Component-Based Software Engineering	**Exercise Sheet No. 2**
Dipl.-Medieninf. Christian Piechnick	Software Technology Group
INF 2080	Institute for SMT
christian.piechnick@tu-dresden.de	Department of Computer Science
	Technische Universität Dresden
	01062 Dresden

Implementing Component-Based Systems

Task 1: Transparency Problems

A transparency problem describes software concerns that should be transparent (invisible, hidden) when you write, deploy a component. This task repeats the different kinds of transparency problems.

1a) What can be subject of secret w.r.t. transparency problems of component-based systems?

1b) What aspects of transparancy do you know? How are they aligned with the secret subjects?

1c) What is language transparency and how can it be achieved?

1d) Why is location transparency important? Give an example.

Task 2: Enterprise Java Beans (EJB)

Enterprise Java Beans is a server-side component architecture for building distributed business applications with Java.

2a) Is EJB a composition system?

2b) What is a component in EJB? What component types does EJB support?

2c) What is the EJB composition technique?

2d) What is the EJB composition language?

2e) Which transparency problems does EJB address?

Task 3: Implementation of the Appointment Manager

In the last task you designed a simple appointment manager application for researchers. In this task you are supposed to implement your design in Enterprise Java Beans. Try to implement your design as it is and if it does not fit the properties of the composition system of EJB, please update your design artifacts and try to implement your adjusted design. The result should be a prototype which does not necessarily catches all errors but works for all identified use cases, when a regular input is expected. We will deal with error handling later on.

3a) Implement a prototypical version of the appointment managing application using EJB and the JBoss application server.

3b) Implement a front-end (user interface) that supports ONE of the use-cases and uses the server backend application.

3c) Send your binary solution along together with installation instructions to christian.piechnick(at)tu-dresden.de until July 9th 2015.

3d) Prepare a short presentation and demo (maximum 10 minutes)!