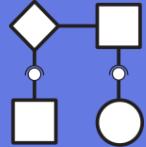




VSR://EDU/SSE



Software Service  
Engineering

# Software Service Engineering

WS 2025/2026 – 7. Tutorial

Valentin Siegert M.Sc.

Maheshika Walpola M.Sc.

*VSR.Informatik.TU-Chemnitz.de*

1

# Task 1

- Answer the following questions:
  - **What is an architecture style?**
  - What are the REST constraints?  
Which properties do they induce?
  - What is the “Hypermedia” constraint and  
which examples of its application do you know?

- What is an architecture style?  
An architectural style is a coordinated **set of architectural constraints** that restricts the roles/features of architectural elements and the **allowed relationships** among those elements within any architecture that conforms to that style.
- Why someone decides to adhere to certain architecture style?

- Answer the following questions:
  - What is an architecture style?
  - **What are the REST constraints?  
Which properties do they induce?**
  - What is the “Hypermedia” constraint and  
which examples of its application do you know?

- What are the REST constraints?
  - Client-Server
  - Stateless
  - Cache
  - Uniform interface
    - Identification of resources
    - Manipulation of resources through representations
    - Self-descriptive messages
    - Hypermedia as the engine of application state
  - Layered system
  - Code-on-demand

- Client-Server
  - Portability (server responses are understood by all clients)
  - Scalability (server outsources part of logic to client)
- Stateless
  - Visibility (request analysis is possible)
  - Reliability (request can be easily repeated in case of failures)
  - Scalability (no persistent storage on the web needed)
  - But: amount of data transferred increases, correct client implementation required
- Cache
  - Scalability (servers are requested not so often)
  - But: reliability can decrease (if cached data differs from original)

- Uniform interface
  - Simplicity (one protocol to learn)
  - Visibility (everyone understands the protocol)
  - Independent evolvability (implementations behind interface can change)
  - But: performance drawbacks because of interface transformation
- Layered System
  - Simplicity (layering of services, request optimization)
  - Scalability (load balancing)
  - But: additional latency
- Code-on-Demand
  - Simplicity (clients can be enriched with functionality)
  - But: reduced visibility (code can be arbitrary)

# Performance

...The primary measures for user-perceived performance are latency and completion time...

# Scalability

...ability of the architecture to support large numbers of components, or interactions among components, within an active configuration...

# Simplicity

...individual components are substantially less complex, then they will be easier to understand and implement..

# **Modifiability (evolvability, extensibility, customizability, configurability, reusability)**

...the ease with which a change can be made to an application architecture...

## **Visibility**

...ability of a component to monitor or mediate the interaction between two other components...

## **Portability**

...Software is portable if it can run in different environments...

## **Reliability**

...the degree to which an architecture is susceptible to failure at the system level in the presence of partial failures within components, connectors, or data...

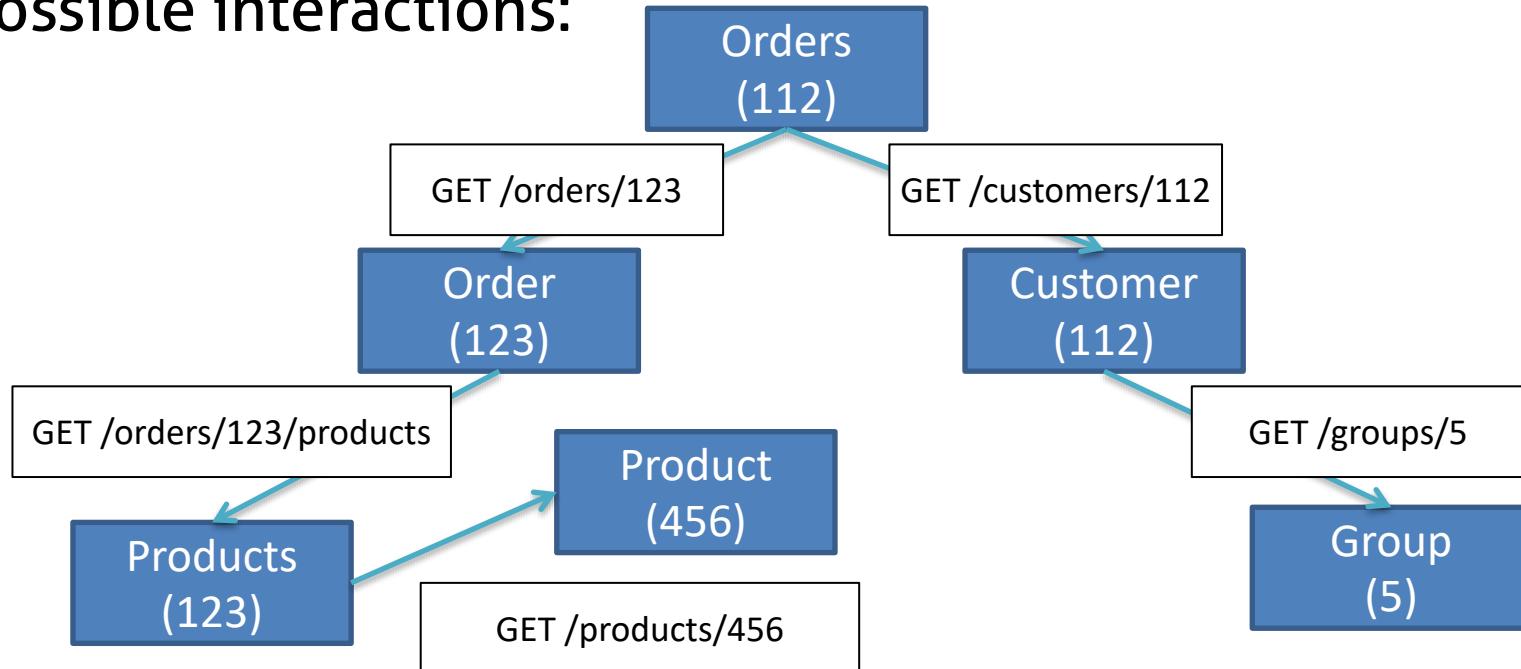


	Client-Server	Stateless Server	Cache	Uniform Interface	Layered System	Code on demand
Performance						
Scalability						
Simplicity						
Modifiability						
Visibility						
Portability						
Reliability						

- Answer the following questions:
  - What is an architecture style?
  - What are the REST constraints?  
Which properties do they induce?
  - **What is the “Hypermedia” constraint and which examples of its application do you know?**

# Part of the uniform interface constraint

Resource representations should expose further possible interactions:



# Content-Negotiation:

- Clients indicate their preferences on representation format, language, time of the representation:
  - Accept: text/html;
  - Accept-Language: de;
  - Accept-Datetime: Thu, 31 May 2007 20:35:00 GMT
- Servers responds according to the client request

# Task 2

SOAP Operation	HTTP Verb	URL	Request Header	Request Body	Response Header	Response Body
getAllBooks()	GET	/books	Accept: */*	-	Content-Type: application/xml Content-Length: ..	<books><book><title>...</title><authors>...</authors></book></books>
getAllBooksAsJson()	GET	/books	Accept: application/json	-	Content-Type: application/json Content-Length: ..	{"books": [{"book": {"title": "...", "authors": "..."}}]}]
getBookById( bookId: int; language: {en, de, fr, nl} )	GET	/books/{bookId}	Accept-Language: en-US; Accept: */*	-	Content-Type: application/xml Content-Length: ..	<book><title>...</title><authors>...</authors></book>
updateBook( bookId: int; book: (title, authors) )	PUT	/books/{bookId}	Content-Type: application/xml Content-Length: ..	<book><title>...</title><authors>...</authors></book>	Content-Type: application/xml Content-Length: ..	<book><title>...</title><authors>...</authors></book>
getAllCategories()	GET	/categories	Accept: */*	-	Content-Type: application/xml Content-Length: ..	<category><title>...</title></category>
getBooksInCategory( categoryId: int )	GET	/categories/{categoryId}/books	Accept: */*	-	Content-Type: application/xml Content-Length: ..	<books><book><title>...</title><authors>...</authors></book></books>
addBook( categoryId: int; book: (title, authors) )	POST	/categories/{categoryId}/books	Content-Type: application/xml Content-Length: ..	<book><title>...</title><authors>...</authors></book>	Content-Location: /books/{newId}	<book><title>...</title><authors>...</authors></book>

3

# Task 3

In the template *Task3-Template.zip* you will find a REST/HTTP Web service operating on a single resource – User = (**int id, string name**).

Extend the service towards management of user bookmarks, with Bookmark = (**int id, string url**).

Following operations should be implemented:

- Read all bookmarks / Search bookmarks by keyword
- Create a bookmark assigned to a user
- Delete a bookmark
- Read bookmarks of a given user

Extend the service and the client from the task 3 with the following functionality:

- Bookmarks should be able to be assigned to Categories = (**int id, string name**).
- All categories can be listed
- Bookmarks of a given category can be listed
- Categories can be searched by a keyword



VSR

Your feedback on today's session:



[mytuc.org/ttbw](http://mytuc.org/ttbw)

# Questions?

[vsr-sse@informatik.tu-chemnitz.de](mailto:vsr-sse@informatik.tu-chemnitz.de)

*VSR.Informatik.TU-Chemnitz.de*