



Web Services und Recommender Systeme

Proseminar Semantic Web

UNI Kaiserslautern

Ralf Kretschmann



- **Web Services**
 - Motivation für (Semantic) Web Services
 - Grundlagen:
 - Zusammenspiel Beschreibungssprachen SOAP, WSDL, UDDI
 - SOAP
 - WSDL
 - UDDI
 - Realisierung
 - Zusammenfassung
- **Recommender Systeme**
 - Motivation für Recommender Systeme
 - Verschiedene Methoden
 - Content-Based Filtering
 - Collaborative Filtering
 - Beispiele
 - Zusammenfassung

- Web Dienste realisieren verteilte Anwendungen
- Nachteil „normaler“ Web Dienste:
 - unterschiedliche Standards
 - nicht von Rechnern auffindbar; **Mensch** muss eigenhändig Informationen bei verschiedenen Quellen suchen
 - „teile und herrsche“ durch den Menschen
 - Zusammenfügen der Informationen durch Menschen
- Beispiel Internet-Recherche
 - Aufruf der entsprechenden Webseite
 - Ausfüllen von Web Formularen (Entscheidungen selbst treffen)
 - Absenden der Formulare
 - Ergebnisse auswerten und zusammenfassen

- Web Services, die Techniken des Semantic Webs nutzen
 - **Für Programme** wiederverwendbare Dienste
 - **Selbstbeschreibende** Dienste
 - **Standardisierung**
 - **Automatische** Auswahl, Kombination, Interaktion, Kommunikation einzelner primitiver WS ohne dass Mensch die komplexen Zusammenhänge kennen muss
 - Komplexe WS als Komposition von komplexen oder primitiven WS durch *while, if – then – else, ..*
 - Automatische Erstellung und Ausführung der Anfrage
- Beispiel
 - Anfrage: „Organisiere mir Ferienreise nach Spanien!“

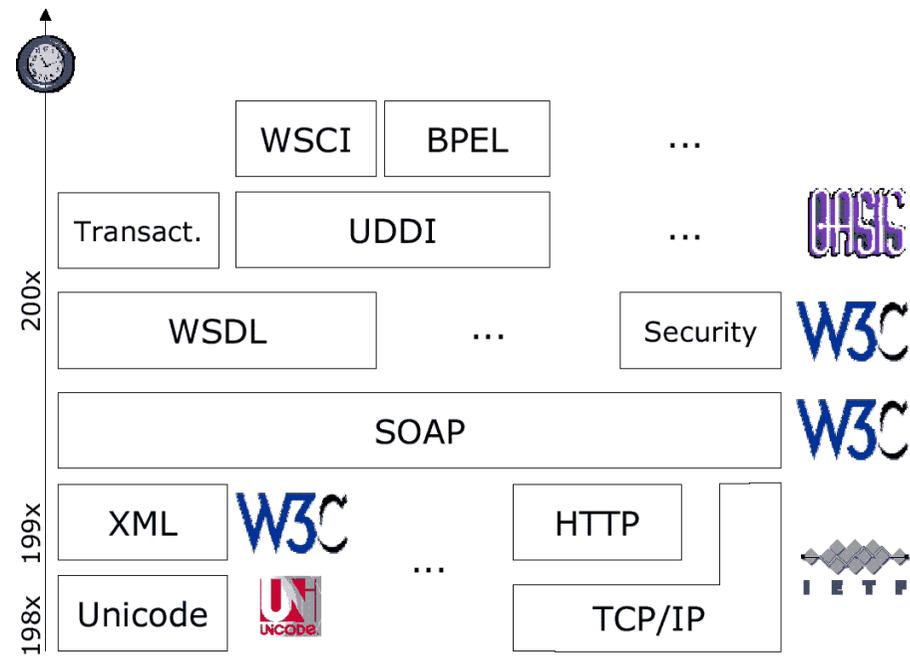
- **Inhalt** von Nachrichten: XML-Darstellung
- **Verpackung + Übertragung** des Inhalts: SOAP
- Beschreibung der **Schnittstelle** eines Dienstes: WSDL
- **Auffinden** der Dienste: UDDI



Schichtenmodell und zeitliche Entwicklung



Schichtenmodell



Zeitliche Entwicklung

Simple Object Access Protocol

- Entwicklung von Microsoft 1997 gestartet
- Aktuell in Verwendung: Version 1.2
- Nachrichtenaustausch über XML-Dokumente
- Transport meist über HTTP, auch SMTP, FTP
- Für fast alle Programmiersprachen verfügbar:
 - Apache SOAP für JAVA
 - Microsoft .NET
 - SOAP::Lite for Perl
- Plattform- und sprachübergreifende Kommunikation

Aufbau einer SOAP-Nachricht

```
<s:Envelope xmlns:s=http://www.w3.org/2001/06/soap-envelope>  
  <s:Header>  
    ....  
  </s:Header>  
  <s:Body>  
    ...  
  </s:Body>  
</s:Envelope>
```

- Namespace -> SOAP v 1.2 Nachricht
- Header kann entfallen
- Header und Body bestehen aus wohlgeformtem namespace qualifiziertem XML

Beispiel

Funktionsaufruf: public int quadriere(int zahl)

```
<n:quadriere xmlns:n="urn:Rechnen">  
  <zahl xsi:type="xsd:int">  
    2  
  </zahl>  
</n:quadriere>
```

Web Service Description Language

- Von IBM, Microsoft
- Beschreibt, wie Zugriff auf Web Server funktioniert
- Client holt sich Anweisungen über WSDL
- Client muss nicht wegen Änderungen am Server neu angepasst werden, sondern nur die WSDL-Beschreibung des Dienstes
- Nutzt XML-Schema
- Implementierungen in
 - Microsoft .NET
 - IBM Web Services Toolkit
 - Apache Axis Project

Aufbau einer WSDL-Beschreibung

Data Types <wsdl:types/>	Typdeklarationen
Messages <wsdl:message/>	Nachrichtendeklarationen (mit Parameternamen+Typ)
Interfaces <wsdl:portType/>	Nachrichten zu logischen Operationen zusammengefasst (Ein-/Ausgabe)
Services <wsdl:binding/> <wsdl:service/>	Binding: Verknüpfung logisch zu physischen Operationen +Encoding Service: Adresse(n) (Mirrors)

Universal Description, Discovery and Integration Protocol

- Von Ariba, IBM, Microsoft
- Verzeichnis über Web Dienste und wo sie spezifiziert sind
 - analog zu Gelbe Seiten; anstatt Telnummern Verweis auf WSDL-Beschreibung
- Zugriff auf UDDI-Verzeichnis selbst wieder über WSDL-Beschreibung
- Sowohl öffentliche als auch private Verzeichnisse möglich (Intranets)

UDDI Zugriff:

- Durch SOAP-Nachricht mit Body:

```
<find_business generic=„1.0“ xmlns=„urn:uddi-org:api“>  
  <name> NameDerFirma </name>  
</find_business>
```
- Antwort: XML-Nachricht mit Informationen über angebotene Dienste in Form von businessKeys und serviceKeys
- Keys heißen Universally Unique Identifiers (UUID)

UDDI Zugriff:

● Inquiry API

➤ Find things

- find_business
- find_service
- find_binding
- find_tModel

➤ Get Details

- get_businessDetail
- get_serviceDetail
- get_bindingDetail
- get_tModelDetail

● Publishers API

➤ Save things

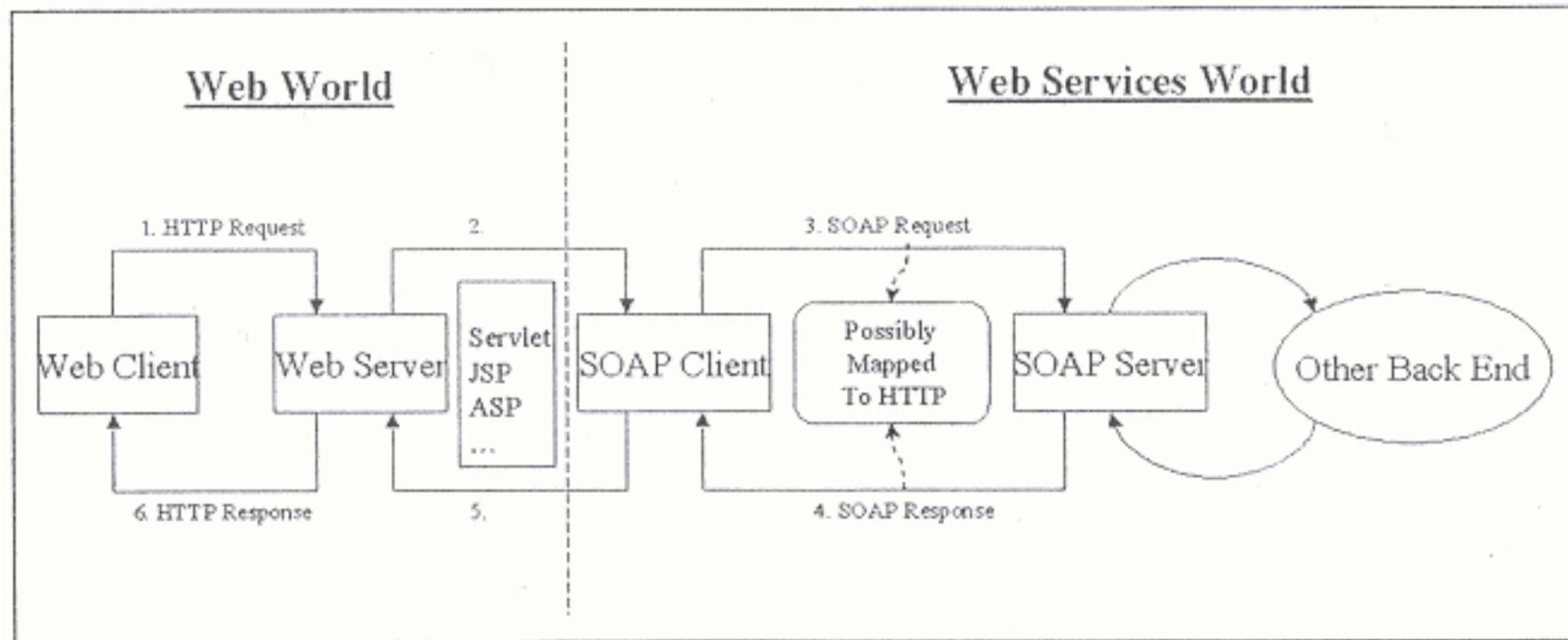
- save_business
- save_service
- save_binding
- save_tModel

➤ Delete things

- delete_business
- delete_service
- delete_binding
- delete_tModel

➤ security...

- get_authToken
- discard_authToken



Kommunikationsfluss zum Ausführen eines Web Services

Details: siehe „Visualizing the Semantic Web“

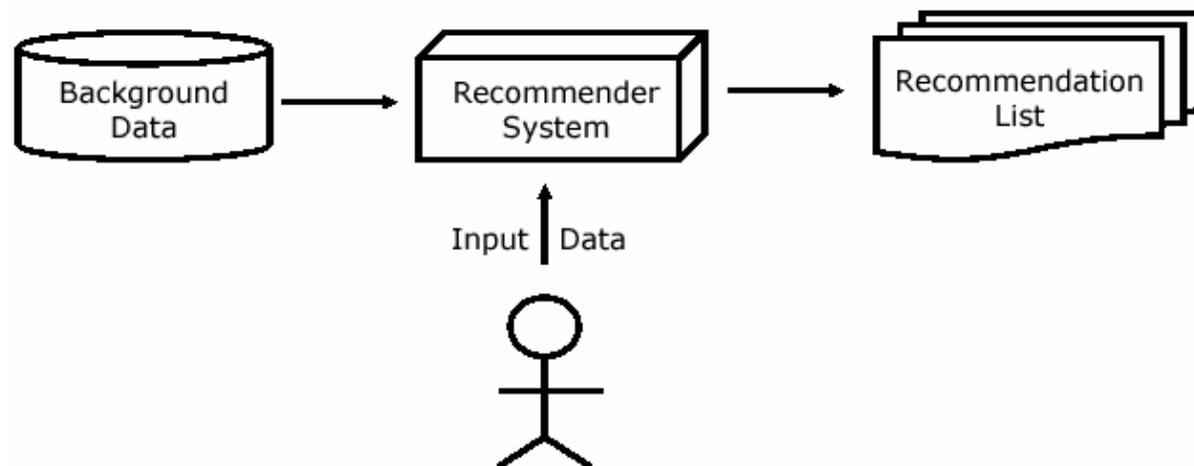
Ausblick

- Zukünftig breite Unterstützung der Protokolle von Web Services in Aussicht, auch durch Standardisierung durch W3C
- Ansätze zur Verstärkung der Sicherheit (WS-Security) und komplexere Ablaufbeschreibungen (Web Service Choreography Interface WSCI) in Entwicklung
- Letztendlich entscheidet Qualität der angebotenen Dienste über die Verbreitung von (Semantic) Web Services

- Problem: Internet Recherche ohne Fremde Hilfe kaum noch machbar
- Effizientere Nutzung der Informationsmenge des WWW notwendig
- Beispiel: Suchmaschinen können Informationen nicht personalisiert finden
- Einsatz sinnvoll, wenn Nutzerzahl und Informationsmenge hoch
- Einsatz Beispiele: Informations- / Nachrichtenportale, E-Commerce, E-Learning

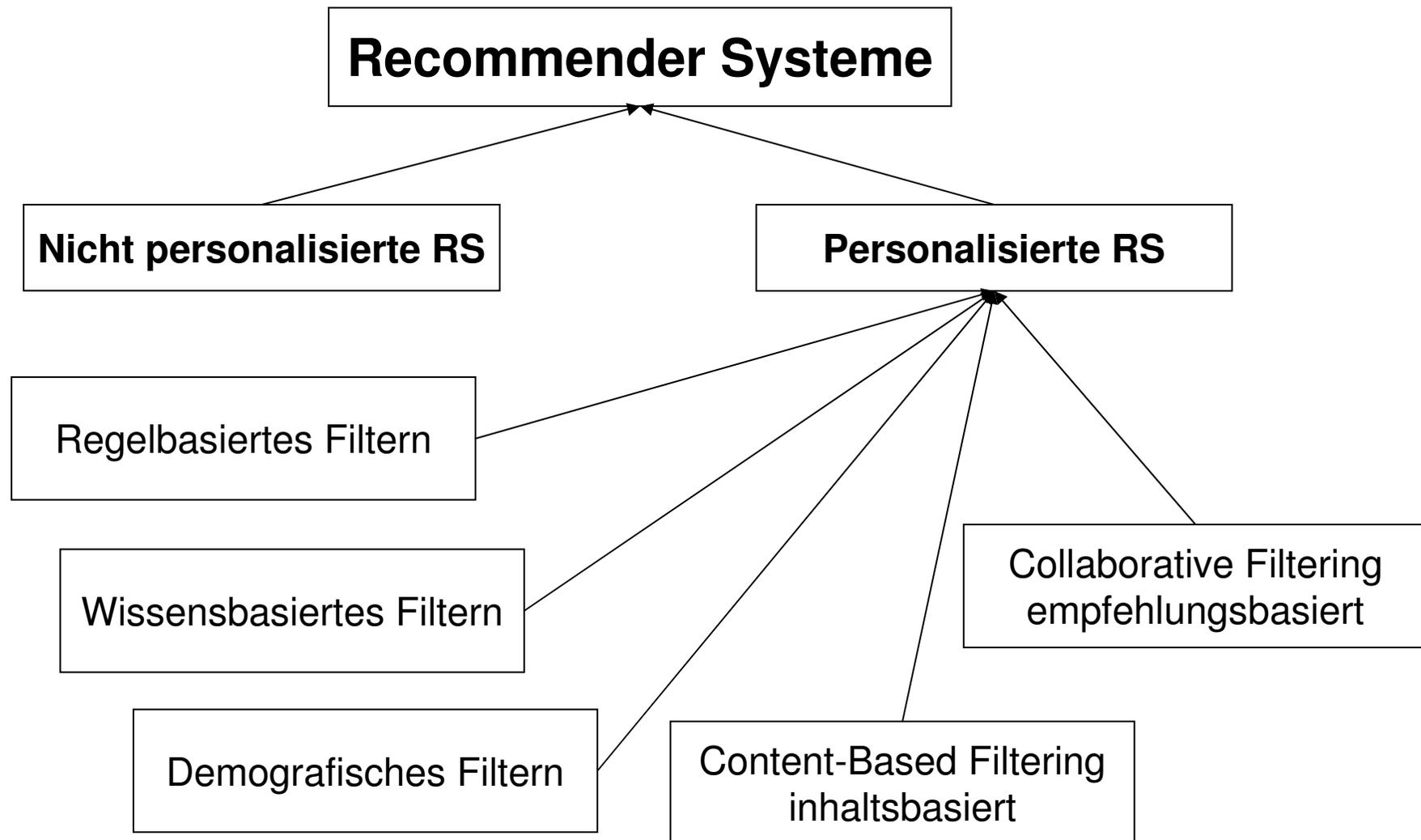
Recommender System

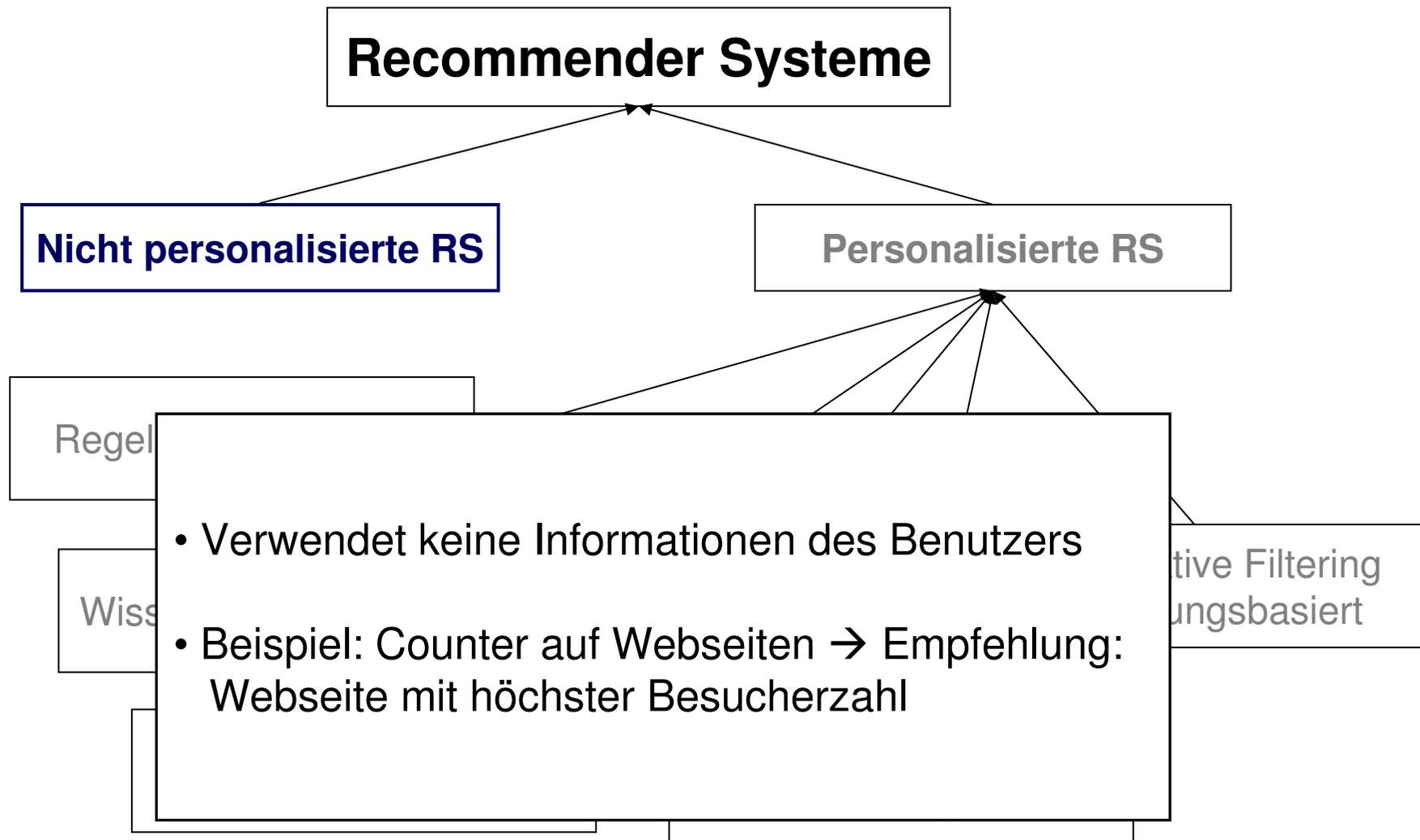
- Generiert automatisch Empfehlungen
- Schließt von vorhandenen Informationen auf neue Daten

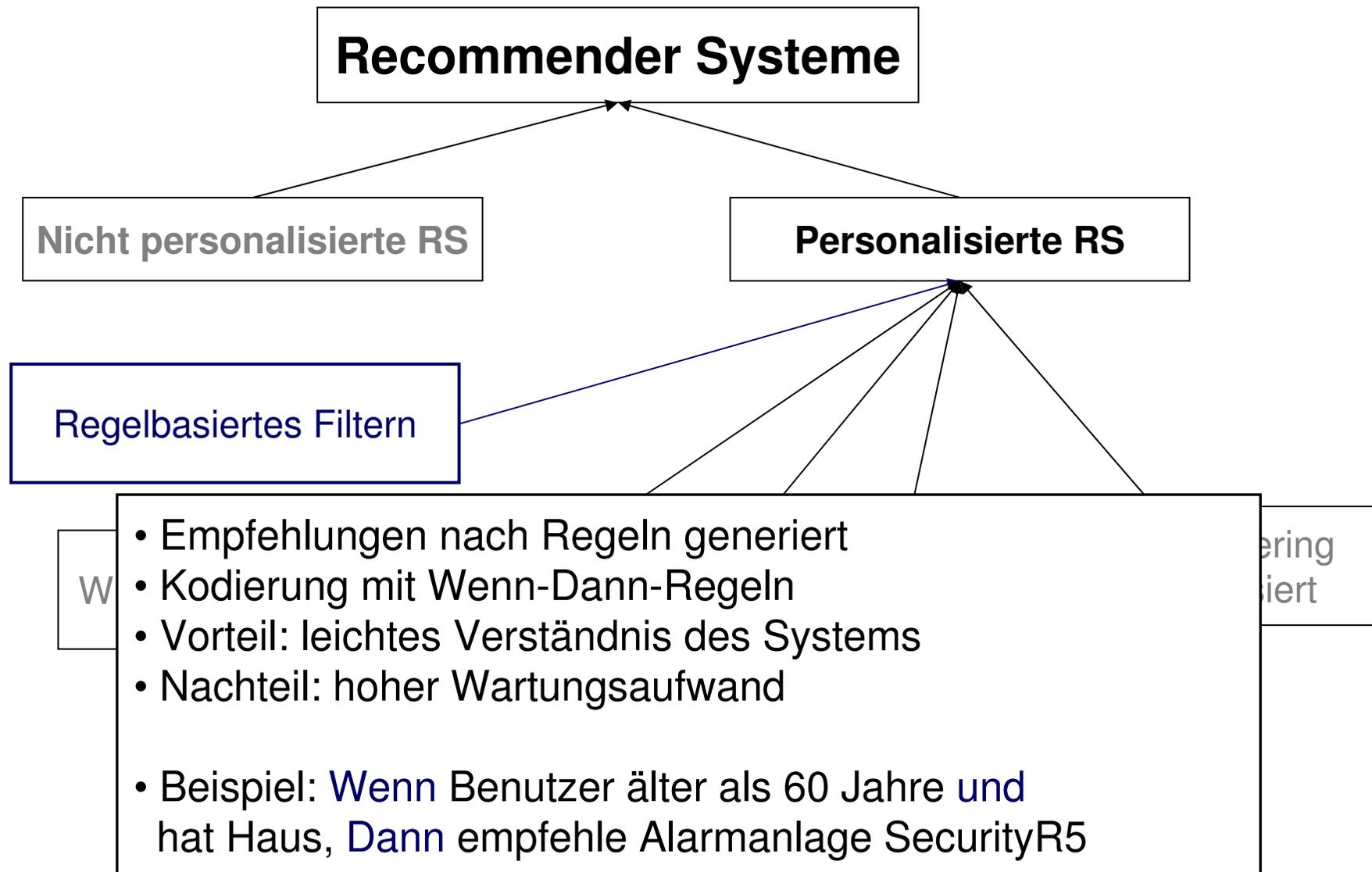


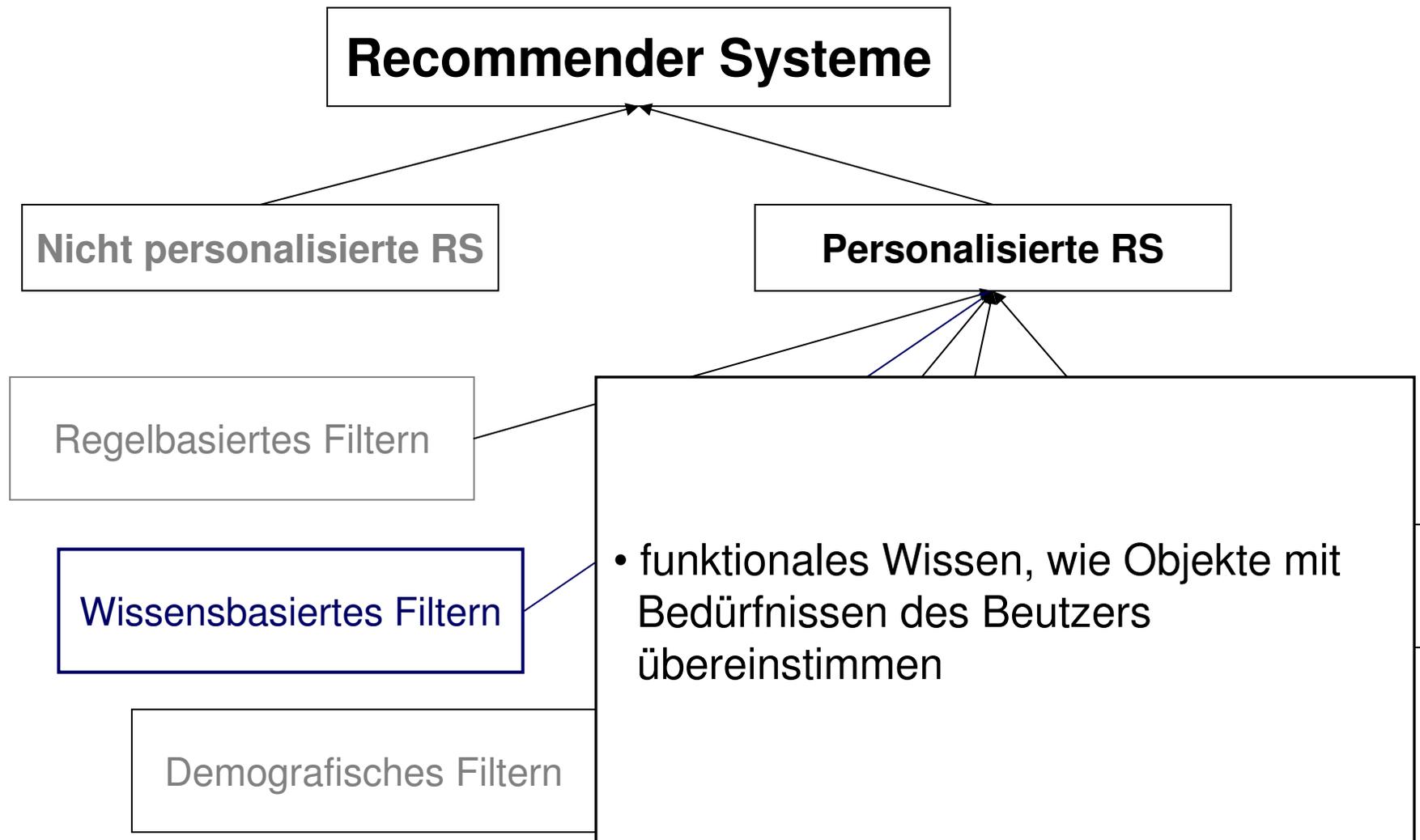
Geschichtliche Entwicklung

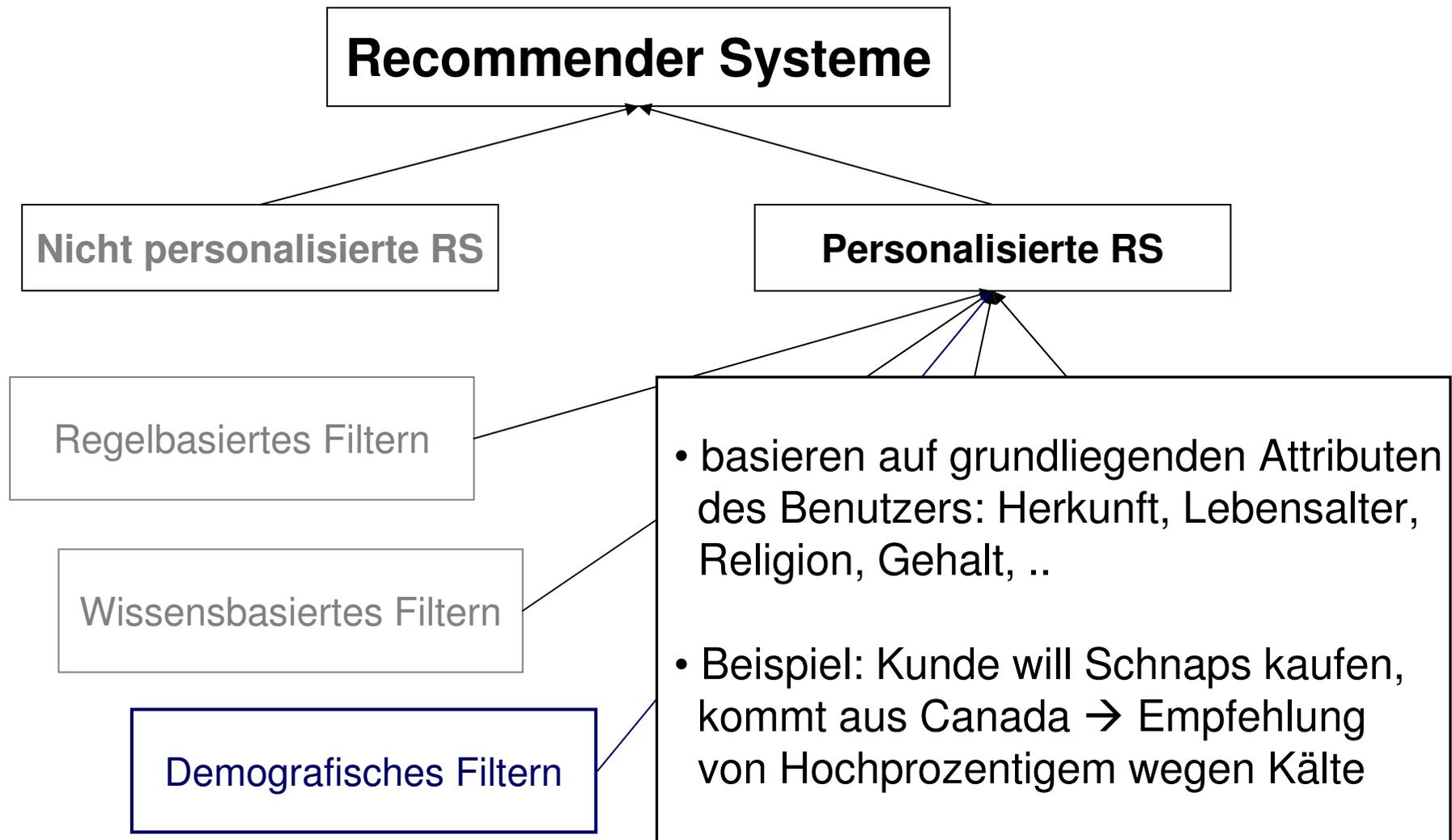
- Aus Decision Support Systeme entwickelt (50er Jahre)
- Collaborative Filtering (1987 patentiert)
- Tapestry: Mailfiltersystem
- Universitäre Forschungsprojekte: Group Lens, Open Sesame!

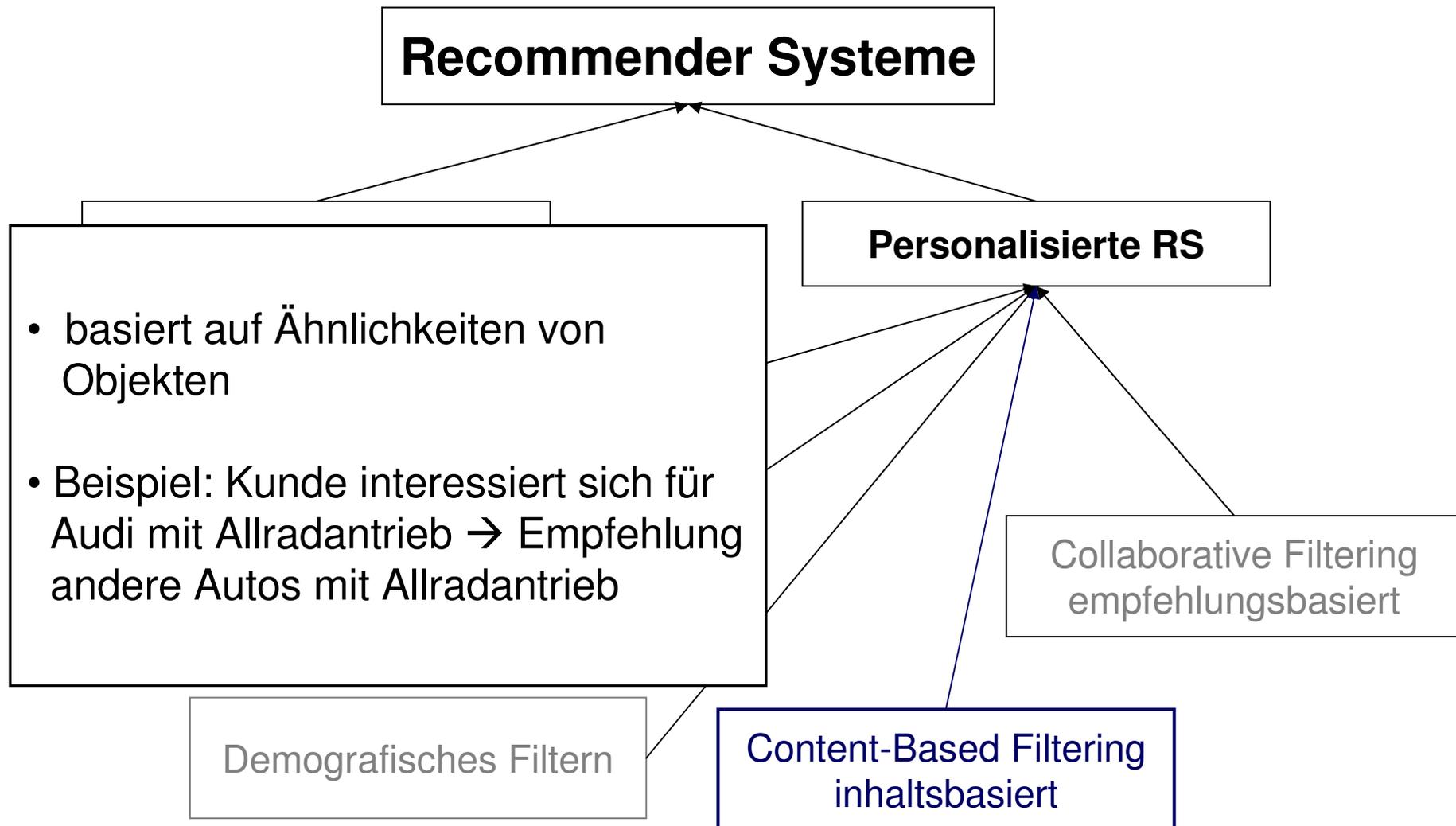


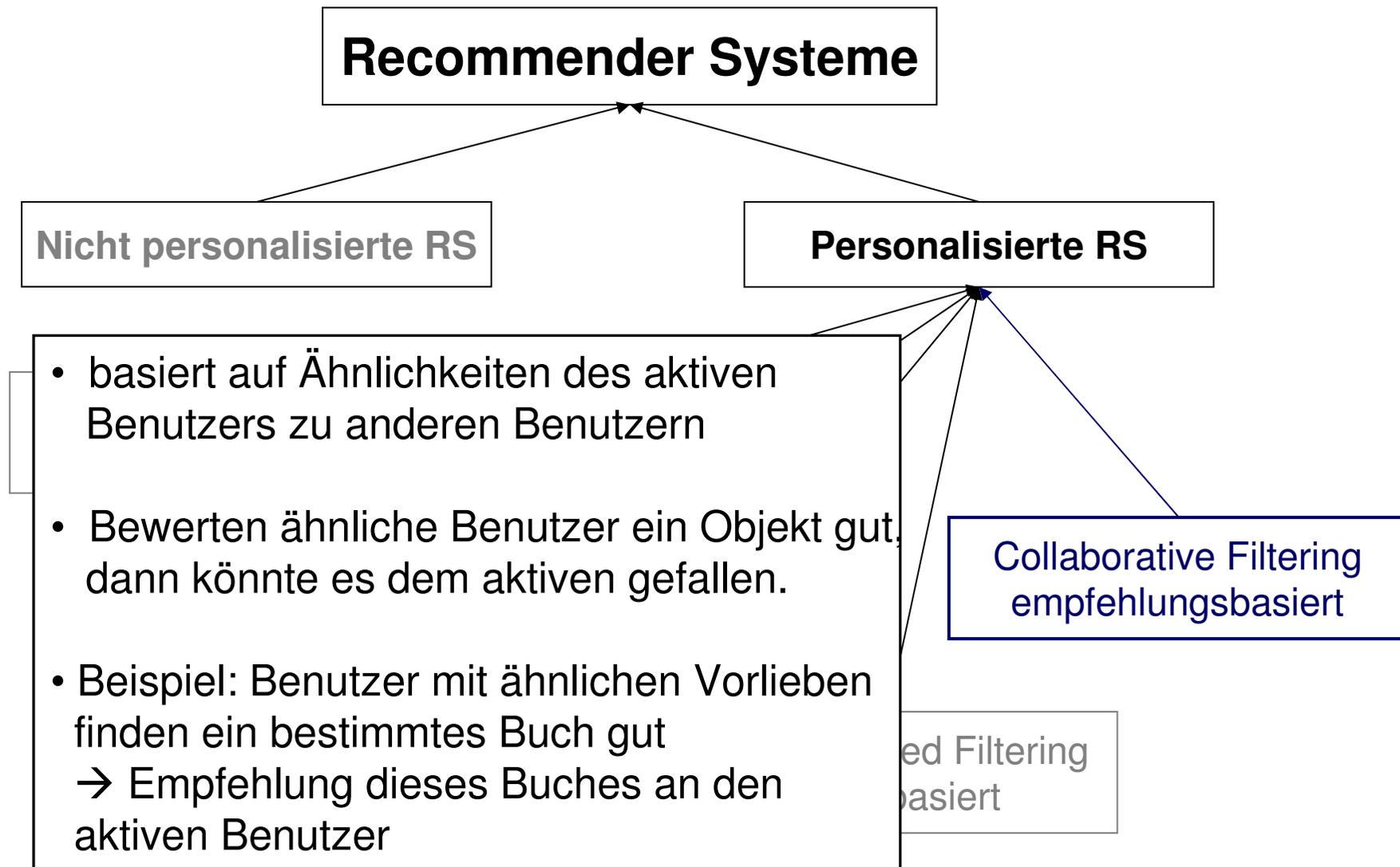












Content-Based Filtering Prinzip

Schritte zur Empfehlung für Textdokumente:

1. Klassifikation der Textdokumente, mit denen Benutzer Kontakt hatte anhand **Schlüsselwörter** im Text
2. Sammlung gewichteter Schlüsselwörter → Beschreibung des Interessensgebietes → **Benutzerprofil**
3. Verstärkung des Profils durch **weiteren Benutzerinput**:
 - Explizit: Fragebögen,..
 - Implizit: Verweildauer auf Webseiten, ..
4. **Vergleich** aller Dokumente mit Profil → Auswahl der ähnlichsten Dokumente

Beispiel: Filme

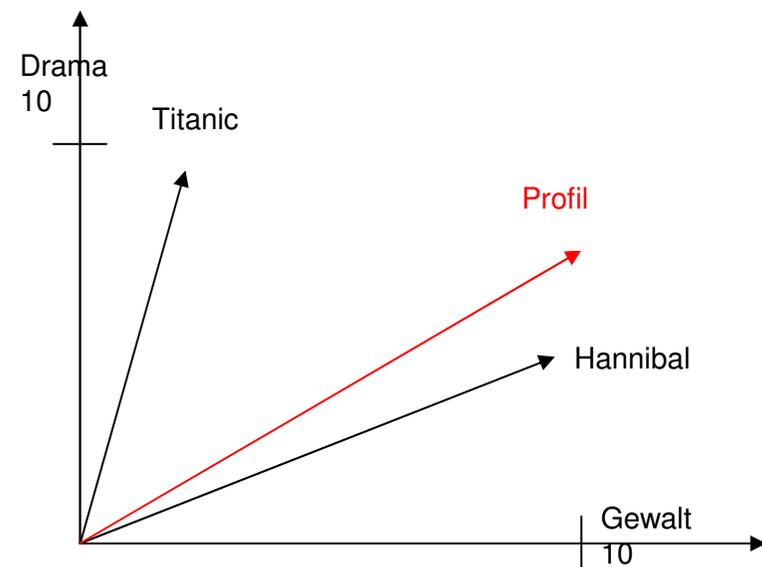
Benutzer hat „Sieben“ gesehen, was gefällt ihm, was empfiehlt ein Content-Based RS?

Film\Eigenschaft	Drama	Liebe	Gewalt	Humor	Action
Sieben	5	1	10	1	5
Hannibal	4	0	9	2	3
Titanic	8	10	1	0	2

Ergebnis: Hannibal. Wie berechnet?

Auffinden zum Profil ähnlicher Objekte

- Exact Match: Verwendung Boolescher Operationen (and, or, not)
 - Bewertung eines Objekts mit 0 oder 1
 - starke Einschränkung
- Best Match: Verwendung des Vektorraummodells
 - n Schlüsselwörter → n Dimensionen
 - Abstand → Ähnlichkeit
 - Geringster Abstand → Empfehlung
 - Geringere Einschränkung



Bewertung von Content-Based Filtering

- + Funktioniert schon mit geringen Nutzerzahlen
- + Empfehlungen für Betreiber nachvollziehbar
- Keine subjektiven Bewertungen erfassbar
- Überspezialisierung

Collaborative Filtering Prinzip

Schritte zur Empfehlung von Objekten:

1. Bestimme die Ähnlichkeit der übrigen Benutzer zum aktiven Nutzers (Profile)
2. Wähle Benutzer aus, die
 - ähnlich zum aktiven Nutzer sind
 - Weitere Objekte bewertet haben
3. Summiere die gewichteten (z.B. abhängig vom Grad der Ähnlichkeit) Bewertungen dieser Benutzer auf
→ Empfehlung

Beispiel: Filme

Nutzer 1 bis 4 haben Filme bewertet, welche Nutzer sind sich ähnlich?

	Terminator	Rambo	Titanic	Dirty Dancing	Star Wars
Nutzer 1	5	-	1	2	5
Nutzer 2	3	5	-	4	5
Nutzer 3	3	2	4	4	3
Nutzer 4	5	4	1	2	4

Ergebnis: Nutzer 3 - 4 sehr unähnlich, Nutzer 1 - 4 sehr ähnlich. Wie berechnet?

Auffinden von ähnlichen Nutzern

- Bestimmung der Distanz der Nutzer in Matrix, z.B. durch euklidischen Abstand
- Ergebnis: **Abstandsmatrix**

	Nutzer 2	Nutzer 3	Nutzer 4
Nutzer 1	0	-0,47	0,97
Nutzer 2		0,85	-0,32
Nutzer 3			-0,65

- Abstandswert < Schwelle: Nutzer fällt heraus (Schritt 2)
- Abstandswert kann als Gewicht für Summierung der Bewertungen dienen (Schritt 3)

Bewertung Collaborative Filtering

- + Geringer Initialaufwand
- + Berücksichtigung subjektiver Bewertungen
- + Flexibel
- + Hochwertige Empfehlungen (Cross-Genre)

- Minimum an Profilen notwendig (New-User-Problem)
- Objekte müssen bewertet sein (New-Item-Problem)
- Empfehlungsprozess nicht transparent für Betreiber

Beispiele

- Collaborative Filtering: <http://movielens.umn.edu>

Predictions for you ↕	Your Ratings	Movie Information	Wish List
★★★★	Not seen	Butterfly Effect, The (2004) info imdb Drama, Sci-Fi, Thriller	<input type="checkbox"/>
★★★★	Hide this	Identity (2003) DVD , VHS , info imdb Crime, Horror, Mystery, Thriller	<input type="checkbox"/>
★★★★	0.5 stars	Salton Sea, The (2002) DVD , VHS , info imdb Crime, Drama, Thriller	<input type="checkbox"/>
★★★★	1.0 stars	Red Dragon (2002) DVD , VHS , info imdb Crime, Mystery, Thriller	<input type="checkbox"/>
★★★★	1.5 stars	Phone Booth (2002) DVD , VHS , info imdb Drama, Thriller	<input type="checkbox"/>
★★★★	2.0 stars	One Hour Photo (2002) DVD , VHS , info imdb Thriller	<input type="checkbox"/>
★★★★	2.5 stars	Spy Game (2001) DVD , VHS , info imdb Action, Crime, Drama, Thriller	<input type="checkbox"/>
★★★★	3.0 stars		
★★★★	3.5 stars		
★★★★	4.0 stars		
★★★★	4.5 stars		
★★★★	5.0 stars		

- Gemischte Methoden: amazon.de

Zusammenfassung

- Verschiedene Filtermethoden haben alle Vor- und Nachteile
- Forderung: in Echtzeit, trotz riesiger Nutzdaten (Amazon: 29 mio Benutzer, Millionen Produkte) gute und schnelle Empfehlungen generieren
- Nachteile beseitigen durch kombinierte, hybride Filtermethoden (Item Based Collaborative Filtering: Amazon)
- Probleme wie Arbeiten auf leeren Datenbeständen bestehen weiterhin (Coldstart)
- Nur wenige, personalisierte Webseiten nutzen Recommender Systeme