



Integration Architectures

Chapter 1: Introduction and Organization, First View on Integration Architecture

**A lecture for the study program Bachelor Computer Science,
based in the specialization “Komplexe Software-Systeme”**

Wintersemester (Winter Term) 2024 / 2025

Prof. Dr. Sascha Alda
(sascha.alda@h-brs.de)

Vorstellung Prof. Dr. Sascha Alda



Professor für **Software-Architektur** am Fachbereich Informatik (FB02)
der Hochschule Bonn-Rhein-Sieg. Dekan des Fachbereichs.

- Raum: C 103 (Dekanat)
- (nur selten) C 218
- Telefon: 02241 – 865 – 201
- E-Mail:
sascha.alda@h-brs.de
- Web:
<http://www.sascha-alda.de>
- Offene Sprechstunde:
Do., 12:00 -13:00 Uhr



Finisher beim Bonn-Halbmarathon
(14. April 2024)



Besuch beim BSI am 26. Juli 2024
(mit Hartmut Ihne und Claudia Plattner)


Some general details on this course



- A **specialization course** for the study programs Bachelor Computer Science (BI) and Bachelor Information System (BWI), based in the specialization “Complex Software Development”. Open also for exchange students from Partner Universities.
- Date: Tuesdays (starting from October, 8th 2024) in C 117. The whole lecture will be recorded and published on my YouTube channel.
- A **regular participation** in the whole course (lecture and exercise) is expected! No compulsory attendance, however!
- In the lecture, the **primary language of instruction** is English 😊
 - Feel free to ask questions or give contributions in English 😊!
 - Face-to-Face discussions e.g., in exercise courses will be done in German or English.
 - All material (slides, exercise sheets etc.) will be provided in English

Some general details on this course – further contact person



- Luca Ringhausen, B. Sc. Computer Science
Research Assistant (CS department; Project DIArchitect).
 - Contact hours:
 - Tuesday, 15:00 - 16:30 (not on holidays)
 - Location: according to prior agreement (C 164 or remote)
 - E-Mail: luca.ringhausen@smail.inf.h-brs.de
 - Topics: JavaScript, Node.js, MEAN, CI/CD. Technical questions concerning the semester project, DIArchitect
- 
- The project **DIArchitect** aims at providing a digital platform for students to attend to the course „Integration Architectures“ remotely.
 - Use also the **forum of the LEA** page for discussing questions:
 - https://lea.hochschule-bonn-rhein-sieg.de/ilias.php?ref_id=1620530&cmd=showThreads&cmdClass=ilrepositorygui&cmdNode=y1&baseClass=ilRepositoryGUI
 - Another forum is implemented in DIArchitect (should be mainly used)

Organization of the Exercise courses



- The lecture course will be accompanied by an **exercise course**, which is used to deepen the material that is taught during the lecture courses
- Despite of the formal separation of the modules, there will **mostly be mixed modules**
 - Example: 30 min lecture, break, 30 min exercise, 30 min lecture etc.
 - Occasionally, both lecture and exercise course will be used as a working workshop
- 13 sessions (until the end / middle of January 2025). Time schedule:

Type of course	Date	Video-Recording	Room for on-site lecture
Lecture	Tuesdays, 17:00 – 18:30	Yes, full	C 117
Exercise	Tuesdays, 18:45 – 20:15	Yes, partial	C 117

- No lecture on October, 29th 2024.
- Access Data for WebEx will be announced in an urgent case.



- For some exercise courses, students need to work on **exercise sheets** containing several assignments that are part of a continuous semester project (see later)
 - Exercise courses will be used to actively work on the assignments, yielding to first preliminary results
 - The completion of the assignments must be done by the students at home
 - There are six exercise sheets.
- **Publication date** of the exercise sheets is Tuesday 15:00 (3:00 pm) prior to the lecture. Location:
 - DIArchitect server (assignments); LEA server will be used in the beginning, occasionally
- Submission date of the assignments: ca. 1-3 weeks time for preparation
 - Submission my means of an upload on the DIArchitect server
 - Submission can be done **on your own or in teams of (up to) three students** (no more!)
 - Formats depending on the assignment: PDF (for models, text etc.), raw texts (source codes); perhaps link to your public Git-Repro
 - Solutions will be commented and discussed in the exercises

Semester project „HighPerformance“



- Goal of the project:

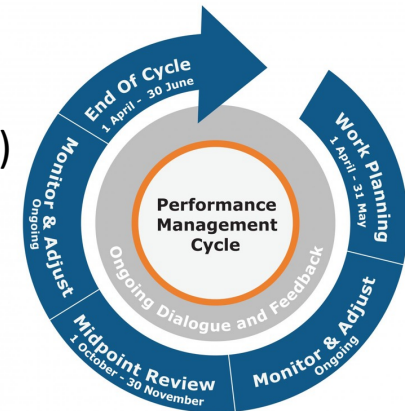
Automation of the performance evaluation for salesmen working at the company SmartHoover Ltd.
(Project „HighPerformance“)

- Basic conditions:

- Integration of given systems for maintaining employees (OrangeHRM, Odoo) as well as for management of both clients and orders (OpenCRX)
- Implementation of a performance cockpit based on the **MEAN Stack**

- Concrete Tasks:

- **Analysis** of a given software architecture (IST Architecture)
- Understanding the problems of the actual process; **Design** a new one (target process)
- Development of a new software architecture for the **Integration** of the existing software
- **Implementation** a prototype of Performance Cockpit for HR personal
- **Continuous Delivery** of the final software on the SEPP CI / CD pipeline



- The semester project is **continuously driven by the exercise sheets**

Foto: <https://hr.un.org/page/performance-management-home>

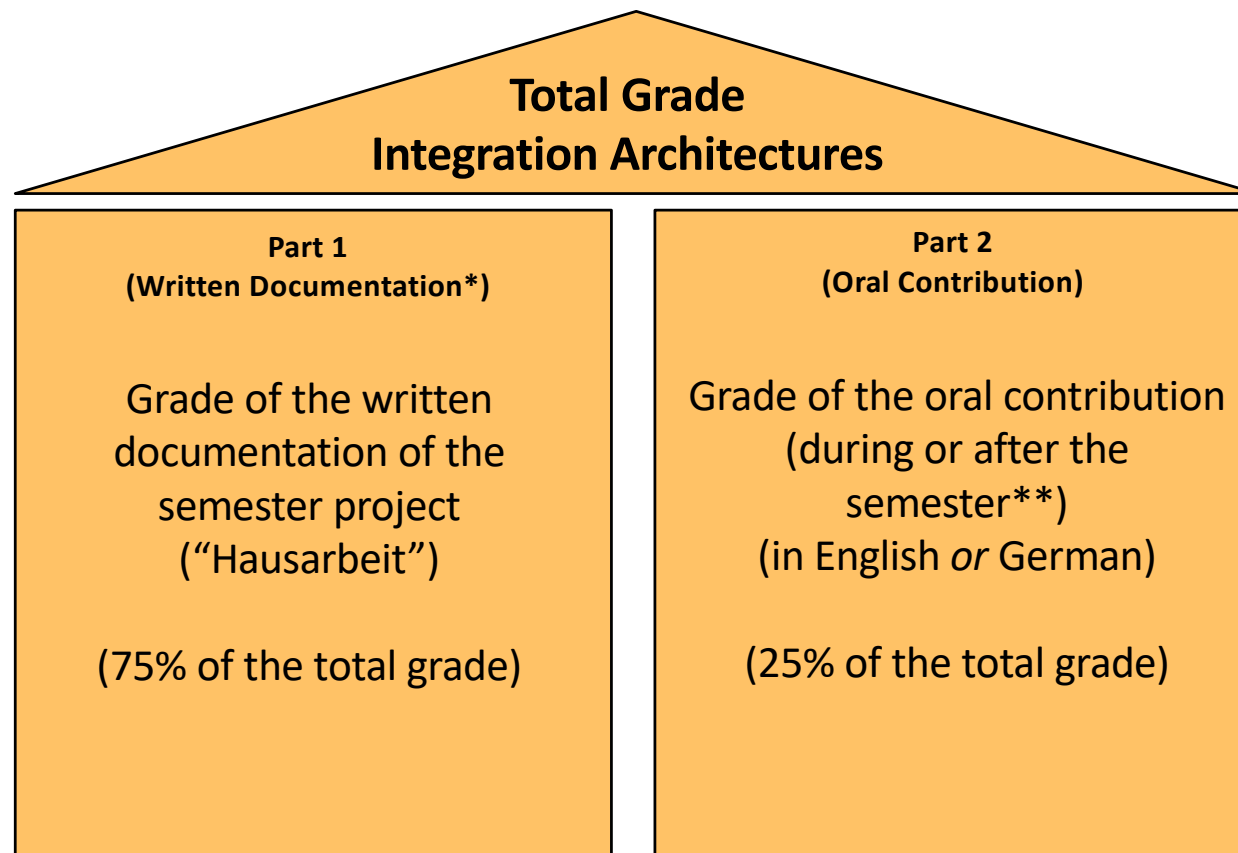


- Prerequisites for getting the 6 ECTS-Points: passing a **written documentation** of the semester project including **a short demo of the prototype**.
 - Due Date: February, 15th 2025. Short demo: according to plan in February / March 2025
 - Teamwork is possible and highly recommended (max. 2-3 students per team)
 - Duration: approx. 10-15 min *in total*.
- Also, an **oral contribution** must be made. There are two ways:
 - Presentation and oral discussion of a solution of an exercise sheet (at least one assignment) during the semester (single or team)
 - Elaborated presentation and oral discussion of the semester project after the semester (single or team, will take place in February / March 2025).
 - Duration: approx. 10-15 min *per student*
- **Criteria for receiving** the admission for the final exam (“Zulassungskriterien”):
 - 1/3 of the assignments must be passed (i.e., the evaluation must be positive)
- The oral contribution can be performed in English *or* in German language. Also, the written documentation can be prepared in English *or* in German.

Composition of the total grade



- The final total grade will be composed of two parts:



Both components must be passed with a grade better than 4,0!

* Re-Take of the written documentation: an oral exam of 20-25 minutes in the summer term (grade of the oral contribution valid for one year (including SS 2025))

** Single presentation: 10 minutes. Team presentation: approx. 10 min per student

What are the goals of this lecture?



- Learn basic and advanced concepts of **modern software architectures**
- **Learn and apply architectural integration principles and solutions**
- Apply your knowledge and competencies to a complex **real life case study based on enterprise application systems**
- Deepen your knowledge in programming by learning the language **JavaScript**
- Learn and apply the basic and advanced principles of the **MEAN stack**
- Learn and apply basic principles of *interoperable* interface design based on **REST**
- Intensify your knowledge in **software integration testing**
- Intensify your knowledge on modern **CI / CD pipelines**
- Improve your **English skills**, of course!

Prerequisites for the module



- A strict prerequisite that is checked during the first two weeks:
 - Students **MUST have passed the lecture Software Engineering (SE) 1 successfully!**
 - Foreign students: please prove the passing of an equivalent module from your home University
- Further recommended lectures that **SHOULD** have been passed (no proof necessary)
 - Introduction to Programming (1 and 2)
 - Introduction to Databases
 - Web Engineering (at least some knowledge, e.g., from SE-2 (Prof. Alda))
 - Software Engineering 2 (well, some content will be used, which will be introduced here)
- Further required competencies in the beginning (not all are necessary):
 - Java (SE), maybe also a bit JavaScript
 - Object-oriented principles (design patterns)
 - Basic knowledge on software architectures
 - Case Tools (especially an IDE like IntelliJ or Eclipse)
 - Basic understanding on User Interface development
 - Some (Business) English

My expectations to the participating students



- **Attention** and motivation within all the courses during the complete term!
- **Active** participation (ask questions, make comments, present solutions)
- Be ready to independently introduce yourself in new tool and frameworks
- A **deep** interest in complex software development **MUST** be given (there will be a lot of programming!)



Active participation
(the guy in the blue shirt ;-))

I ask all students to show both an appropriate **social behavior** and a **professional attitude** in both lecture and exercise in the course
Integration Architectures!



- Hoppe, G. and Woolf, B.: Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions. Addison-Wesley, 2004
- Bass, L., Clements, P., and Kazman, R.: Software Architecture in Practice. Addison-Wesley, 4rd edition. 2021.
- Starke, G.: Effektive Software-Architekturen. Hanser, 10. Auflage, 2024
- Takai, D.: Architektur für Websysteme. Hanser, 2017.
- Spichale, Kai.: API-Design – Praxishandbuch für Java und Webservice-Entwickler. Dpunkt-Verlag. 2017

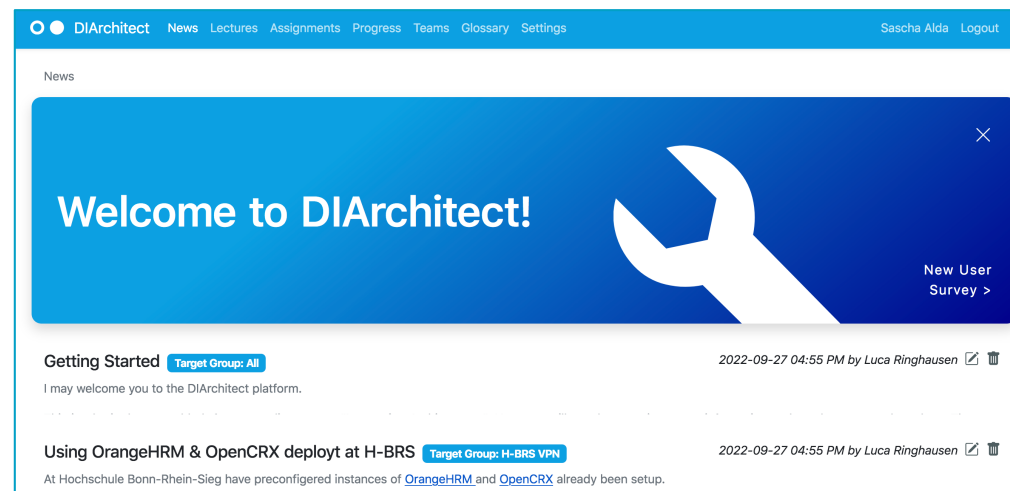


Further resources will be recommended in the lecture



- All resources for the lecture (e.g., slides) can be downloaded from DIArchitect:

<https://diarchitect.inf.h-brs.de>



- In the beginning, as a backup, all resources can also be gathered from LEA.
- The slides will be uploaded prior to lecture on Tuesday, ca. 16:58 am ;-)
- Usually a pre-final version, minor errors will be corrected afterwards...
- Source codes and basic frameworks will be pushed to a GitHub repository (tba)
- There will be an introduction to DIArchitect on October 15th, 2024!

Structure of the lecture



Chapter	Topic	
1	Introduction and Organization, First View on Integration Architecture	
2	Introduction to Software Architectures and Architectural Integration	
3	Design and Development of interoperable Interfaces with Techniques and Solutions from State-of-the-Art (REST, Node.js)	
4	A deeper look on Architectural Integration: Basic Techniques, Patterns, and Methods	
5	A further look on Architectural Integration: Practical Solutions from State-of-the-Art (Kafka, Camunda, Mule)	
6	Patterns and Tools for Interface Integration Testing	
	Guest lectures: January 14 th , 2025: SAP Lean-IX (Jan Nonnen): “Event-based integration of business applications”	

Student participation time!!



- Now, the students should introduce themselves 😊
- Please indicate (in German *or* English):
 - Who are you?
 - What is your major? (Hauptfach)?
 - What are your personal **goals** for the lecture?
- Students / the teams will also have a look on the first exercise sheets. Make yourself familiar with the semester project (→ assignment 1.1)





Chapter 1: Introduction and Organization, First View on Integration Architecture

1	Organizational Aspects, Introduction of Students	✓
2	First View on Integration Architecture	
3	Conclusion	



- So, a lot of definitions on the term “integration architecture” can be found (e.g., (Luisi, 2014), (Murer et al., 2011))
- ... Also, for the bit old term „Enterprise Application Integration“ (EAI) (e.g., Hohpe and Woolf, 2004)
- But let’s start we a novel definition:

An integration architecture defines **global design decisions** for the integration of potentially *distributed* and *heterogeneous* software components. An integration architecture also makes **implications for tools and frameworks** for developing, testing, and maintaining *interoperable* software. (Alda, 2019)

The corresponding **method** for applying those tools and frameworks is referred as **architectural integration**. (Alda, 2019)



The architecture of a system is the **shape** given to that system by those who build it. The form of that shape is the division of that system into components, the **arrangement of those components**, and the **ways** in which those **components communicate** with each other. (Martin, 2017)

- Integration is the **basis** for component communication! (Hohpe and, Woolf, 2004)!
- The arrangement of component can be mastered by the application of architectural patterns (Fowler, 2012)
- A common architectural pattern is called **layer pattern** (Evans, 2004)
- Let's discuss, how communication (and, thus, integration!) takes place on a layered software system...
 - and at last, let's see, where this MEAN stack might enter the scene ;-)



Chapter 1: Introduction and Organization, First View on Integration Architecture

1	Organizational Aspects, Introduction of Students	✓
2	First View on Integration Architecture	✓
3	Conclusion	



- This chapter gave you a first understanding about the needs to integrate software components
- Yet, software integration of heterogeneous software components is one of the **biggest challenges** in software projects in the industry!



- Murer S., Bonati B., Furrer F.J.: "Integration Architecture", in: Murer, S. and Bonati, B.: Managed Evolution- A Strategy for Very Large Information Systems. Springer, Berlin, Heidelberg. 2011.
- Luisi, J.V.: Pragmatic Enterprise Architecture - Strategies to Transform Information Systems in the Era of Big Data. Morgan Kaufmann, 2014.
- Martin, R.C.: Clean Architecture: A Craftsman's Guide to Software Structure and Design. Prentice Hall, First Edition. 2017.
- Fowler, M.: Patterns of Enterprise Application Architecture: Pattern Enterprise Application Architecture. Addison-Wesley, First Edition. 2012.
- Evans, E.: Domain-Driven Design: Tackling Complexity in the Heart of Software. Addison-Wesley, 2004.