Bonn-Rhein-Sieg
University of Applied Sciences

Software Architectures – Perspectives on Research and Education

Facultat d'Informatica de Barcelona (FIB)
Universitat Politecnica de Catalunya (UPC)

Barcelona, March, 13th 2013

Prof. Dr. Sascha Alda

Bonn-Rhein-Sieg University of Applied Sciences

(sascha.alda@h-brs.de)

Presentation Prof. Dr. Sascha Alda

Prof. Dr. Sascha Alda

Bonn-Rhein-Sieg University of
Applied Sciences

Grantham-Allee 20
53754 Sankt Augustin, Germany

Phone:+49-2241-865-760
Mail: sascha.alda@h-brs.de

Marathon Bonn, April 2012

Barcelona Marathon (March 2013)

A short CV:

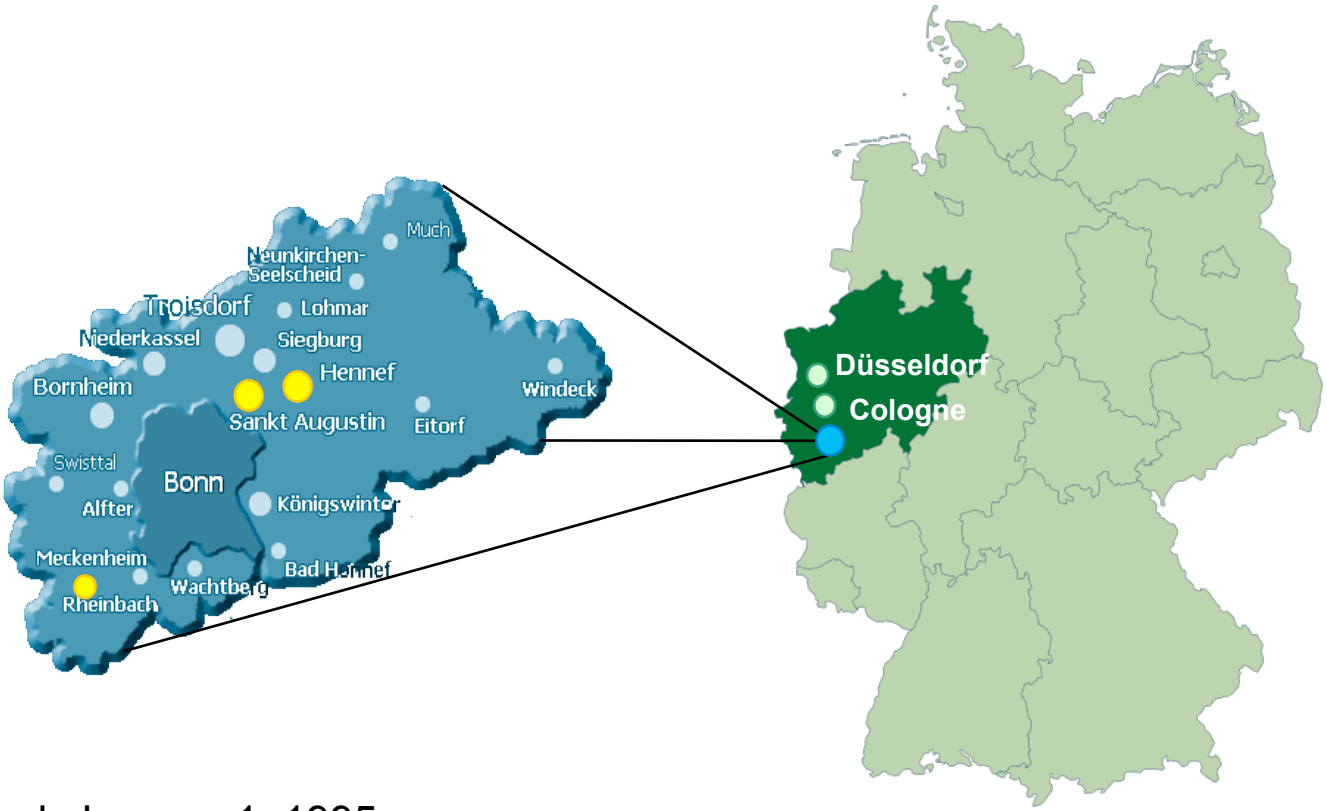
- Professor of Computer Science, University of Applied Sciences, Bonn since 2010.
- Main areas of interests: software engineering, *software architecture*, and modern database systems.
- Three years of Industrial experience (IBM and Accenture).
- Doctoral degree, University of Bonn (2006).
- Diploma in computer science, University of Koblenz (2000).

Structure of the Presentation

Software Architectures – Perspectives on Research and Education		
1	Introduction to the Speaker	✓
2	**The Bonn-Rhein-Sieg University of Applied Sciences**	
3	Software Architecture – Some Assumptions	
4	Software Architecture – Perspectives on Research	
5	Software Architecture – Perspectives on Education	
6	Discussion	

Our University – Some basic Information

Bonn-Rhein-Sieg University of Applied Sciences
Our location in Germany



- Founded: January 1, 1995
- Compensation means for the move of the capital from Bonn to Berlin
- Study programs are offered on three campuses

Our University – Facts and Figures

- **More than 6.500 Students**
- 139 Professors

- Language Training Center
- University and County Library
- International Office
- Graduate Institute

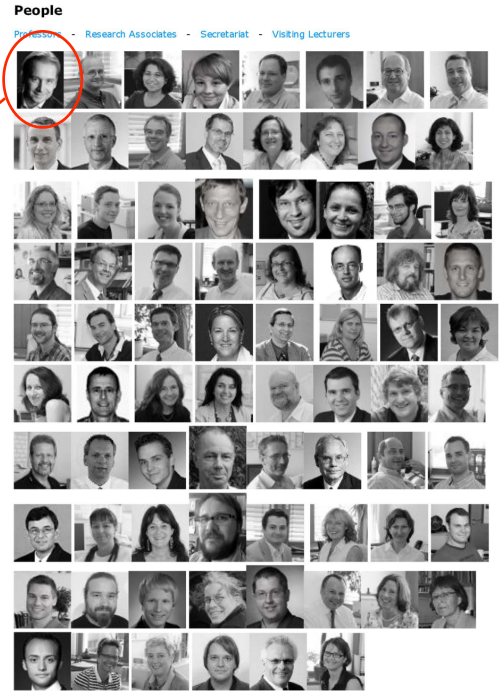
- **The study programs (overview):**
  - Business Administration
  - **Applied Computer Science**
  - Electrical and Mechanical Engineering
  - Applied Biology and Chemistry
  - Technical Journalism

Main Campus in Bonn (Sankt Augustin)

Further Campus in Bonn (Rheinbach)

Department of Computer Science – Facts and Figures

- **> 1.500 students** (ca. 180 graduates per year)
- 31 professors fulltime
- ca. 45 scientific staff
- Research areas and centers:
  - Robotics & Autonomous Systems
  - Computer Vision & Graphics
  - Information Security
  - Business Information Systems
  - **Software Engineering (5 Professors)**
  - Biomedical Computing …. and many more

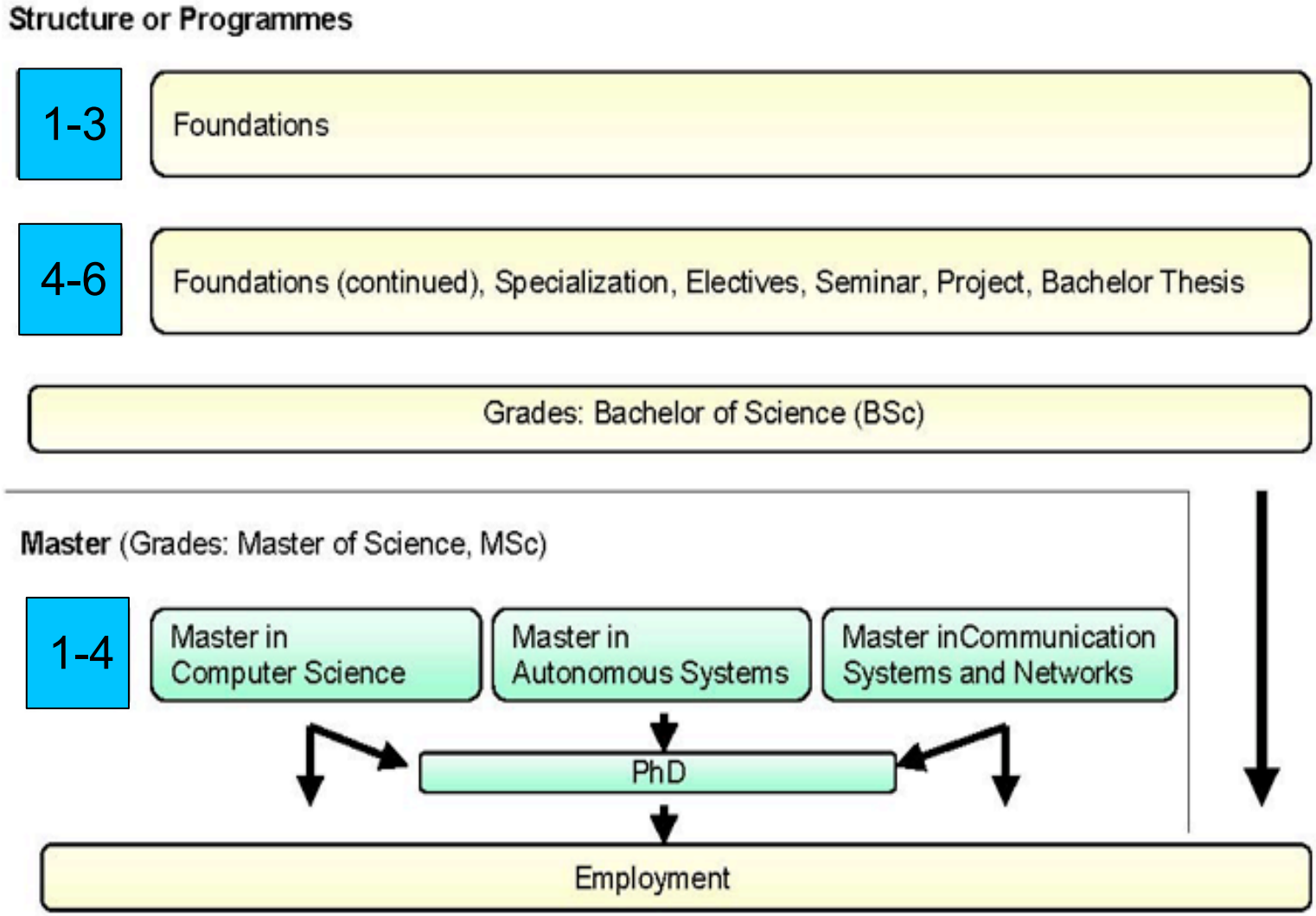


Some collegues

- **Interdisciplinary** with industrial partners (Post, Telekom)
- **Internationalization** (study programs in English, exchange)

…and Johnny
(1st place in the RoboCup@Home)

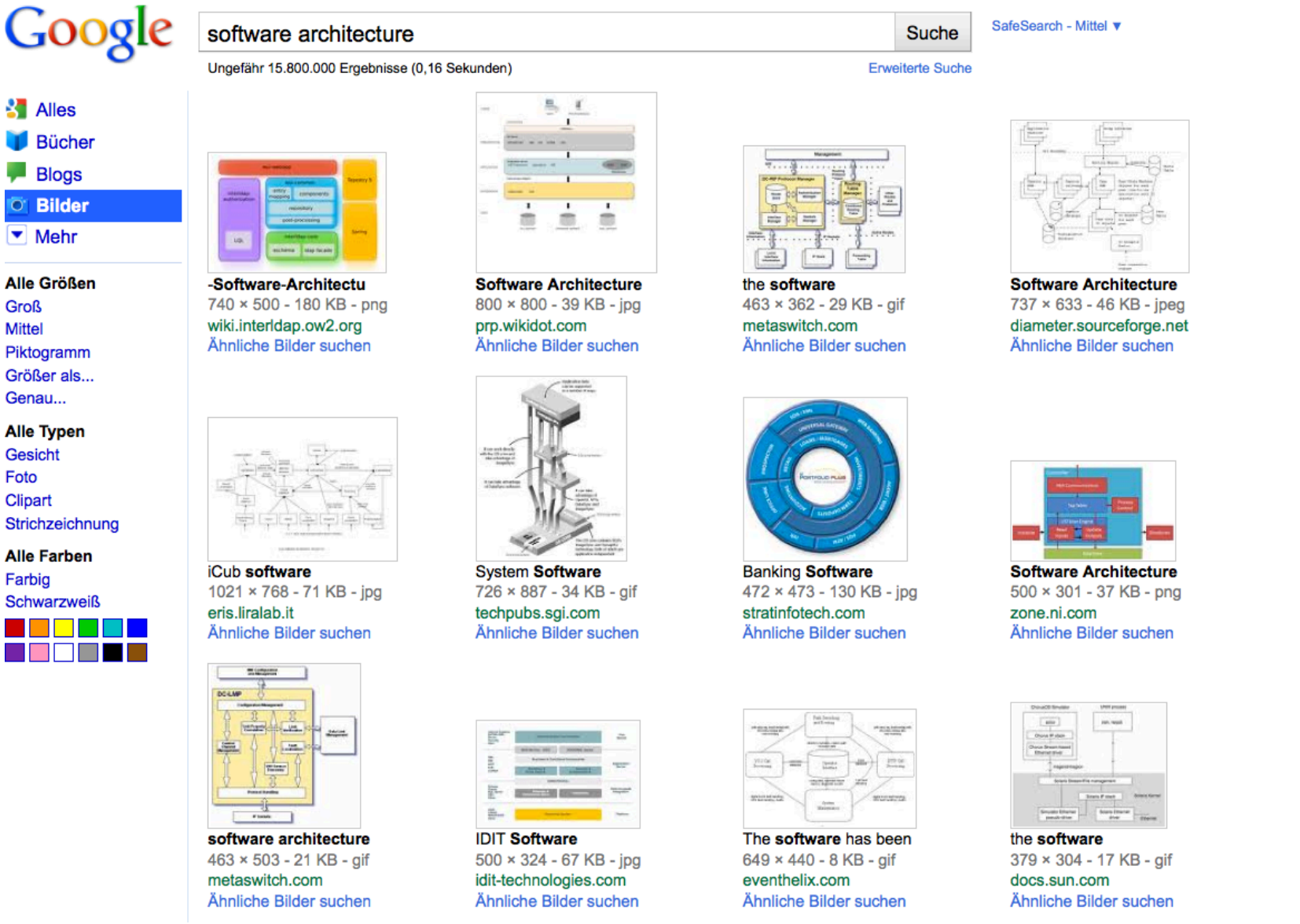
Structure of our program (Computer Sc.)



Structure of the Presentation

Software Architectures – Perspectives on Research and Education		
1	Introduction to the Speaker	✓
2	The Bonn-Rhein-Sieg University of Applied Sciences	✓
3	**Software Architecture – Some Assumptions**	
4	Software Architecture – Perspectives on Research	
5	Software Architecture – Perspectives on Education	
6	Discussion	

What is a Software Architecture?

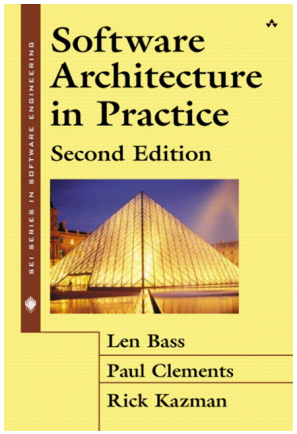


Some Definitions of a Software Architecture from Literature

- „Architecture is the fundamental organization of a system embodied in its components, their relationships to each other and to the environment, and the principles guiding its design and evolution.“

- “The software architecture of a program or computing system is the structure or structures of the system, which comprises:
  - software elements
  - the externally visible properties of those elements,
  - and the relationships among them”

  [Bass et al., 2003]



- Fundamental work has been carried out by the SEI (Software Engineering Institute) at Carnegie Mellon:

  http://www.sei.cmu.edu/architecture/

Zusammenfassende Definition Software-Architektur

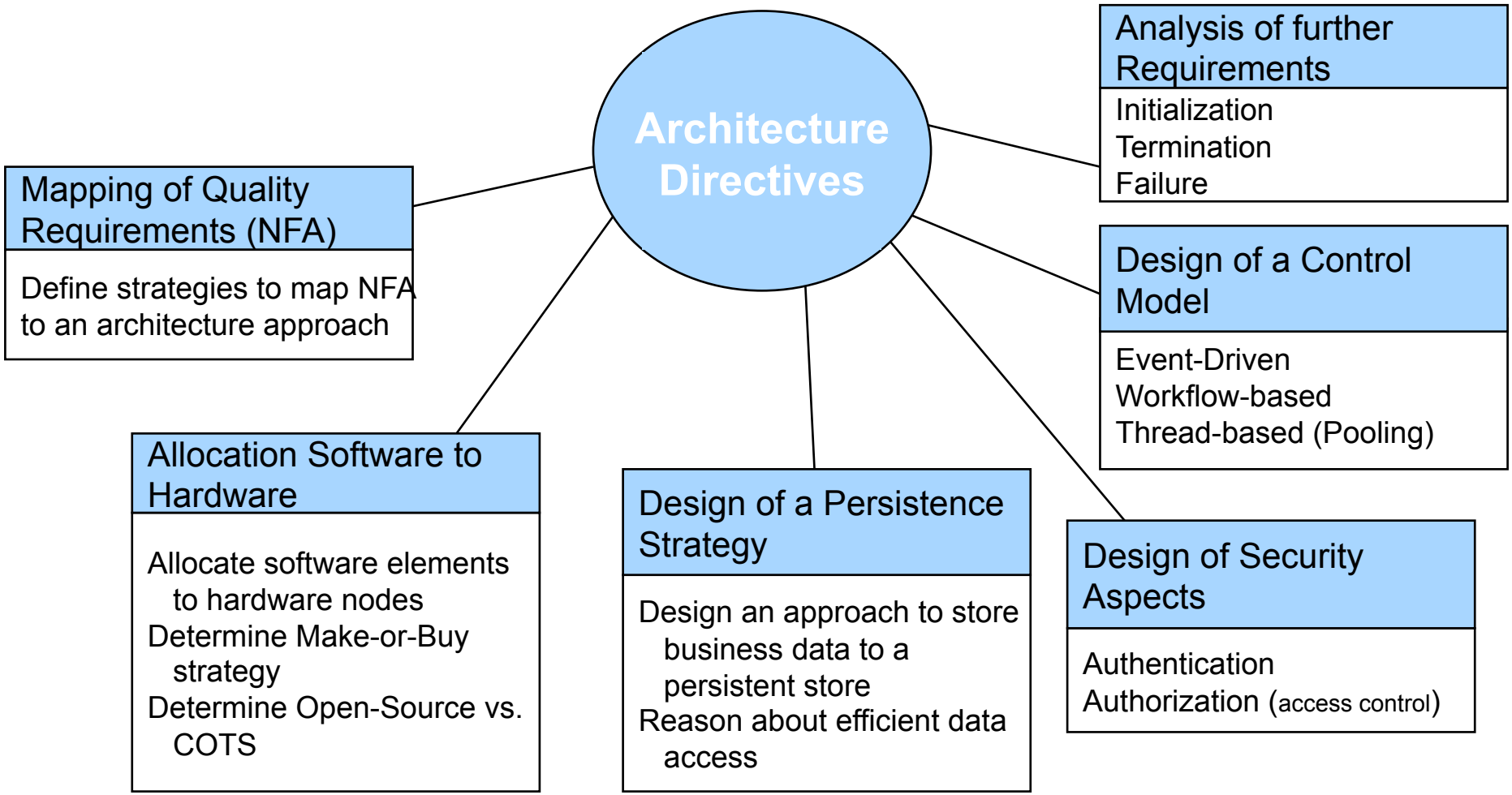
- A software architecture describes the decomposition of a software system that follows the specification of a pertaining architectural style. The description of a software architecture comprises the following constituting elements:
  - the fundamental architecture elements and their interfaces,
  - the interaction relationships among those architecture elements

  as well as
  - the architecture directives, and
  - the characteristic factors

  of the whole software architecture.

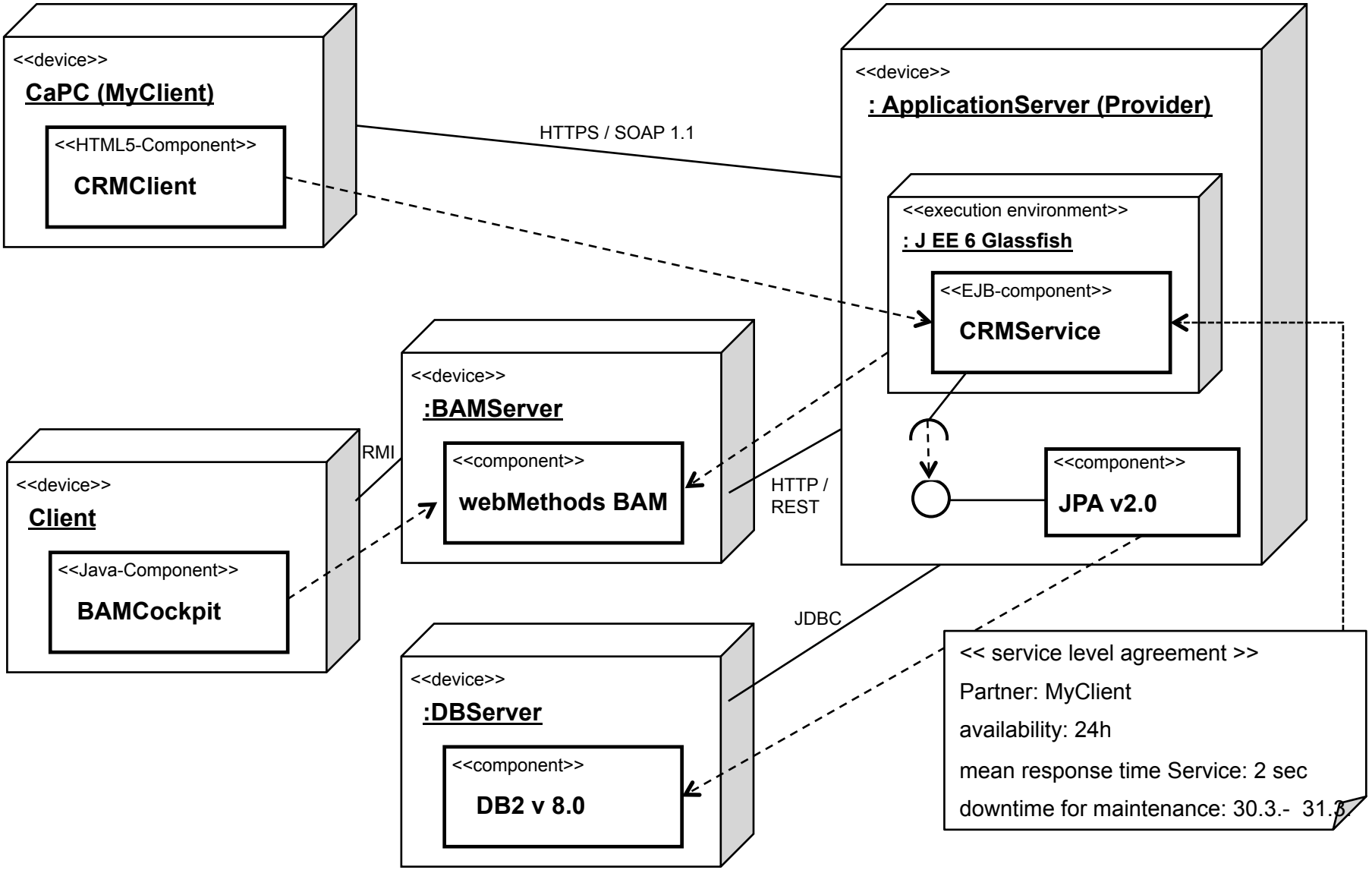
Architecture Directives – Typical Cases

- Architecture directives are typically derived by the following activities:

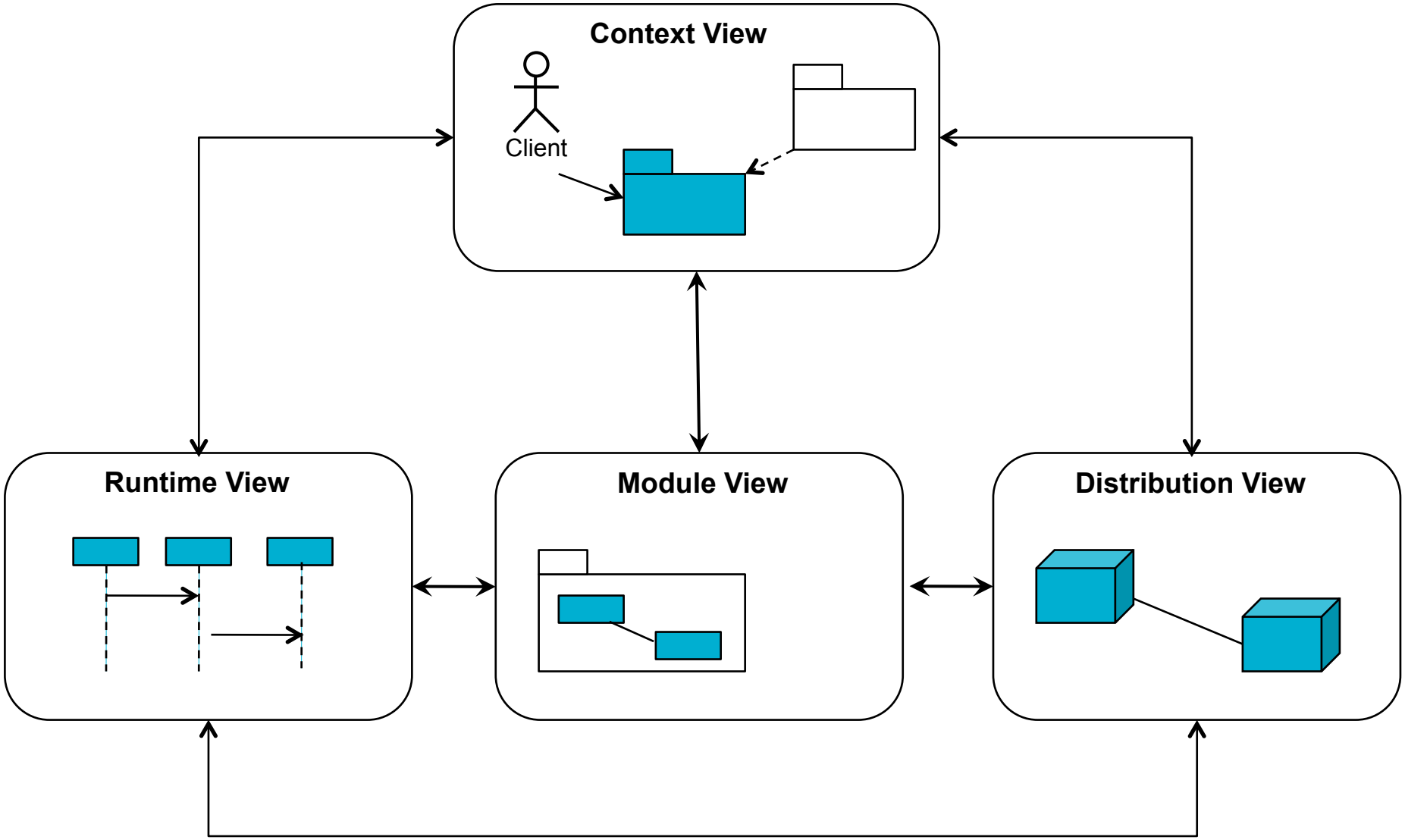


w.r.t. [Brügge and Dutoit, 2004]

Documentation of Software Architectures (UML)



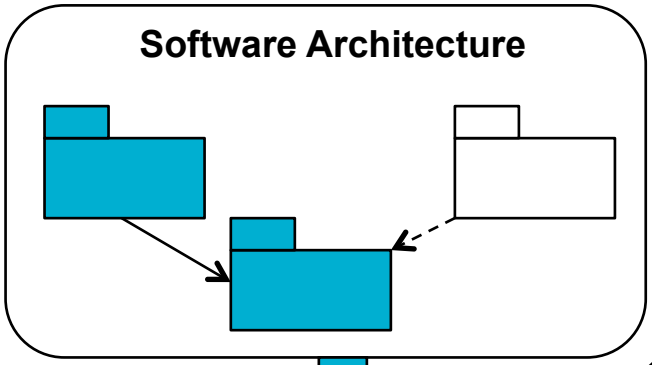
Documentation: The 4 Views Model (Starke, 2011)



Structure of the Presentation

Software Architectures – Perspectives on Research and Education		
1	Introduction to the Speaker	✓
2	The Bonn-Rhein-Sieg University of Applied Sciences	✓
3	Software Architecture – Some Assumptions	✓
4	**Software Architecture – Perspectives on Research**	
5	Software Architecture – Perspectives on Education	
6	Discussion	

Research Projects motivated by three Questions



What are Application Scenarios of *modern* Software Architectures?

Project
ProSyWis
(2013-2016+)

What are the Risks within a Software Architecture?

Project
MoRE Architecture
(since 2011)

How to compose the architectural elements in a Software Architecture?

Project
BUCO
(since 2012)

Structure of the Presentation

Software Architectures – Perspectives on Research and Education		
1	Introduction to the Speaker	✓
2	The Bonn-Rhein-Sieg University of Applied Sciences	✓
3	Software Architecture – Some Assumptions	✓
4	**Software Architecture – Perspectives on Research**	
	4.1 – Project ProSyWis	
	4.2 – Project BUCO	
5	Software Architecture – Perspectives on Education	
6	Discussion	

Project ProSyWis – An Overview

- ProSyWis
  - Prozessunterstützende Systeme für Wissensarbeiter
  - (in English: **Process-based System for Knowledge Workers**)

- Principle Participants
  - Bonn-Rhein-Sieg University of Applied Sciences, Department of Computer Science
  - University of Magdeburg, Department of Knowledge and Language Engineering
  - Software AG, Darmstadt, Germany

- Project Lead
  - Prof. Dr. Rüdiger Buck-Emden (Main Project Lead)
  - Prof. Dr. Sascha Alda (Technical Project Lead)

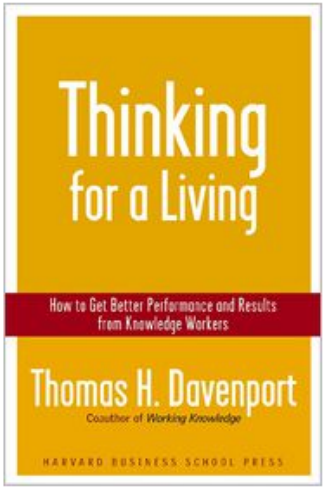
- Project Status
  - Preliminary projects and tasks are running
  - Research proposal is under review (Ministry of Research and Education, Germany, Program “FHProfFund 2013”)

Knowledge Worker (Davenport 2005)

- „Knowledge workers have high degrees of expertise, education, or experience, and the primary purpose of their jobs involves the creation, distribution, or application of knowledge."

- "Companies with a high proportion of knowledge workers […] are the fastest growing and most successful companies in […] leading economies."

- Typical knowledge workers are:
  - Scientists
  - Lawyers
  - Consultant
  - Doctors
  - Decision makers in general... and many more



- Knowledge workers work on individual and complex cases (e.g. a law case)
- Exact replication of workflows are uncommon (no established structure)
- Context-based information, decisions, and collaborations are necessary
- The result and the sequence of work can seldom be foreseen

Our Observation

Conclusion:

- Effective and efficient knowledge work is a critical success factor in our current and future work society

However, there are problems and limitations:

- The sequence of work of knowledge workers is insufficiently supported by modern information technology (IT)
- Yet, E-Mail and telephone constitutes the most relevant tool work-sharing knowledge work

- Workflow-Management and Enterprise Resource Planning Systems are too restrictive and cannot provide adequate support for flexible knowledge work
- Tools of other fields (such as CSCW) are well-known, but hardly adopted in Industry and integrated with dominant systems (i.e. WFMS and ERP)

Project ProSyWis– Goals and Vision

Goal:

- Overcome the existing restrictions to support knowledge workers
- Context-based information will be derived from previous knowledge (closed cases) and is based on current cases and events
- Representation of cases will be funded on a semi-formal meta model.
- Extension of an existing Workflow Management System (aka BPMS) towards an adaptive platform that supports less structured, dynamic, and knowledge intensive processes

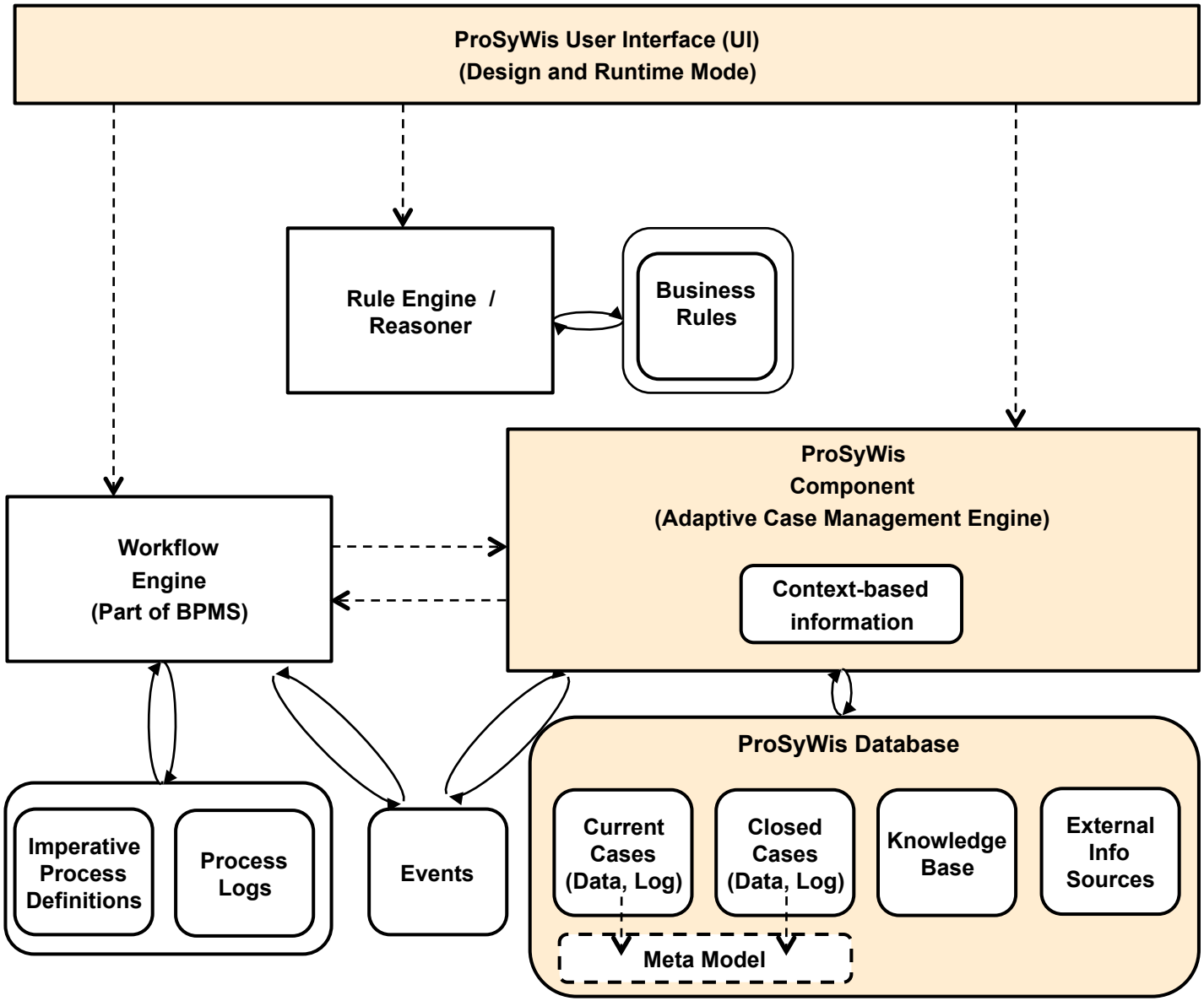
Vision:

- ProSyWis will provide an integrated working environment for knowledge workers through the combination of diverse fields from computer science:
  - Service-Oriented Architecture (SOA)
  - Adaptive Case Management (ACM)
  - Collaboration Tools (CSCW)
  - Knowledge Management
  - Complex Event Processing (CEP)
  - Case-Based Reasoning

Research Questions / Working packages

- What are the *real* functional requirements of knowledge workers from industry for an IT-based system to support their work?
- What is the structure of the resulting software architecture?
- How can existing WFMS be extended to support the structured and unstructured activities of knowledge workers?
  - Additive Component? (→ see First Draft)
  - Concrete Extension?
  - Common Notation (e.g. based on BPMN 2.0 or BPEL)?
  - Design Pattern?
- What are the next steps (recommended procedures) for the provision of ProSyWis as a Cloud (SaaS) solution?
  - Software Architecture?
  - Business Case?

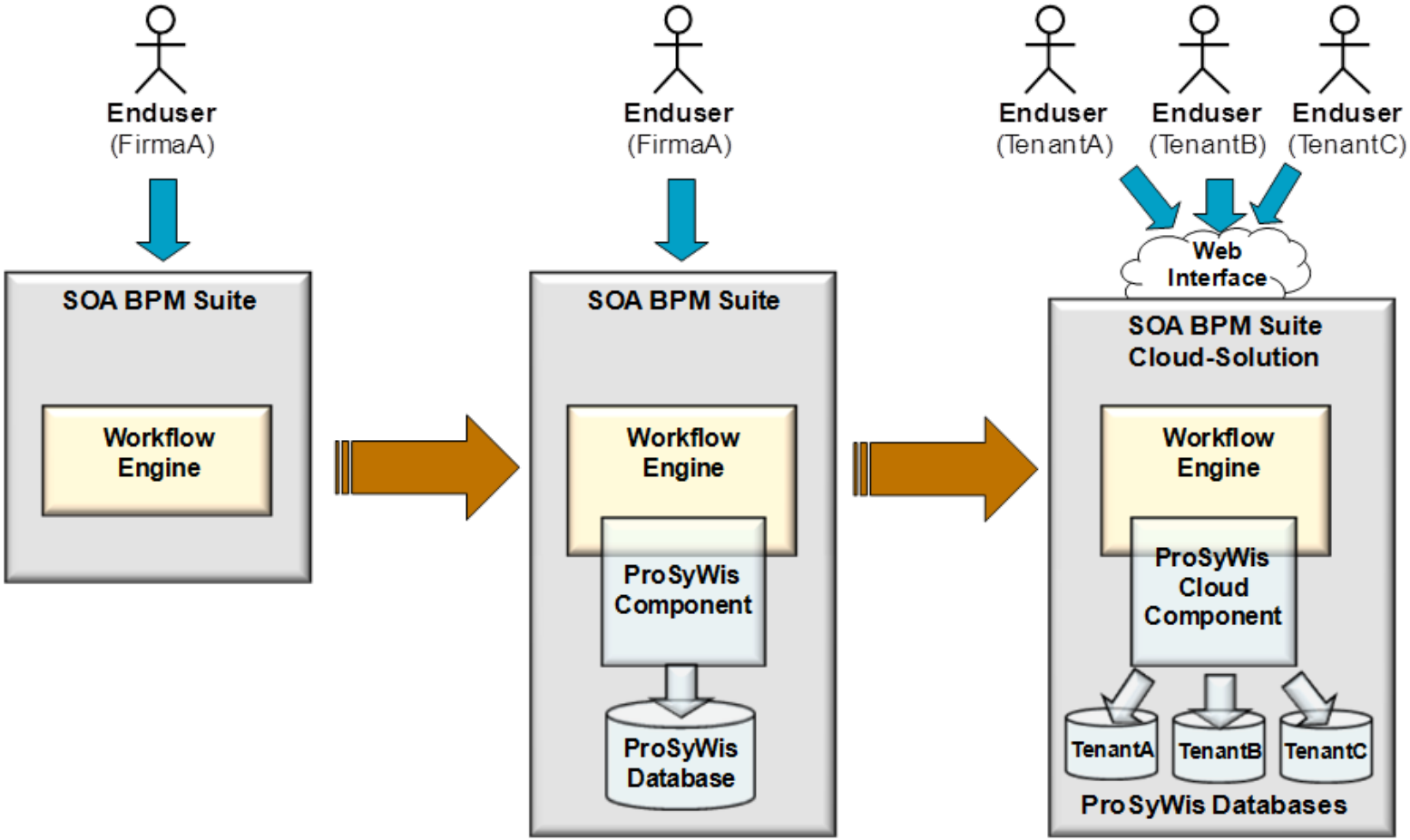
First Draft of a Software Architecture



Paper Prototype: Elements of the ProSyWis User Interface

- Visualization of necessary context-based information:
  - Activities
    - currently running
    - upcoming (some prerequisites are missing)
    - next possible ones (prerequisites are fulfilled)
    - recommended activities
  - Context-based Information Objects
    - Access to all relevant information / data to the current case
    - Short list of the highly relevant data / information w.r.t. to critical cases
    - Overview of all relevant events (internal, external)
  - Cases
    - Short list to all notable previous cases that are in context to the current case
  - Roles and Person
    - Short list to all relevant person being involved in the current case
  - Forms
    - Gathering of all relevant data

ProSyWis – a phase model



Initial situation at project start (2013)

Concept, architecture, and prototype of ProSyWis, Integration in a SOA BPM Suite (currently: webMethods) (2013 - 2016)

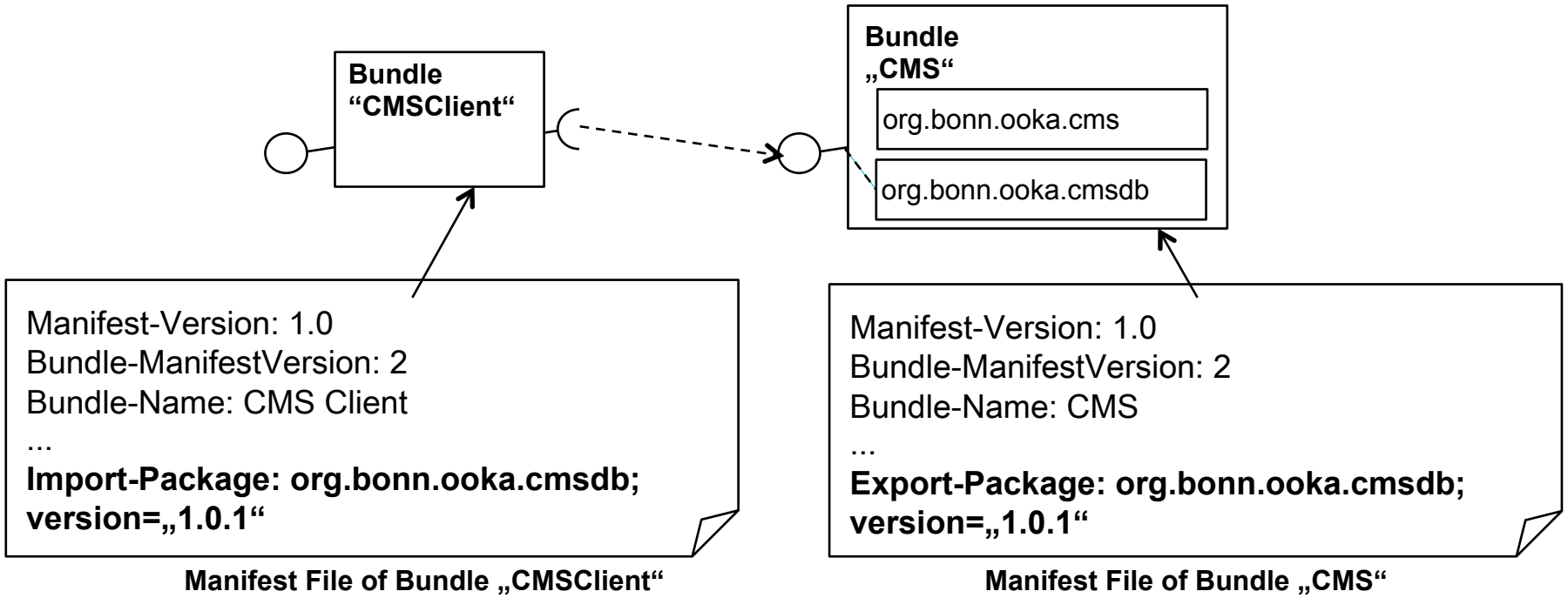
Recommended procedures for a Cloud-based ProSyWis (2013-2016) Implementation in a follow-up project (2016+)

Structure of the Presentation

Software Architectures – Perspectives on Research and Education		
1	Introduction to the Speaker	✓
2	The Bonn-Rhein-Sieg University of Applied Sciences	✓
3	Software Architecture – Some Assumptions	✓
4	**Software Architecture – Perspectives on Research**	
	4.1 – Project ProSyWis	✓
	4.2 – Project BUCO	
5	Software Architecture – Perspectives on Education	
6	Discussion	

The component model of OSGi

- A component (*bundle*) is a set of Java-based packages (with version [opt])
- Each bundle comes with a manifest file
- Packages and manifest file are assembled in a .jar file
- Compositions of bundle are expressed on package level by means of import / export statements.
- Dependencies are resolved at runtime

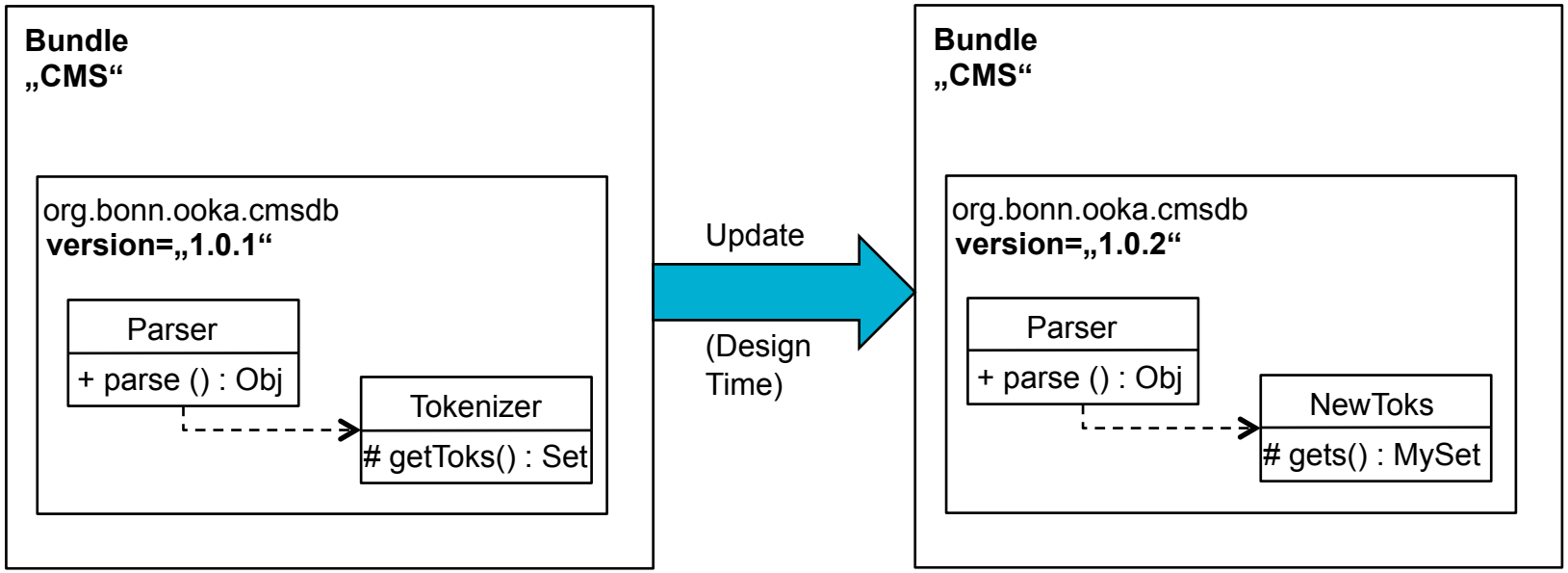


Manifest File of Bundle „CMSClient“

Manifest File of Bundle „CMS“

The component model of OSGi - Discussion

- Component assembler defines the composition based on package information only
- If bundles are updated at design time, inconsistencies can occur when the internal structure (*semantics*) of bundles have changed
- Drawback of OSGi: *no* semantic integrity constraints can be defined to express import / export statements based on the internal structure of bundles



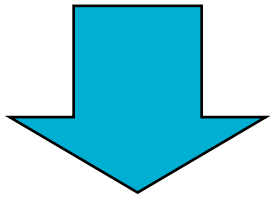
Semantic integrity

Definition (Blom and Nordby, 2000)

- "By *Semantic Integrity,* we mean the preservation of the collective semantic properties of a program, a module or a system."
- "If the semantic properties are violated, the system will enter unstable or inconsistent states, which will, eventually, lead to some kind of malfunction"

Related Work / Previous Work

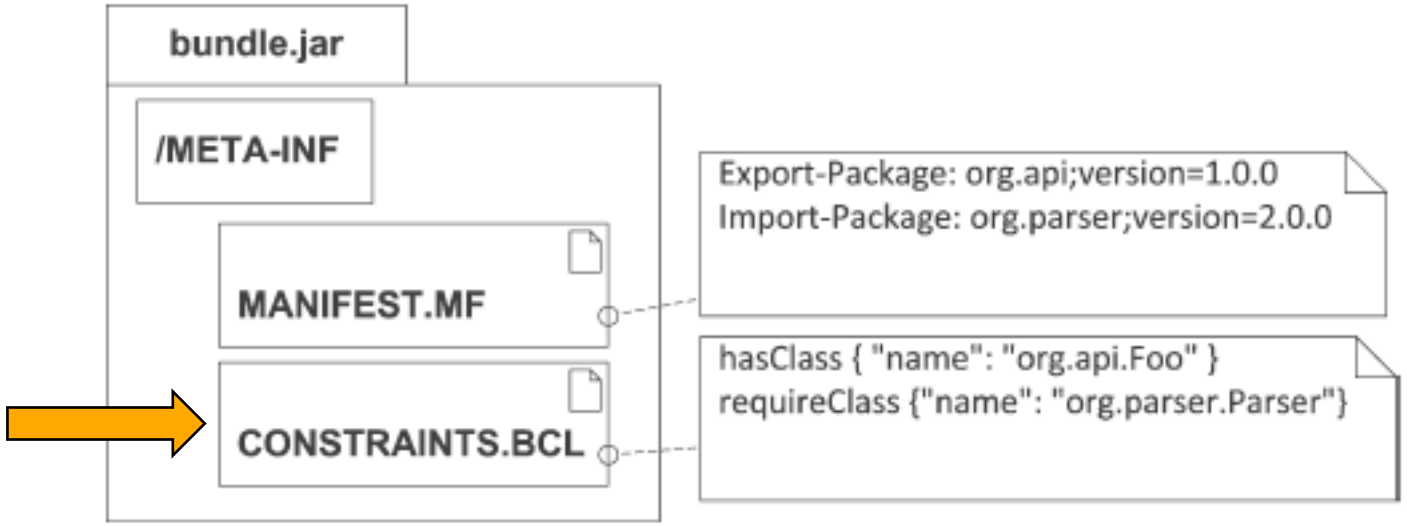
- Adoption of semantic integrity to define valid states of component-based compositions:
  - The FreEvole Platform (Won, 2004), (Won and Krüger 2002)
  - The DeEvolve Platofrm (Alda, 2006)



Adoption of previous findings to the OSGi platform
(Project BUCO)
(Pawelzik, 2012)

Semantic integrity in BUCO

- Semantic integrity constraints can be integrated in a separate deployment descriptor file to a given bundle within the META-INF directory:



- Semantic integrity constraints can be defined based on the domain-specific language BCL (Bundle Composition Constraint Language)
  - Based on JSON meta language
  - Checking algorithm implemented as a separate Java component
- Adaptation environment checks constraints during user interaction

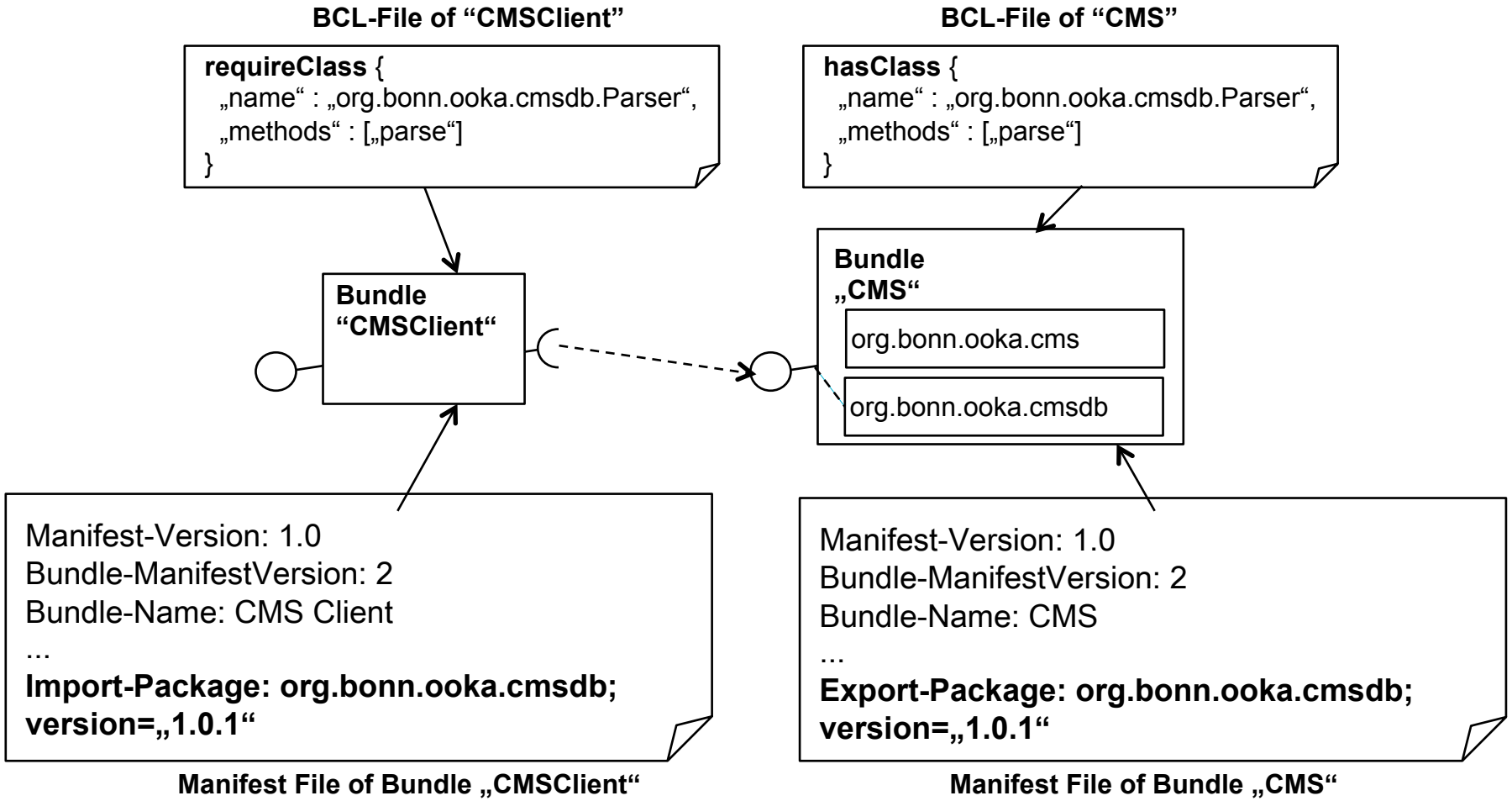
Semantic integrity in BUCO - Types

- Currently, six different semantic integrity constraints are implemented:

Name	Intention
Class Integrity	Request / announce the existence of distinct classes and methods
Type Hierarchy Integrity	Request / announce that classes should be part of a distinct type hierarchy; Ensure type conversions
Communication Integrity	Defines communication channels to preserve architectural compliance
Assertion Integrity	Define assertions to the system environment (e.g. existence of distinct bundles)
Trigger Integrity	Define a procedure that is invoked around the arise of specific events
Behavior Integrity	Defines the order of methods calls that have to be / should be invoked

Semantic integrity in BUCO – Class Integrity

- The class integrity allows the composition of bundles based upon the existence of distinct classes and methods:



Semantic integrity in BUCO – Behavior Integrity

- The behavior integrity allows the composition of bundles based upon the order, in which selected methods are executed:

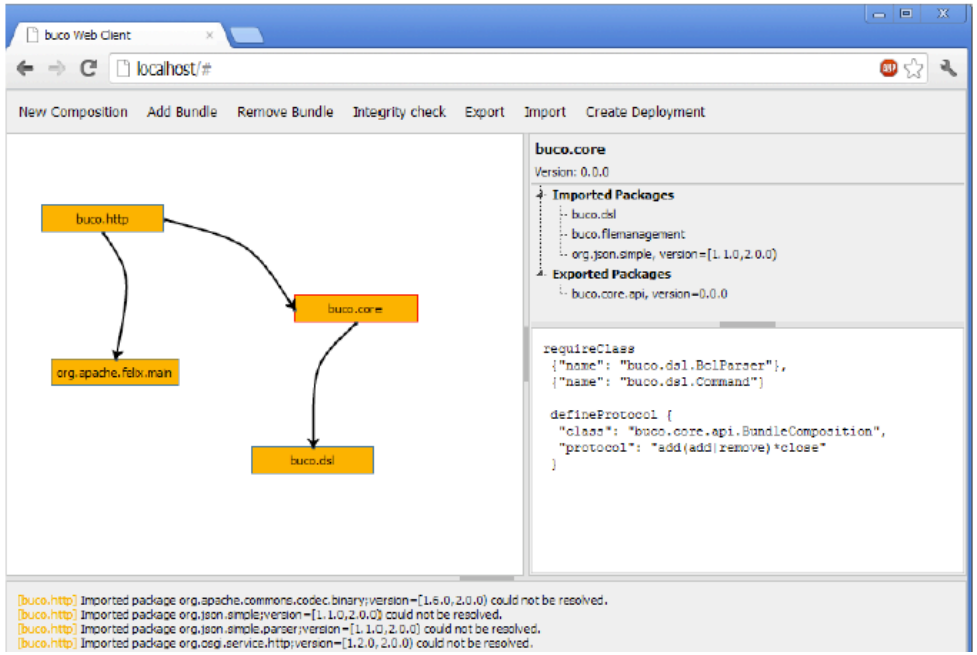
```
1  // Definition des Protokolls beim Schnittstellenanbieter
2  defineProtocol {
3   "class": "A",
4   "protocol": "init(a|b|c)*close"
5  }
6  // Prüfen des Protokolls durch den Schnittstellennutzer
7  checkProtocol {
8   "class": "A",
9   "sequence": ["init", "a", "b", "b", "close"]
10 }
```

Protocol of Component Provider

Protocol of Component Consumer

Semantic integrity in BUCO – Implementation

- The BUCO adaptation environment has been implemented as a Client-Server application
- Client is built as a Rich Internet Application (RIA, HTML5, jQuery)



- Future Work:
  - Performing integrity checks at runtime
  - Automatic extraction of BCL-based constraints (code analysis)

Structure of the Presentation

Software Architectures – Perspectives on Research and Education		
1	Introduction to the Speaker	✓
2	The Bonn-Rhein-Sieg University of Applied Sciences	✓
3	Software Architecture – Some Assumptions	✓
4	**Software Architecture – Perspectives on Research**	
	4.1 – Project ProSyWis	✓
	4.2 – Project BUCO	✓
5	Software Architecture – Perspectives on Education	
6	Discussion	

Structure of the Presentation

Software Architectures – Perspectives on Research and Education		
1	Introduction to the Speaker	✓
2	The Bonn-Rhein-Sieg University of Applied Sciences	✓
3	Software Architecture – Some Assumptions	✓
4	Software Architecture – Perspectives on Research	✓
5	**Software Architecture – Perspectives on Education**	
6	Discussion	

Software Architecture in Education (Bachelor)

Database Systems	Bachelor, 2nd Semester
Principle architecture of Database Systems, ER/EER Modeling, Relational data model, Relational Algebra, SQL DML, SQL DDL, PostgreSQL, JDBC, Transactions, NoSQL Databases (MongoDB)	

Software Engineering I	Bachelor, 3nd Semester
Stakeholder in Software projects, Waterfall model, agile software methods and processes (Scrum), User Stories, cost estimation, prioritization, Use Case modeling (UML and text), analysis object model (UML), foundations of software architecture, basic architecture patterns (MVC, layers), design patterns, delegation vs. inheritance, test-driven development (JUnit), refactoring	

Software Architecture in Education (Master)

Service- Oriented Architectures	Master, 1st 2nd Semester
Basics of SOA, reference architectures, key components, service design, enterprise service bus architecture, modeling of SOA, design principles, architectural patterns, loose coupling, scalability, Web Services (SOAP vs. REST), W3C protocols, AJAX2, WSDL, BPEL, BPMN	

Object-Oriented Component Architectures	Master , 1st or 2nd Semester
Definition of software components, ingredients of component models, modeling of software components, process models and patterns for developing component-based applications, EJB 3.1 (component models, JPA, transactions), OSGi (basic component model, service model, execution environment, version model, hot deployment)	

The primary goal of the summer school is to deepen the student's knowledge of applying agile software methods within a four weeks self-contained project

Further Sub Goals

- Applying the Scrum process model
  - Initializing project set-up, roles, and responsibilities
  - Elicitating users stories and tasks
  - Conducting a self-paced sprint
  - Organizing daily scrum meetings
  - Presenting project's results (poster presentation, demo)
- Applying object-oriented modeling, implementation, and testing techniques
- Learning and deepening of agile development methods
- Open to foreign students (Bachelor with experience in SE, Java)

The Project

The aim of the project is to provide an architecture and based on that a first prototype of a Campus Information System for Foreign students (CIS)

Potential Functional Requirements	Technical Requirements and ToDos
• Enabling foreign guest students to easily navigate through the campus by means of a mobile handheld or a notebook • Implementing an interactive FAQ • Integrating appropriate location-based services and assisting symbols (e.g. QR-codes) • Anticipating multi language support • …	• Providing an flexible software architecture describing the decomposition of the entire system • Analysis of functional requirements • Implementing a client prototype (e.g. Android OS) • Implementing a prototype of a backend (e.g. Java, MongoDB, Node.js) • Providing a solid test-methodology • …

The Setup

The Summer School will be organized and conducted by an international team that brings together years of experiences in developing software projects

Prof. Dr. Sascha Alda
(Main Organizer)

Bonn-Rhein-Sieg University
Grantham-Allee 20
53754 Sankt Augustin
Germany

Phone: +49-2241-865-760
sascha.alda@h-brs.de

Dr. Izzat Alsmadi
(Subject Matter Expert)

Yarmouk University
Jordan

alsmadi@gmail.com

- The Summer School will also be supported by:
  - Dr. Grigory Makeev (Teaching and technical suppport) – TO BE CONFIRMED
  - Nadine Froebel (administrative stuff and summer school angel :-))
  - N.N. - Assisting students (technical support)
  - Capgemini GmbH (leading software consultany company in Germany) – Guest presentation

The Social Aspect

The Summer School will be accompanied by various social events and excursions bringing nearer the cultural life in Germany

Candidate events and excursions (tba)

- Arithmeum Museum, Bonn (museum that reflects the history of modern digital systems)
- Tour around Cologne (including historical cathedral)
- Tour to the Nürburgring (most famous racing course in Europe)
- Hiking Trip through the Siebengebirge (one of most beautiful mountains in West-Germany)
- Final Summer party at the end of the summer school (including German dishes and beers ☺)

Discussion

- Do you have any further questions?
- ... comments?
- ... suggestions?

Backup