

## Assignment 1:

### Architectures and System Software for Net-based Applications

- There is no meeting to introduce this assignment, i.e., there is **no meeting on October 18<sup>th</sup>** at 11:30. (But there will be a meeting for the grading of this task; see below.)
- You may work on the assignments in groups of up to three people.
- All members of a group have to show up together for the grading of the assignment. For the grading, each group will need to provide a written document with the answers to the questions, which is collected by the tutor. Each group member will be asked questions about the solution.
- The grading of this exercise will take place in seminar room 0.124 between 11:30 and 13:00 on October 25<sup>th</sup>, 2016.
- If you have questions, send an email to the tutor to make an appointment.

#### Task 1 – Three Tier Architecture

In this task, a suitable architecture for a web-based client/server application shall be designed. The service to be considered is an online shop similar to amazon.com that can be accessed via the World-wide Web (WWW).

Describe a suitable three-tier architecture for this service. Your design should give answers to the following questions:

- Which types of components do you need? Briefly describe their functionality.
- At which tier are these components located according to the three-tier architecture style?
- Describe the relations between your components by describing the typical interactions between the components when a user searches for a certain item (e.g. a book) in the online store.

#### Task 2 – System Software for Net-based Applications

Assume you have to implement a client/server system for printing documents. This system consists of a networked print server that receives commands from applications running on different PCs in the network. Typical commands include sending a file to the print server (print request returning a print job id), deleting a pending print job (delete request), listing the currently pending print jobs at the server (list request).

Compare the complexity of implementing such a service using either a network operating system *without* addition RPC middleware and *with* RPC middleware. In particular, consider the difficulties arising in heterogeneous systems, for instance, a client running on an Intel machine, and a server running on a PowerPC machine (different internal byte-orders). Describe the functionality, the programmer has to implement with and without RPC middleware (in other words: which additional functionality does the RPC middleware provide?).