

Stefan Tilkov, innoQ Deutschland GmbH

# RESTful HTTP

<http://www.innoq.com>

**Stefan Tilkov**

**stefan.tilkov@innoq.com**

**[http://www.innoq.com/blog/st/  
@stilkov](http://www.innoq.com/blog/st/@stilkov)**



**innoQ Deutschland GmbH**

Halskestraße 17

D-40880 Ratingen

Phone +49 21 02 77 162-100

[info@innoq.com](mailto:info@innoq.com) · [www.innoq.com](http://www.innoq.com)

**innoQ Schweiz GmbH**

Gewerbestrasse 11

CH-6330 Cham

Phone +41 41 743 01 11

# Software Architecture Consultancy

~50 Employees

in Germany

(Ratingen, Darmstadt)

and Switzerland (Zurich)

Strategic IT Consulting, Architectural  
Consulting, Development

Service-oriented Architecture (SOA)

(WS-\*, REST, OSS/Lightweight Solutions)

Efficient Software Development

(MDA, MDSD, Java EE, Ruby on Rails)



<http://soa-expertenwissen.de>



<http://rest-http.info>

**<http://Heise.de/developer/podcast>**

**What is REST?**



**1:**

# **REST, an Architectural Style**

**1:  
REST, an Architectural Style**

**Described by Roy T. Fielding  
in his dissertation**

**1:**

# **REST, an Architectural Style**

**One of a number of  
architectural styles**

**1:**

# **REST, an Architectural Style**

**Defined via constraints**

**1:**

**REST, an Architectural Style**

**Architectural principles  
underlying HTTP,  
defined *a posteriori***

# **1: REST, an Architectural Style**

**The Web as one  
particular instance**

**2:**

**REST, the Web Used Correctly**

**2:**

**REST, the Web Used Correctly**

**Use HTTP, URIs, other Web  
standards in line with the  
model**



**2:**

**REST, the Web Used Correctly**

**“On” the Web, not tunneled  
through it**

**2:**

**REST, the Web Used Correctly**

**WOA**

**2:**

**REST, the Web Used Correctly**

**ROA**

**2:**

**REST, the Web Used Correctly**

**RESTful HTTP**

**3:**

**~~XML over HTTP without SOAP~~**

4:

~~Method calls through HTTP GET~~

# The REST Uniform Interface

identification  
of resources

resource  
manipulation  
through  
representations

hypermedia as  
the engine of  
application  
state

self-descriptive  
messages

# The REST Uniform Interface

identification  
of resources

resource  
manipulation  
through  
representations

hypermedia as  
the engine of  
application  
state

self-descriptive  
messages

<http://example.com/orders?year=2008>

<http://example.com/customers/1234>

<http://example.com/orders/2007/10/776654>

<http://example.com/products/4554>

<http://example.com/processes/sal-increase-234>



# The REST Uniform Interface

identification  
of resources

resource  
manipulation  
through  
representations

hypermedia as  
the engine of  
application  
state

self-descriptive  
messages

GET /customers/1234

Host: example.com

Accept: application/vnd.mycompany.customer+xml

<customer>...</customer>

GET /customers/1234

Host: example.com

Accept: text/x-vcard

begin:vcard

...

end:vcard

# The REST Uniform Interface

identification  
of resources

resource  
manipulation  
through  
representations

hypermedia as  
the engine of  
application  
state

self-descriptive  
messages

```
<order self='http://example.com/orders/3321'>  
  <amount>23</amount>  
  <product ref='http://example.com/products/4554' />  
  <customer ref='http://example.com/customers/1234' />  
  <link rel='edit'  
    ref='http://example.com/order-edit/ACDB' />  
</order>
```

# The REST Uniform Interface

identification  
of resources

resource  
manipulation  
through  
representations

hypermedia as  
the engine of  
application  
state

self-descriptive  
messages

Standard  
Method

```
GET /service/customers/1234 HTTP 1.1
Host: www.example.com
User-Agent: XYZ 1.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Keep-Alive: 300
Connection: keep-alive
If-Modified-Since: Fri, 02 Oct 2009 16:47:31 GMT
If-None-Match: "600028c-59fb-474f6852c9dab"
Cache-Control: max-age=60
```

Media Type

# visibility

Control Data

```
HTTP/1.1 200 OK
Date: Sat, 04 Oct 2009 19:05:25 GMT
Server: Apache/2.2.11 Debian
Last-Modified: Fri, 02 Oct 2009 16:47:31 GMT
Etag: "600028c-59fb-474f6852c9dab"
Cache-Control: max-age=300
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 7160
Keep-Alive: timeout=15,max=91
Connection: Keep-Alive
Content-Type: application/xml
```

Data

```
<?xml version='1.0' encoding='utf-8' ?>
```

...

getOrderDetails()

submitApplicationData()

updateQuote()

findMatchingBid()

initiateProcess()

cancelSubscription()

listAuctions()

getUsers()

getOrderDetails()

findMatchingBid()

listAuctions()

getUsers()

**GET**

initiateProcess()

submitApplicationData()

**POST**

**PUT**

updateQuote()

**DELETE**

cancelSubscription()

**generic**

```
interface Resource {  
    Resource(URI u)  
    Response get()  
    Response post(Request r)  
    Response put(Request r)  
    Response delete()  
}
```

Any HTTP client  
(Firefox, IE, curl, wget)

Any HTTP server

Caches

Proxies

Google, Yahoo!, MSN

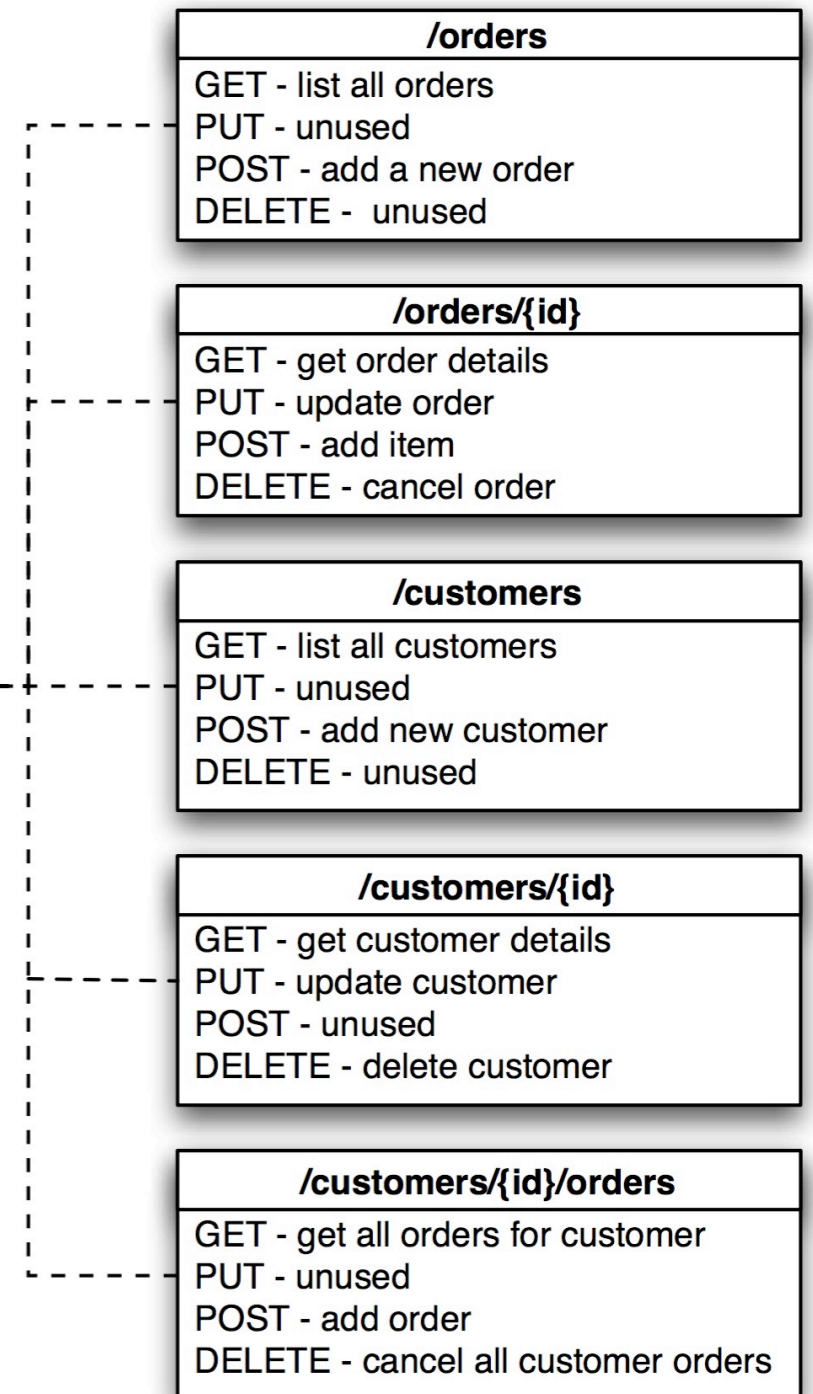
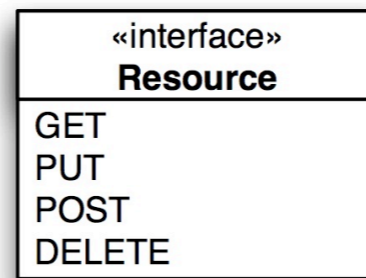
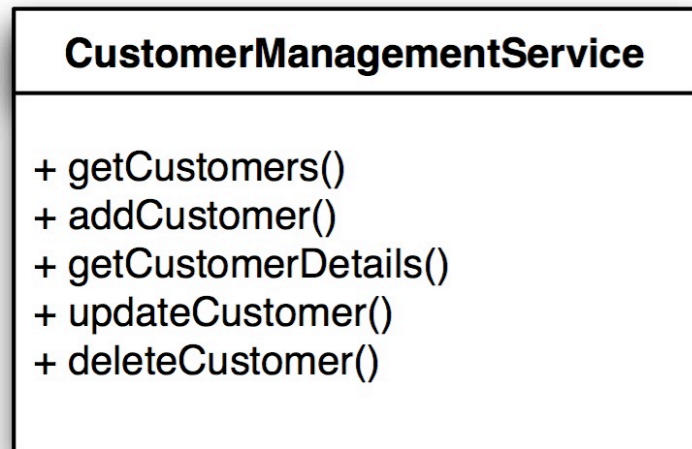
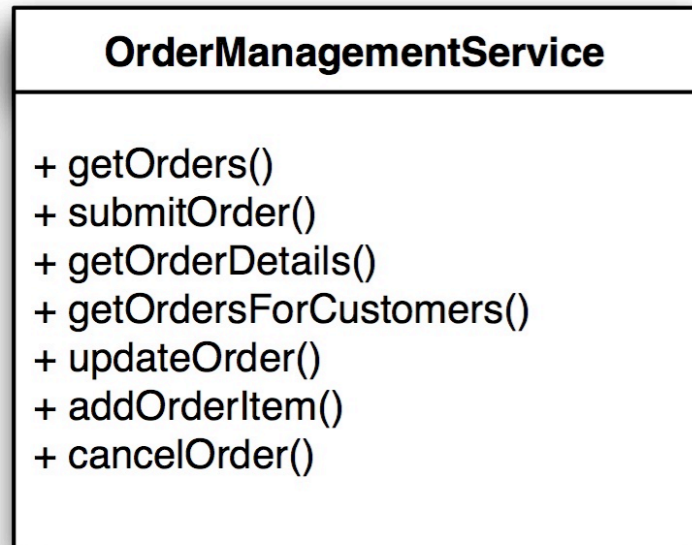
---

```
class CustomerCollection : Resource {  
    ...  
    Response post(Request r) {  
        id = createCustomer(r)  
        return new Response(201, r)  
    }  
    ...  
}
```

Anything that knows  
your app

**specific**

# Consequences





# Web Services

## OrderManagementService

- + getOrders()
- + submitOrder()
- + getOrderDetails()
- + getOrdersForCustomers()
- + updateOrder()
- + addOrderItem()
- + cancelOrder()
- + cancelAllOrders()

## CustomerManagementService

- + getCustomers()
- + addCustomer()
- + getCustomerDetails()
- + updateCustomer()
- + deleteCustomer()
- + deleteAllCustomers()

A separate interface (façade) for each purpose

As known from CORBA, DCOM, RMI/EJB

Often used for SOA (“CORBA w/ angle brackets)

Application-specific protocol

# Contribution to the Net's Value

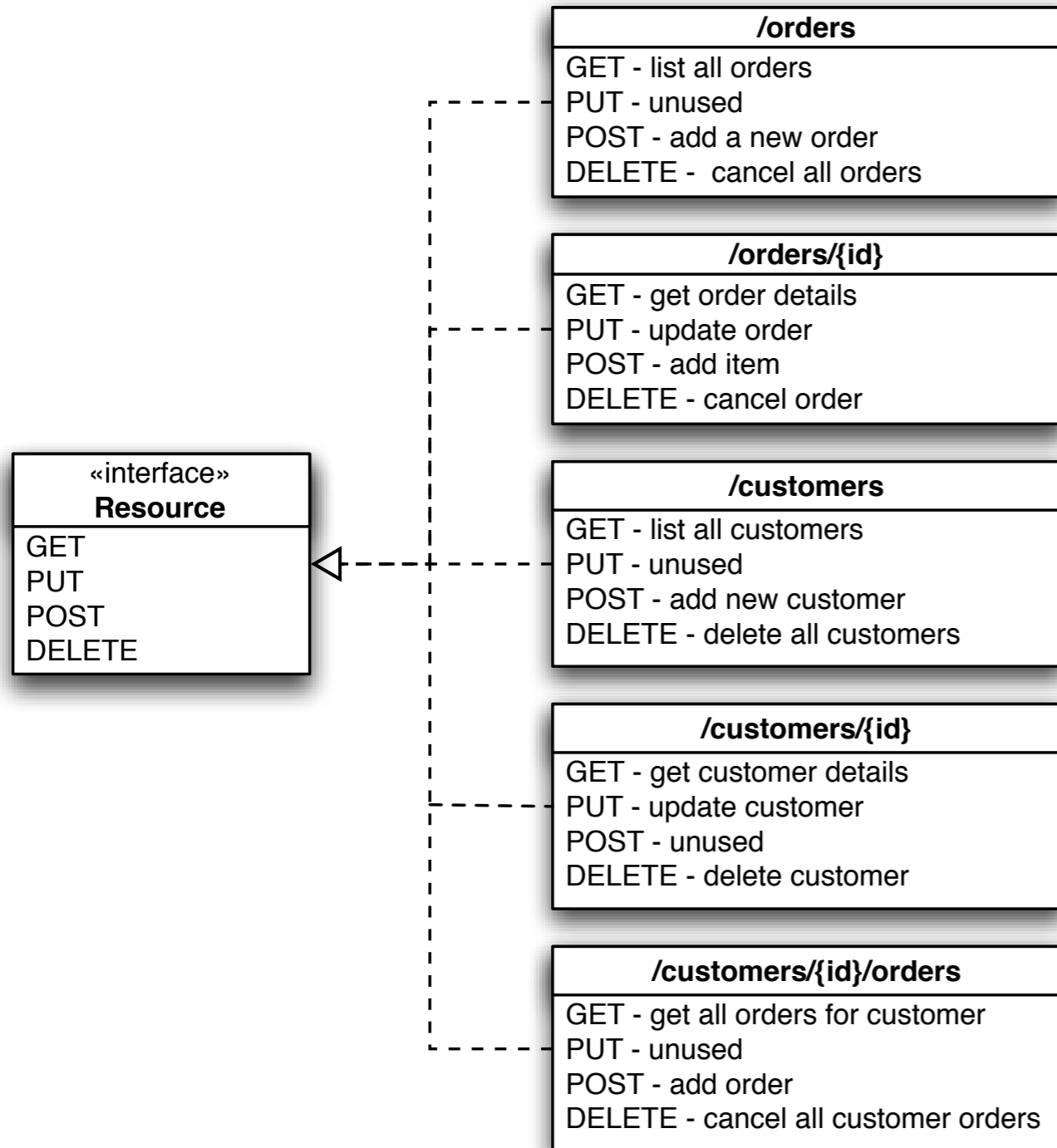
## 2 URLs

- ▶ <http://example.com/customerservice>
- ▶ <http://example.com/orderservice>

## 1 method

- ▶ POST

# REST Approach



A single *generic* (uniform) interface for everything

Generic verbs mapped to resource semantics

A standard application protocol (e.g. HTTP)

# Contribution to the Net's Value

*Millions of URLs*

- ▶ every customer
- ▶ every order

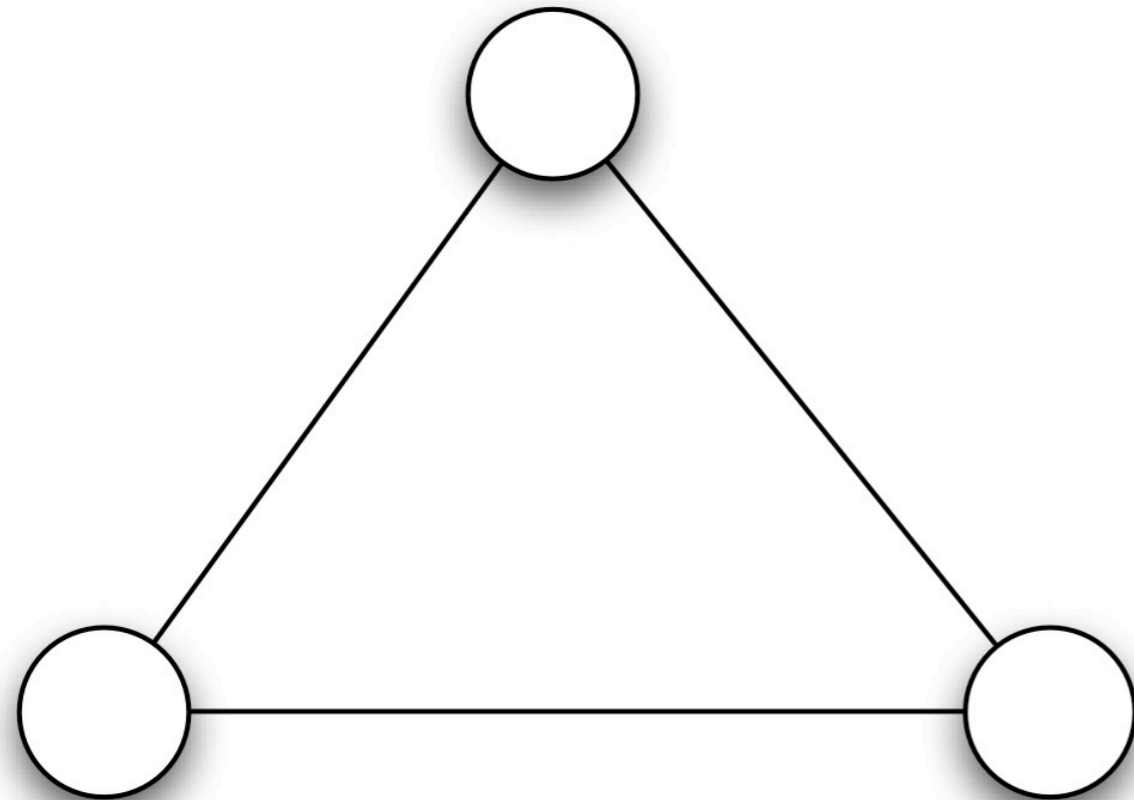
4-6 supported methods per resource

- ▶ GET, PUT, POST, DELETE, OPTIONS, HEAD

Cacheable, addressable, linkable, ...

many

Data types



Operations

many

Instances

very few  
(one per service)

**OrderManagementService**

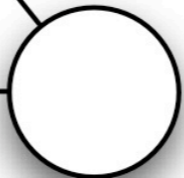
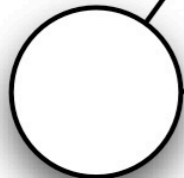
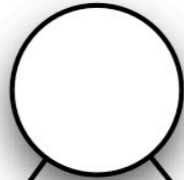
```
+ getOrders()  
+ submitOrder()  
+ getOrderDetails()  
+ getOrdersForCustomers()  
+ updateOrder()  
+ addOrderItem()  
+ cancelOrder()
```

**CustomerManagementService**

```
+ getCustomers()  
+ addCustomer()  
+ getCustomerDetails()  
+ updateCustomer()  
+ deleteCustomer()
```

many

Data types

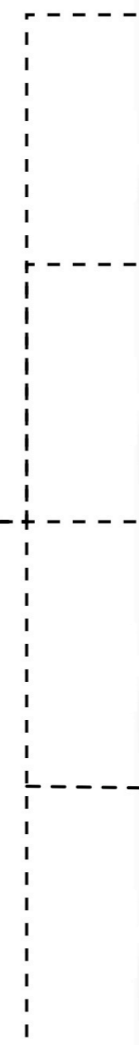
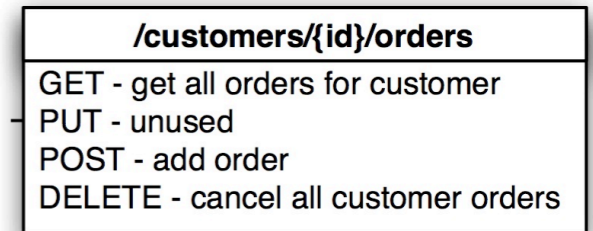
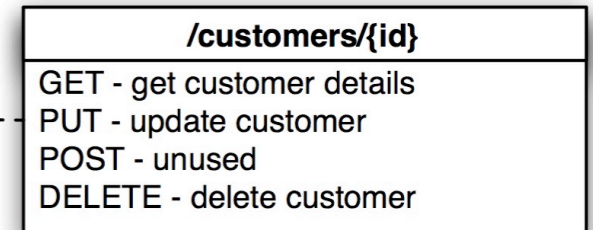
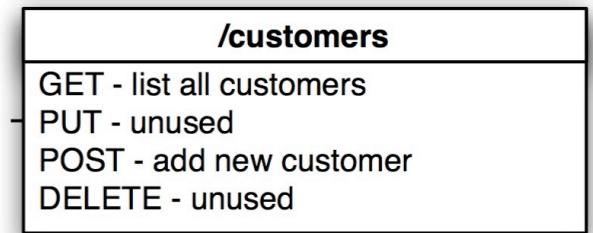
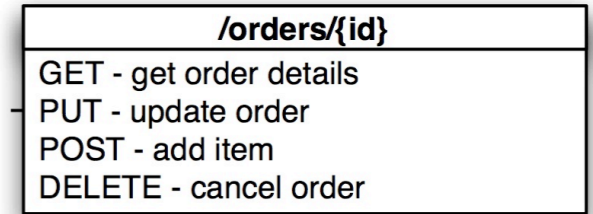
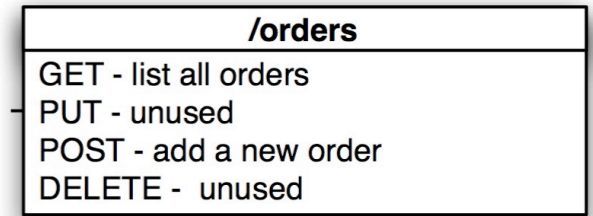
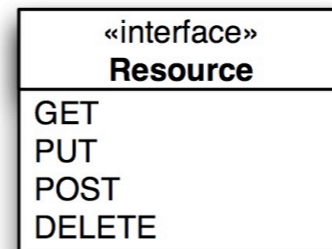


Operations

Instances

very few  
(fixed)

many



# Mapping Examples

getFreeTimeSlots(Person) → GET /people/{id}/timeslots?state=free

---

rejectApplication(Application) → POST /rejections ←  
<application>http://...</application> ←  
<reason>Unsuitable for us!</reason>

---

performTariffCalculation(Data) → POST /contracts ←  
Data  
← Location: http://.../contracts/47 | |  
→ GET /contracts/47 | | /tariff  
← Result

---

shipOrder(ID) → PUT /orders/08 | 5/status ←  
<status>shipped</status>

---

shipOrder(ID) [variation] → POST /shipments ←  
Data  
← Location: http://.../shipments/47 | |

---

# **REST Anti-Patterns**





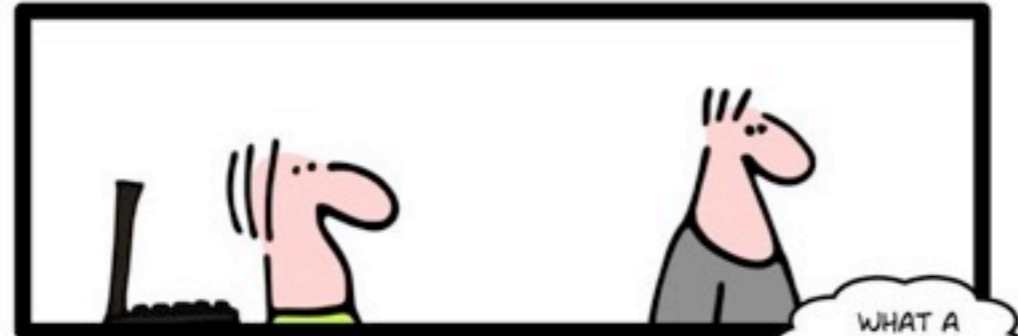
**Tunneling  
Through  
GET**

`http://example.com/some-api?method=insert&name=Smith`

`http://example.com/some-api?method=deleteCustomer&id=13`

GEEK FOR DUMMIES

AND WHEN YOU WANNA SEE THE INVOICE WITH THE ID 42 JUST TYPE  
`http://hitchhiker.lifeofbrian:8080/services/invoice?id=42`  
INTO THE ADDRESS BAR!  
ISN'T THAT UBERCOOL?



CHAPTER 1: BE AWARE THAT NOT EVERYBODY SHARES YOUR ENTHUSIASM ABOUT RESTFUL APIS

NOTICE



Stop Worrying  
About URI Design

`http://example.com/some-api?method=findCustomer&id=13`

`http://example.com/customers/13`

# Accidentally RESTful

<http://www.markbaker.ca/blog/2005/04/14/accidentally-restful/>



# Tunneling Through POST

**(a.k.a. The SOAP Way)**

POST **http://example.com/CustomerMgmt**

```
<soap:Envelope  
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
  <soap:Body>  
    <deleteCustomer xmlns="http://example.com/ns1">  
      <customerId>13</customerId>  
    </ns:deleteCustomer>  
  </soap:Body>  
</soap:Envelope>
```

**Method**

**ID**

**Endpoint**

**“Endpoint”?**



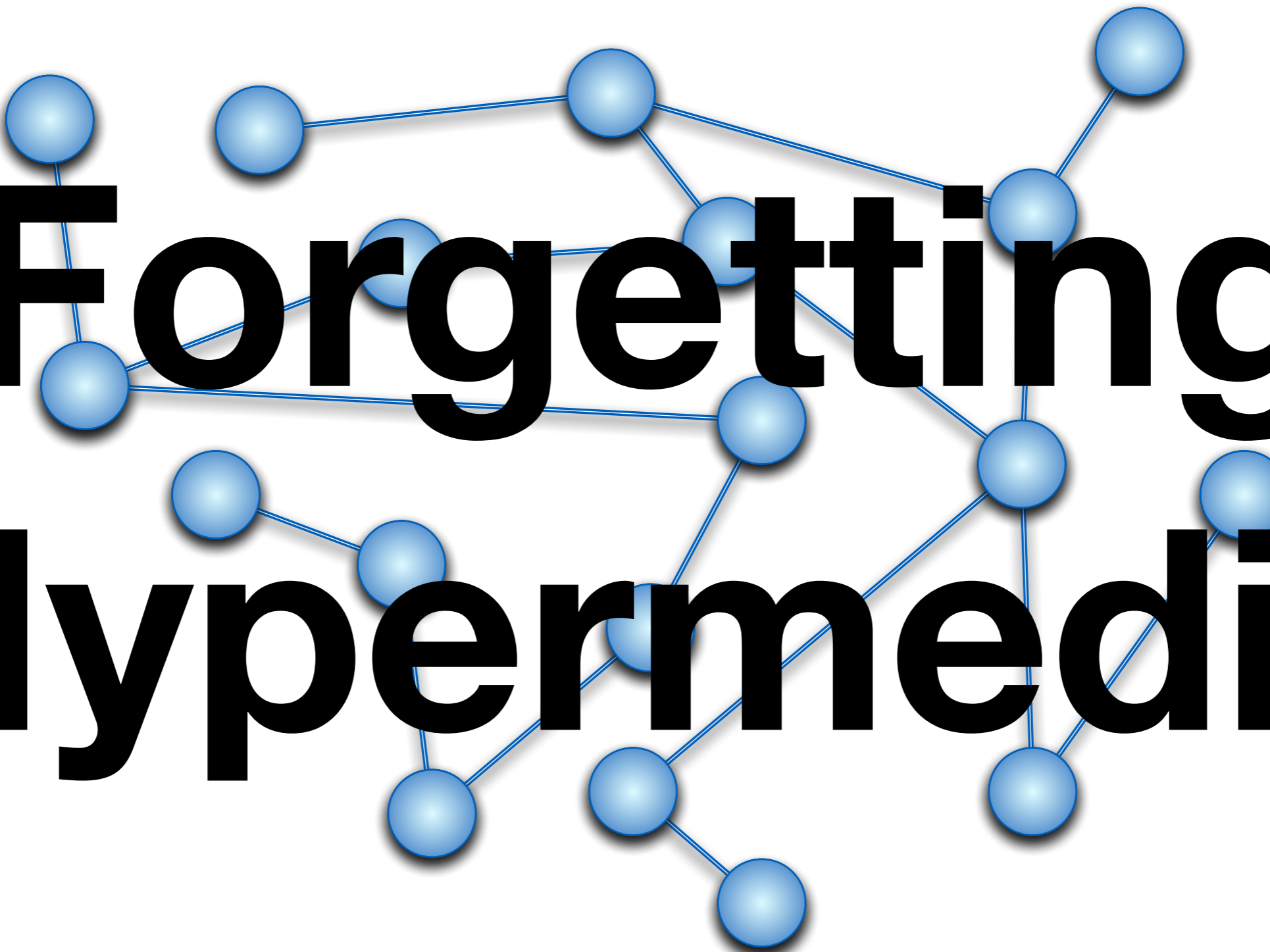
DO NOT

ENTER

DEAD

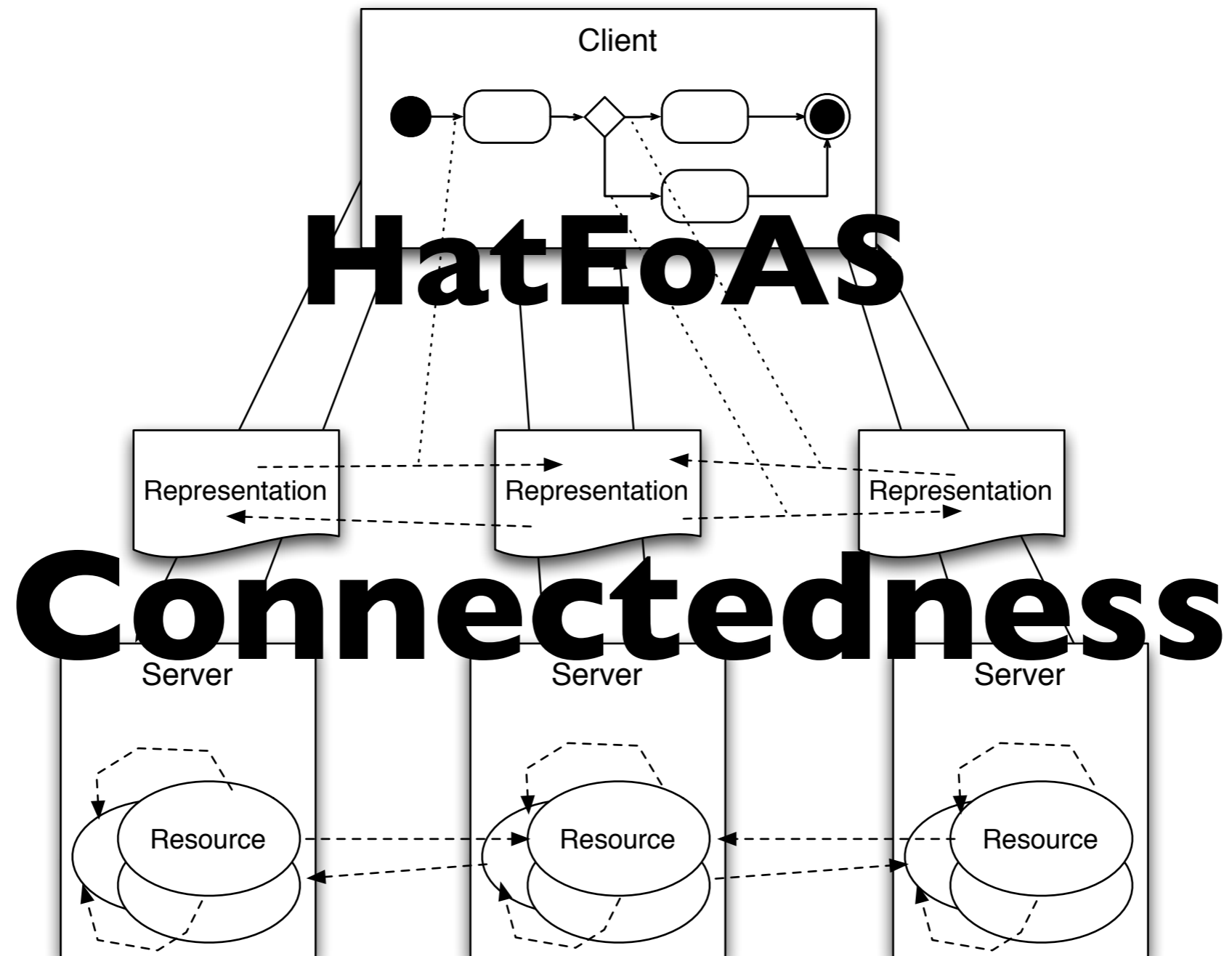
END





**Forgetting  
Hypermedia**

# Hypermedia Levels



**Why you should care**

# The Web & SOA

Standard protocols	
Standard formats	
Library and tool support	
Mature and useful intermediaries	
Support for loose coupling	
Wide availability and adoption	
Well-defined architectural model	

# The Web & SOA

Standard protocols	✓
Standard formats	✓
Library and tool support	✓
Mature and useful intermediaries	✓
Support for loose coupling	✓
Wide availability and adoption	✓
Well-defined architectural model	✓

## APIs and Web Services

### ADVERTISING & BUSINESS SERVICES



**APT from Yahoo!**  
Interact with our advertising platform and ad exchange-based technology.



**Y! Search Marketing**  
Create and manage campaigns, pull reports, set bids



AUTH



[Google Apps APIs](#)



[Google Finance Portfolio Data API](#)



[Google Base Data API](#)



[Google Health Data API](#)



[Blogger Data API](#)



[Google Notebook Data API](#)



[Google Book Search Data API](#)



[Picasa Web Albums Data API](#)



[Google Calendar Data API](#)



[Google Spreadsheets Data API](#)



[Google Code Search Data API](#)



[Webmaster Tools Data API](#)



[Google Contacts Data API](#)



[YouTube Data API](#)



[Google Documents List Data API](#)

COM



Access and update contacts in the Yahoo! Address Book.



Build applications to list messages, display folders, compose and send messages.



Create add-on features that run inside Yahoo! Messenger.

CONTENT



**Finance**  
RSS feeds deliver timely data on



**HotJobs**  
Post, edit, refresh and delete public



**MyBlogLog**  
Build social networking services



#### Infrastructure Services

- Amazon Elastic Compute Cloud (Amazon EC2)
- Amazon SimpleDB
- Amazon Simple Storage Service (Amazon S3)
- Amazon CloudFront
- Amazon Simple Queue Service (Amazon SQS)
- AWS Premium Support

#### Payments & Billing

#### On-Demand Workforce

#### Alexa Web Services

#### Amazon Fulfillment & Associates



***“My Internet is bigger than your enterprise.”***

Paraphrasing Dare Obasanjo,  
see <http://tinyurl.com/dare-enterprise>

**We've been there  
before ...**



[Advanced Search](#)  
[Preferences](#)**Web**Results **1 - 10** of about **61**

### [Basel II – Wikipedia](#) - [ [Translate this page](#) ]

Der Terminus **Basel II** bezeichnet die Gesamtheit der Eigenkapitalvorschriften, die vom **Basler** Ausschuss für Bankenaufsicht in den letzten Jahren ...

[de.wikipedia.org/wiki/Basel\\_II](http://de.wikipedia.org/wiki/Basel_II) - 72k - [Cached](#) - [Similar pages](#) -

### [Bundesbank - Bankenaufsicht - Basel II](#) - [ [Translate this page](#) ]

**Basel II** - Die neue Baseler Eigenkapitalvereinbarung ... Während der im Jahr 1998 begonnenen Entwicklung des **Basel II** Regelwerkes standen die Aufsicht und ...

[www.bundesbank.de/bankenaufsicht/bankenaufsicht\\_basel.php](http://www.bundesbank.de/bankenaufsicht/bankenaufsicht_basel.php) - 27k -

[Cached](#) - [Similar pages](#) -

### [Bundesbank - Bankenaufsicht - Basel II - Säule 2: Aufsichtliches ...](#) - [ [Translate this page](#) ]

Die drei Säulen von **Basel 2** Im Rahmen dieser so genannten zweiten Säule, die als integraler Bestandteil des neuen Kapitalakkords gleichberechtigt neben den ...

[www.bundesbank.de/bankenaufsicht/bankenaufsicht\\_basel\\_saeule2.php](http://www.bundesbank.de/bankenaufsicht/bankenaufsicht_basel_saeule2.php) - 17k -

[Cached](#) - [Similar pages](#) -

[More results from www.bundesbank.de »](#)

### [Basel-II.info - Steuerkanzlei Christian Reichling, Köln](#) - [ [Translate this page](#) ]

Dipl.-Kfm.(FH) Christian Reichling, Steuerberater aus Köln (Cologne) berät Sie gerne in allen Fragen der Bilanzierung und Besteuerung von Gesellschaften ...

[www.basel-ii.info/](http://www.basel-ii.info/) - 12k - [Cached](#) - [Similar pages](#) -

### [Definition Basel 2, Basel II](#) - [ [Translate this page](#) ]

Ausführliche Informationen zum Thema **Basel II** – neue Bedingungen für Firmenkredite, Rating und Tipps.

[www.foerderland.de/353.0.html](http://www.foerderland.de/353.0.html) - 48k - [Cached](#) - [Similar pages](#) -

### [Deutscher Industrie- und Handelskammertag / Homepage](#) - [ [Translate this page](#) ]

Der Deutsche Industrie- und Handelskammertag (DIHK) ist die Spitzenorganisation der 80 Industrie- und Handelskammern (IHKs) in Deutschland.

[www.dihk.de/inhalt/informationen/news/schwerpunkte/rating/basel.html](http://www.dihk.de/inhalt/informationen/news/schwerpunkte/rating/basel.html) - 6k -

[Cached](#) - [Similar pages](#) -

# Unlock information as resources

VersicherungX

Search

[Advanced Search](#)  
[Preferences](#)

Web

Results 1 - 10 of about 1

[Versicherungsnehmer: Hans Müller](#) - [ [Translate this page](#) ]

Hans Müller, 40880 Ratingen, Kunde seit dem 1.3.2001, letzte Aktualisierung 1.1.2009 ...  
[crm.example.com/customers/4711](#) - 72k - [Cached](#) - [Similar pages](#) -

[Kfz-Haftpflicht Hans Müller](#) - [ [Translate this page](#) ]

Kfz.-Haftpflicht für ME-HM 123, abgeschlossen 1.1.2005, Halter: Hans Müller, Audi A4 ...  
[bestand.example.com/lv/26621](#) - 72k - [Cached](#) - [Similar pages](#) -

[Risiko LV Anne Müller](#) - [ [Translate this page](#) ]

Risiko LV Anne Müller, 40880 Ratingen, abgeschlossen 1.1.2004, Begünstiger: Hans Müller,  
Versicherungssumme: €200.000 ...  
[bestand.example.com/lv/26621](#) - 72k - [Cached](#) - [Similar pages](#) -

**Pragmatic  
Recommendations  
from an Enterprisey  
RESTafarian**

**#1: Ensure your Web apps  
are RESTful**

**#2: Expose machine-  
readable information  
via HTTP GET**

“I do think the REST-afarians are missing an opportunity by not driving home the secret sauce that is HTTP GET. [...] **GET is one of the most optimized pieces of distributed systems plumbing in the world.** It's an absolute/objective slam dunk. No arguing/evangelism needed IMO. GET is the classic ‘the first bag is free’ kind of feature a platform builder dreams about.”

***Don Box, Co-inventor of SOAP***



# **#3: Manage Your Metadata with RESTful HTTP**

**#4: Use WS-\* for  
read/write interactions if  
politics or legacy force you**

**#5: Draw your own  
Conclusions Watching  
the Adoption of WS-\* vs.  
RESTful HTTP**

# Q&A



**innoQ Deutschland GmbH**

Halskestraße 17  
D-40880 Ratingen

Phone +49 21 02 77 162-100

[info@innoq.com](mailto:info@innoq.com) · [www.innoq.com](http://www.innoq.com)

**innoQ Schweiz GmbH**

Gewerbestrasse 11  
CH-6330 Cham

Phone +41 41 743 01 11

**Stefan Tilkov**

[stefan.tilkov@innoq.com](mailto:stefan.tilkov@innoq.com)

<http://www.innoq.com/blog/st/>

[@stilkov](#)

Phone: +49 170 471 2625