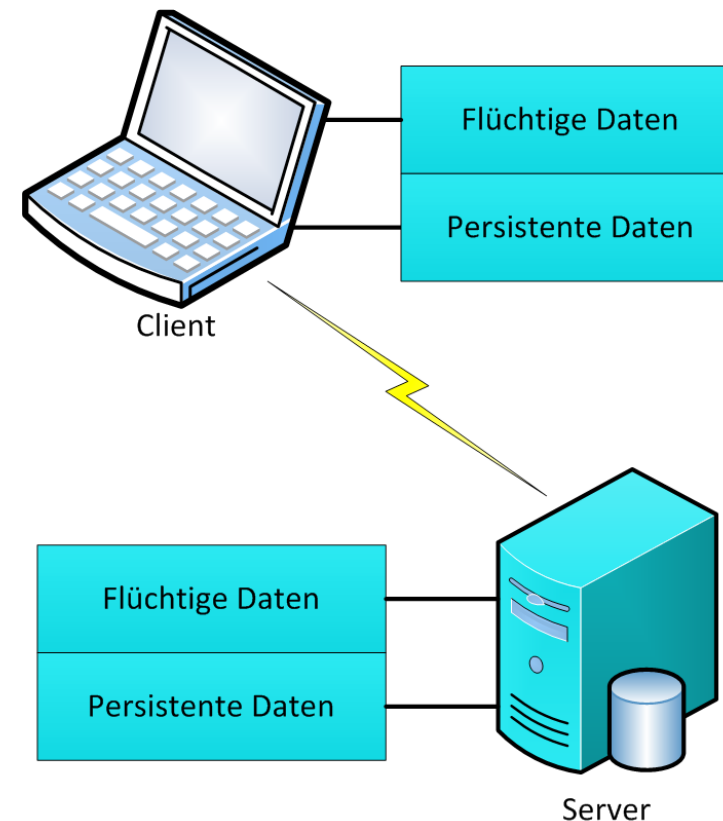
Daten

► Klassisch:

- Persistente Daten nur auf Serverseite
 - Ausnahme: Cookies
- Nur Formulardaten werden übertragen

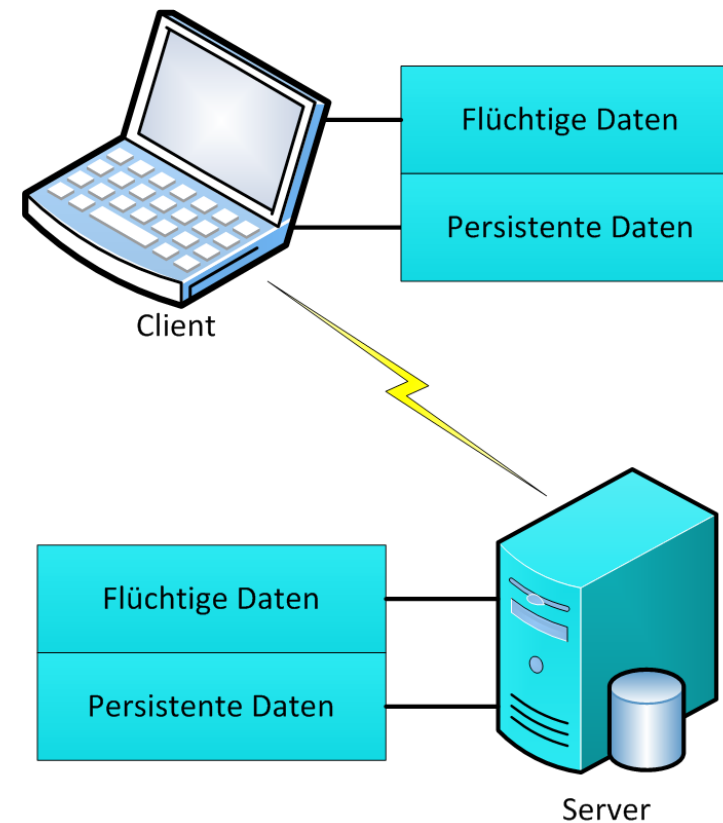
► RIA:

- Flüchtige und persistente Daten auf Client und Server



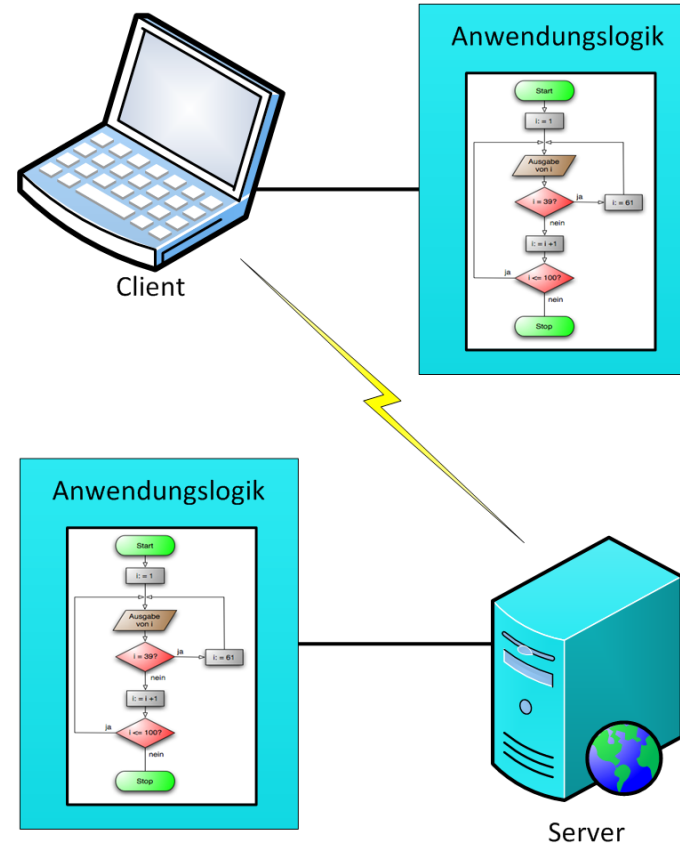
Daten

- ▶ Datenobjekte serialisieren und übertragen
- ▶ Datenintegrität
- ▶ Validierung
- ▶ Modellierung:
 - ▶ Persistenzebenen mit einbeziehen



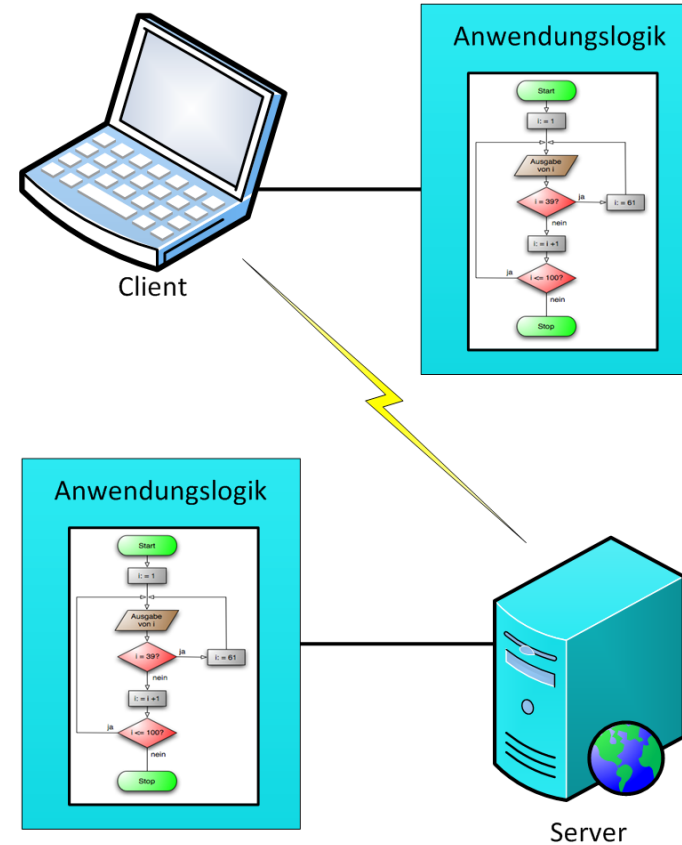
Anwendungslogik

- ▶ Klassisch:
 - ▶ Nur auf Serverseite
- ▶ RIA:
 - ▶ Clientseitig
 - ▶ Serverseitig
 - ▶ „Gemischt“



Anwendungslogik

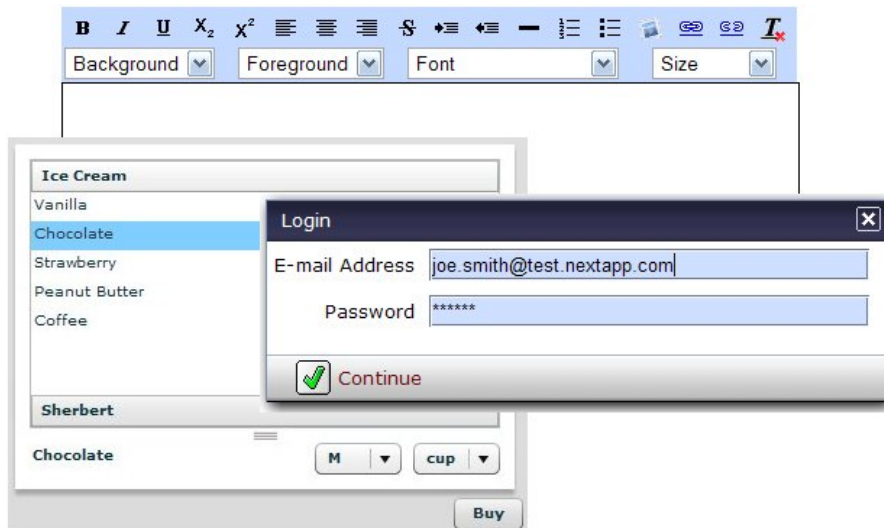
- ▶ Nicht zu viel Clientlogik
 - ▶ Kein Zugriff auf Serverdaten
- ▶ Modellierung:
 - ▶ Verteilungsunabhängige Sicht
 - ▶ Verteilungsabhängige Sicht



Präsentation

► Klassisch:

- HTML, einfache Bedienelemente
- Viele Page-Refreshes

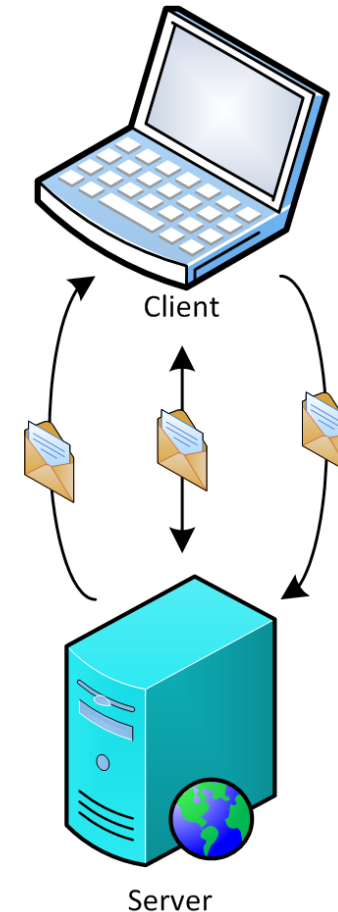


► RIA:

- Vielfältige Bedienelemente
- Erweitertes Benutzerinteraktion
 - Drag and Drop
 - Animationen
- Unterschiedliche Funktionalitäten des Clients
 - Bildschirmgröße
 - Multimedia-Unterstützung

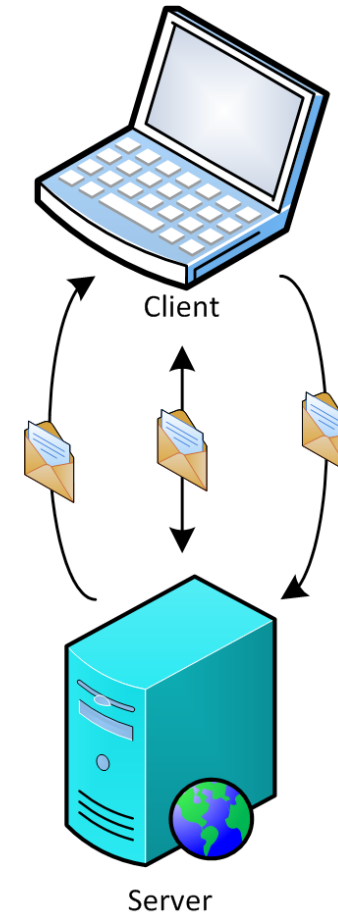
Kommunikation

- ▶ Klassisch:
 - ▶ Synchron
 - ▶ Serverantworten in Form von kompletten Seiten
- ▶ RIA:
 - ▶ Synchron oder asynchron
 - ▶ I.d.R. keine kompletten Seiten als Serverantwort
 - ▶ Geringere Datenübertragung
 - ▶ Pulling, Pushing möglich



Kommunikation

- ▶ Offline-Funktionalität
- ▶ Verschiedene Kommunikationsmodelle:
 - ▶ Message-based
 - ▶ State replication
 - ▶ Method replay
- ▶ Clients werden „von außen“ benachrichtigt
- ▶ GUI und Logik kann komplett beim Start oder zur Laufzeit (nach)geladen werden



Testen

Beispiel: WebTest

```
...  
<webtest name="Test DWR 2.0 RC3 demo App">  
  <steps>  
    <invoke url="http://localhost:8080/dwr/simpletext"/>  
    <verifyXPath xpath="//*[@id= 'demoReply']" text=""/>  
    <clickButton label="Send"/>  
    <sleep seconds="2"  
      description="Wait for completion of async call"/>  
    <verifyXPath xpath="//*[@id= 'demoReply']" text="Hello, Joe"/>  
  </steps>  
</webtest>  
...
```

Testen

Beispiel: WebTest

#	Result	Name	Parameter
1	✓	Invoke Resulting page	url http://localhost:8080/dwr/simpletext
2	✓	verifyXPath text	xpath //*[@id= 'demoReply']
3	✓	clickButton	label Send
4	✓	sleep Wait for completion of async call	seconds 2
5	✓	verifyXPath text	Hello, Joe xpath //*[@id= 'demoReply']

Sicherheit

- ▶ Wenig Unterschiede zu klassischen Webanwendungen
 - ▶ Kommunikation, Authentifizierung etc.
- ▶ Daten und Anwendungslogik
 - ▶ Client darf nicht auf alle Daten Zugriff haben
- ▶ Client und Server müssen „Aufräum-Mechanismen“ zur Verfügung stellen
 - ▶ Nicht nur Cookies

Anforderungen

- ▶ Modellierung
 - ▶ Sowohl Client- als auch Serverseite
 - ▶ Technologie unabhängig
 - ▶ Code-Generierung
 - ▶ Entwicklungswerkzeuge
 - ▶ CASE-Tool
 - ▶ IDE
 - ▶ Wartbarer Code (Refactoring)
 - ▶ Testmöglichkeiten
 - ▶ GUI-Beschreibungssprache
-

Anforderungen

- ▶ Offline-Funktionalität
 - ▶ Anwendungscode offline
 - ▶ Persistente Daten auf Clientseite
- ▶ Robuste Clientplattform (Kompatibilität)
- ▶ Möglichst hohe Verbreitung der Clientplattform

Fazit

Mögliche Thematik für Masterarbeit:

- ▶ Was ändert sich für Entwickler im Hinblick auf die vorgestellten Problemstellungen?
- ▶ Untersuchung der Problemstellung anhand einer mittelgroßen Anwendung

Risiken

- ▶ Viele Technologien
- ▶ Viele Methodiken und Modellierungssprachen
- ▶ Kaum Erfahrung mit großen Softwaresystemen

- ▶ Daher: Eingrenzung auf Schwerpunkte
 - ▶ Modellierung
 - ▶ Code-Generierung
 - ▶ Offline-Funktionalität
 - ▶ Sicherheit
 - ▶ ...

Quellen

- [1] J.C. Preciado, M. Linaje, F. Sánchez, S. Comai – *Necessity of methodologies to model Rich Internet Applications* – In Proceedings of the 2005 Seventh IEEE International Symposium on Web Site Evolution (WSE'05) – 2005
- [2] J.C Preciado, M. Linaje, S. Comai, F. Sánchez-Figueroa – *Designing Rich Internet Applications with Web Engineering Methodologies* – In 9th IEEE International Workshop on Web Site Evolution (WSE 2007) – 2007
- [3] J. Duhl – *White paper: Rich Internet Applications* – IDC, 2003
- [4] A. Bozzon, S. Comai, P. Fraternali, G. T. Carughi – *Conceptual modeling and code generation for rich internet applications* – Proceedings of the 6th international conference on Web engineering – ACM – 2006
- [5] A. Bozzon, S. Comai, P. Fraternali, G. T. Carughi – *Capturing RIA Concepts in a Web Modeling Language* – Proceedings of the 15th international conference on World Wide Web – ACM –2006

Quellen

- [6] J. Farell, G. S. Nezlek – *Rich Internet Applications: The Next Stage of Application Development* – In Proceedings of the ITI 2007 29 Int. Conf. on Information Technology Interfaces – 2007
- [7] A. Leff, J. Rayfield – *Programming Model Alternatives for Disconnected Business Applications* – In Internet Computing, IEEE – 2006
- [8] M. Guillemot, D. König – *Web Testing Made Easy* – OOPSLA '06: Companion to the 21st ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications – 2006
- [9] A. Tanenbaum, M. van Steen – *Verteilte Systeme* – Pearson Studium – 2003 – ISBN 3-8273-7057-4

Fragen?

Leif Hartmann
INF-M3 - Seminar/Ringvorlesung - Wintersemester 2007/2008
07. Dezember 2007