FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Architecture and Integration

Prof. Dr. Jürgen Jung

jung.juergen@fb2.fra-uas.de

Fachbereich 2 Informatik und Ingenieurwissenschaften

Prof. Dr. Jürgen Jung

Academia
- Dipl.-Inform. Uni Koblenz
- Dr. rer. pol. Uni Duisburg-Essen

➢ Software Engineering
➢ Enterprise Modelling

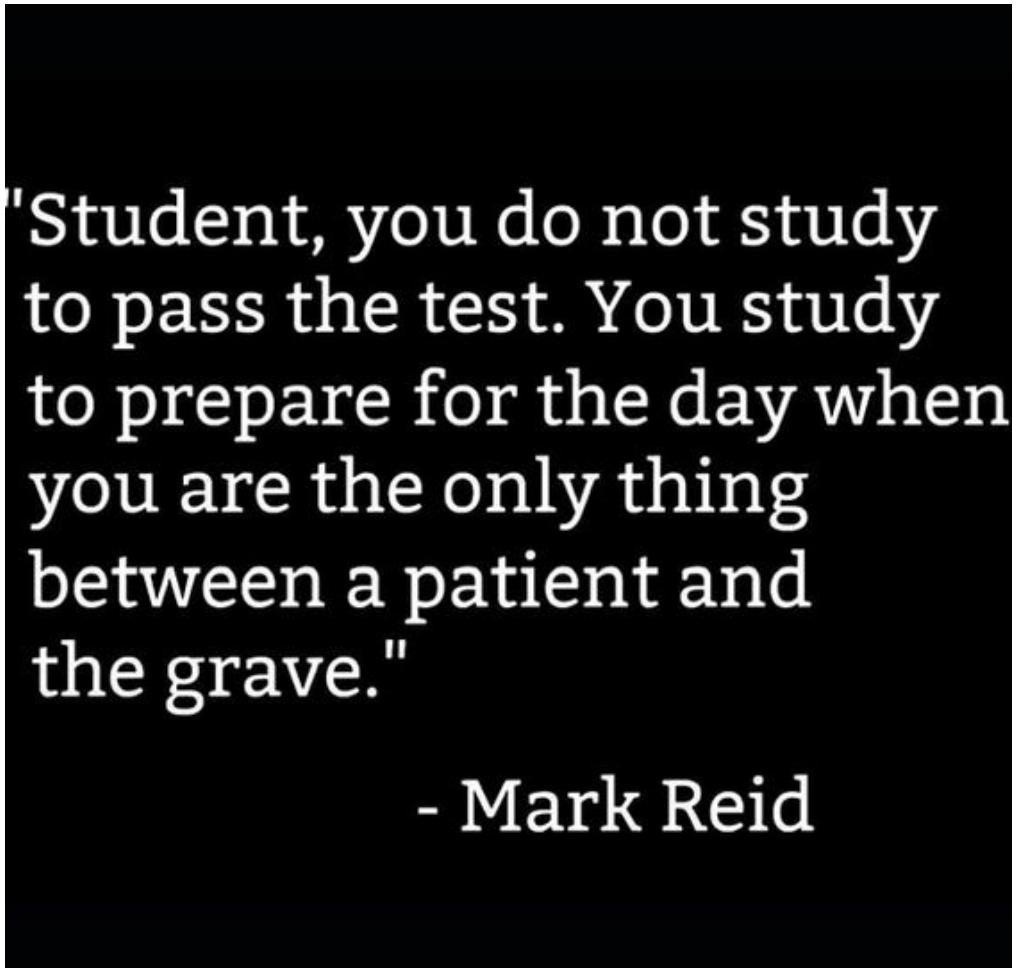
Deutsche Post DHL
- Post & Parcel
- Global Forwarding
- eCommerce

➢ Business Process Management
➢ Enterprise Architecture

Frankfurt University of Applied Sciences
➢ Business Information Systems
➢ Process and Architecture Management

2007 2017

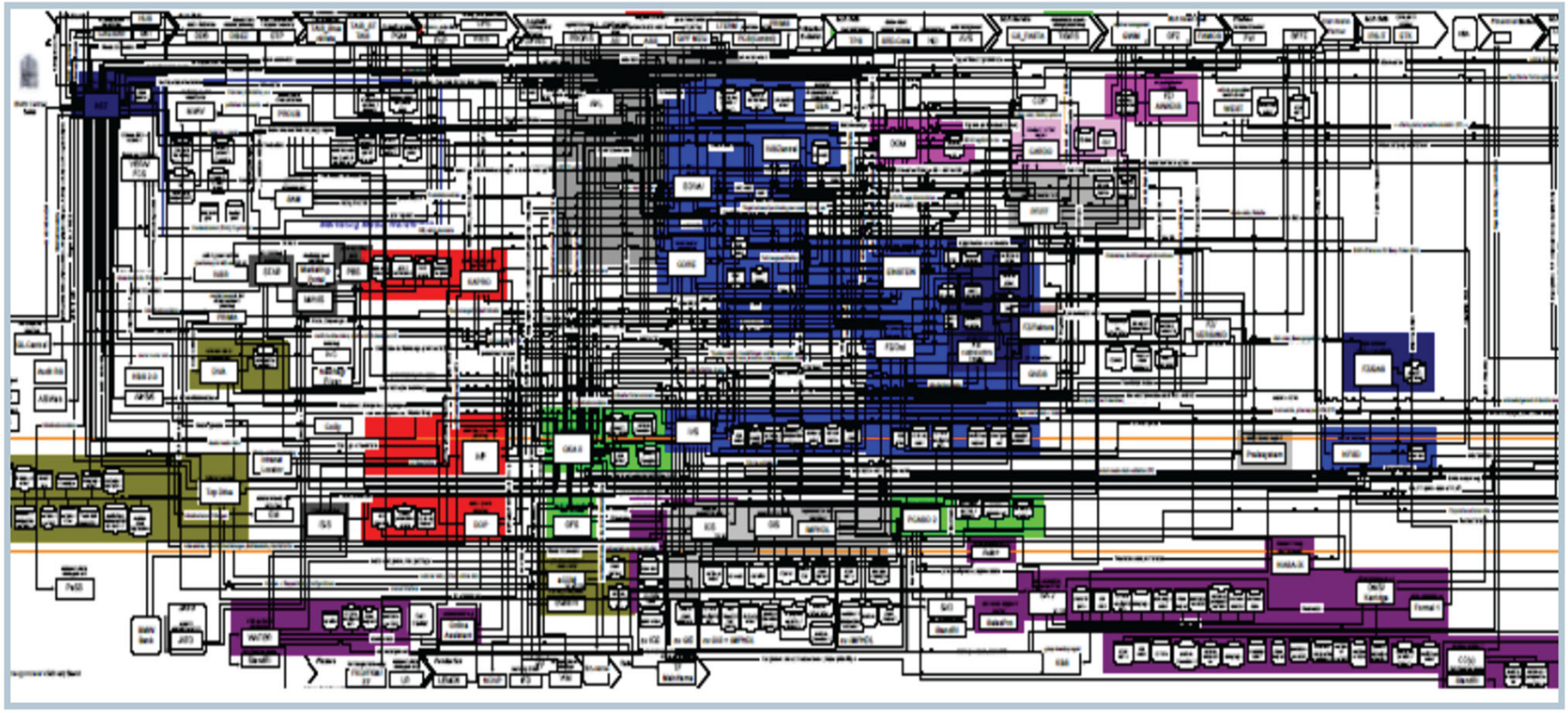
Purpose of a Lecture



Mark B. Reid, MD
Twitter, Jan. 2016

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Context



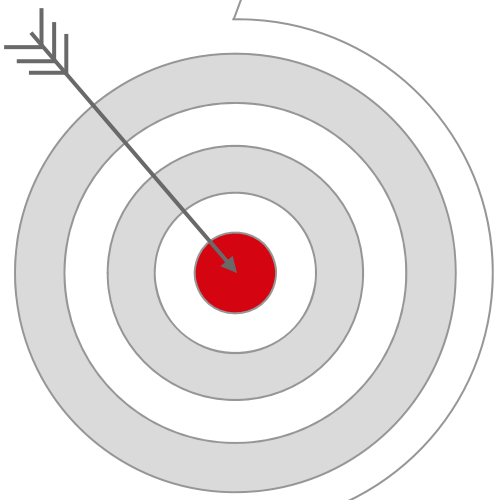
Motivation

Corporate Information Systems are …	Implication
… large (several hundred software systems)	Hard to maintain the complete overview
… changing constantly	Software will be updated often by IT teams
… complex (thousands of relationships)	Change in one system might affect many others
… subject to regulations (e.g. DSGVO)	Challenge for project management
… business relevant!	No trial and error!

Topics

Architecture	• Terminology and basic concepts • Architecture modelling (ArchiMate)
Integration	• Motivation, terminology and concepts • Kinds of integration and principles
Integration technologies	• Middleware for integrating software systems • Examples: Message Queue, Workflow, Enterprise Service Bus
Architecture quality and principles	• Criteria for determining architectural quality • Architecture principles
E-business and Supply Chain	• Architecture and technologies for E-Business • Inter-company Supply Chain Integration
Enterprise Architecture	• Information System Architecture and the Business • Preview: Business Architecture

General Learning Objectives



Applying relevant terminology for IT architecture and business integration

Describing an IT architecture for given organisation

Explaining different integration technologies and their business use

Implementing a system using integration technology

Context in Curriculum

Semester 4 30 CP	18) Enterprise Resource Planning (5)	19) Data Warehousing (5)	20) Digital Business and E-Commerce (5)	21) IS Project Management (5)	24) Seminar Digital Management (5)	27) Service Integration und Management (5)
					23) Seminar Information Systems Architecture (5)	26) Architecture and Integration (5)
					22) Seminar Business Analytics (5)	25) Programming Lab (5)
Semester 3 30 CP	12) Statistik (5)	13) Software Engineering (5)	14) Logistik und Produktion (5)	15) Geschäftsprozessmanagement (5)	16) Datenschutz- und Internetrecht (5)	17) Web-basierte Anwendungssysteme (5)
Semester 2 30 CP	6) Analysis (5)	7) Betriebssysteme und Rechnernetze (5)	8) Datenbanken (5)	9) Schlüsselkompetenzen (5)	10) Wirtschaftsprivatrecht (5)	11) Business Englisch (5)
Semester 1 30 CP	1) Algebra (5)	2) Wirtschaftsinformatik (5)	3) Objektorientierte Programmierung (10)		4) Betriebswirtschaftslehre (5)	5) Rechnungswesen (5)

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Lecture and Seminar

From the module handbook:

Module contents	Architecture and Integration – Lecture Architecture and Integration – Seminar
Module teaching methods	Lecture, Seminar
Module language	English

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Competencies Required for Learning

Competency	Self-study	Apply knowledge	Critical reflection
Student	• Actively preparing • Self reliance • Organisation	• Work on project • Assess own results • Clarify issues	• Critical thinking • Transfer knowledge • Reflect own learning
Lecturer	• Provide structure • Provide material • Prioritise topics • Support students	• Provide case study • Provide feedback • Assess results	• Provide topics • Define criteria • Provide feedback

Examination

Project work	Individual assessment

Case study work
• Team of 3 students
• Working time: 8 weeks
• Project results:
  ○ Architecture model and description
  ○ Implementation using selected integration technology
  ○ Final presentation (5 minutes per person)
• Max. 60 points altogether
	Written examination
	• 60 minutes
	• Testing knowledge
	  ○ Explaining concepts
	  ○ Using proper terminology
	  ○ Explaining technologies
	  ○ ArchiMate modeling language
	• Max. 40 points altogether

Project work

Context	• Small business scenario provided May 8 • Choosing integration technology • Objective: prototypical implementation of scenario using integration technology
Model	• Architecture model for prototype • Blueprint for implementation
Technology	• Integration technology understood • Preparing implementation of prototype
Prototype	• Running instance of integration technology • Dummy implementation for connected systems
Submission	• Upload in CampUAS • July 3 eob.
Presentation	• Overview and demonstration • July 10/17

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Interaction during Lecture/Seminar

Particify

Room number: 3701 4753

https://fra-uas.particifyapp.net/p/37014753

Selected Reading

- Laudon, K.; Laudon, J.: “Management Information Systems: Managing the Digital Firm.” 7th ed., Pearson, 2022
- Wierda, G.: “Mastering ArchiMate Edition 3.1: A serious introduction to the ArchiMate® enterprise architecture modeling language.” R&A, 2021
- Kurbel, K.: “The Making of Information Systems: Software Engineering and Management in a Globalized World.” Springer, 2008
- Sousa, P.; Vasconcelos, A.: “Enterprise Architecture and Cartography: From Practice to Theory; From Representation to Design.” Springer 2022
- The Open Group: “The TOGAF Standard 10: Introduction and Core Concepts.” online: https://pubs.opengroup.org/togaf-standard/introduction/index.html

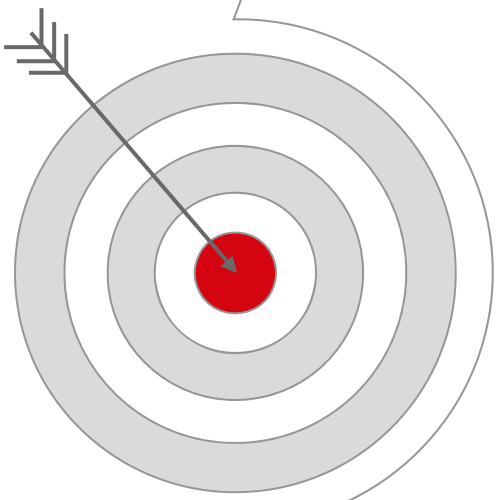
FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Architecture and Integration

Architecture: Motivation and Introduction

Fachbereich 2 Informatik und Ingenieurwissenschaften

Learning Objectives: Architecture



Defining “architecture”

Explaining different kinds of architecture

Describing elements of an architecture

Short Characterisation

Architecture is about describing the structure of a system.

The structure of an information system is usually not visible, so that it needs to be documented as an **architecture model**. Such a model can be

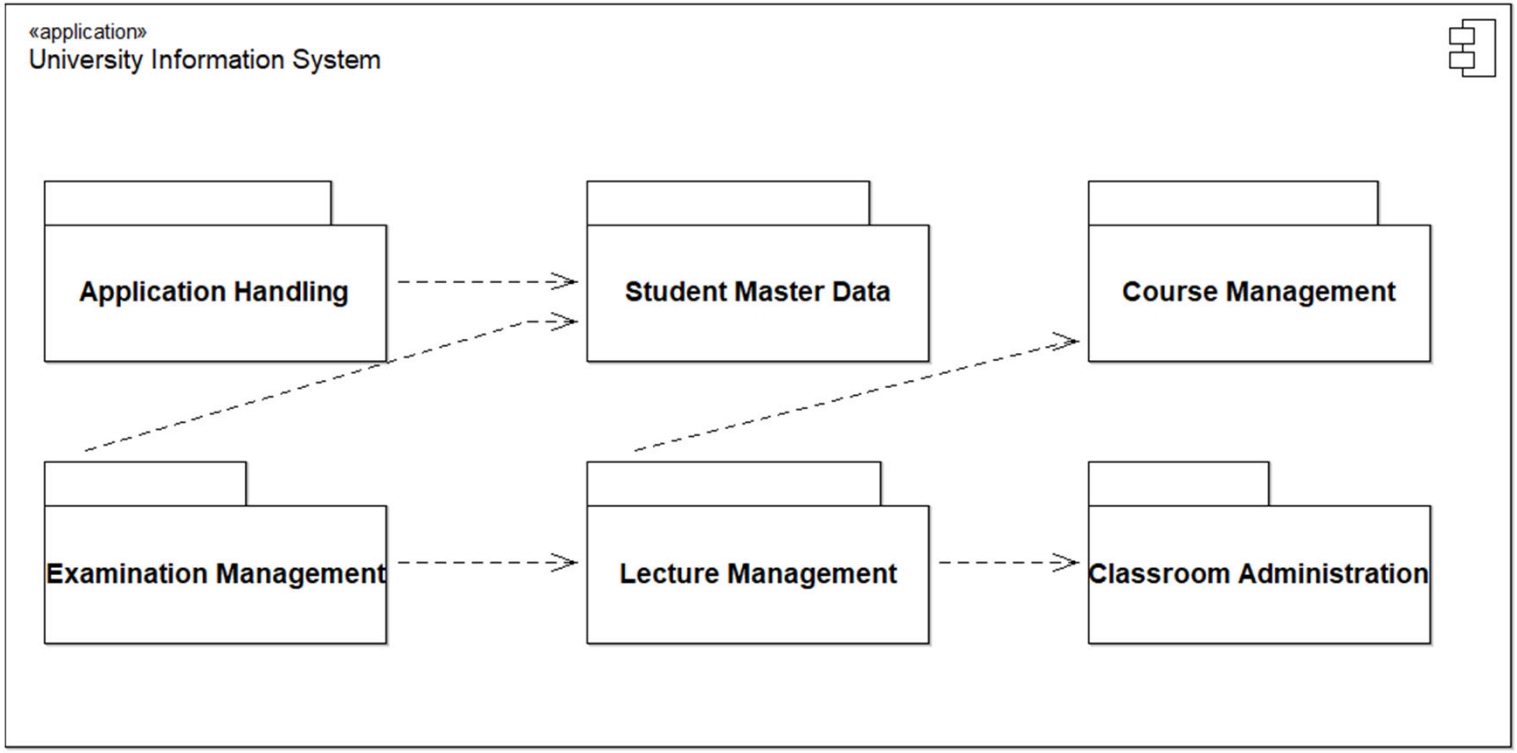
- **Descriptive**: describing an existing system
- **Prescriptive**: blueprint for a future system

Particify

Which architectures do you know (examples)?

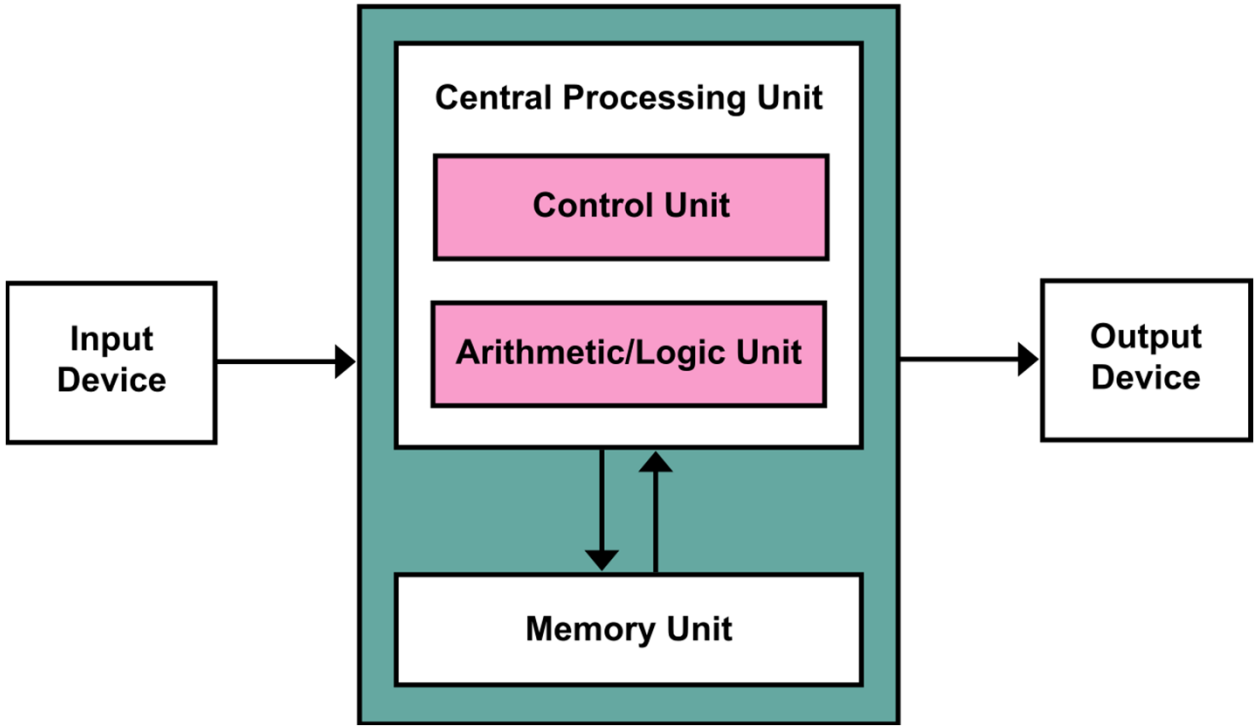
Software Architecture (example)

Architecture of a software system consisting of different modules (e.g. Java packages) and their dependencies.



von Neumann Architecture (Computer)

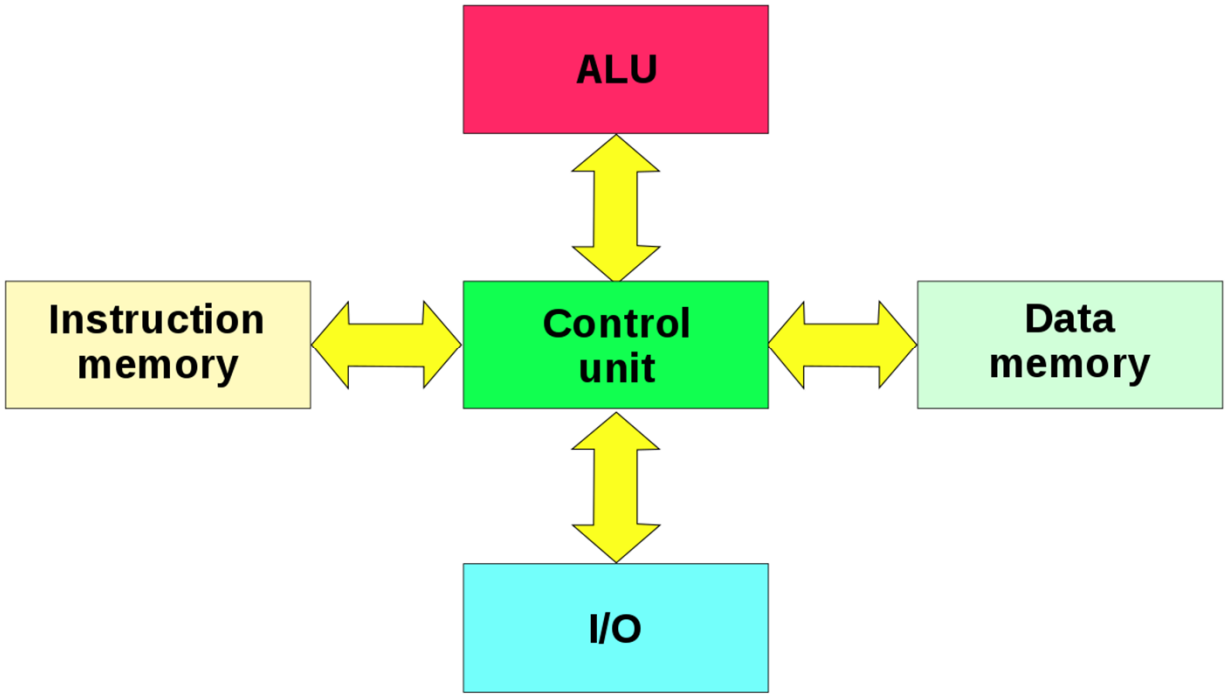
Basic architecture of a computer with its Central Processing Unit (CPU) and its interfaces to memory and devices.



Source: https://en.wikipedia.org/wiki/Von_Neumann_architecture

Harvard Architecture (Computer)

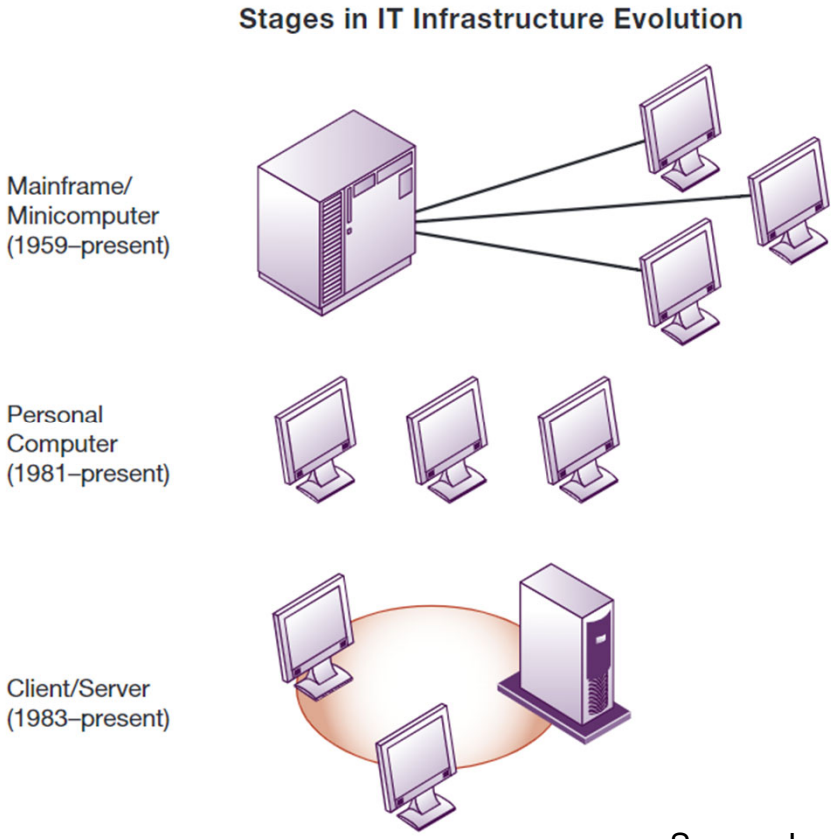
Basic architecture of a computer with its control unit and its interfaces to Arithmetic Logic Unit (ALU), memory and devices.



Source: https://en.wikipedia.org/wiki/Harvard_architecture

IT Architecture (early stages)

Architecture of computing devices in a business environment.

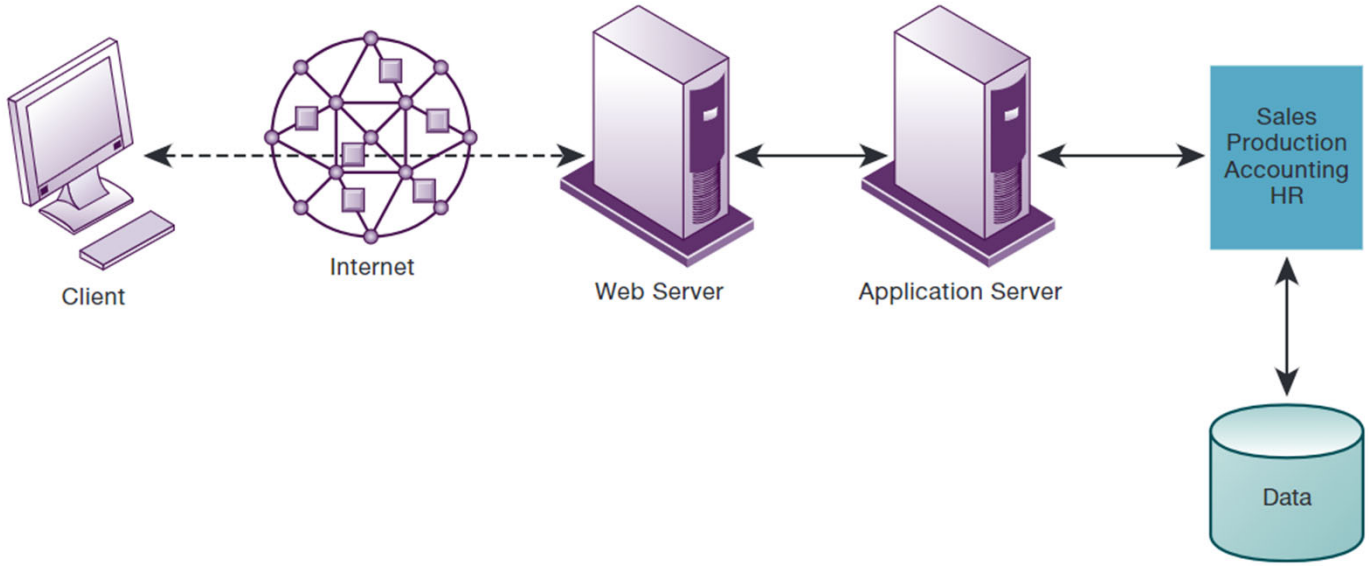


- Central computing device (mainframe) with connected terminals.
- Personal computers for individual business tasks (no computer network)
- Computer network with central server and connected personal computers.

Source: Laudon/Laudon: Management Information Systems, 2022, p.200

N-Tier Architecture

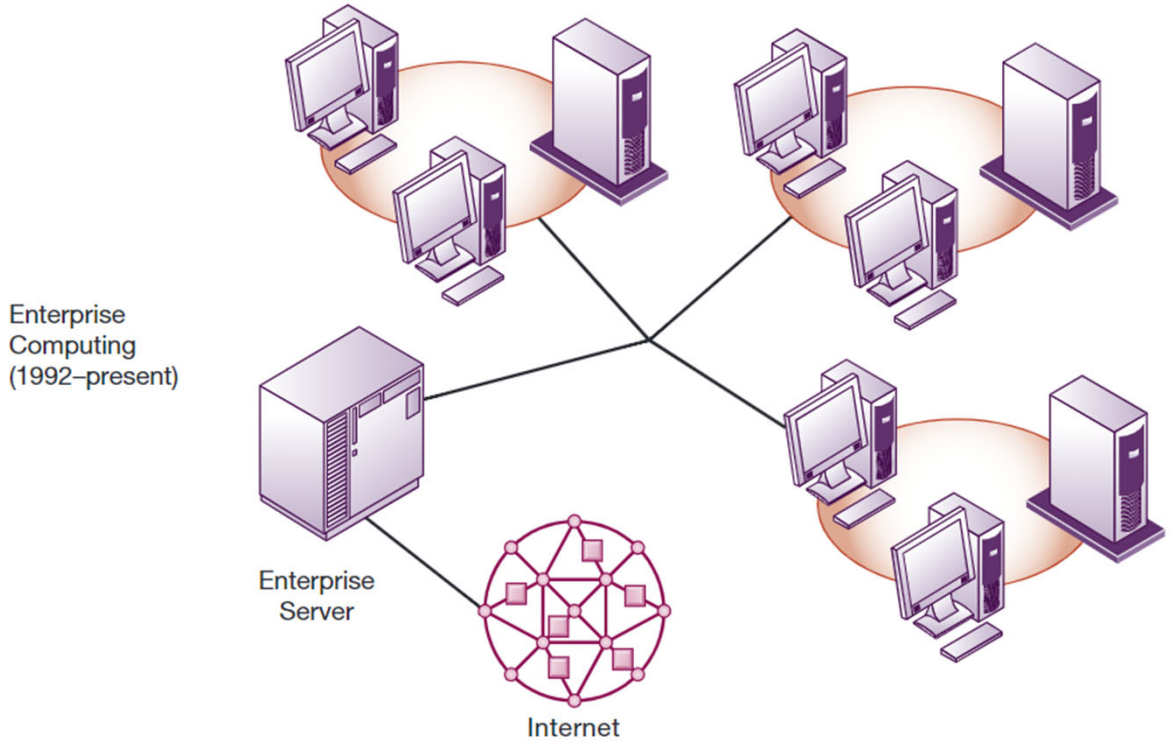
Client/server architecture with several specialised services.



Source: Laudon/Laudon: Management Information Systems, 2022, p.202

IT Architecture

A corporate IT architecture consists of several integrated computer networks.



Source: Laudon/Laudon: Management Information Systems, 2022, p.200

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Particify

What is architecture?

Architecture – Definition

Architecture: The structure of components, their inter-relationships, and the principles and guidelines governing their design and evolution over time.

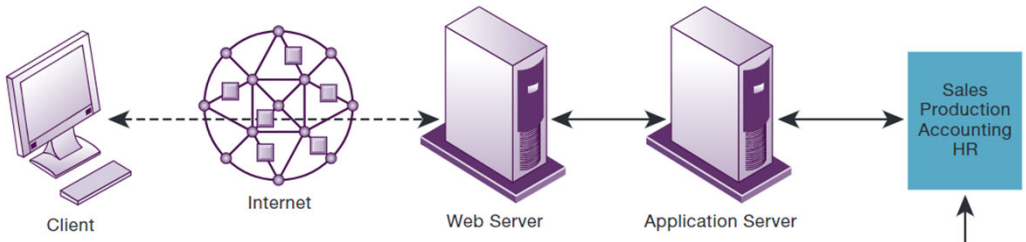
The Open Group: The TOGAF Standard 10

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Particify

Components of architecture?

Elements of N-Tier Architecture



Architecture: The structure of components, their inter-relationships, and the principles and guidelines governing their design and evolution over time.

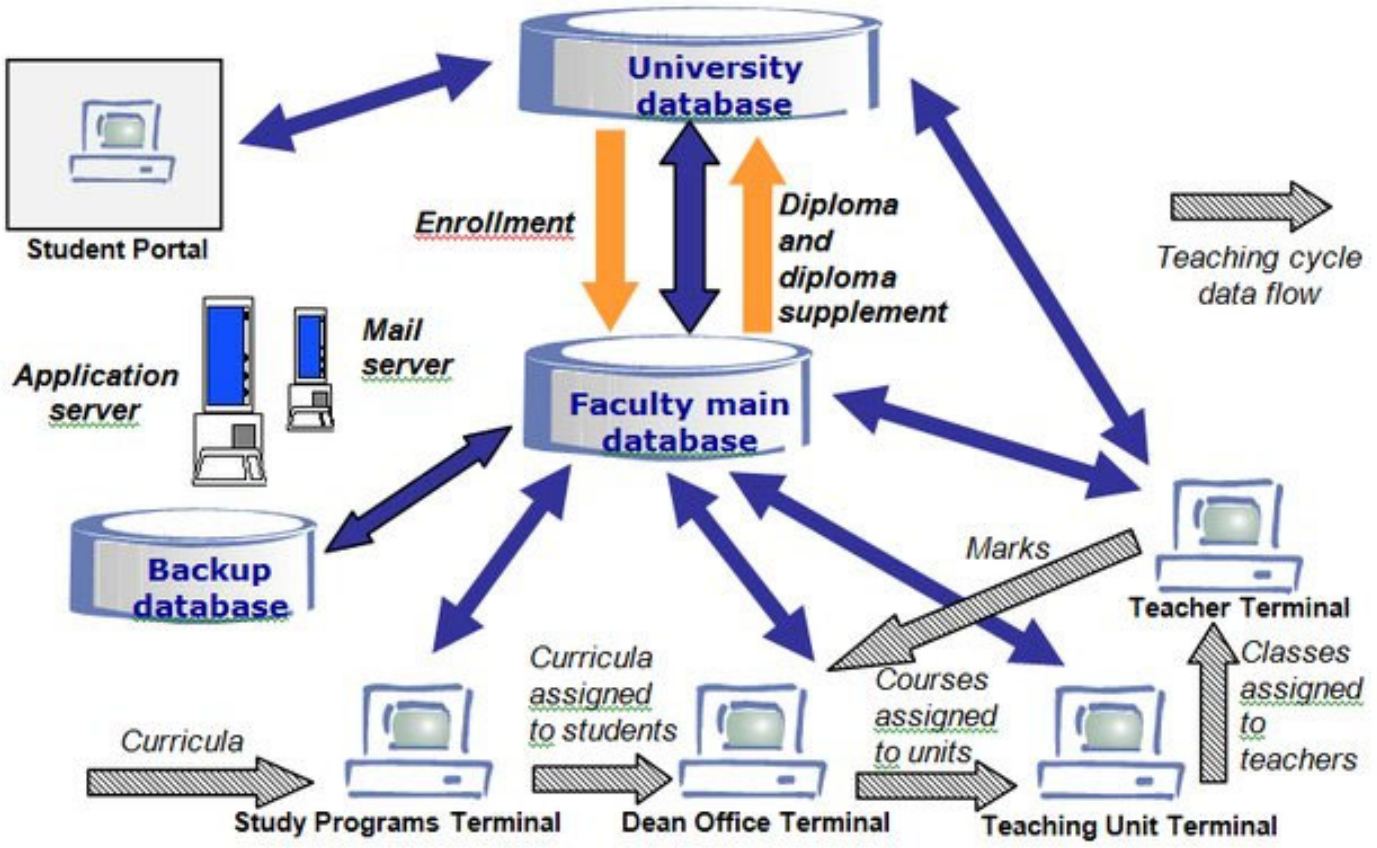
The Open Group: The TOGAF Standard 10

Components: Computing devices with role as *client* or *server*.

Inter-relationships: Network connections via internet or local *network*.

Principles: Client request a service from a server. Servers have dedicated purposes (e.g. web, application, database)

Example Architecture



FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Elaborate the Following Questions

1. What is the process supported by this architecture?
2. What is the meaning of each symbol?
3. Why do symbols have different sizes?
4. What is the meaning of each arrow and its color?
   - Blue
   - Orange
   - Grey

- Prepare a short presentation (any format)
- Team of three students
- Time 20 minutes

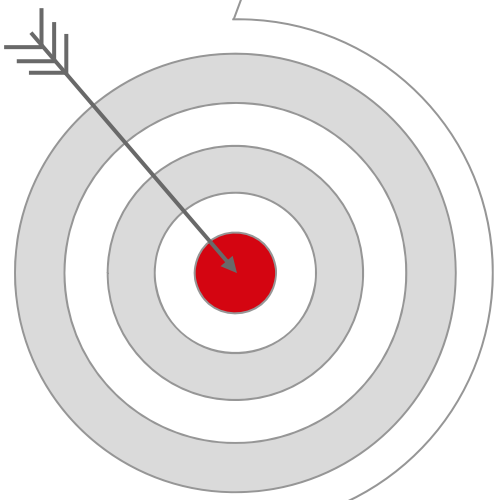
FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Architecture and Integration

Information System Architecture and ArchiMate

Fachbereich 2 Informatik und Ingenieurwissenschaften

Learning Objectives: IT Architecture



Explaining typical elements of an information system architecture

Modelling simple architecture with ArchiMate

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Particify

What is an information system (IS)??

Information System – Definition

Information System: An information system can be defined [..] as a set of interrelated components that collect (or retrieve), process, store, and distribute information to support decision making and control in an organization.

Laudon/Laudon: Management Information Systems, 2022, p.246

IS Architecture – Definition

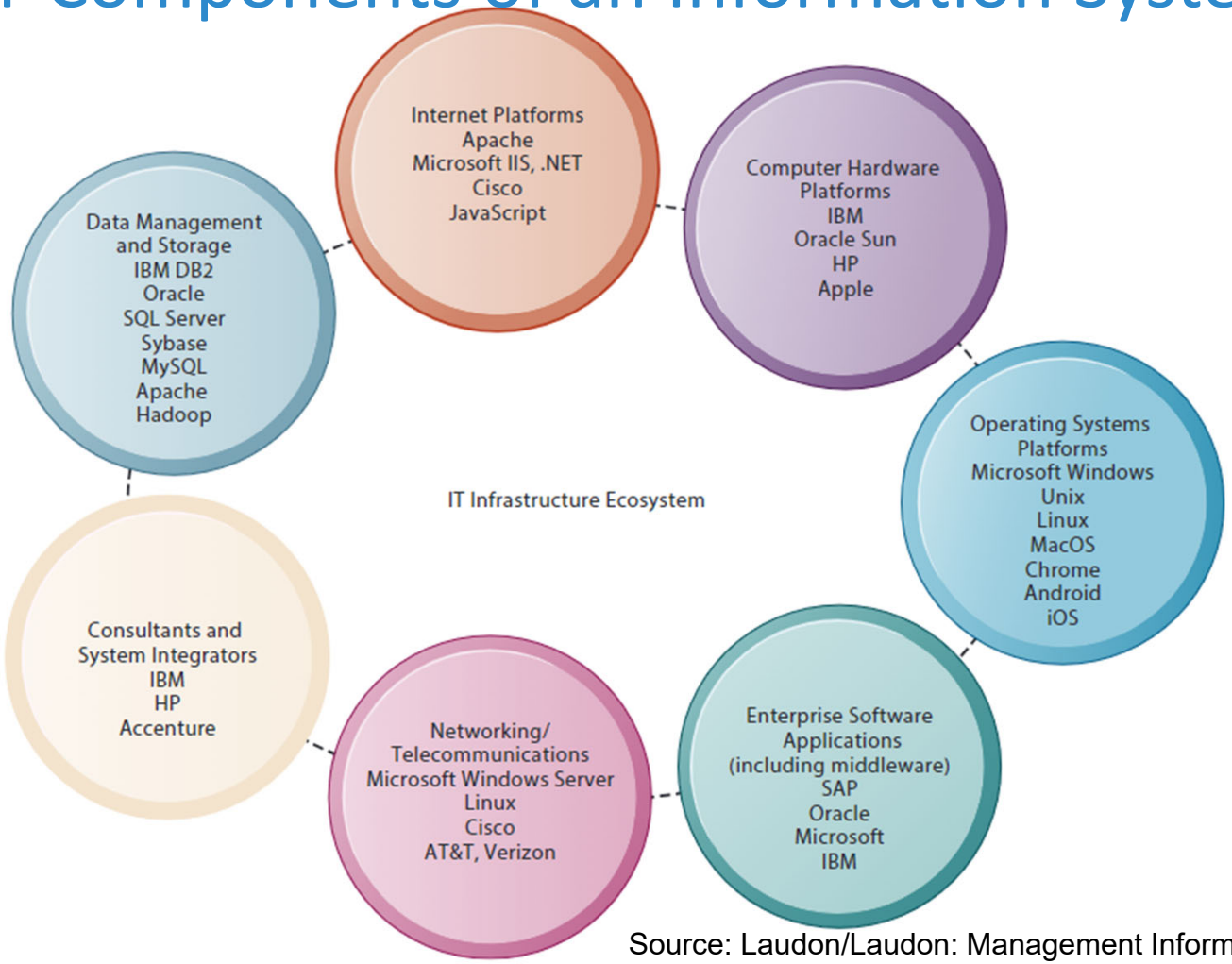
IS Architecture: An Information System Architecture (IS Architecture) is the architecture of an information system in a corporate environment.

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Particify

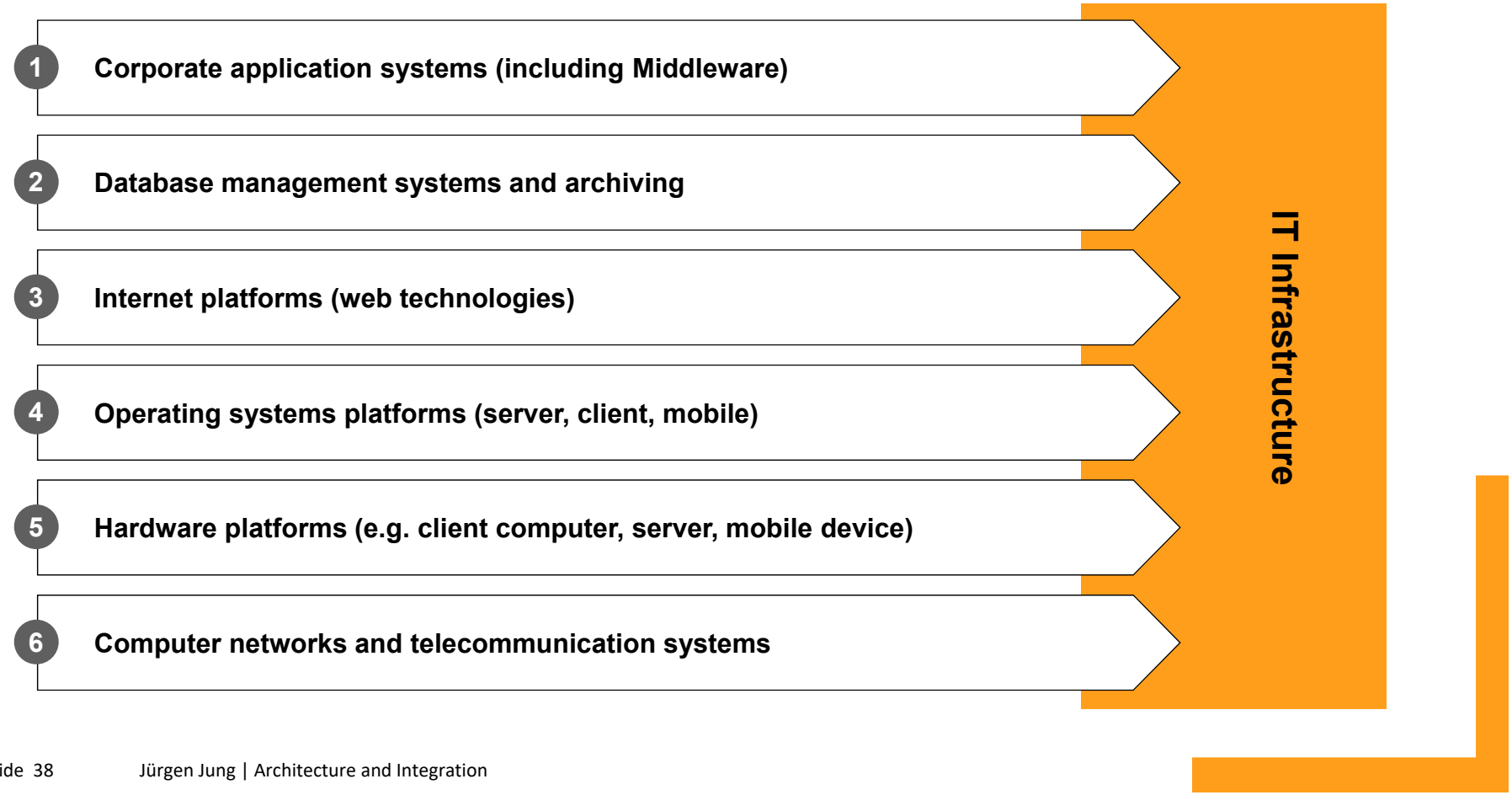
Which kinds of components (elements, objects) does an information system consist of?

IT Components of an Information System



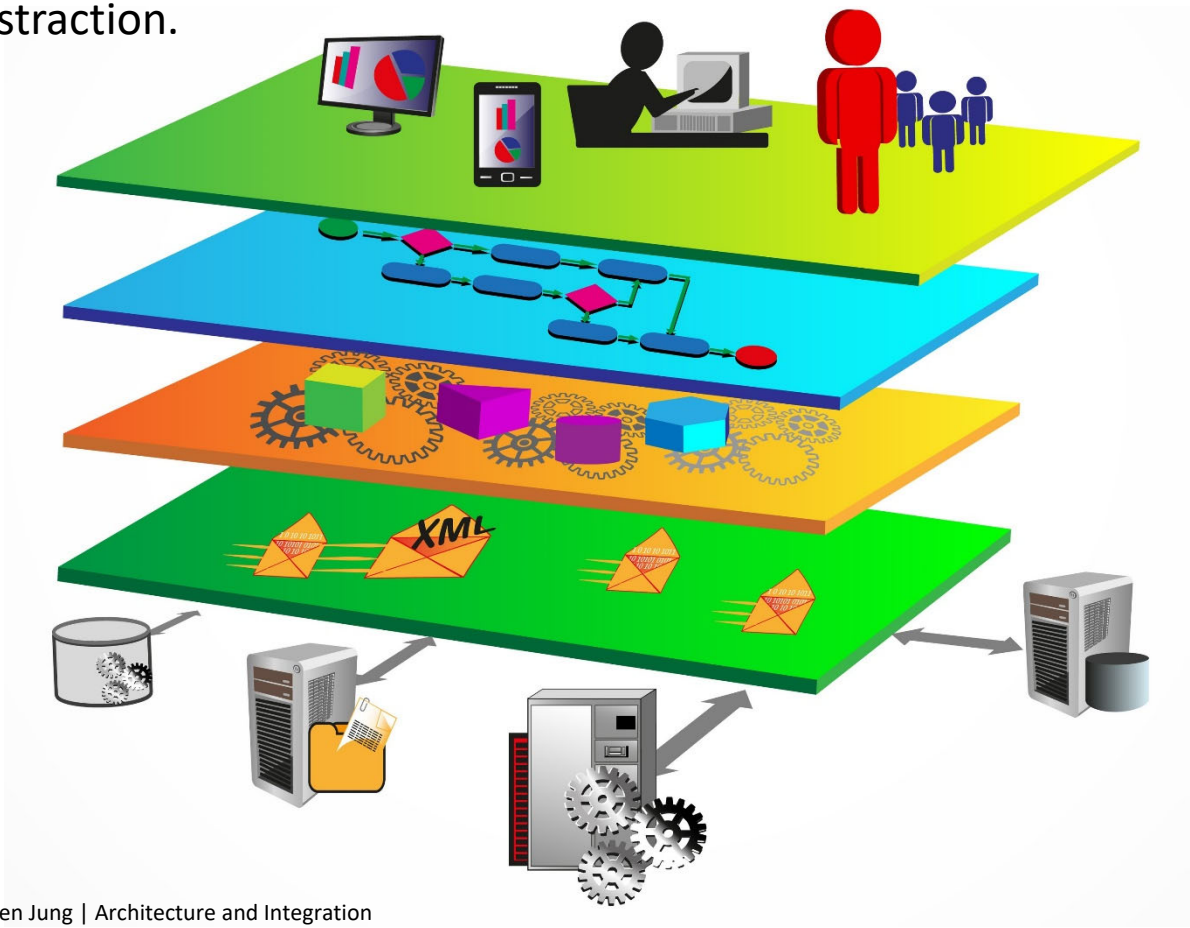
Source: Laudon/Laudon: Management Information Systems, 2022, p.209

IT Components of an Information System

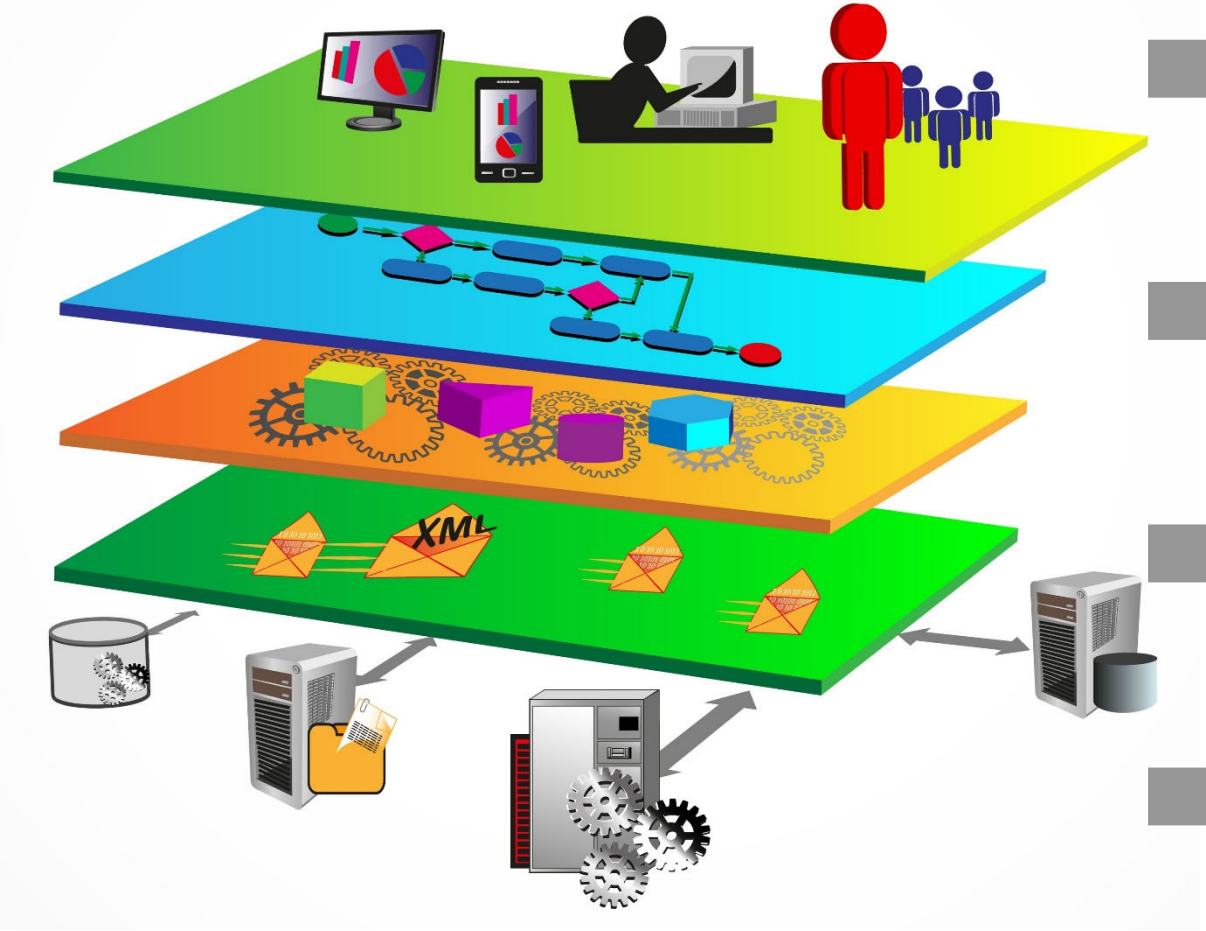


Information System – Levels

An information system in a corporate environment consists of elements on different levels of abstraction.



Information System – Concept Overview



Business environment (e.g. user, role, process, business function, location)

Applications (e.g. standard software, reporting tool, data warehouse, web server)

Data objects (e.g. product, customer, order, invoice, complaint, transaction)

Technology (e.g. hardware device, network element, operating system)

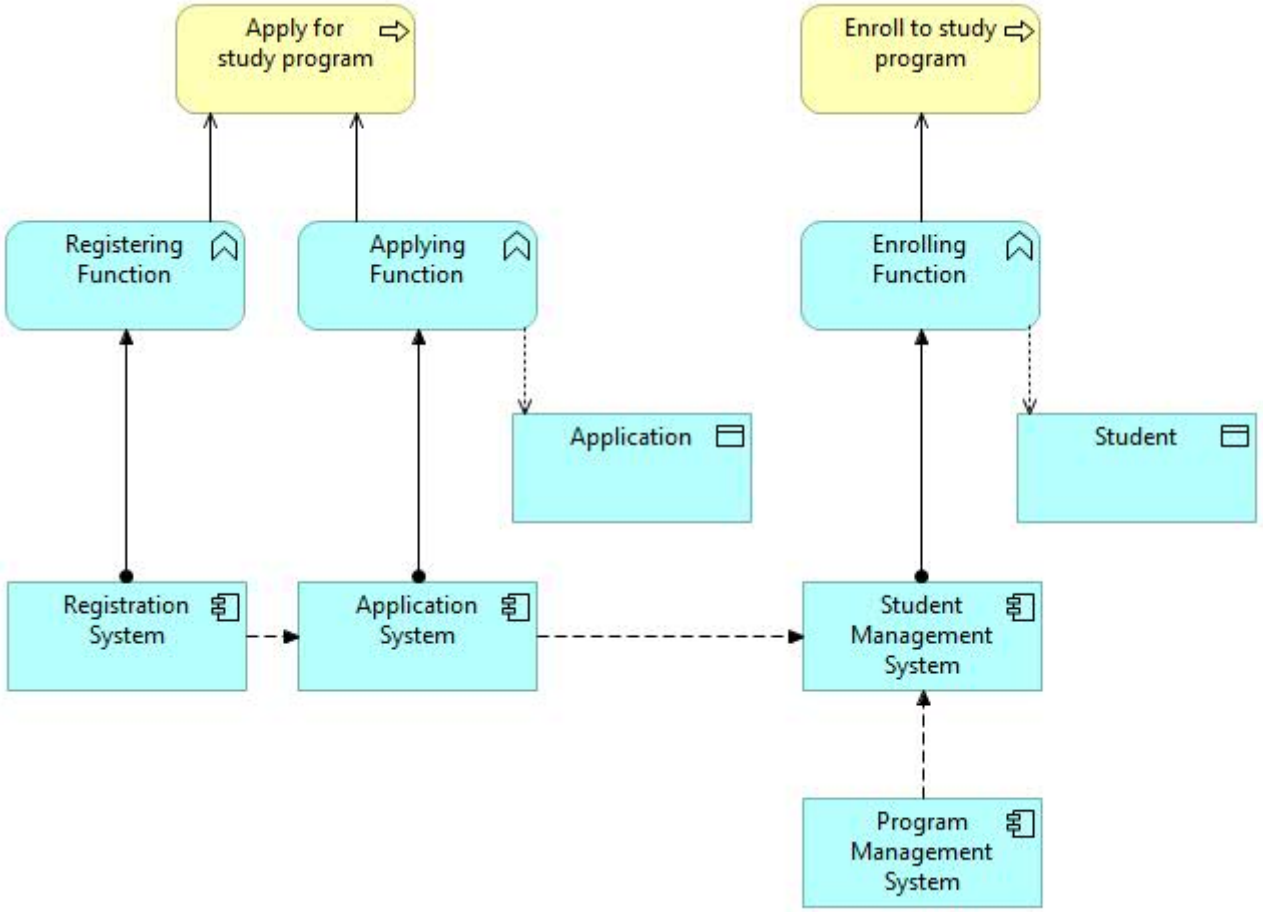
Architecture Model – Definition

Architecture Model: An architecture model is the (graphical) representation of an architecture. It can be described on different levels of detail and from various perspectives. Each perspective represents the for a dedicated stakeholder (e.g. management, developer, system architect or IT infrastructure manager).

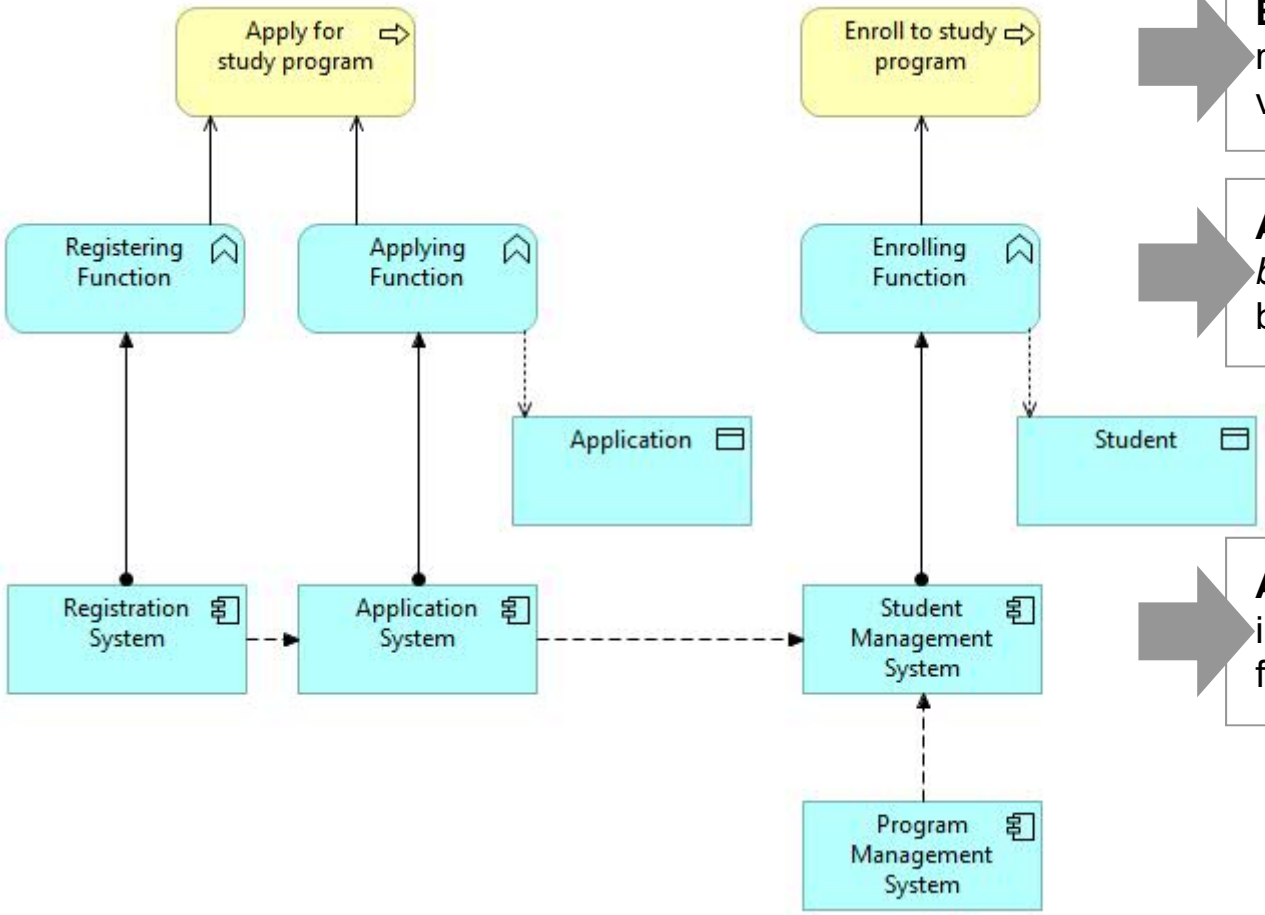
Modelling IS Architecture using ArchiMate

- Modelling language for Enterprise Architecture
- Standardised by *The Open Group* (vendor neutral)
- Current version 3.2 since October 2022
- Originated in a Dutch project during 2002--2004
- Supported by various modelling tools (e.g. Archi)

ArchiMate – Introductory Example



ArchiMate – Concepts Explained

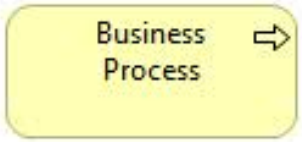
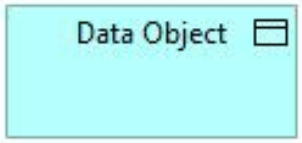


Business process: set of related activities to create value in the organisation

Application function: *behaviour* being implemented by software applications

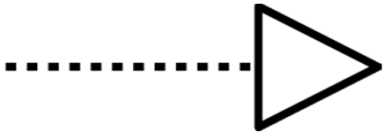
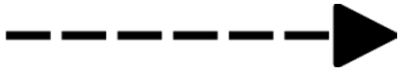
Application component: *actor* implementing an application function (i.e. software)

ArchiMate – Symbols (excerpt)

	A **business process** represents a sequence of business behaviors that achieves a specific result such as a defined set of products or business services.
	An **application component** represents an encapsulation of application functionality aligned to implementation structure, which is modular and replaceable.
	An **application function** represents automated behavior that can be performed by an application component (i.e. being implemented by the application component).
	A **data object** represents data structured for automated processing.

Source: ArchiMate 3.2 Specification, sections 8 and 9

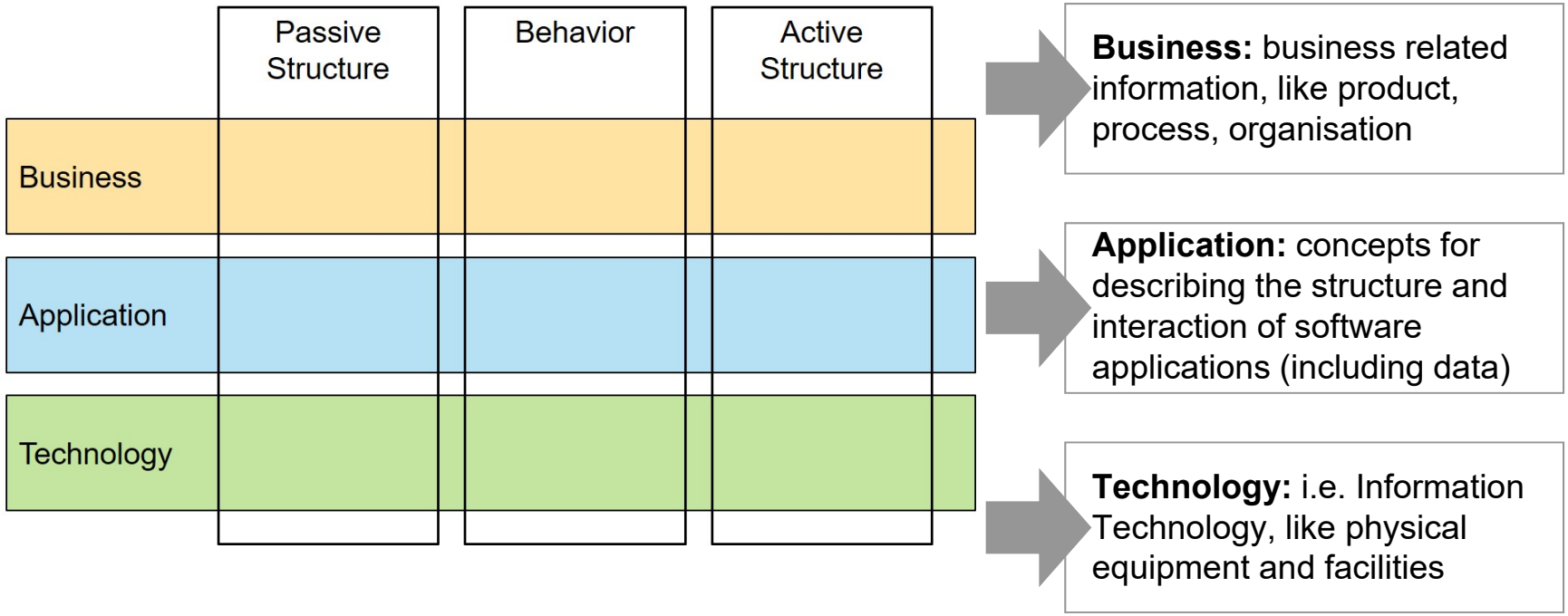
ArchiMate – Relationships (excerpt)

	The **assignment** relationship represents the allocation of responsibility, performance of behavior, storage, or execution.
	The **realization** relationship represents that an element plays a critical role in the creation, achievement, sustenance, or operation of a more *abstract* element.
	The **access** relationship represents the ability of behavior and active structure elements to observe or act upon passive structure elements.
	The **flow** relationship represents transfer from one element to another.

Source: ArchiMate 3.2 Specification, section 5

ArchiMate – Core Framework

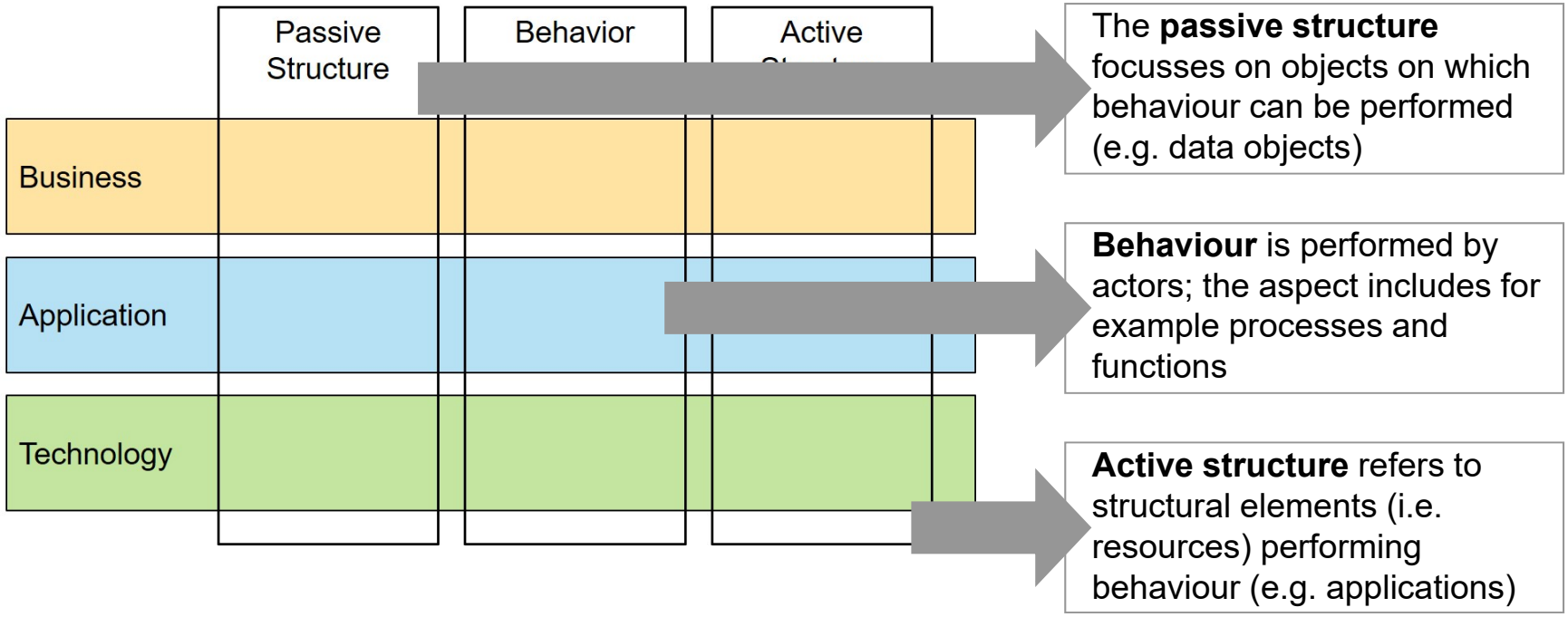
ArchiMate is based on a framework separating the models into different layers.



Source: Jung/Fraunholz: Masterclass EAM, 2021, p.188

ArchiMate – Aspects in Core Framework

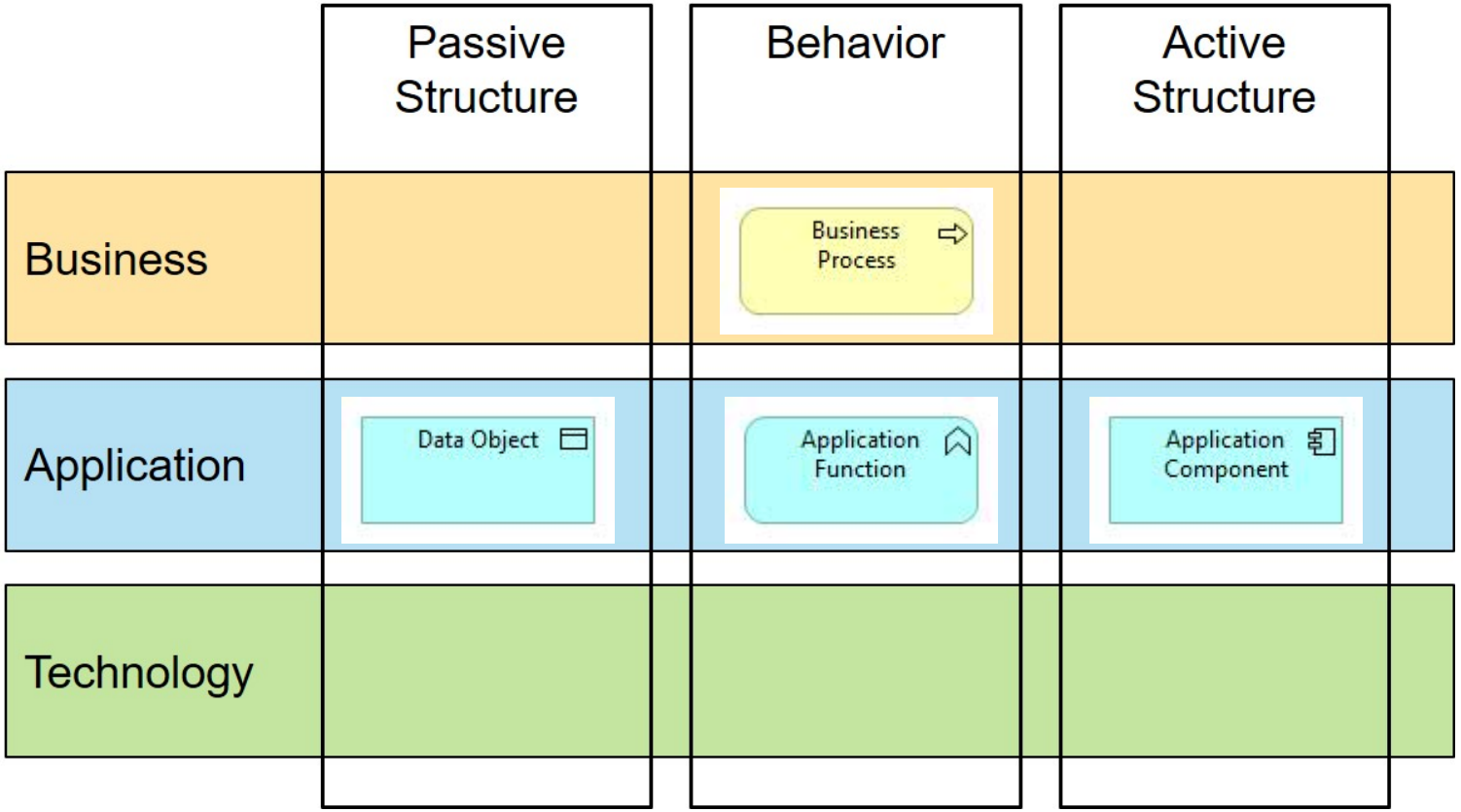
Aspects differentiate between data, functions and resources.



Source: Jung/Fraunholz: Masterclass EAM, 2021, p.188

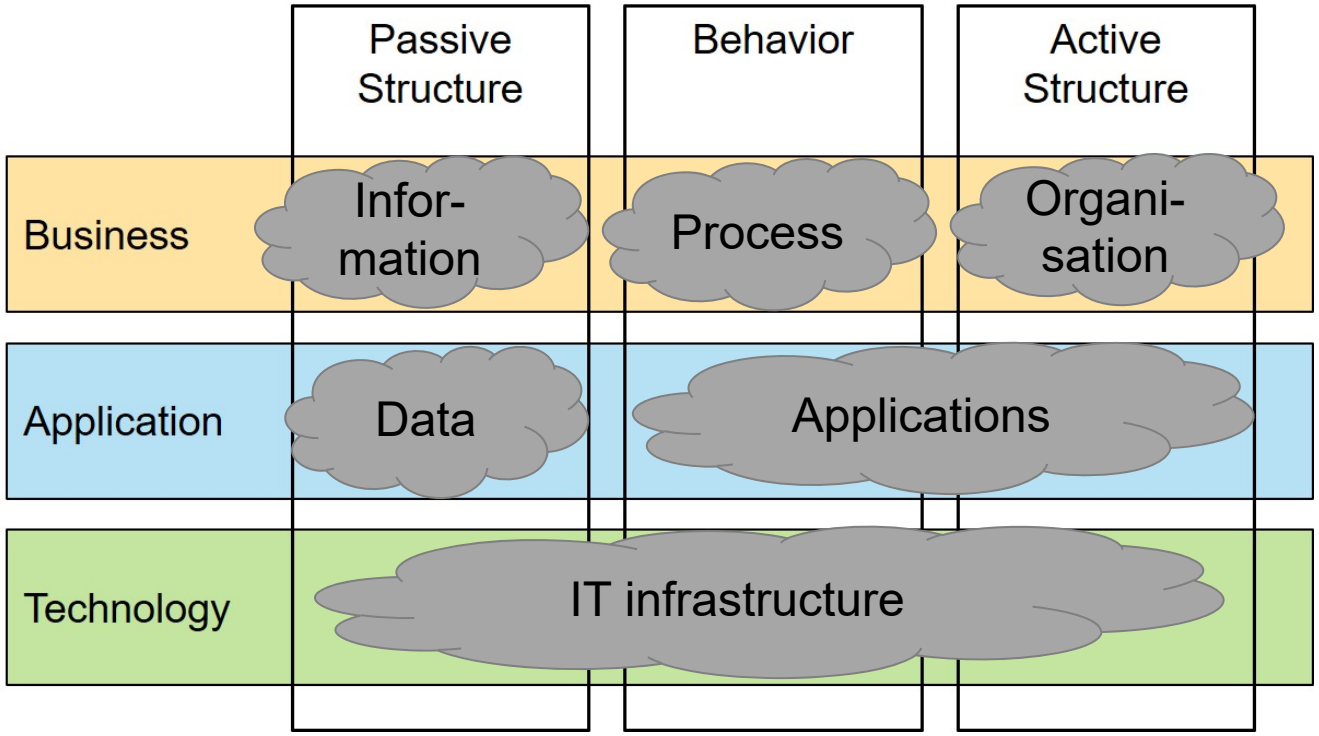
ArchiMate – Today's Elements

The four elements introduced today fit to the framework as follows:

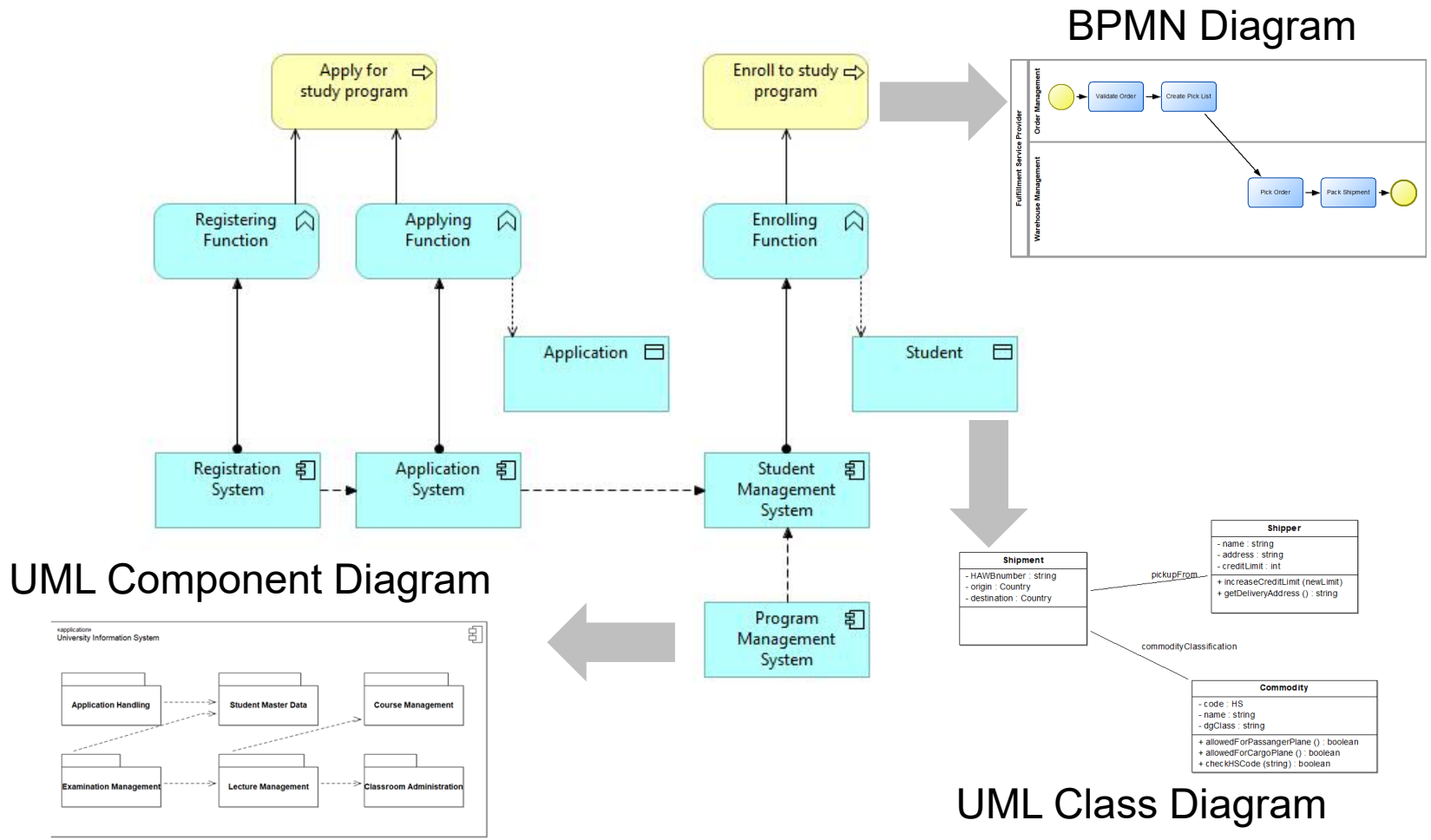


ArchiMate – Typical Elements

Aspects and layers correspond to typical elements in IS architecture.



ArchiMate – … and other Standards



FRANKFURT UNIVERSITY OF APPLIED SCIENCES

ArchiMate 3.2 Resources

- ArchiMate textbook
  Wierda, Gerben: "Mastering ArchiMate." R&A, 2017

- ArchiMate 3.2 reference cards
  https://www.opengroup.org/sites/default/files/docs/downloads/n221p.pdf

- Archi (ArchiMate modelling tool)
  https://www.archimatetool.com/

- ArchiMate 3.2 specification
  https://pubs.opengroup.org/architecture/archimate32-doc/index.html

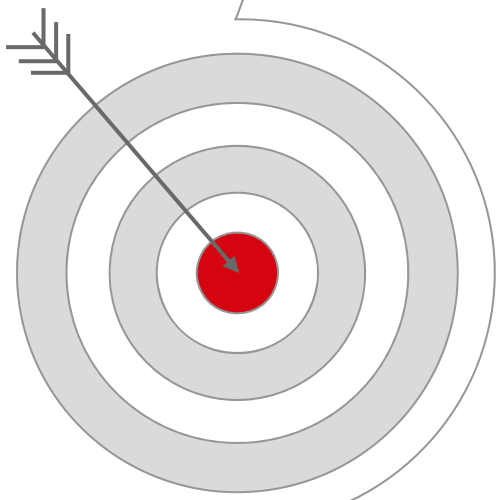
FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Architecture and Integration

Determining Architecture Quality

Fachbereich 2 Informatik und Ingenieurwissenschaften

Learning Objectives: Determining Architecture Quality



Explaining properties of a good architecture

Explaining typical criteria for evaluating an information system architecture

Assessing the quality of an architecture

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

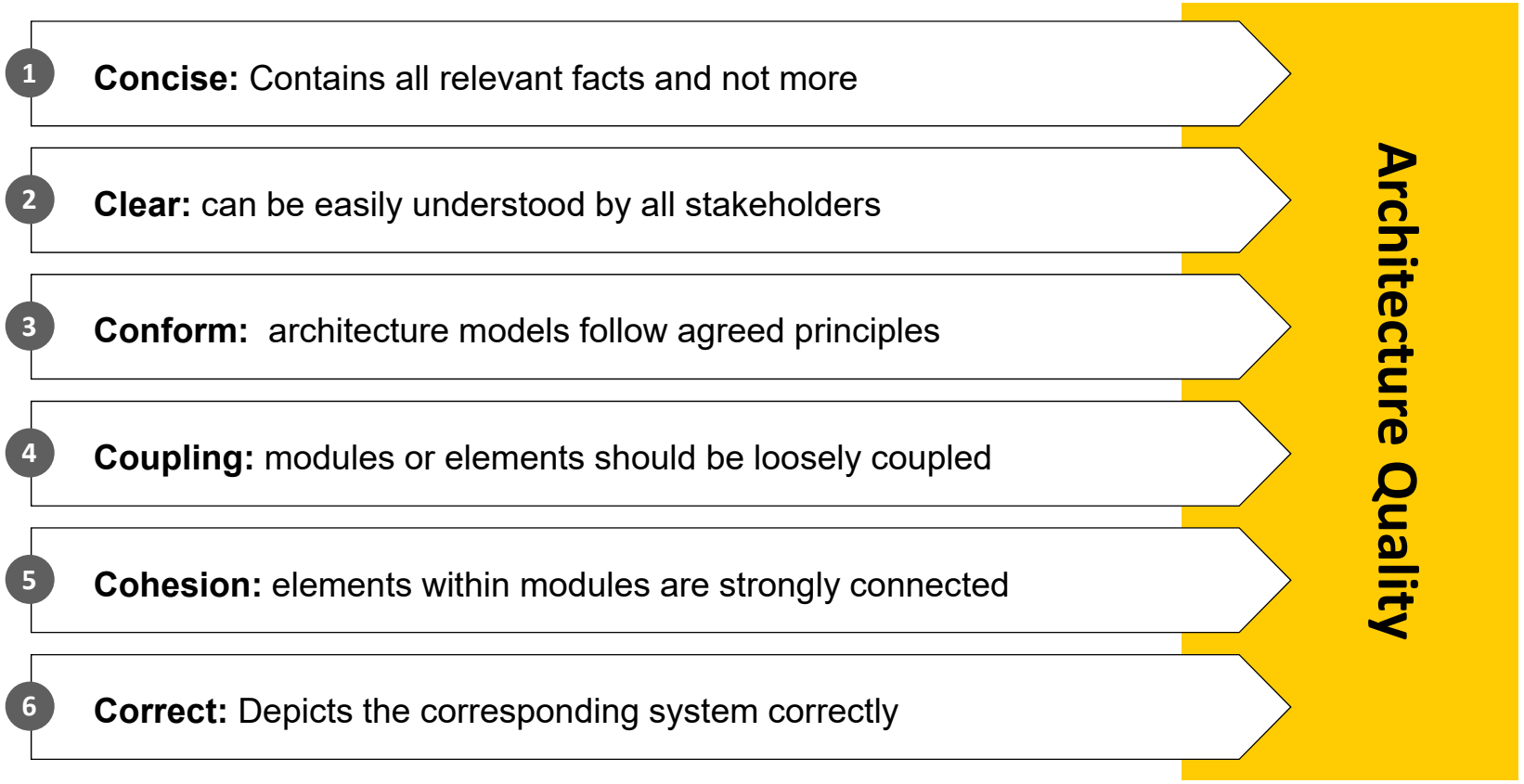
Particify

How would you rate this architecture? Likert, 5 examples

Which criteria can we use for assessing the quality of an architecture?

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Quality Aspects of an Architecture



FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Benefits of Concise Architecture

Issues from exhaustive architectures	Benefits of concise architectures
Architecture • Contains unnecessary elements • Explosion of relationships • Architecture hard to change	**Architecture** • Focuses on required elements • Only required relationships • Architecture easier to maintain
Architecture model • Contains irrelevant information • Effort for modelling complete information • Effort for keeping information up-to-date • Hard to find relevant information	**Architecture model** • Focuses on relevant information only • Reasonable effort for modelling • Easy to keep models up-to-date • Model shows basic information

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Benefits of Clear Architecture

Issues from chaotic architectures	Benefits of clear architectures
Architecture • Missing structure • No common patterns • Plethora of relationships • Architecture hard to change **Architecture model** • Model hard to read • Model is hard to maintain • Relevant information hard to find	**Architecture** • Clear structure • Follows common principles • Contains relevant relationships only • Architecture easier to change **Architecture model** • Models tend to be easy to understand • Models can be changed easier • Common structure helps finding information

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Conformance: Example Principles

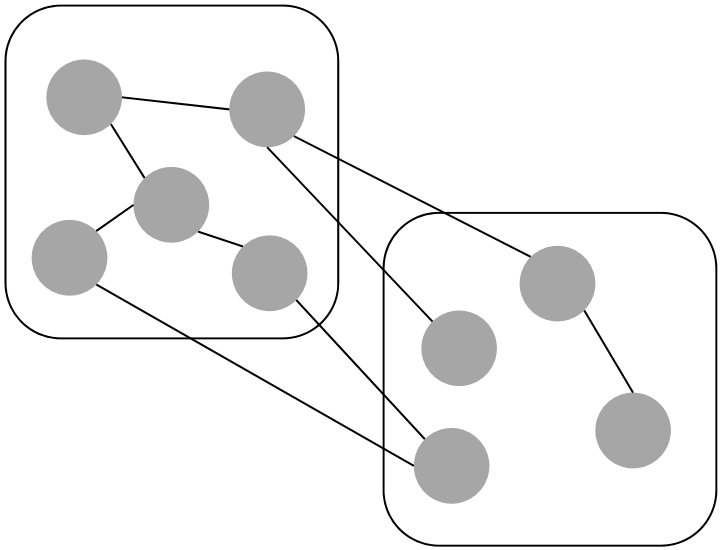
Principle	Descriptions
IT systems adhere to open standards	• No individual or proprietary solutions • Open to communicate with partners and systems • Reuse of common knowledge
Software applications are preferably open source	• No vendor lock-in • Software can be customized to individual needs • Maintenance together with community
Data is provided by the source	• Enter data where it gets available first • Those who generate data, enter the data
Each kind of data is stored in single dedicated IT system	• Avoid data redundancies • Reduce data quality issues • Provide single source of truth

Particify

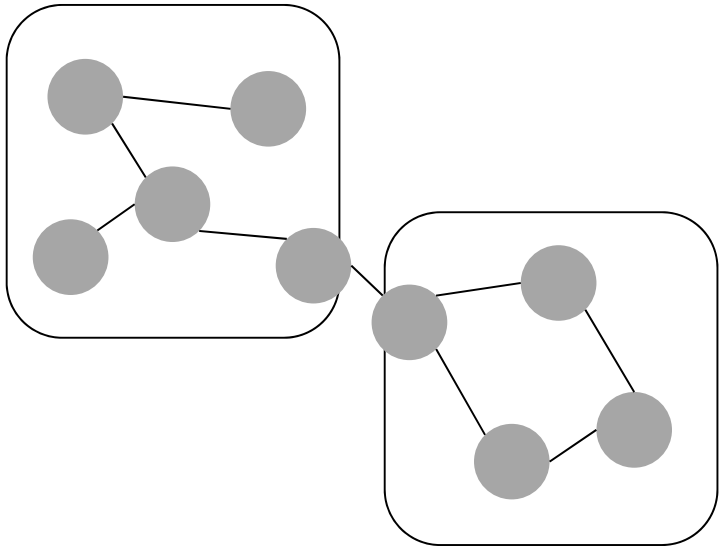
Which kind of principles could you imagine for an architecture?

Coupling: Overview

- *Coupling* refers to the number of dependencies between two or more systems.
- Low coupling is usually recommended as it fosters modularisation



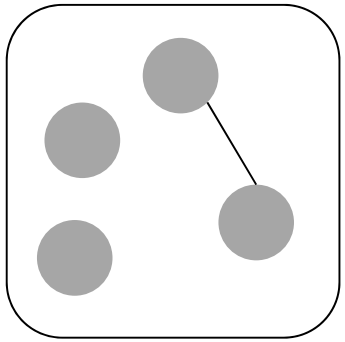
High coupling



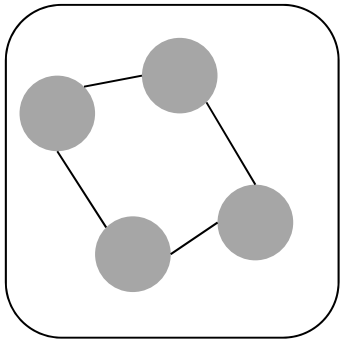
Low coupling

Cohesion: Overview

- *Cohesion* refers to the number of dependencies between element s within one system.
- High cohesion is usually recommended as internal elements are strongly related
- Low coupling usually determines high cohesion (and vice versa)

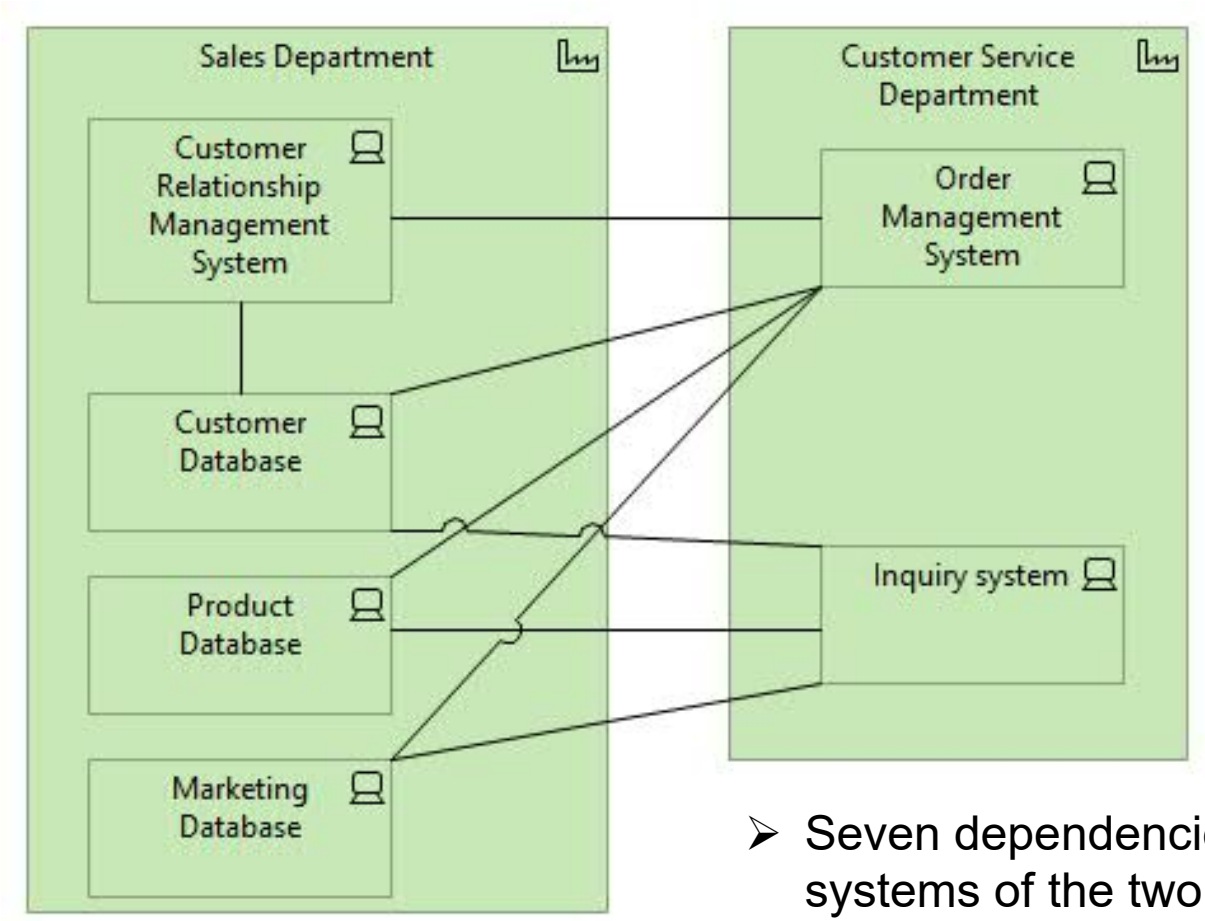


Low cohesion



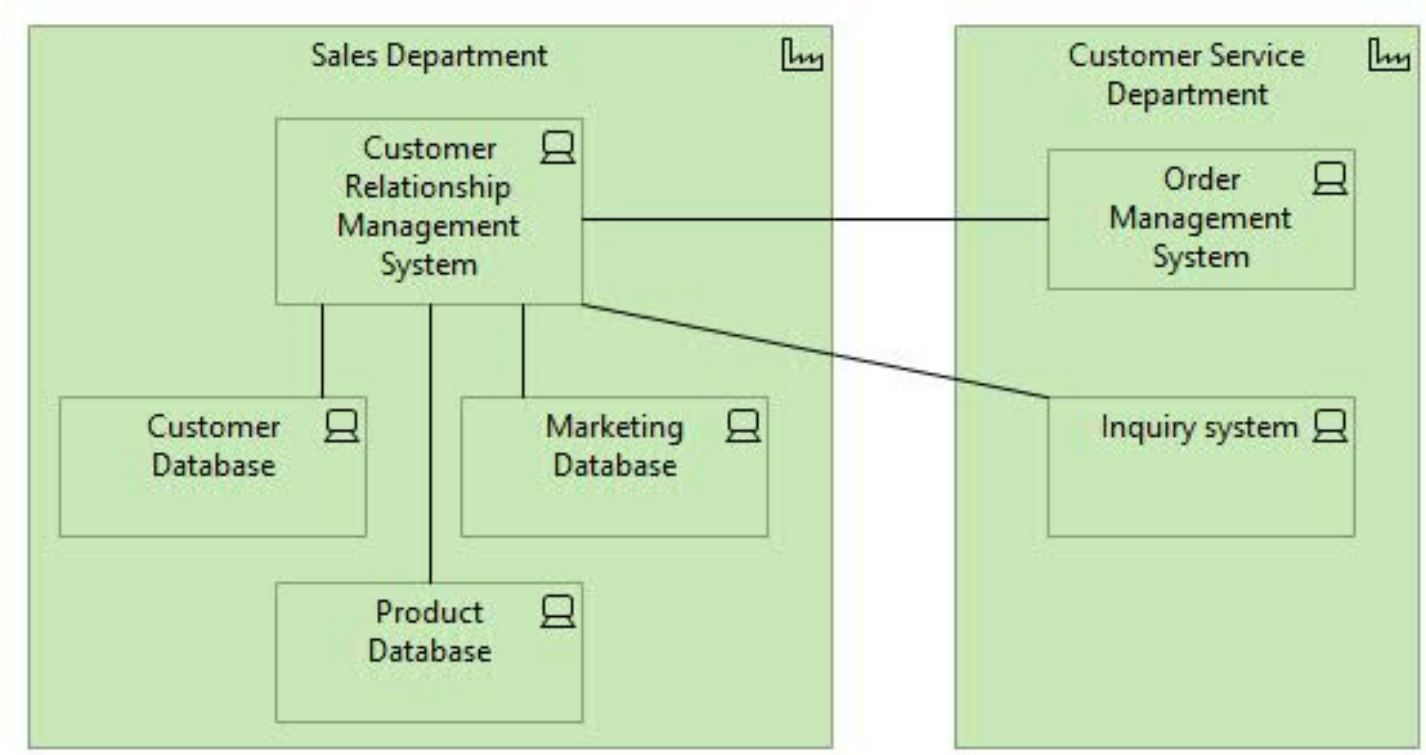
High cohesion

High Coupling: Example



- Seven dependencies between IT systems of the two departments.

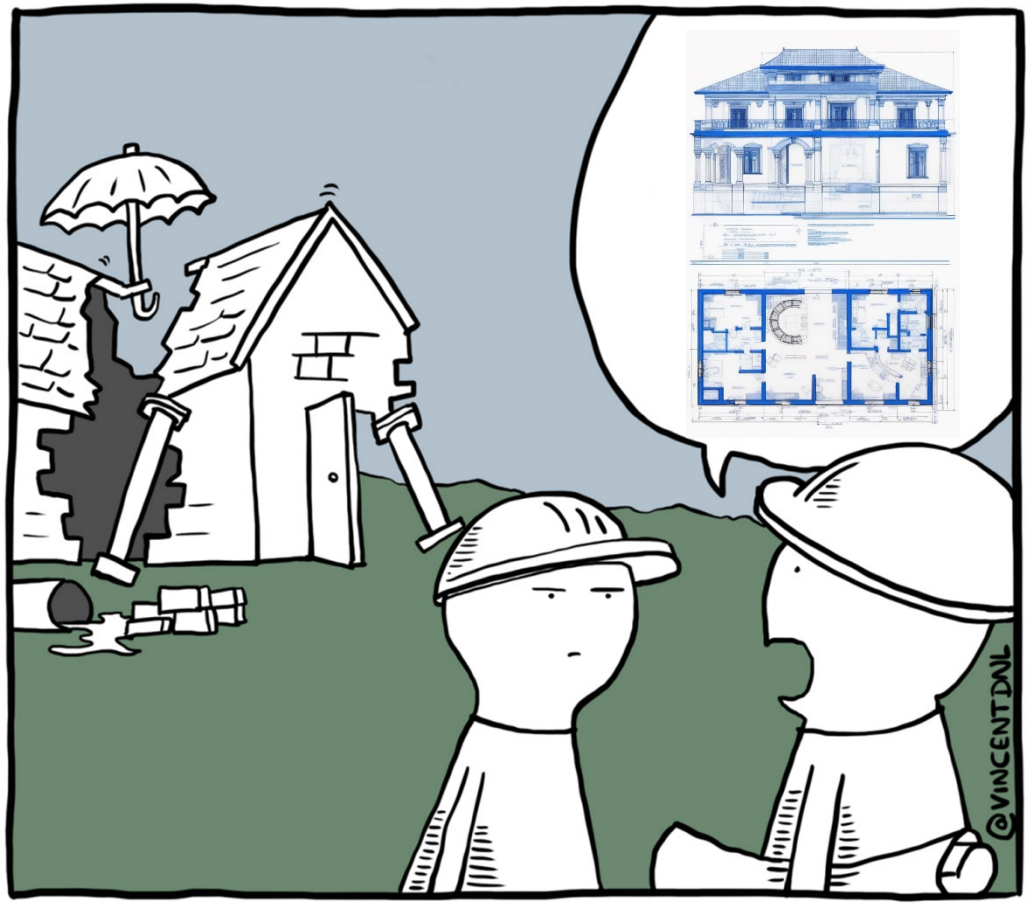
Low Coupling: Example



- Two dependencies between IT systems of the two departments
- High cohesion

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Correctness of an Architecture Model



- An architecture model has to depict the corresponding architecture as it is
- A wrong picture will not help with fixing issues but just hides them

Correctness of an Architecture Model

A correct **architecture model** needs to represent

- Elements as they exist in reality
- Complete set of relationships
- Correct information concerning elements and relationships
- Underlying principles and assumptions
- Any weakness (for as-is models)
- Improvements (for to-be models)

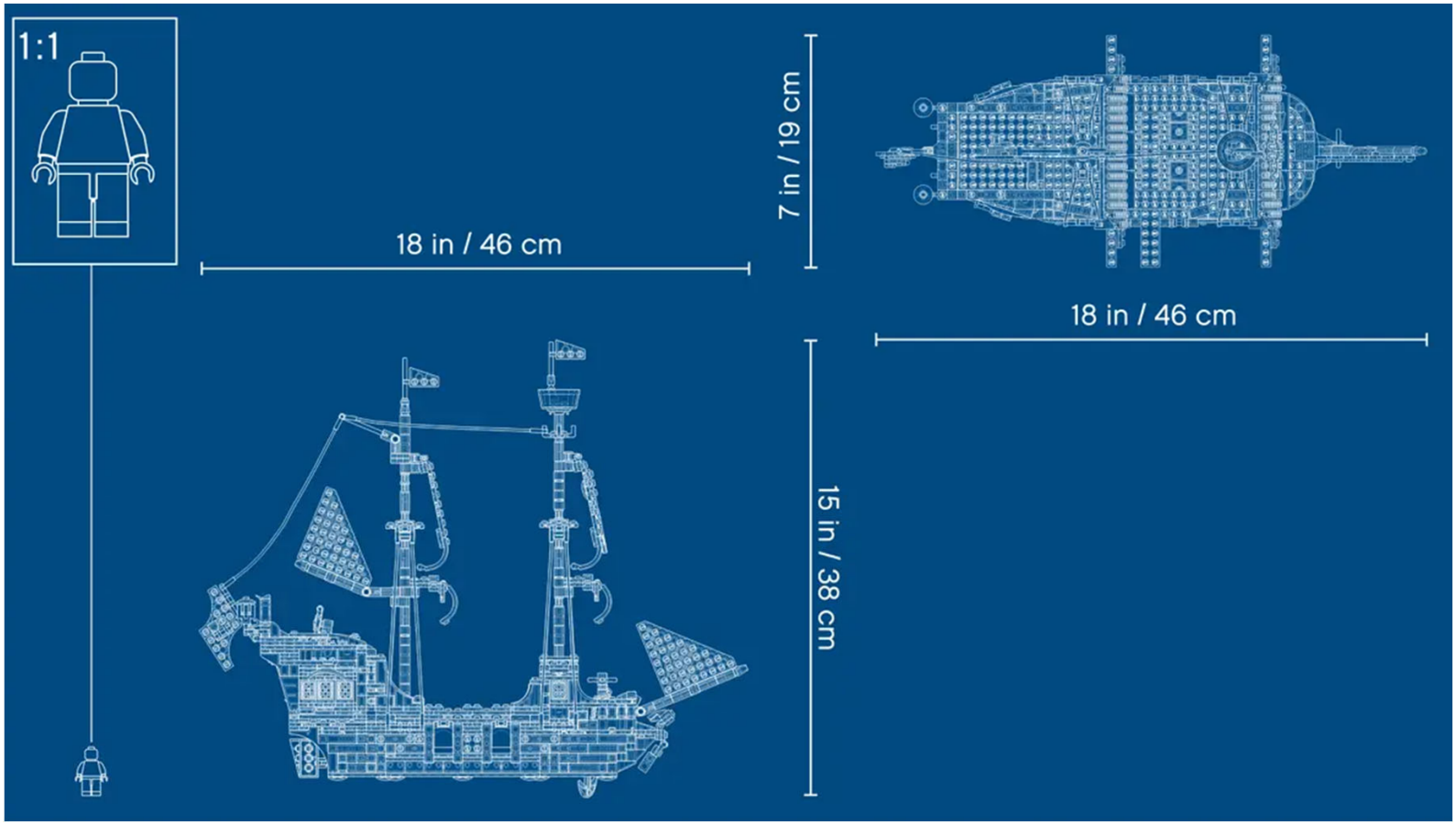
➢ Otherwise, it cannot be used for managing architecture!

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

What is this Architecture About?

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Complete Picture: Architecture of a Pirate Ship



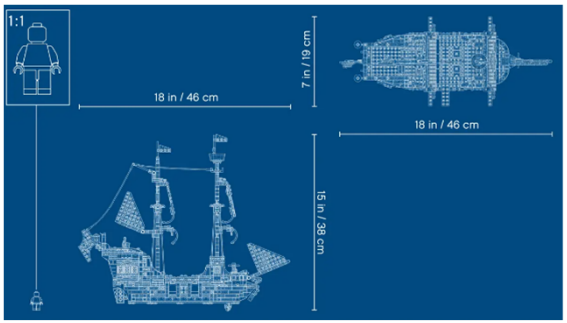
FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Purpose Matters: Architecture of a Pirate Ship

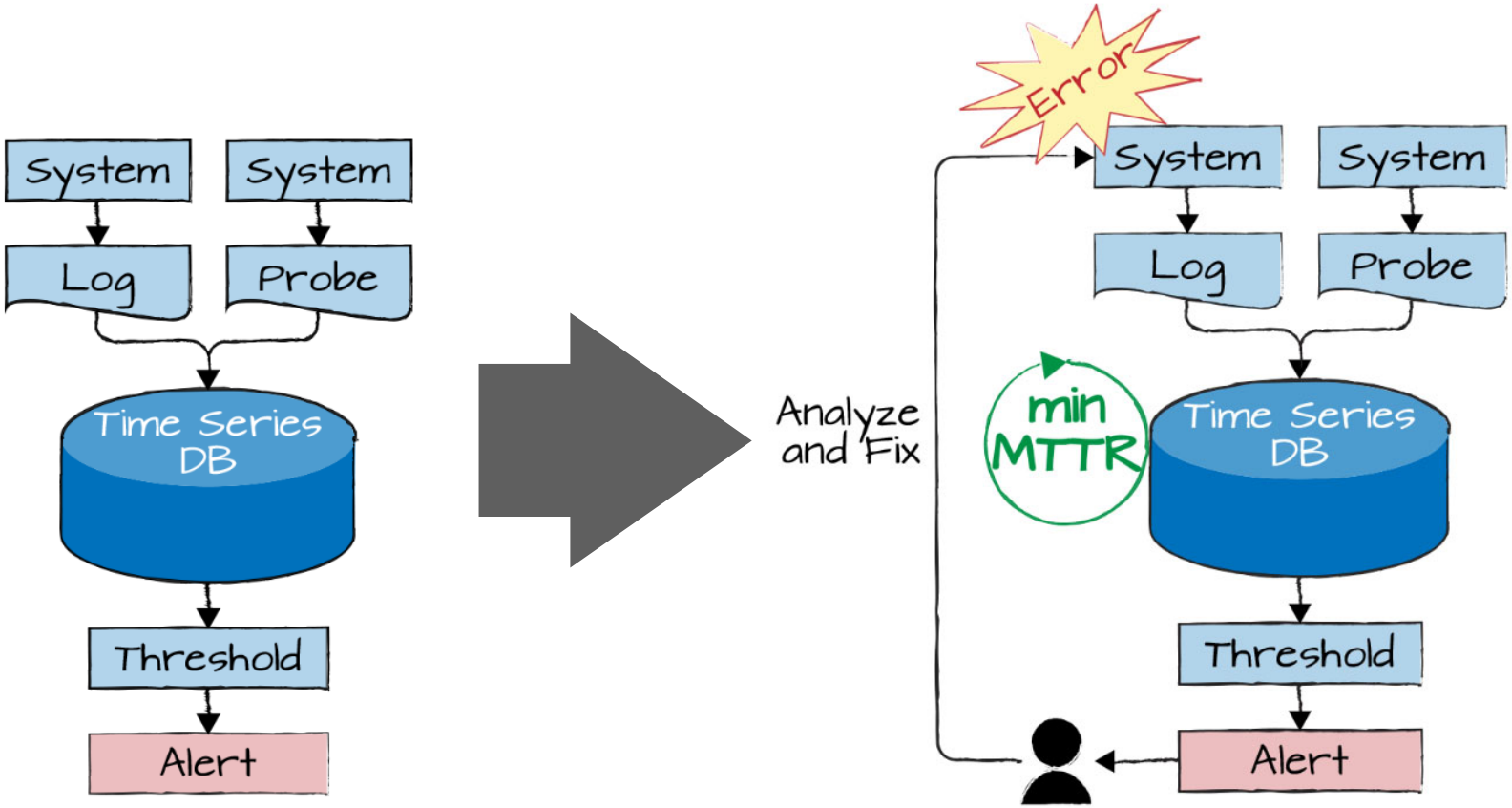
FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Architecture Follows a Purpose

Elements of an architecture → Elements and their relationships → Purpose of an architecture

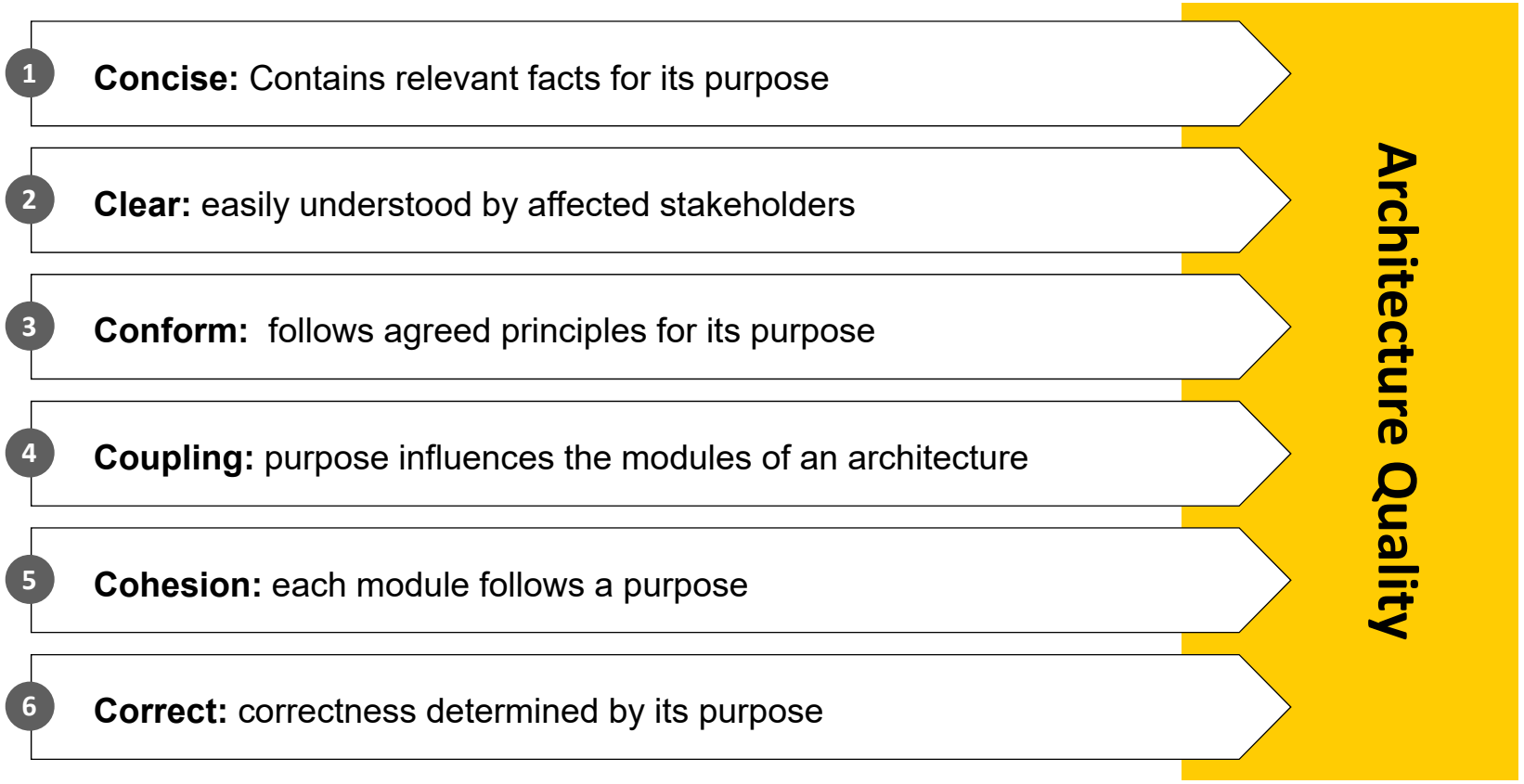


Show the Pirate Ship



Source: Hohpe: The Software Architect Elevator, 2020, pp.154

Quality Aspects with respect to Purpose



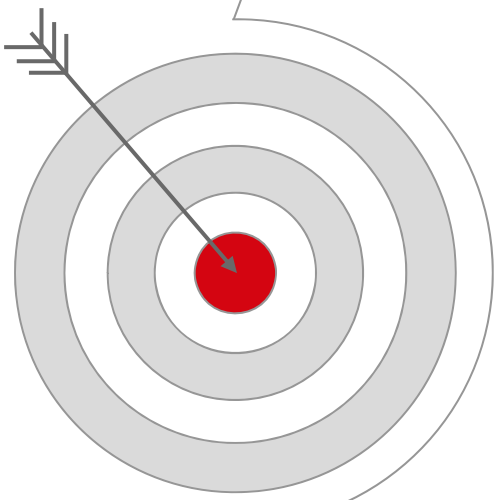
FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Architecture and Integration

Integration

Fachbereich 2 Informatik und Ingenieurwissenschaften

Learning Objectives: Integration



Explain integration paradigms based on

- Data
- Functions
- Process

Describe integration with respect to number of partners:

- Point-to-Point
- Multiple partners

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Particify

We have two systems: A customer relationship management system (CRM) and an order management system (OMS). Customer data is only stored in the CRM system. Customer data is also needed for executing order? How can we get customer data into the OMS?

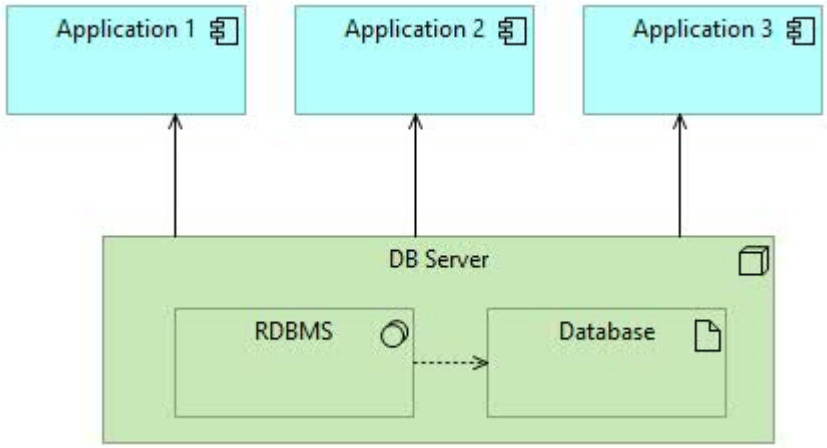
Integration Based on Data

Motivation	Prerequisites	Solution
• Several applications in an organization use the same data • Data should not be entered manually into each system • Example: Product data required for marketing, sales and manufacturing	• Common data format – Types and attributes – Representation • Communication channel for data exchange	• Common database • Automatic data exchange

Integration Based on Data: Solutions

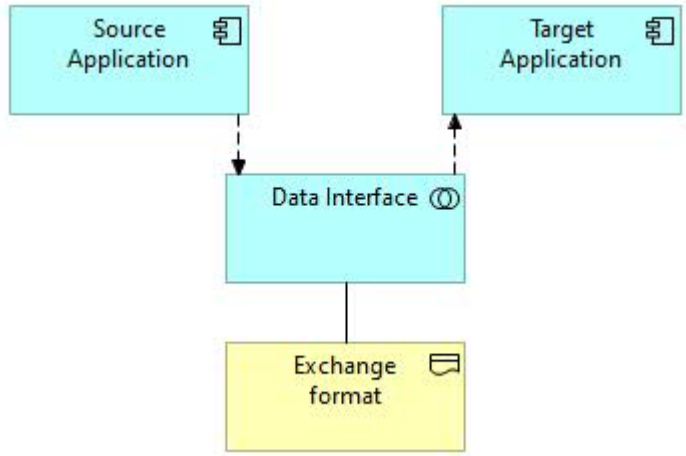
Integration with shared database

- Data is stored in single database
- Format determined by database schema
- Individual applications can read
- Only one application should create data



(Automated) Data exchange

- Data is stored in each applications
- Data exchanged through interface (individual software component)
- Exchange format required



FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Particify

We are in DHL: There is one system that can create shipping labels (the ones attached onto the parcels). We now want to introduce a web shop system that can also create shipping labels. Do we need to implement the same functionality twice? Which possibilities are you aware of?

Integration Based on Functions

Motivation

- Functionality implemented by an application can be re-used by others
- Example: Parcel tracking used by web portal, customer service, operations

Prerequisites

- Common data format
  - Types and attributes
  - Representation
- Communication channel for service invocation

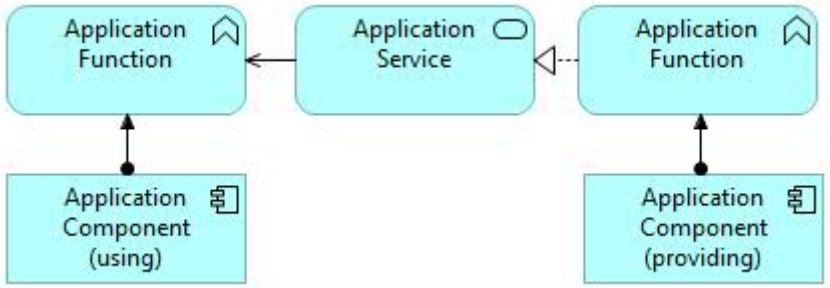
Solution

- Software service
  - Provided at endpoint
  - Having dedicated name
  - Data in- and output

Integration Based on Functions: Solution

Application service

- An application (providing) implements a function
- Function is offered as an application service
- A function implemented by another application (using) is served by the service



Example technologies

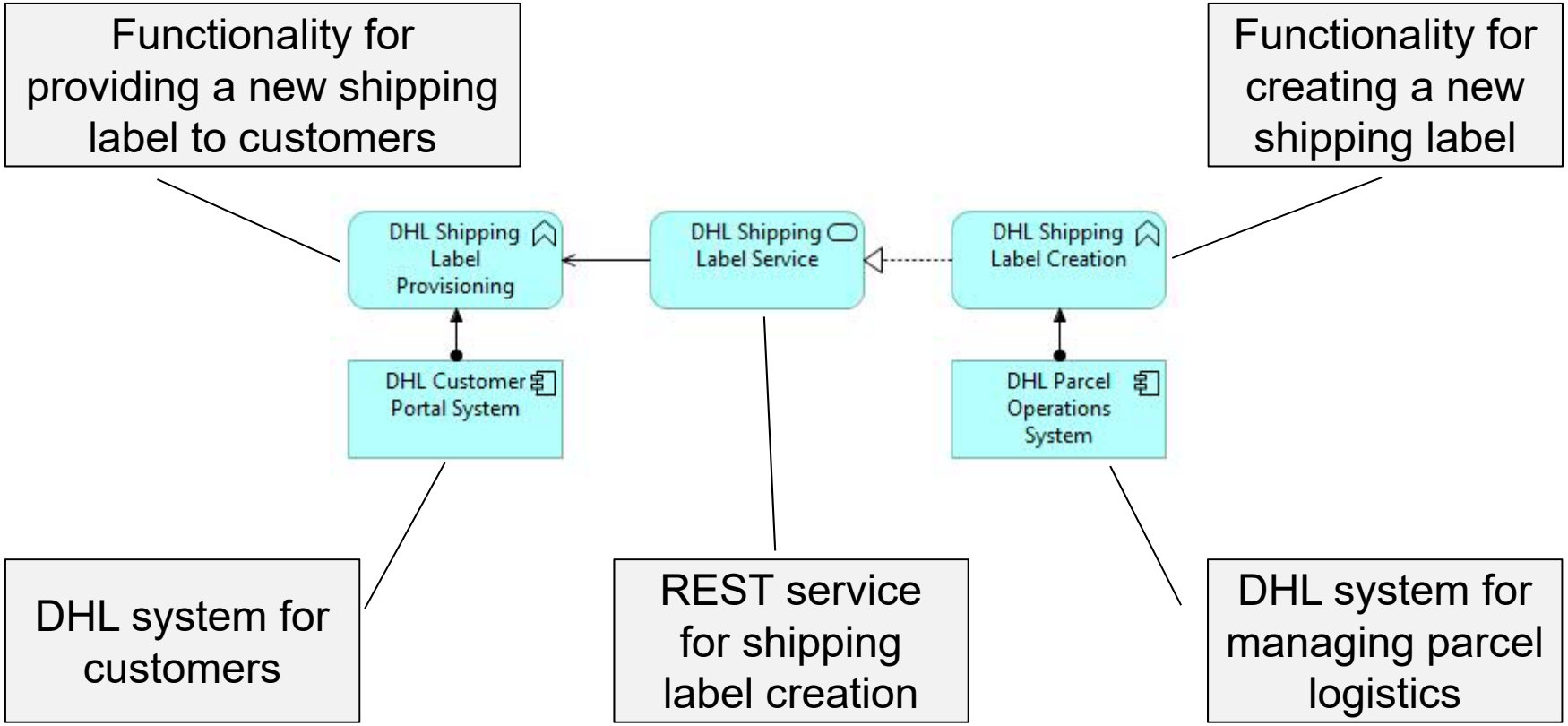
Programming languages

- Remote Method Invocation (RMI)
  - Java
- Remote Procedure Call (RPC)
  - Go, Python
- Remote Function Call: SAP

Language-independent

- Webservices (REST)
  - HTTP
- Service-oriented Architecture (SOA)

Integration Based on Functions: Example

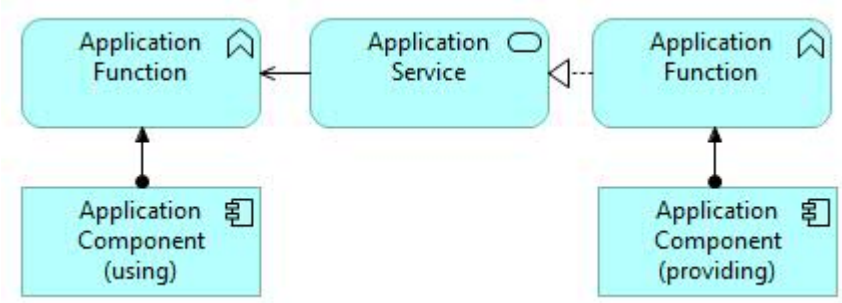


Integration – Definition

Integration: The term *Integration* in the context of Information Systems refers to the combination of several applications so that they can share data and use common functionality. Integration is driven by business processes creating business value.

Integration: Realisation as Function

- Integration usually requires changing a software system (i.e. programming)
- Changes are conducted within a software development project



Program code for
- Invoking the remote service (Application Service)
- Processing the result

Program code for
- Functionality (Application Function)
- Service interface (Application Service)

Integration Based on Processes

Motivation

- Processes define workflow
- Different software applications are required for workflow
- Example: Order management

Prerequisites

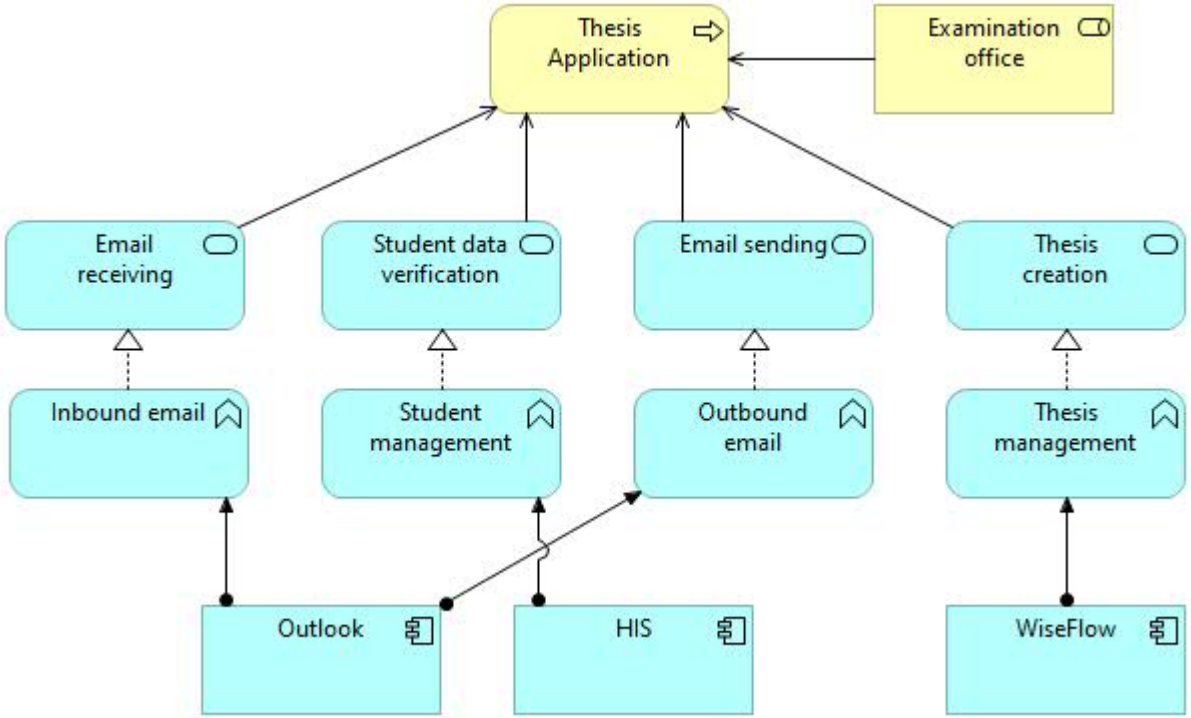
- Common data format
  - Types and attributes
  - Representation
- Communication channel for service invocation
- Processes well-defined

Solution

- Process automation
  - Process engine executes process
  - Uses existing software systems (services)

Integration Based on Processes: Example

- Before writing their final thesis, students need to send an application
- The examination office executes the process by using different applications



FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Integration Based on Processes: Overview

- Process defines workflow execution (e.g. via BPMN diagram)
- Process automation tool executes process as defined
  - Triggering activities of different actors
  - Invoking several software applications
  - Coordinates data exchange between actors and applications
  - Combination of manual and automated activities possible
- Various systems available on the market, e.g.
  - Camunda BPM (open source & commercial)
  - SAP Business Workflow (commercial)
  - jBPM (open source)

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Integration: Summary

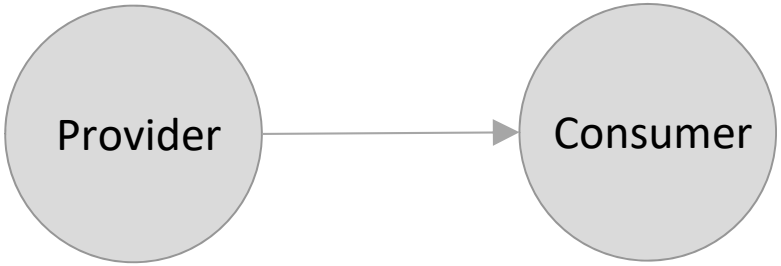
Type of integration	Data-oriented	Function-based	Process-based
Central concepts	• Data	• Function • Service	• Process • Event
Prerequisites	• Data format	• Function • Data format	• Process • Function • Data format
Examples	• Database • File exchange	• RMI • REST	• Workflow Management

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Particify

How many systems can be integrated?

Partners: Point-to-Point (P2P) unidirectional



- Two systems involved
  - Provider: source for information
  - Consumer: drain for information

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

P2P unidirectional: Examples

Hardware

- MP3 player ⇨ headphones (music)
- Keyboard / mouse ⇨ computer (input)
- Computer ⇨ display (screen)

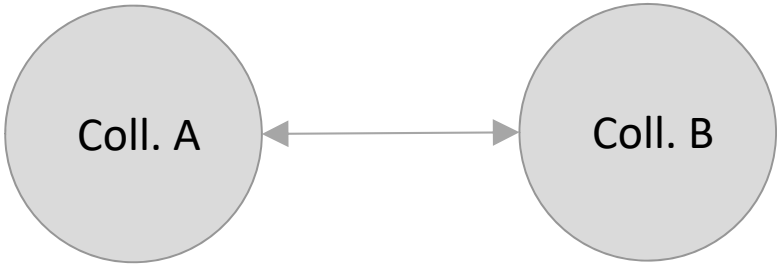
Software

- Address book ⇨ email client (contact)
- http server ⇨ http client (web page)

Other

- Voyager spacecraft ⇨ base station (pictures)

Partners: P2P bidirectional



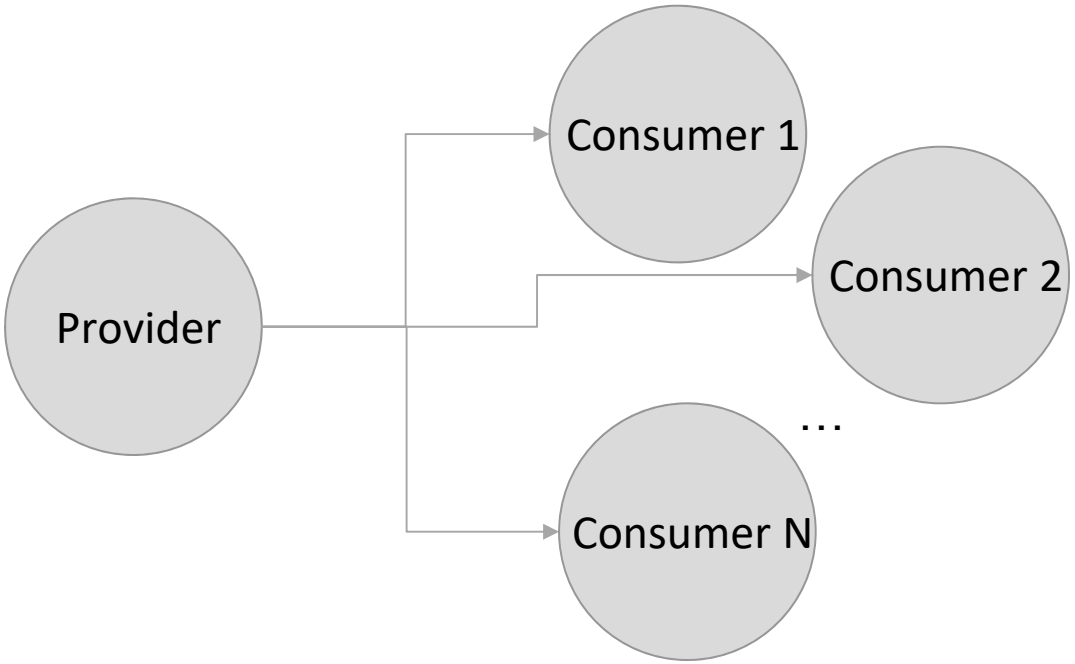
- Two collaborators (Coll. A and Coll. B) involved
- Both exchange information

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Particify

Could you imagine examples for a bidirectional P2P integration?

Partners: Broadcast

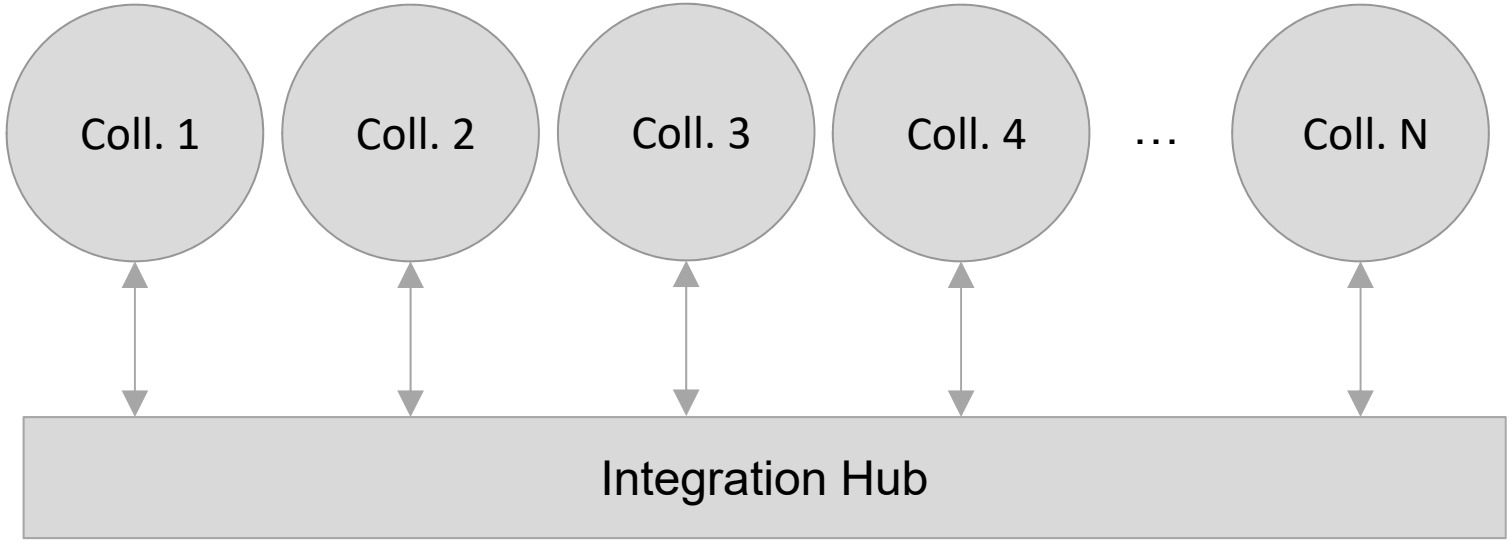


- Broadcast: Provider is distributing information to several consumers
- Publish-Subscribe: Consumers can register for receiving information

Particify

Could you imagine examples for a broadcast integration?

Partners: Hub



- Several collaborators are connected to a single hub
- A collaborator can communicate with any other

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Particify

Could you imagine examples for a hub integration?

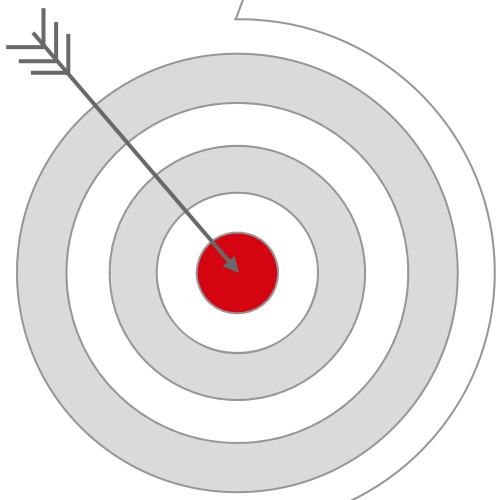
FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Architecture and Integration

Integration Technologies

Fachbereich 2 Informatik und Ingenieurwissenschaften

Learning Objectives: Integration Technologies



Lorem

Ipsum

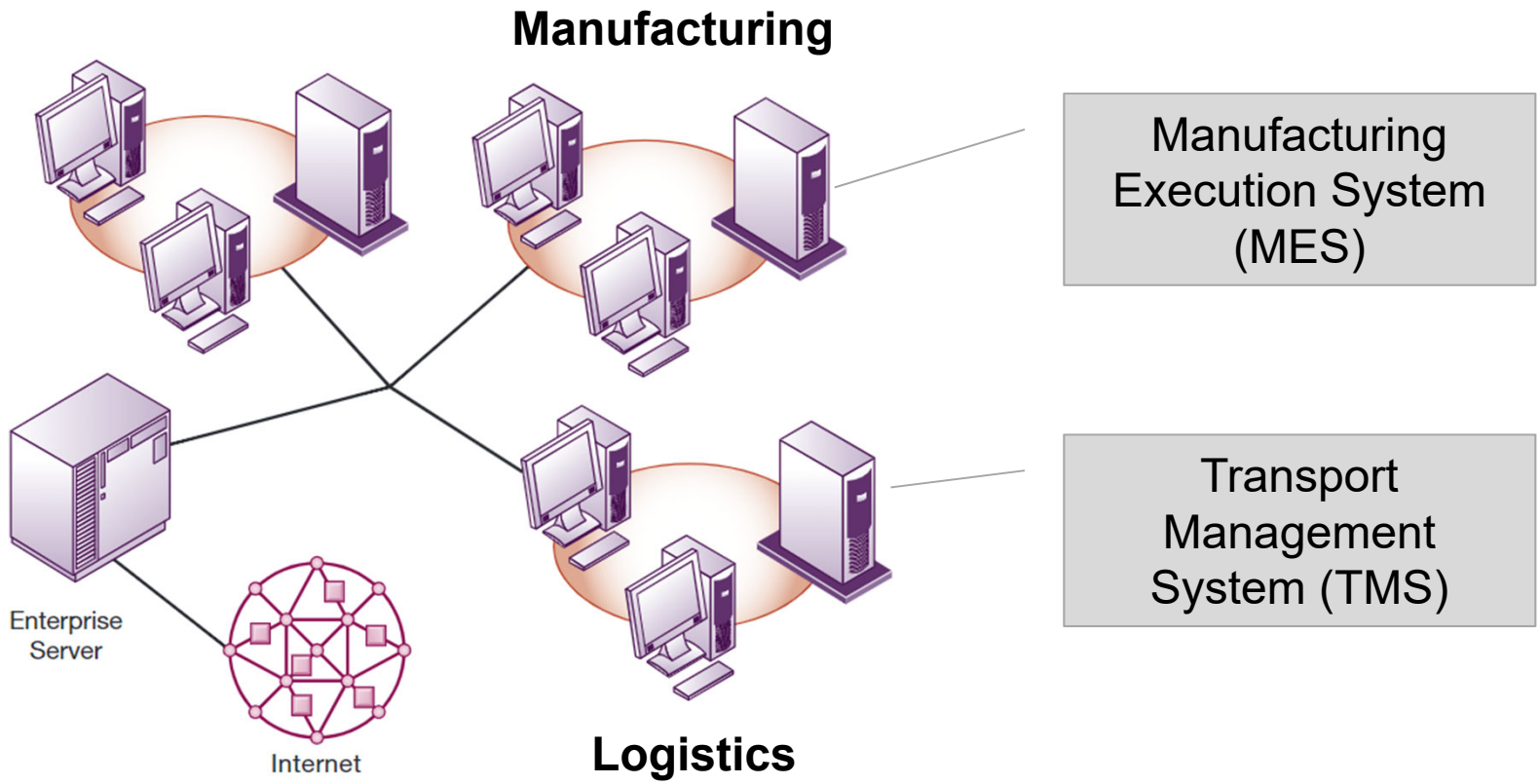
FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Particify

Which kind of technology do we need for integrating software applications?

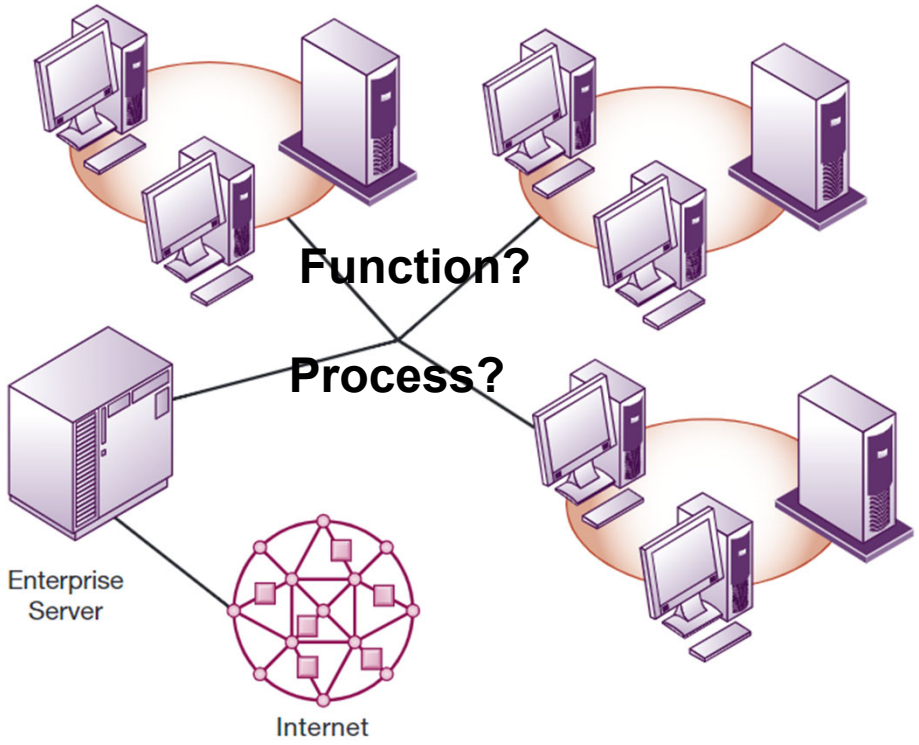
Integration: Computer Network

Computer networks are required for connecting computers. Example:



Integration: Beyond Computer Networks

Basic technology like operating system (OS) or computer networks is not sufficient for function- or process-based integration.



Integration: Benefit

Applications are no isolated islands but need to collaborate in order to support E2E processes.

Sharing data

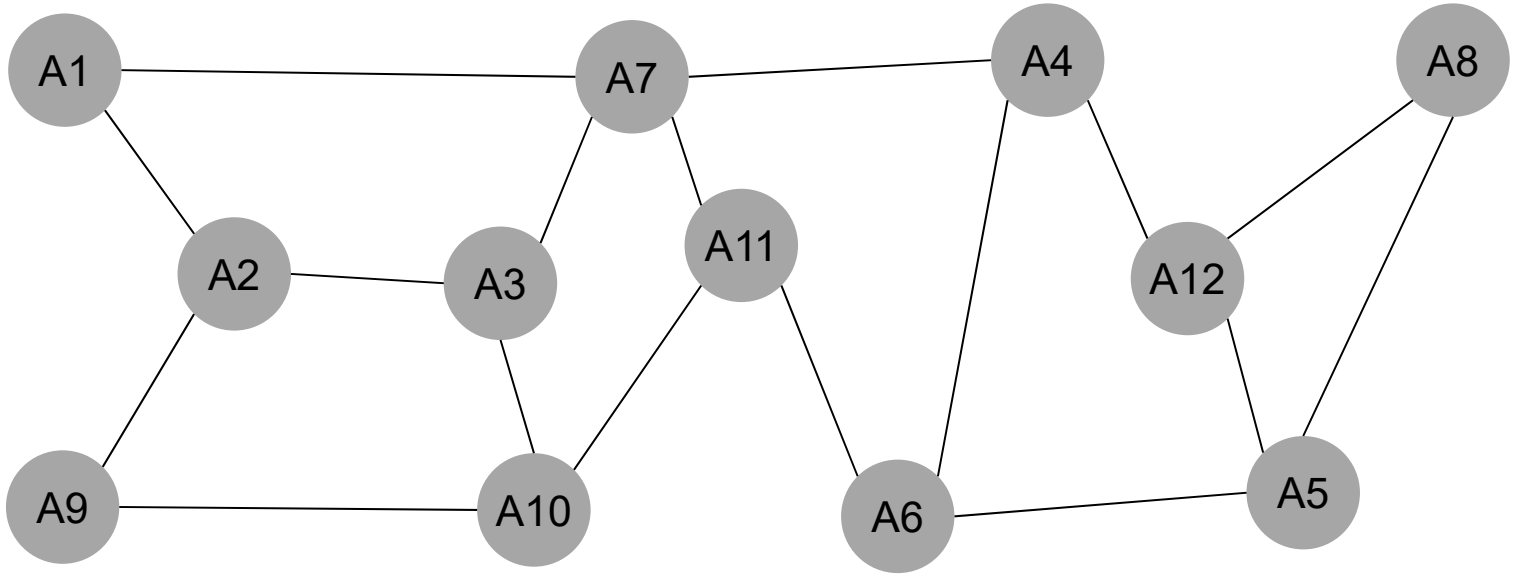
- Common database
  - Standardised data model
  - Consistency managed by Database Management System (DBMS)
  - Concurrency control
- Data ex- and import
  - Easy to implement
  - Different data models
  - No overall consistency

Re-using functionality

- Benefits
  - No redundant implementation
  - Changes only once
  - Functional specialisation
- Requirements
  - Common data model
  - Application Programming Interface (API)
  - Integration infrastructure

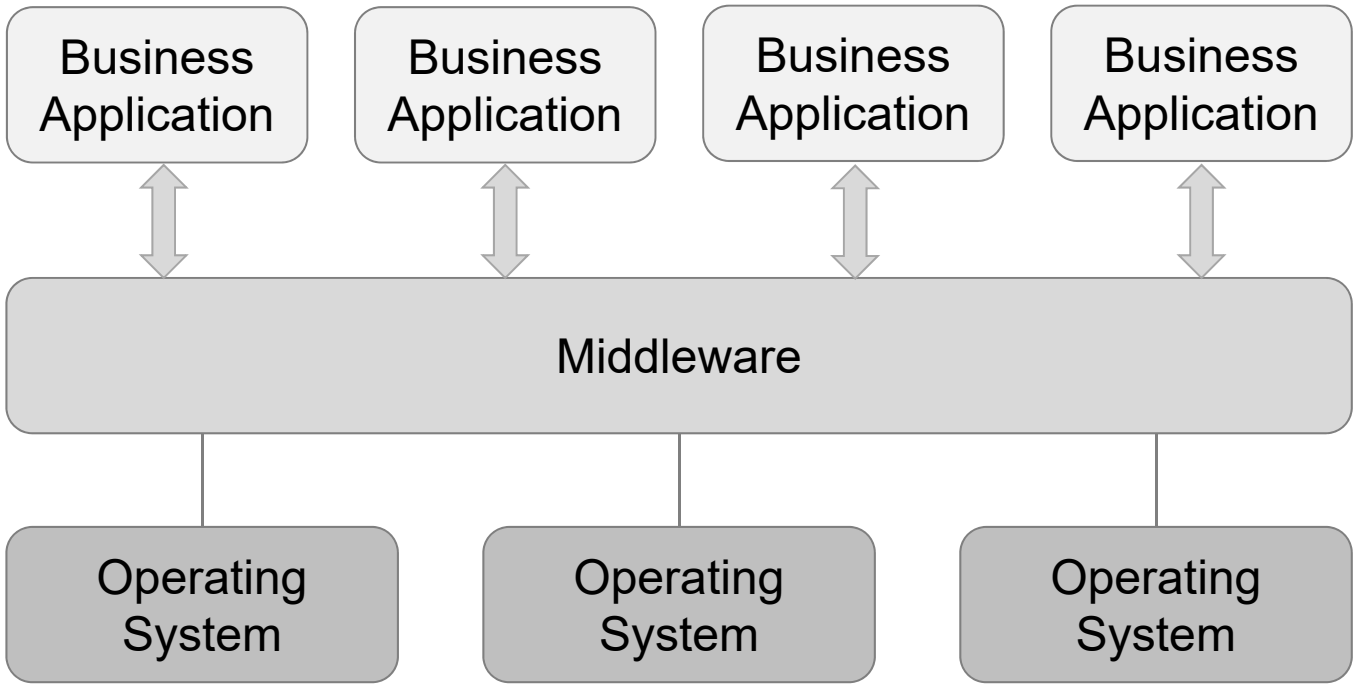
Integration: Challenge

- IS architectures tend to grow large (e.g. several hundreds of applications)
- IS architectures tend to get complex (plethora of interactions between applications)



Middleware: General Idea

- Middleware usually runs on different operating systems

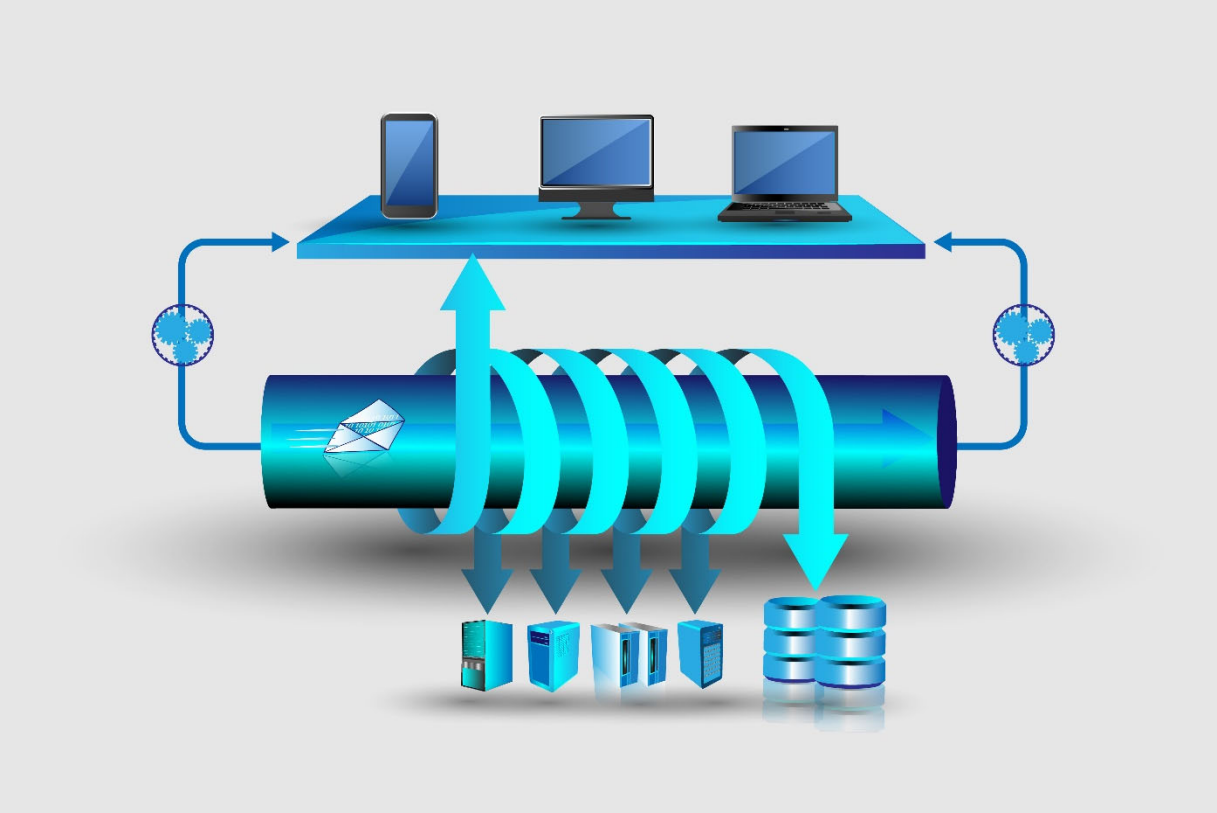


Middleware: Integration

Middleware: The term (*Integration*) *Middleware* refers to a software application which can be used for integrating business software application. It is located between the operating system and the business applications (hence, *middle*ware).

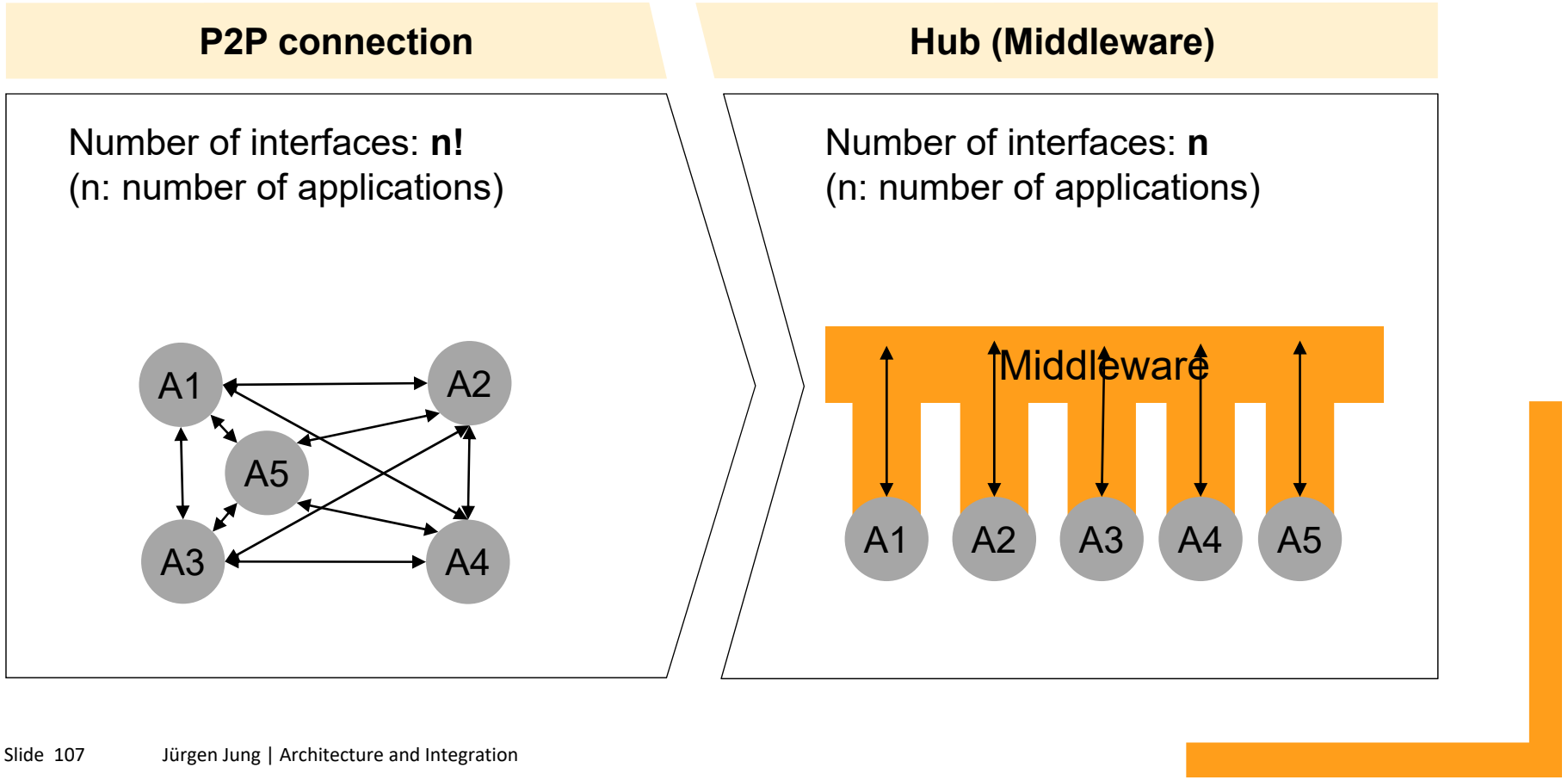
Middleware: General Idea

- Middleware decouples software applications from underlying system software



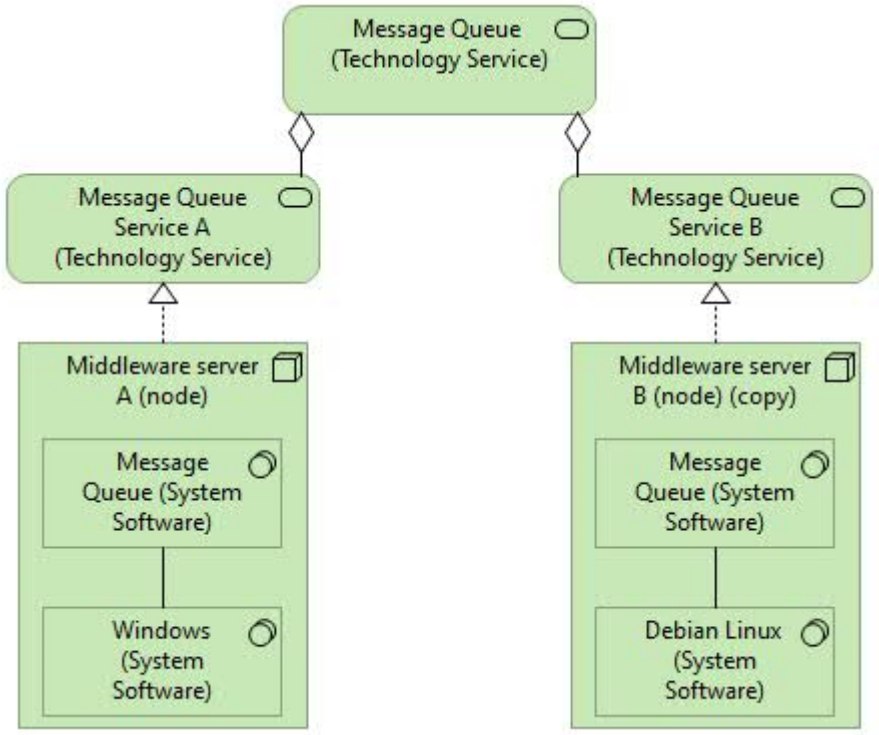
Middleware: Benefit

- Middleware reduces the number of dependencies



Middleware: Example

- One middleware service can consist of software installed on several machines



FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Particify

Which kind of middle can be used for data-oriented integration?

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

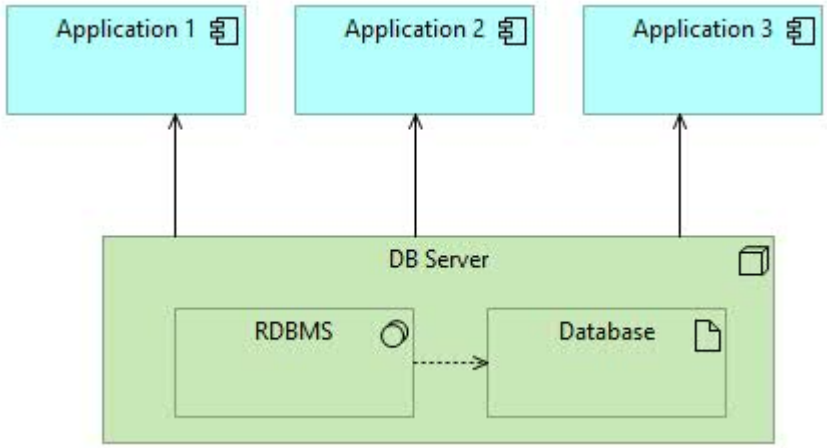
Middleware: Example Technologies

Data	• Shared file storage • Database Management System (DBMS) • Message Queue (MQ)
Function	• Application server (web server) • Service-oriented Architecture (SOA) • Enterprise Service Bus (ESB)
Process	• Workflow Management System (WfMS) • Process Automation • Robotic Process Automation

Integration: Technologies for Data

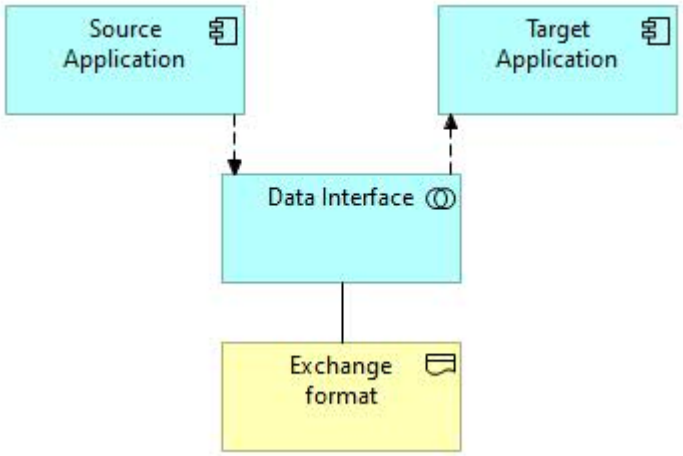
Shared database

- Centralised approach
- DB server provided in the network
- Applications access data remotely



Data exchange

- Decentral approach
- Data is stored as file
- Files distributed to other applications



Particify

Which of the following middleware technologies are you aware of?

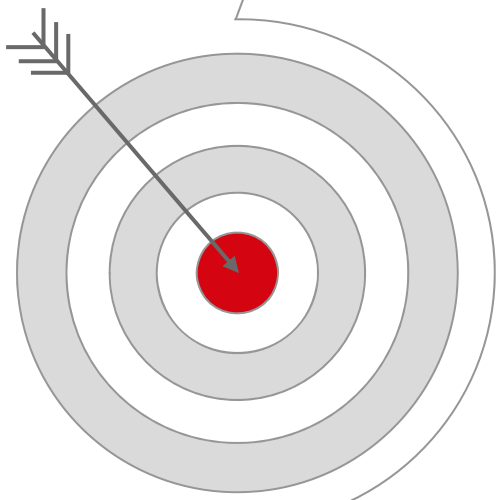
FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Architecture and Integration

Architecture Principles

Fachbereich 2 Informatik und Ingenieurwissenschaften

Learning Objectives: Integration Technologies

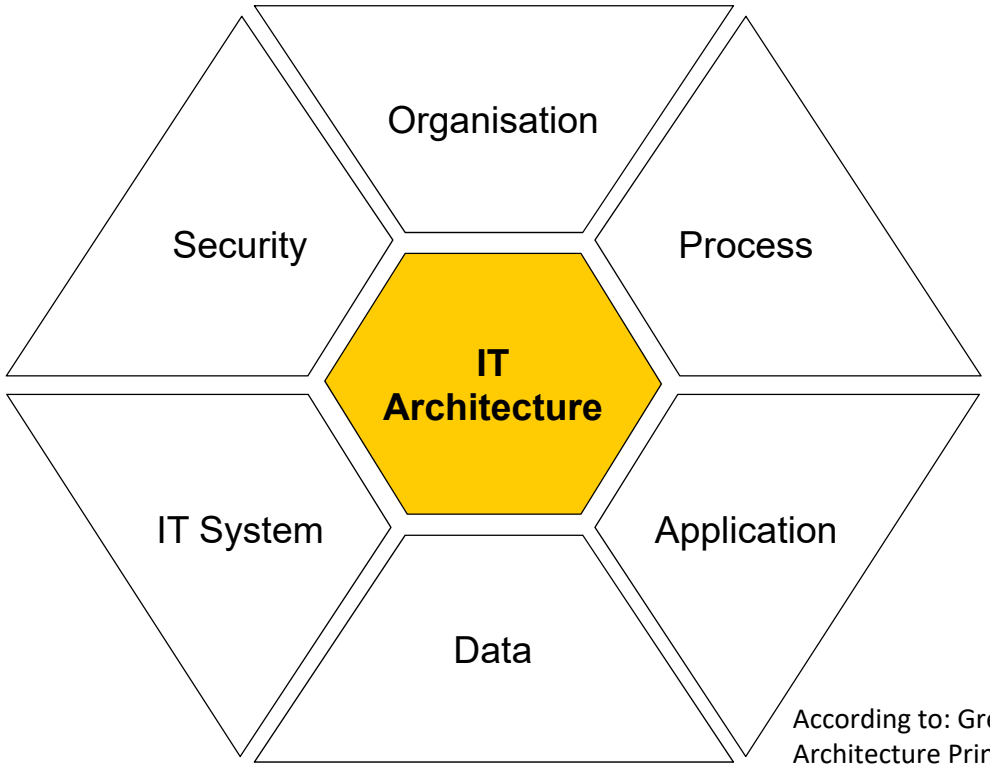


Lorem

Ipsum

Principles: Overview

Architecture principles aim at providing guidance (based on best practices) for designing good architectures in a corporate environment, covering different aspects.



Principles aim at ensuring

- Provision of functionality
- Availability of IT systems
- Performance as required
- Security and data protection
- Scalability with respect to business needs
- Compliance of IT
- Sustainability (technologically and economically)

According to: Greefhorst, D.; Proper, E.: Architecture Principles. Springer, 2011

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Principle: Standardisation

IT Systems Are Standardized and Reused Throughout the Organization

Rationale	Implication
• Cost reduction by preventing redundant investments • Exploitation of economies of scale • Focussing attention, resources, knowledge and investments due to enterprise-wide standards	• Determine standards for any IT functionality • IT systems do not provide functionality that overlaps with other IT systems. • IT systems are reused throughout the organization by all business units. • Concessions may be needed in user requirements

Principle: Availability

IT Systems Are Available at Any Time on Any Location

Rationale	Implication
• People perform work at various locations (office, at client, home) • Work at various times (day and evening) • Mobile employees: No fixed office space and facilities (e.g. workstations) required	• Software is server-based, allowing access from all locations. • Strong authentication services are available to ensure secure

Principle: Partner Integration

Integration with External IT Systems Is Localized in Dedicated IT Components

Rationale	Implication
• More efficient as interface costs are spent only once • Changes limited to one component only • First line of defence for security attacks • B2B integration is complex (special protocols)	• Dedicated integration middleware • Applications contain integration components • Integration components can be reused • Agreements needed for integration with other organizations

Principle: Partner Integration

IT Systems Communicate Through Services

Rationale	Implication
• Reuse of functionality • Assembling new solutions based on services • Shorter time-to-market	• Services are defined for all data and functionality • Services defined as reusable as possible • Hiding implementation details • Adhering to interface standards and formats • Services published in a service directory

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Architecture and Integration

Topics and Appendix

Fachbereich 2 Informatik und Ingenieurwissenschaften

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Topics – Architecture

Architecture -- Introduction

- Motivation (Is this an architecture?) Why architecture?
- Definition: Architecture
- Examples (von Neumann, Software, Client/Server, 3tier, …)

Elements of IS Architecture

- Computer Hardware
- Software and Platforms (OS)
- Definition: IT System
- ArchiMate

Measuring Architecture

- Coupling / Cohesion
- Metrics & Quality

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Topics – Integration

Integration -- Introduction

- Motivation
- Definition: Integration
- P2P, Broadcast, centralised vs. decentralised, Bus

Integration Technologies

- Computer Networking
- MEP
- Technologies: DBMS, MQ, ESB, WfMS, …

Architecture Principles

- Examples
- Organisational principles

-> Übung zu ADR; accso

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Topics – Business

Business functions and applications

E-Business

- Scenarios and principles
- Web services / API
- Organisation

Supply Chain Integration

- Scenarios and Definition
- Challenges (bullwhip)
- Organisation and Management
- SCOR

Enterprise Architecture

- Applications and processes
- Business Architecture