

	<b>Fakultät</b> Elektronik und Informatik	Modulbeschreibung <i>SPO 29 / SoSe</i> <b>2015</b>
	<b>Studiengang</b> Informatik	
	<b>Modulkoordinator</b> Prof. Roy Oberhauser	

Modul-Name		Verteilte Anwendungen				Modul-Nr : 57944	
CP	SWS	Workload	Kontaktzeit	Selbststudium	Angebot Beginn	Sem	Dauer
5	4	150	60	90	<input type="checkbox"/> Wintersemester <input checked="" type="checkbox"/> Sommersemester	7	<input checked="" type="checkbox"/> 1 Semester <input type="checkbox"/> 2 Semester Semester
Angestrebter Abschluss		Modultyp (PM/WPM/WM)		Studienabschnitt		Einsatz in Studiengängen	
Bachelor of Science		PM - Pflichtmodul		HS - Hauptstudium		IN, Schwerpunkt SE	
Form der Wissensvermittlung		<input type="checkbox"/> Vorlesung <input checked="" type="checkbox"/> Übung <input checked="" type="checkbox"/> Labor <input checked="" type="checkbox"/> Selbststudium <input type="checkbox"/> Seminar <input type="checkbox"/> Hausarbeit <input checked="" type="checkbox"/> Projektarbeit <input type="checkbox"/> Sonstiges: Referat, Bericht					
<u>Lernziele / Kompetenzen</u>							
<p><b>Fachkompetenz („Wissen und Verstehen“ und „Fertigkeiten“):</b> Die Studierenden haben fortgeschrittene Softwareengineering Kenntnisse und können diese im Bezug auf die Erstellung von komplexen verteilten Anwendungen einsetzen. Sie können einheitlich und qualitativ verteilte Anwendungen erstellen unter Verwendung von Prinzipien, Mustern, Referenzarchitekturen, und verschiedene aktuellen Middleware, Technologien, Plattformen und Frameworks anhand von gängige Werkzeugen und best Practice Beispielen. Sie lernen aktuellen Middleware Technologien und können diese zweckmäßig anwenden. Studierende planen und führen ein Projekt in Kleingruppen durch, das dann mit Ergebnissen anderer Teams integriert wird und dokumentieren Ihre Ergebnisse. Sie erstellen eines Softwarearchitekturdokuments, das Ihre Softwareanwendung beschreibt.</p> <p><b>Methodenkompetenz:</b> Lernen in ein reales Software-Projekt vorzugehen und es umsetzen.</p> <p><b>Überfachliche Kompetenz („Sozialkompetenz“ und „Selbstständigkeit“):</b> Studierende</p> <ul style="list-style-type: none"> <li>- üben Teamfähigkeit und Selbstorganisation in ein realen Projekt.</li> <li>- üben Kommunikation Anhand Präsentationen und Dokumentation.</li> <li>- recherchieren selbstständig, um die erforderliche Informationen und Ergebnisse für Ihr Projektarbeit zu bekommen.</li> </ul>							
Kompetenzbereich	Schwerpunkt	Teilschwerpunkt		In geringen Anteilen			
Fachkompetenz	<input checked="" type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>			
Methodenkompetenz	<input type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>			
Sozialkompetenz	<input type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>			

<b>Lehrinhalte</b>	
<ul style="list-style-type: none"> <li>- Herausforderungen, Eigenschaften und besondere Merkmale von verteilten Anwendungen</li> <li>- Anwendung von Architektur- und Entwurfsmustern, Plattformen, aktuelle Technologien und Frameworks, und best Practice Beispiele</li> <li>- Vorstellung aktueller Verteilungstechnologien (Middleware, Web Services, Workflow Management Systeme, Cloud Computing, Semantik Computing, Parallel Computing, Agenten Computing, etc.)</li> <li>- Übungen mit aktuellen Technologien</li> <li>- Entwicklungsprojekt einer verteilten Anwendung als Team</li> </ul>	
<b>Zugangsvoraussetzung</b>	<p>Vorbereitung Teilnahme Modul:  Modul: Inhaltlich sollte 57631 Software-Architektur gleichzeitig oder vorher genommen werden. Inhaltlich werden folgende Kenntnisse vorausgesetzt:</p> <ul style="list-style-type: none"> <li>- Software-Engineering (z.B. 57421)</li> <li>- Java und C# Kenntnisse (z.B. 57204, 57429)</li> </ul> <p>Prüfung: Voraussetzung zur Teilnahme an der Prüfung ist ein zugehöriger Übungsschein.</p>

<b>Enthaltene Teilmodule / Lehrveranstaltungen</b>							
Fach-Nr.	Titel des Teilmoduls / Lehrveranstaltung	Lehrende	Art	SWS	CP	Sem	Teilmodulprüfung Art / Dauer / Benotung
57725	Distributed Applications	Prof. Roy Oberhauser	P Ü L	4	5	7	PLP  benotet
	<b>Teilmodultyp (PM/WPM/WM)</b>	<b>Studienabschnitt</b>	<b>Einsatz in Studiengängen</b>				
	PM - Pflichtveranstaltung	HS - Hauptstudium	IN, Schwerpunkt SE				
<b>Zugelassene Hilfsmittel</b>		siehe Projektbeschreibung					

<b>Sprache</b>	<input checked="" type="checkbox"/> Deutsch <input checked="" type="checkbox"/> Englisch <input type="checkbox"/> Spanisch <input type="checkbox"/> Französisch <input type="checkbox"/> Chinesisch <input type="checkbox"/> Portugiesisch <input type="checkbox"/> Russisch
<b>Literatur</b>	<ul style="list-style-type: none"> <li>- Cloud Computing: Concepts, Technology &amp; Architecture by Erl et al.</li> <li>- Cloud Computing Design Patterns by Erl et al.</li> <li>- Grundkurs Verteilte Systeme : Grundlagen und Praxis des Client-Server-Computing von G. Bengel</li> <li>- REST und HTTP: Entwicklung und Integration nach dem Architekturstil des Web von Tilkov et al.</li> <li>- Building Microservices von S. Newman</li> <li>- Microservices: Grundlagen flexibler Softwarearchitekturen von E. Wolff</li> <li>- Advanced Microservices: A Hands-on Approach to Microservice Infrastructure and Tooling by T. Hunter</li> <li>- Microservice Architecture: Aligning Principles, Practices, and Culture by I. Nadareishvili et al.</li> <li>- Docker Up &amp; Running von K. Matthias et al.</li> <li>- Docker: Software entwickeln und deployen mit Containern von A. Mouat</li> <li>- Docker: Praxiseinstieg von Matthias &amp; Kane</li> <li>- Amazon Web Services in Action by Wittig &amp; Wittig</li> <li>- Professionell entwickeln mit Java EE 8: Das umfassende Handbuch von A. Salvanos</li> <li>- Skalierbare Container-Infrastrukturen: Das Handbuch für Administratoren und DevOps-Teams. Inkl. Container-Orchestrierung mit Docker, Rocket, Kubernetes, Rancher &amp; Co. von O. Liebel</li> <li>- Serverless Architectures on AWS: With examples using AWS Lambda by P. Sbarski</li> <li>- An Introduction to MultiAgent Systems by Wooldridge</li> <li>- Cloud Architecture Patterns von B. Wilder</li> <li>- Mastering bitcoin: programming the Open Blockchain by A. Antonopoulos</li> <li>- Mastering Blockchain: Distributed ledgers, decentralization and smart contracts explained by I. Bashir</li> <li>- Spring Boot Messaging: Messaging APIs for Enterprise and Integration Solutions by F. Gutierrez</li> <li>- Spring Microservices: Build scalable microservices with Spring, Docker, and Mesos by RV</li> <li>- Reactive messaging patterns with the Actor model: applications and integration in Scala and Akka</li> <li>- Storm Blueprints: Patterns for Distributed Real-time Computation by Goetz &amp; O' Neill</li> <li>- Developing RESTful Services with JAX-RS 2.0, WebSockets, and JSON by Kalali &amp; Mehta</li> <li>- - WebSockets: Moderne HTML5-Echtzeitanwendungen entwickeln von Gorski et al.</li> <li>- Softwarearchitekturen dokumentieren und kommunizieren von S. Zörner</li> <li>- Distributed and Cloud Computing: From Parallel Processing to the Internet of Things by Hwang and Dongarra</li> </ul>
<b>Zusammensetzung der Endnote</b>	
<b>Bemerkungen / Sonstiges</b>	---
<b>Letzte Aktualisierung</b>	2018-03-02 RO