REST in Pieces

Jörn Clausen
joern@TechFak.Uni-Bielefeld.DE

Worum geht es?

- Dissertation

  Architectural Styles and the Design of Network-based Software Architectures

  von Roy T. Fielding, UC Irvine, 2000

- „[. . .] a model for the modern Web architecture was needed to guide its design, definition, and deployment.“
- Representational State Transfer (REST)
- REST vs. SOAP

Überblick

- Was ist Software Architektur?
  Was sind „architectural styles“?
- Wer ist Roy T. Fielding?
- Was ist REST?
- Was sind die Vor- und Nachteile von REST?
- Was sind die Probleme von REST?

Software Architektur

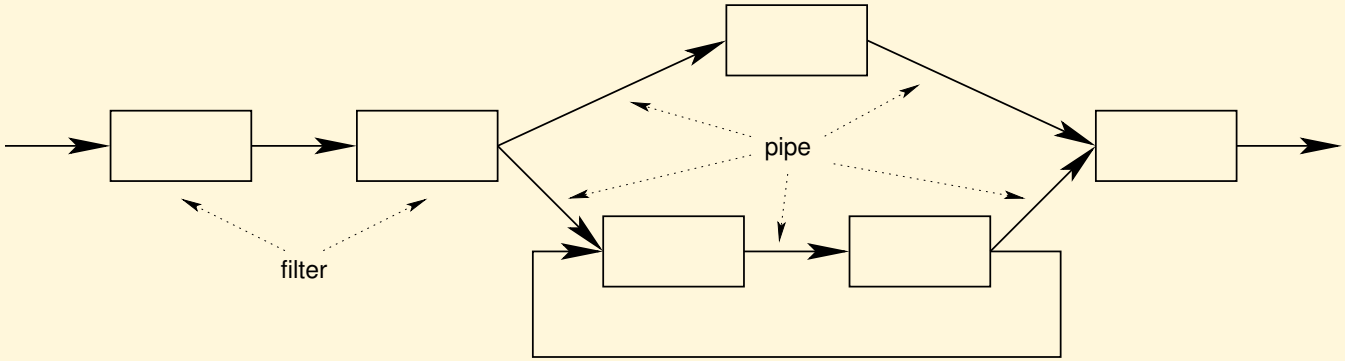
- Beschreibung komplexer Software-Systeme:
  - Zerlegung in Komponenten
  - Kommunikation zwischen Komponenten
- Ziele:
  - Performanz
  - Skalierbarkeit
  - Weiterentwickelbarkeit
  - Wiederverwendbarkeit

architectural styles

An **architectural style** is a coordinated set of architectural contraints that restricts the roles/features of architectural elements and the allowed relationships among those elements within any architecture that conforms to that style.

- *components*: Filter, Schicht, Client, Server, Datenbank, ...
- *connectors*: pipe, Aufruf, Event, Zugriff, ...
- *data*: Byte-Sequenz, Nachricht, ...

pipes and filters



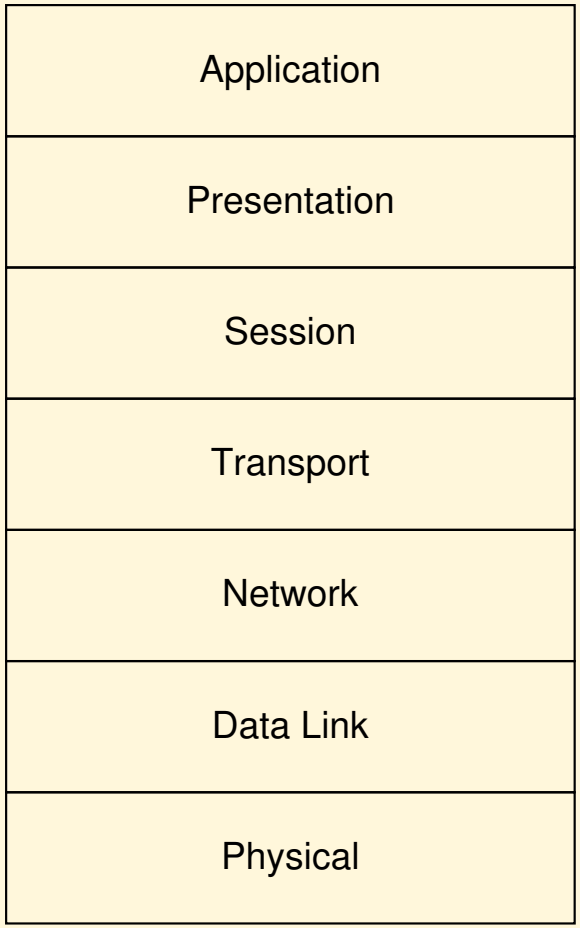
- Beispiel Shell:

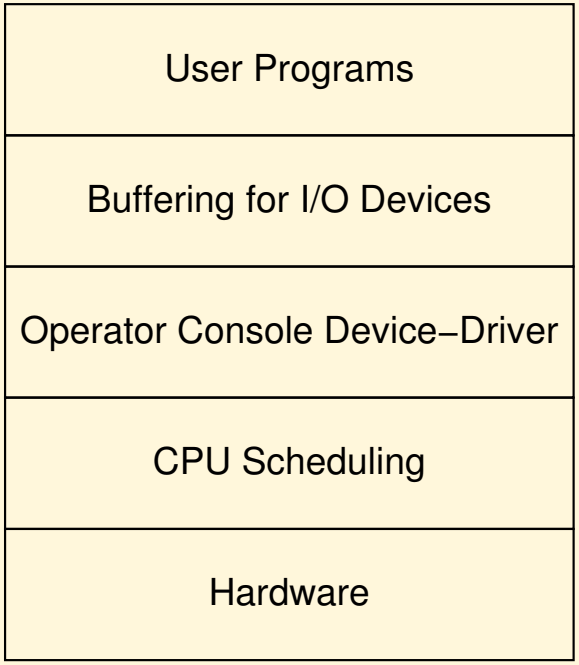
```
ps -e | grep mozilla | awk '{print $1}' | xargs kill
```

```
pgmcrater -n 50000 | pnmsmooth | cjpeg > moon.jpg
```

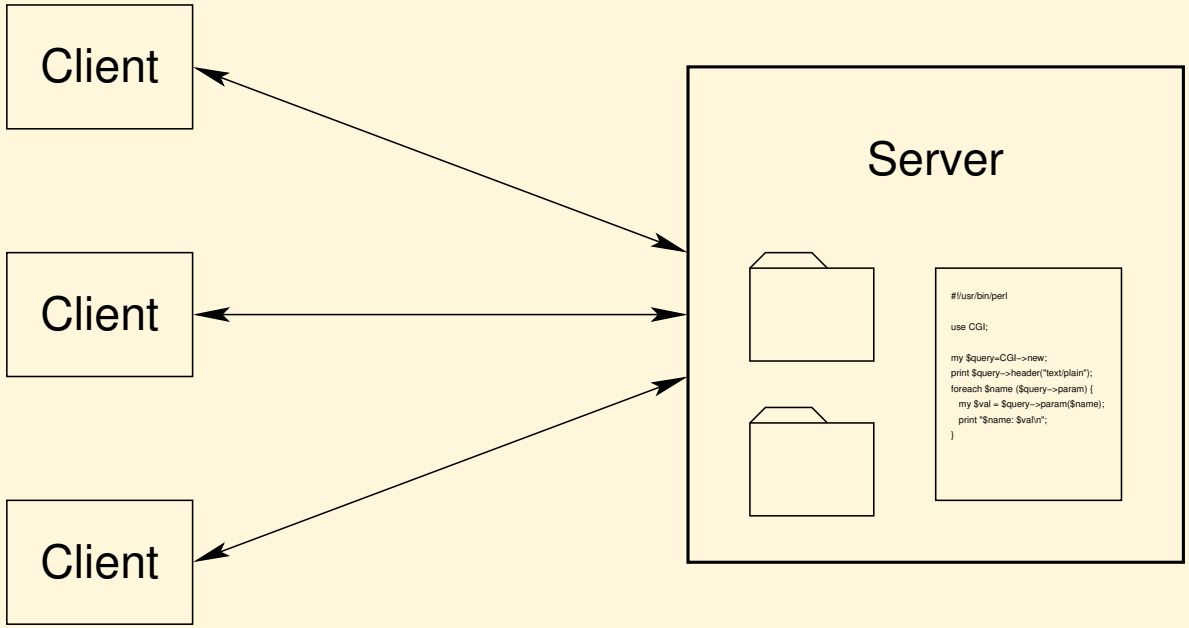
- Khoros, Gimp, ...

Schichtenmodelle





Client-Server



Roy Thomas Fielding

- Apache Software Foundation, Mitgründer und Director
- MOMspider
- libwww-perl, Perl 4
- RFC 1808: Relative Uniform Resource Locators
- RFC 1945: Hypertext Transfer Protocol – HTTP/1.0
- RFC 2068: Hypertext Transfer Protocol – HTTP/1.1
- RFC 2396: Uniform Resource Identifiers (URI): Generic Syntax
- RFC 2616: Hypertext Transfer Protocol – HTTP/1.1

eine Architektur für das WWW

- leichter Zugang für Leser, Autoren und Entwickler
- Erweiterbarkeit
- verteilte Daten
  - Übertragung großer Datenmengen
  - Minimierung der Latenz
- Teil des Internets
  - Zugriffe auf eigene Daten, Verweise auf fremde Daten
  - keine *backpointer* möglich
  - Sicherheit/Vertrauen nur durch geeignete Maßnahmen

Fundamente

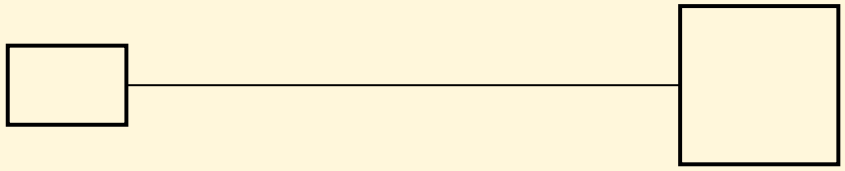
- drei grundlegende Standards:

URI	Uniform Resource Identifiers
HTTP	Hypertext Transfer Protocol
HTML	Hypertext Markup Language

- Adressierung/Lokalisierung durch URIs
- Kommunikation zwischen Komponenten mittels HTTP
- Verknüpfung durch *links* in HTML
- Entwicklung von URIs und HTTP aufgrund von REST

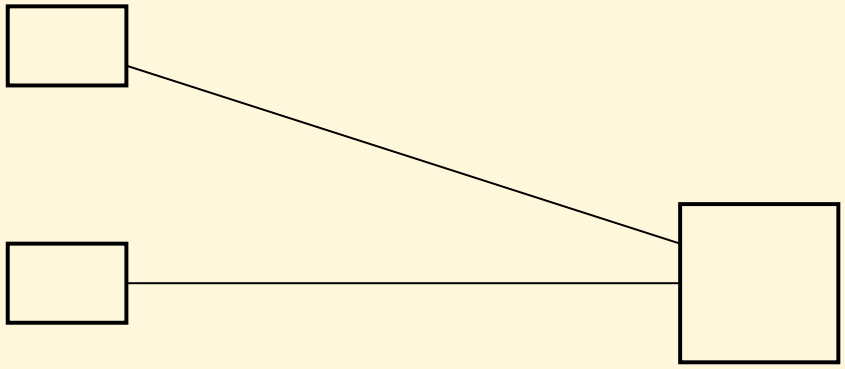
REST

- Client-Server



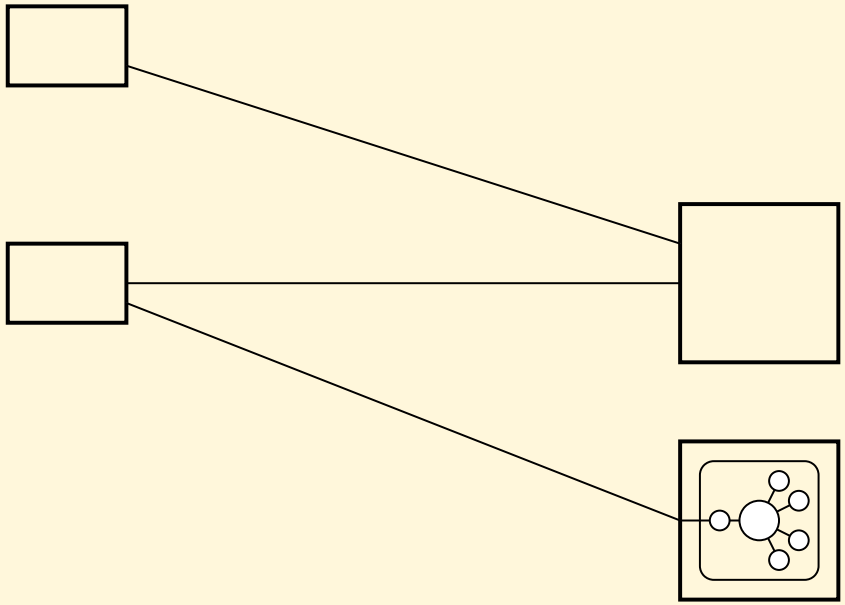
REST

- Client-Server
- zustandslos



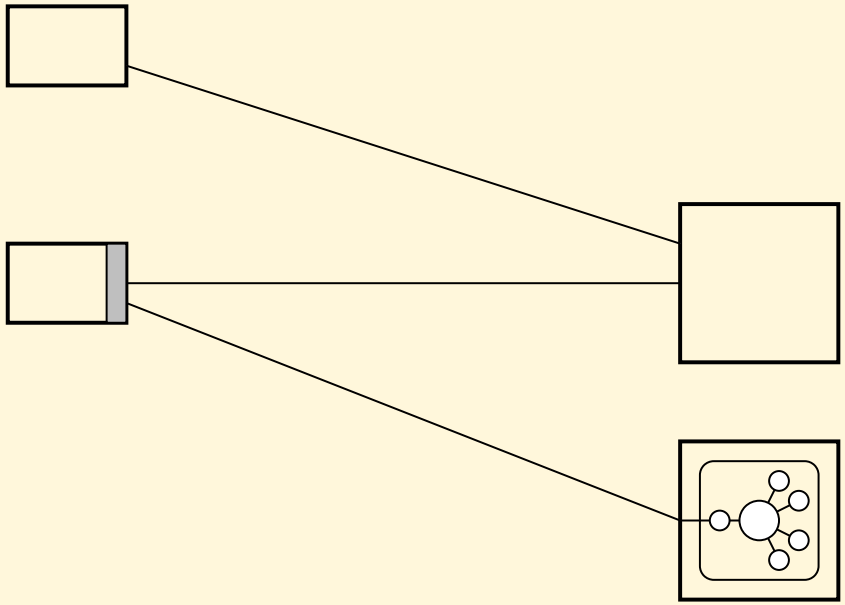
REST

- Client-Server
- zustandslos
- einheitliches Interface



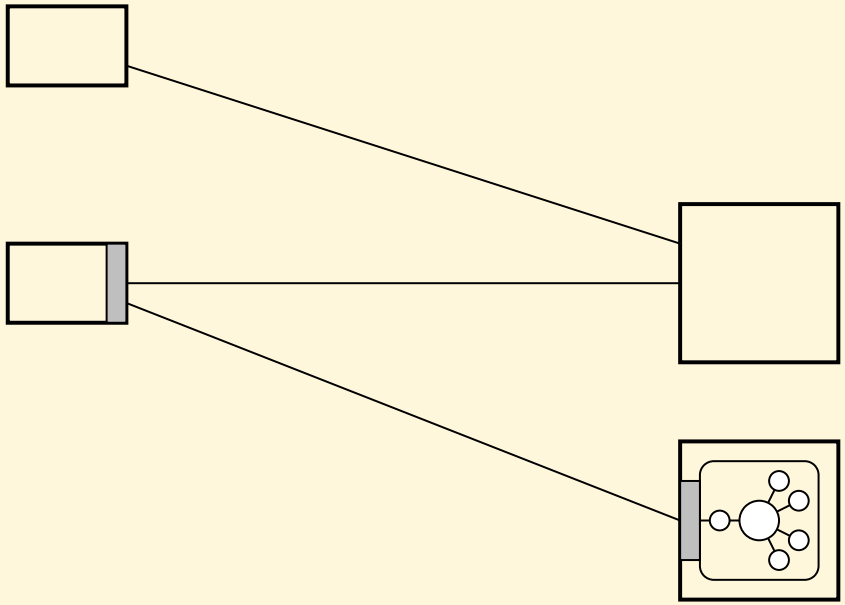
REST

- Client-Server
- zustandslos
- einheitliches Interface
- caching



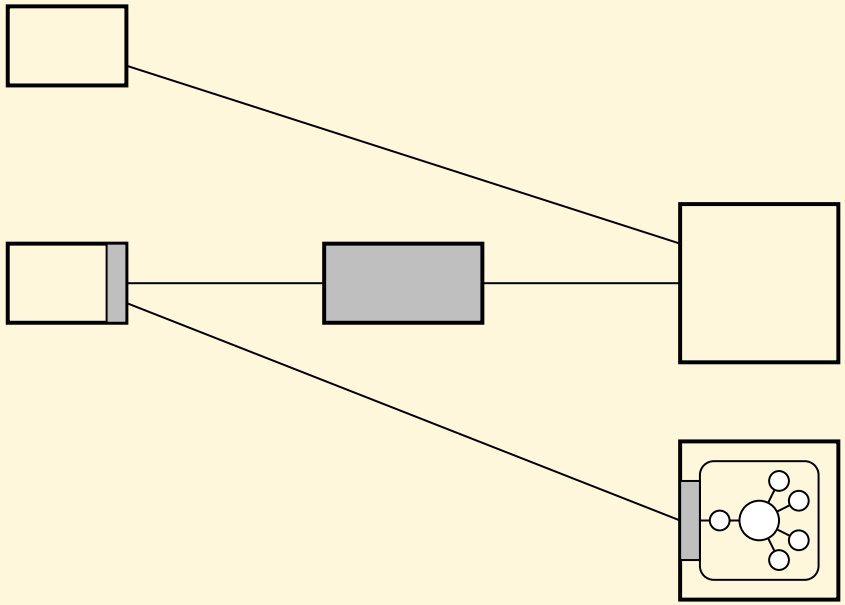
REST

- Client-Server
- zustandslos
- einheitliches Interface
- caching



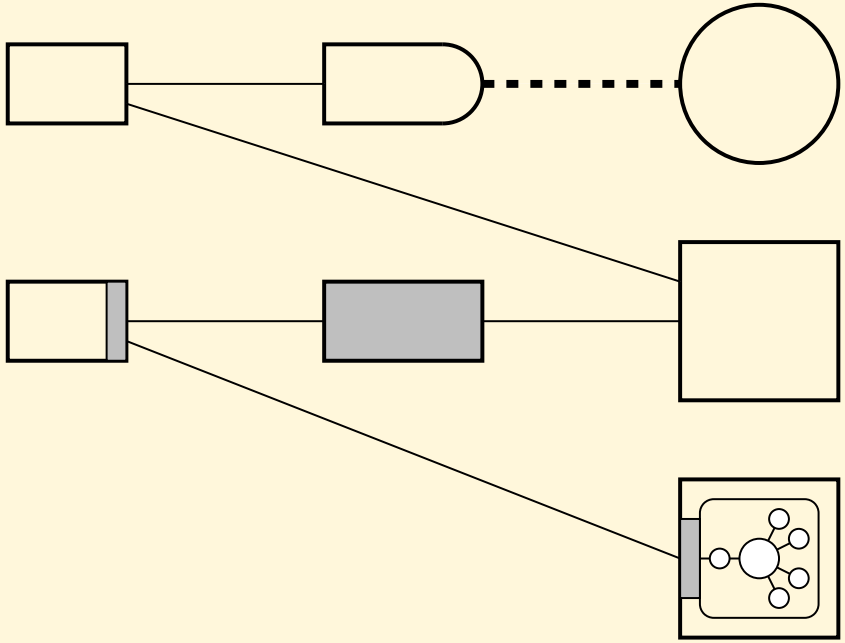
REST

- Client-Server
- zustandslos
- einheitliches Interface
- caching
- Schichten



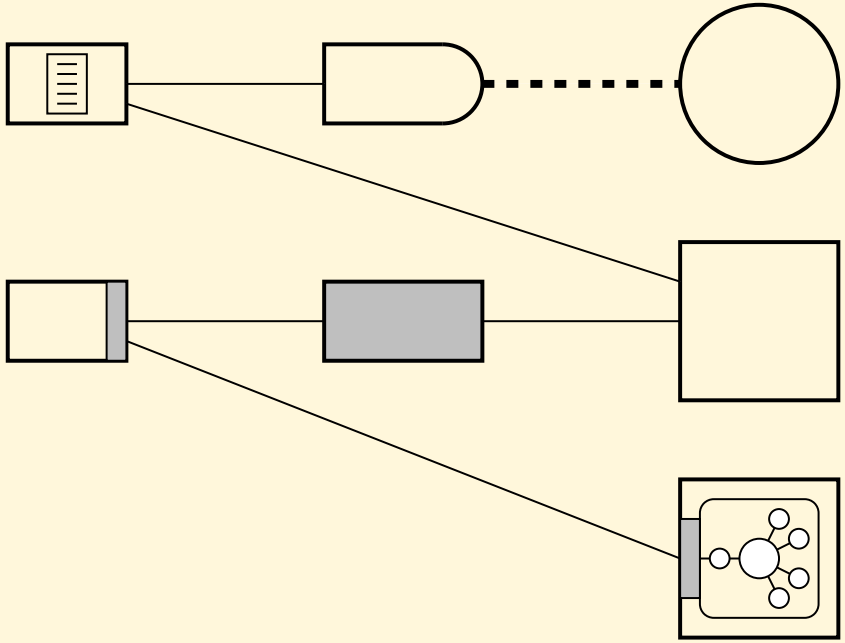
REST

- Client-Server
- zustandslos
- einheitliches Interface
- caching
- Schichten



REST

- Client-Server
- zustandslos
- einheitliches Interface
- caching
- Schichten
- Code-on-Demand



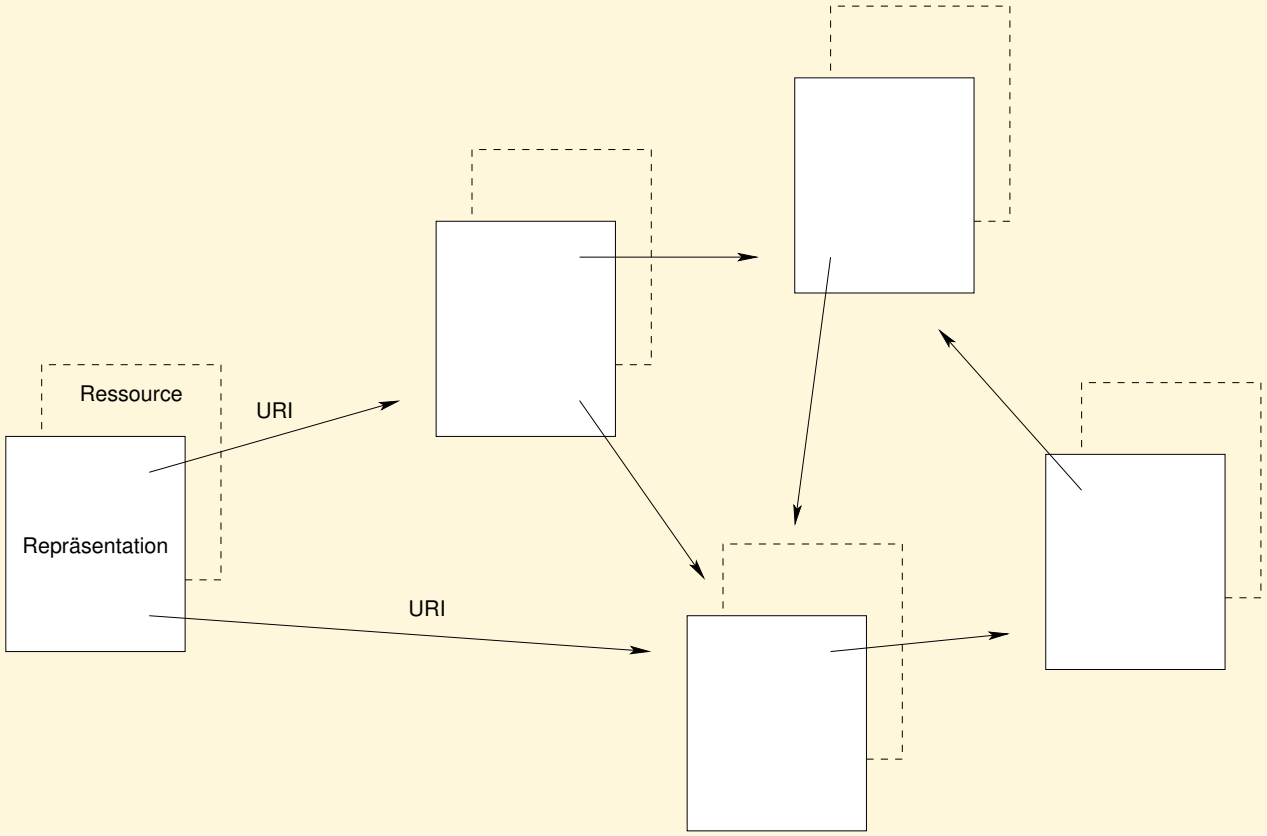
Daten in REST

- zentraler Begriff: Ressource

  > Any information that can be named can be a resource: a document or image, a temporal service (e.g. “today's weather in Los Angeles”), a collection of other resources, a non-virtual object (e.g. a person), and so on.

- Ressource ist Abbildung auf Objekt, nicht das Objekt selber
- Kennzeichnung durch *resource identifier*
- Vergabe durch *naming authority*
- hat Erhaltung der semantischen Korrektheit zu gewährleisten
- Komponenten tauschen Repräsentationen von Ressourcen aus

Representational State Transfer



RESTfulness

- Ressource durch URI gekennzeichnet
- Suche in Google:

  `http://www.google.de/search?q=REST&start=30`

- Ergebnis der Suche hat URI, läßt sich bookmarken
- Verstoß: Parameterübergabe per POST (z.B. `CGI.pm`)
- Länge von URIs nicht durch Standard eingeschränkt
- streaming data?

RESTless

- Frames
  - Konfiguration des Browsers nicht an URI gebunden
- user/session IDs im URL
  - erzeugen „Zustand“
  - verhindern caching
- Cookies
  - erzeugen „Zustand“
  - skalieren schlecht (ganzer Webserver)

Probleme mit REST

- „Ressource“ zu ungenau/falsch definiert
- auf was verweist

  `http://www.w3.org`
- Ressourcen ohne Repräsentation (z.B. namespaces)
- naming authority
- Verflechtung mit HTTP
  - positiv: HTTP nicht als Transport-Protokoll
  - aber: alles wird auf HTTP+URIs abgebildet
  - REST vs. SOAP-Debatte

Probleme mit REST

- „Ressource“ zu ungenau/falsch definiert
- auf was verweist

  `http://www.w3.org/Consortium/`
- Ressourcen ohne Repräsentation (z.B. namespaces)
- naming authority
- Verflechtung mit HTTP
  - positiv: HTTP nicht als Transport-Protokoll
  - aber: alles wird auf HTTP+URIs abgebildet
  - REST vs. SOAP-Debatte

URI vs. URI

Roy T. Fielding, 2003-01-14, www-tag:

> [. . .]
>
> In any case, the reason we had this discussion originally is because some people were complaining about xmlns identifiers being http URIs because they believed that a URI could not be both a name and a way of retrieving a web page. They are wrong, as demonstrated repeatedly by working practice and the REST model, because they were ignoring the difference between a URI and a GET action on a URI.