VSR://edu/2020/evs

Communication in Distributed Systems

//////// Design of Distributed Systems ////////////////

WS 2020/2021 – 1. Tutorial

M.Sc. Mahda Noura

VSR.Informatik.TU-Chemnitz.*de*

Form:

- Detail course knowledge
- Discussion
- Task Solving (by YOU, not me)
- Homework (voluntary)

Own computer is a must-have for this course!

News, Materials:

- http://vsr.informatik.tu-chemnitz.de/news/
- http://vsr.informatik.tu-chemnitz.de/edu/2020/evs/
- Opal: Design of Distributed Systems WS20/21 – Tutorial

Contact:

- mahda.noura@informatik.tu-chemnitz.de
- 1/B203

Organization

- Schedules?
  - Wed., 17:15 – 18:45
  - Fri., 09:15 – 10:45
- Who are you?

The final exam will be in written form and consist out of approx. :

- 50% theoretical knowledge from the lecture
- 50% practical tasks similar as done in the tutorial

Our recommendation:

Do the homework assignments

and hand it in via OPAL

We expect that you already have knowledge in:

- HTML
- Javascript Basics
- XML

1 Distributed Systems

„A distributed system is a collection of independent computers that appears to its users as a single coherent system.“

(Andrew S. Tanenbaum and Maarten van Steen)

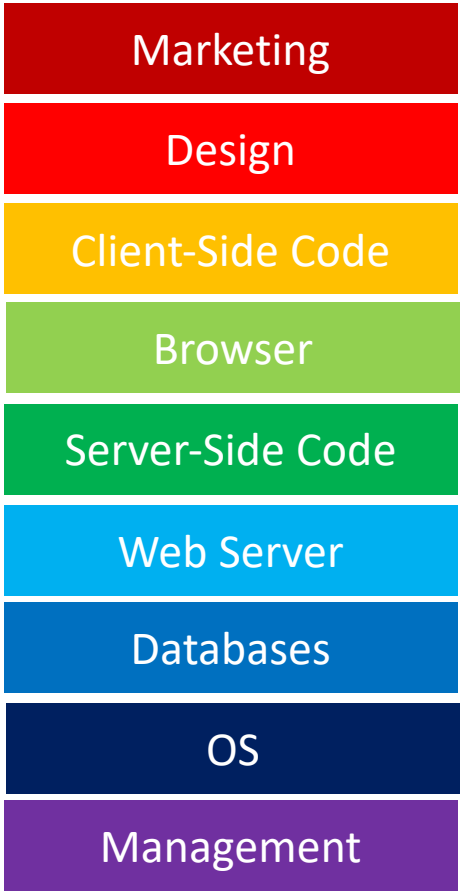
This course mainly focuses on the development of

Web-based systems

2

Full Stack Web Development

In order to build a website, you need knowledge in...



SEO
SEM
Support
UI/UIX Design
Responsive
Photoshop
HTML
CSS
JavaScript
XML
Bootstrap
jQuery
AngularJS
ReactJS
IE, Chrome, Firefox
DOM
AJAX
MobileDev
Webservices
Architecture
OOP
PHP
ASP.NET
Java
Python
nodeJS
Sockets
Load Balancing
Caching
Apache
IIS
Security
Protocols
MySQL
MS SQL
Oracle
NoSQL
Linux
Windows Server
Storage
Routing
DNS
Devices
Tooling
Git / SVN
Documentation

The EVS tutorial will deal with:

- Client-side technologies
- Server-side technologies

In order to build a website, you need knowledge in...

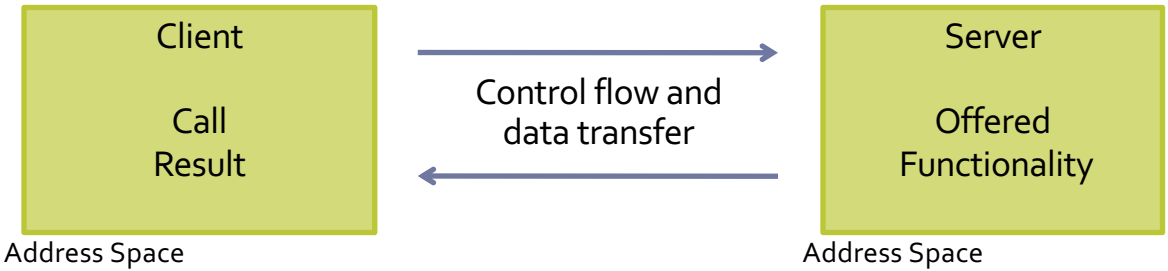
- Marketing
- Design
- Client-Side Code
- Browser
- Server-Side Code
- Web Server
- Databases
- OS
- Management

RPC / Web Services

Distributed system is a loosely coupled set of components, which run on *different computers' host systems* and coordinate by means of message exchange over a communication medium to reach a common goal.

Client/Server Model

- Traditional approach, which is made use of in many others
- Role-based approach
  - Server – Role of a component on the service providers' side
  - Client – Role of a component on the service users' side
- Examples: TCP/IP, Sockets, Web Server

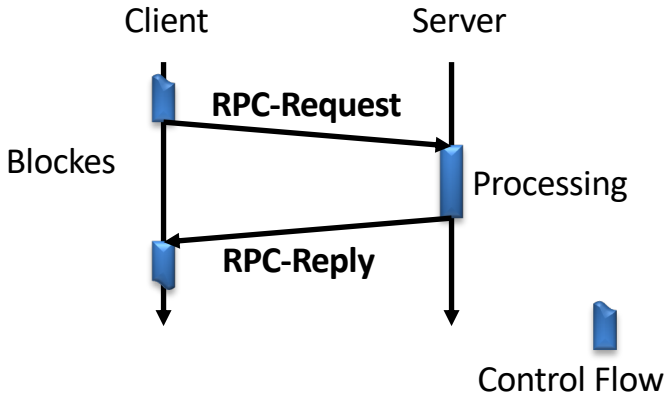


Remote Procedure Call

- Programming language embedding
- Data exchange stays transparent for the programmer
- RPC is located above UDP or TCP in the protocol stack
- Is mostly implemented as a part of the actual application

RPC Execution

Call in waiting state
Parameter- and call transfer to the target system
Procedure execution
Re-registration
Continuation of program execution



Web Services

The communication platform between two different or same platform applications that allows to use their web method

Task 1

Setup

The programming language of this task is C#.

Visual Studio can be downloaded from the DreamSpark-Premium-Portal of TU-Chemnitz.

http://www.tu-chemnitz.de/urz/software/dreamspark.php

Example

Remote call of a “Plus” procedure to retrieve the sum of two numbers

http://vsr-demo.informatik.tu-chemnitz.de/WebServices/PlusMinusService/PlusMinusService.asmx

VSR

mytuc.org/tgxs

Thank You!

mahda.noura@informatik.tu-chemnitz.de

*VSR.Informatik.*TU-Chemnitz.*de*