

Software Architecture

DAT220/DIT544

Truong Ho-Quang

truongh@chalmers.se

Software Engineering Division
Chalmers | GU



Software Architecting

Truong Ho-Quang

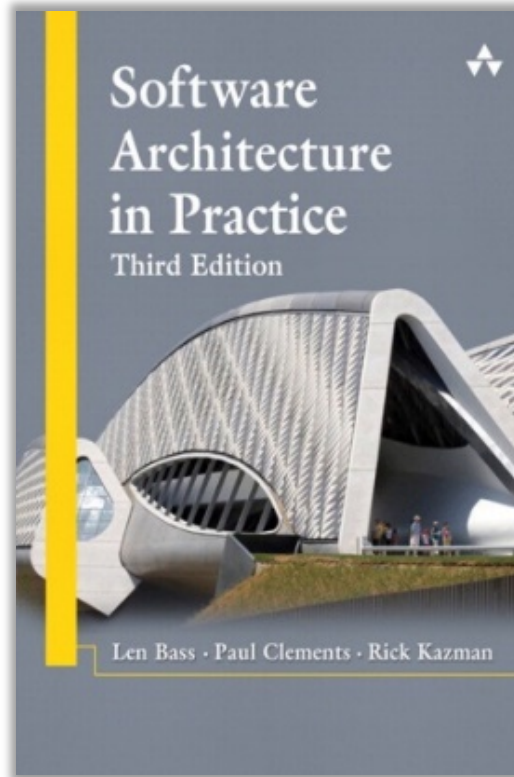
truongh@chalmers.se

Software Engineering Division
Chalmers | GU



Reading material for this lecture

- Ch 1 & 2
from Bass, Clements & Kazman: SW ARCH in Practice (3rd)



About Truong Ho (teacher)

- **Research:**

- PhD. in Software Engineering (CSE, Chalmers|GU)
- Research Interests: Software Architecture, Design & Modeling, Data mining, AI/ML

- **Industrial experiences:**

- System Architect at Volvo Group, Sweden (2019 – 2020)
- Research collaborations with Volvo Cars, Ericsson, and Tetra Pak (2014 – 2019)
- Software Architect at NIICS, Vietnam (2012 – 2014)
- Software Developer in various companies in Vietnam (2009 – 2012)

- **Coordinates:**

truongh@chalmers.se

- Room 476, floor 4, Jupiter building, Campus Lindholmen
- Phone +46 31 772 61 74



About Sam Jobara (teacher)

- Research:
 - Ph.D. in Computer Science and Engineering (CSE, University of South Florida, USA)
 - Research Interests: Testing and Fault Modeling, Information Security, and Business Continuity
 - Also interest in learning and cognitive theories.
- Teaching:
 - Multitude of courses in Computer Architecture, Compilers, TDA594, & DIT824
- Industrial experiences:
 - IT Consultant at SwedQ AB & Maxloyal AB
 - IT Consultant at Zain Telecom Group
 - Planning & Development advisor at GUST university
- Coordinates: jobara@chalmers.se



Teaching Assistant

Mazen Mohamad : mazenm@chalmers.se



Class Representatives

- Drop by or send me an e-mail

Goals of this course

Knowledge:

- concepts & terminology
- engineering standards & pragmatics
- modelling notations & analysis techniques
- ‘standard’ solutions
- ‘soft skills’ are important
- introduction of some research topics

Skills:

- design & documentation of architectures
- assessment of architectures

Goals of this course

Advanced:

- Quantitative analysis techniques
 - Performance
 - Availability & Reliability
- More tactics
- Guest lecture
- Conformance between Architecture and Implementation
- Less focus on requirements & quality

Not a goal

- State of the art technologies / hype / buzzwords
 - AWS-‘Cloud’ (think: storage)
 - 5G (think: network)
 - Node.js (think: Back-end development language)
 - Vue.js (think: Front-end development language)
- Design Patterns



My Hopes and Expectations

- you read the book & papers
- you are constructive
- you are here to do your best to learn something
- you help each other



'eat your veggies'



Your Hopes and Expectations

- Who are you?
 - How many are from a Gbg – BSc SE degree?
 - Degrees in other disciplines?
 - How many are comfortable in UML?
 - Anyone follow a Software Architecture course before?
 - Industrial experience?
- What do you expect / want to learn?

Learning/Teaching in Corona times

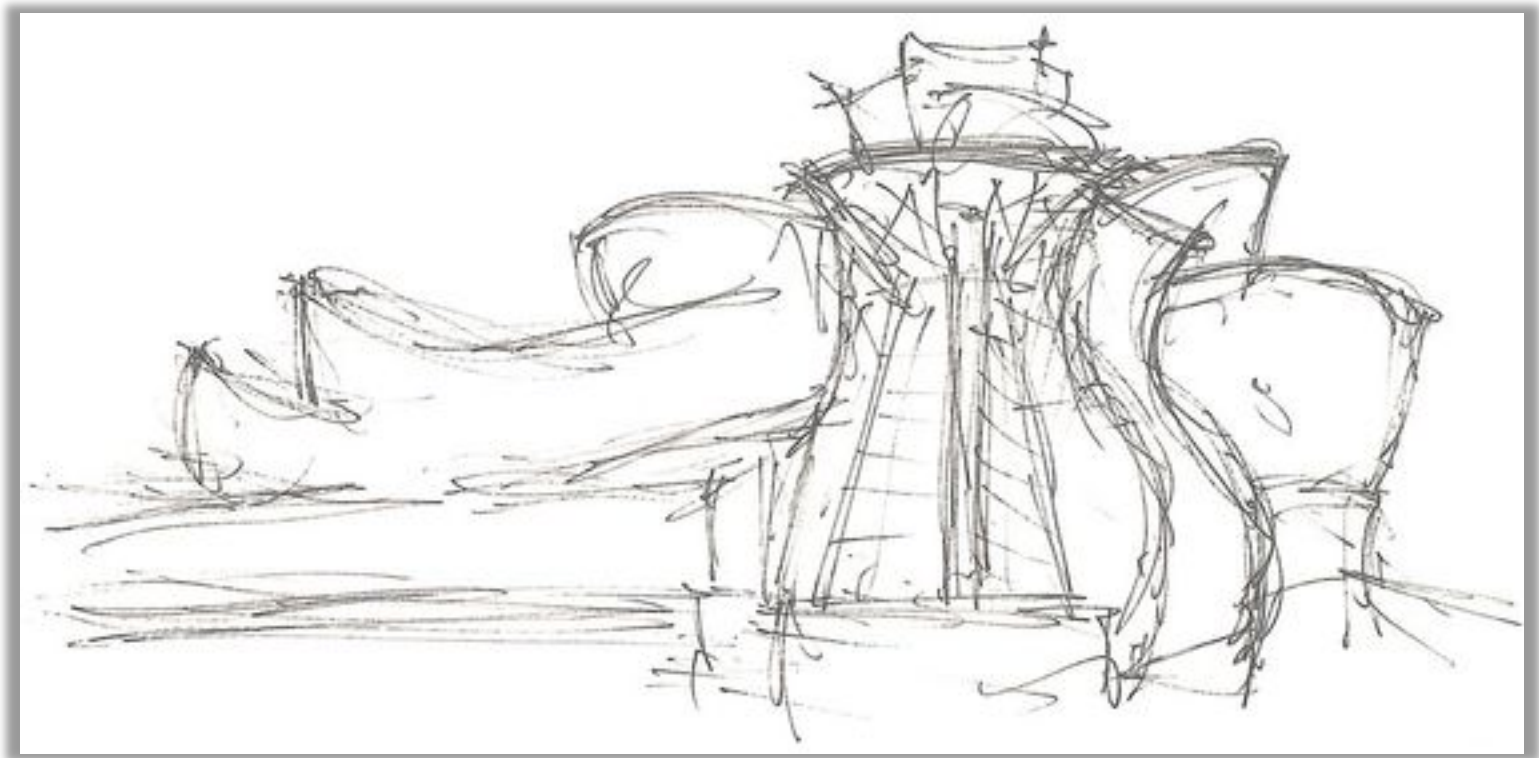
- 100% remote (No physical meetings)
 - Live lectures ('Swedish' timezone - UTC+2)
 - Lectures will NOT be recorded
 - Group activities will be done online
 - Remote written-exam
- Course is administrative in Canvas
 - Be ready to receive lots of "notification" emails

General rules

- Lecture: 2 x 45-50 mins (5 - 15 mins break)
- If you cannot attend a lecture, send me an email in advance.
- Questions are very welcome
 - during & after class (via lecture zoom link)
- Read the book & recommended papers

What is this? What is it for?



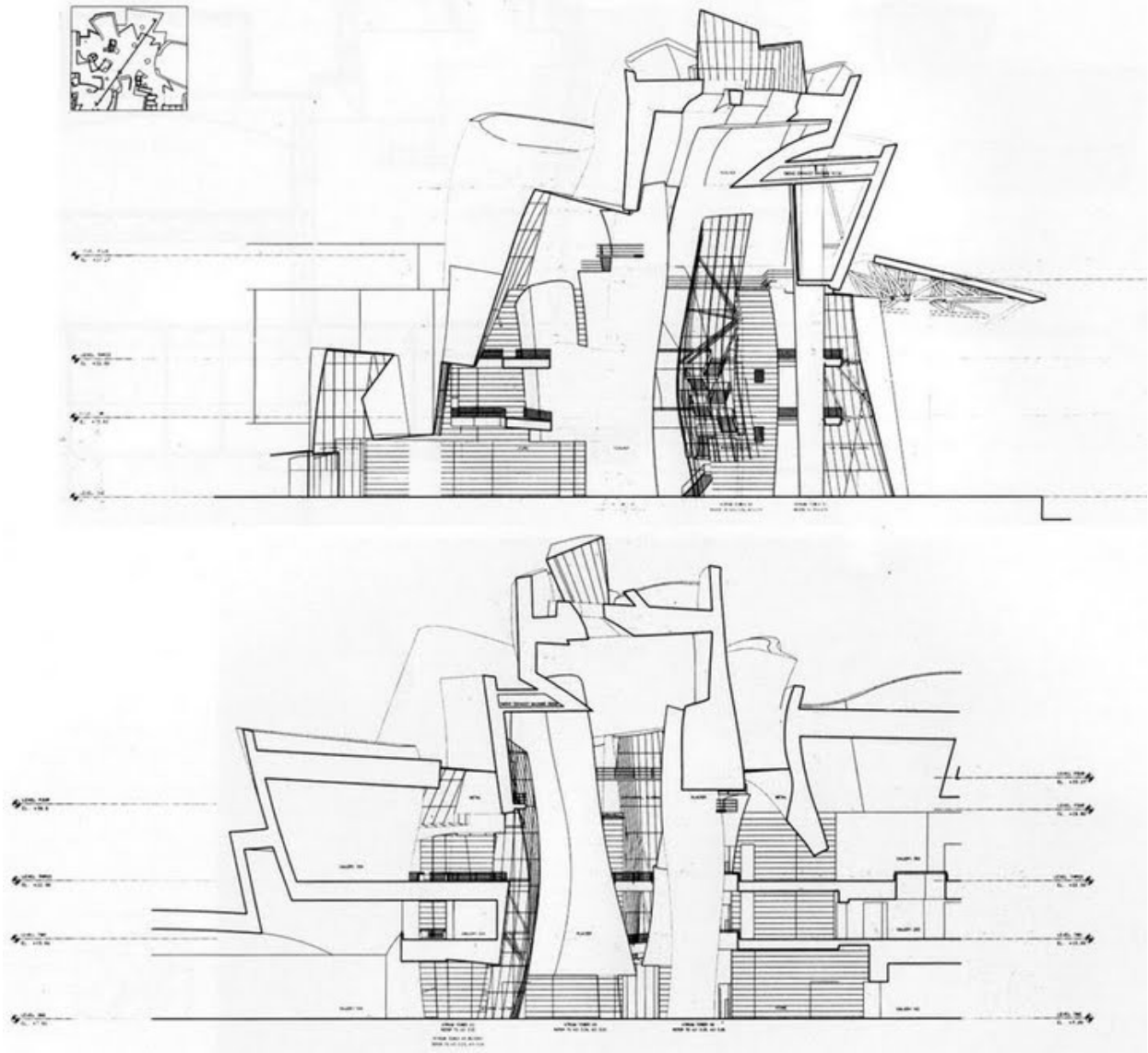




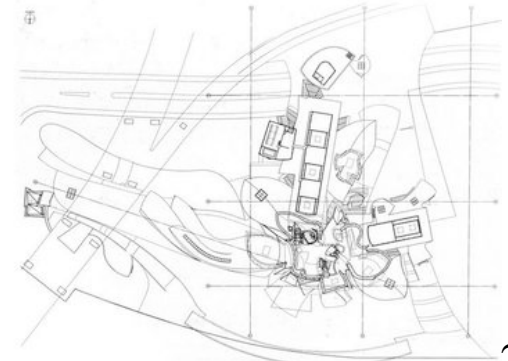
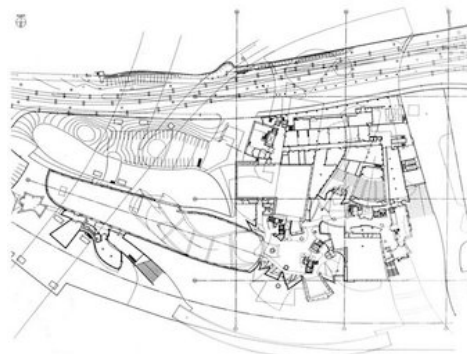
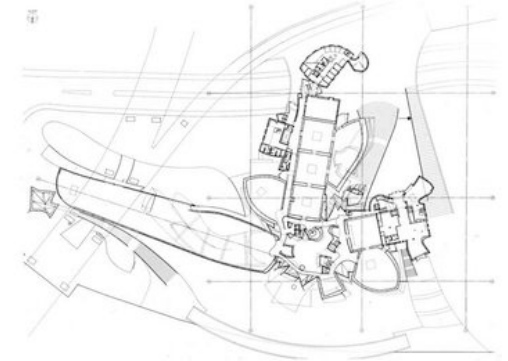
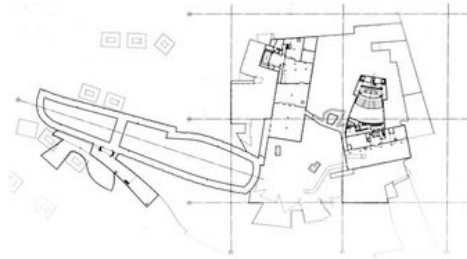
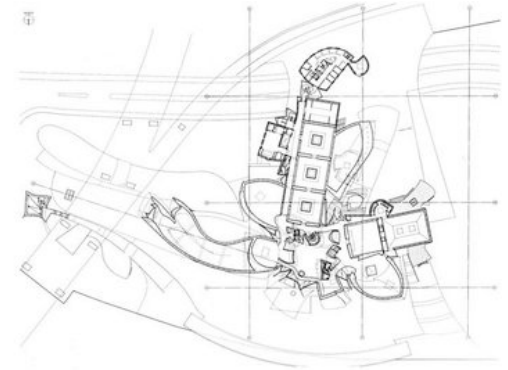
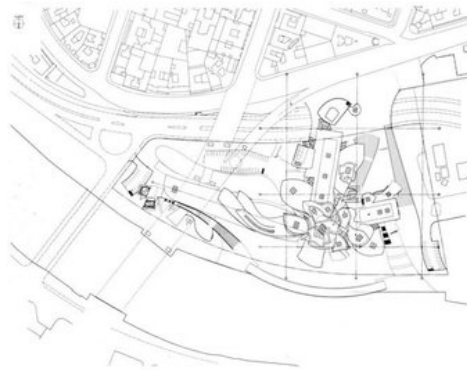
Form & Function



Space, Light



Building in its environment



Outline

- Organization of the lectures
- Importance and use of architecture
- What is Software Architecture?
- Concluding Remarks & References

Schedule

Week		Date	Time	Lecture	Note
3	L1	Wed, 20 Jan	10:15 – 12:00	Introduction & Organization	Truong Ho
3	L2	Thu, 21 Jan	13:15 – 15:00	Architecting Process & Views	Truong Ho
4		Tue, 26 Jan	10:15 – 12:00	Skip	
4	S1	Wed, 27 Jan	10:15 – 12:00	<< Supervision: Launch Assignment 1>>	TAs
4	L3	Thu, 28 Jan	13:15 - 15:00	Roles/Responsibilities & Functional Decomposition	Truong Ho
5	L4	Mon, 1 Feb	10:15 – 12:00	Architectural Styles P1	Truong Ho
5	S2	Wed, 3 Jan	10:15 – 12:00	<< Supervision/Assignment>>	TAs
5	L5	Thu, 4 Jan	13:15 – 15:00	Architectural Styles P2	Truong Ho
6	L6	Mon, 8 Feb	10:15 – 12:00	Architectural Styles P3	Sam Jobara
6	S3	Wed, 10 Feb	13:15 – 15:00	<< Supervision/Assignment>>	TAs
6	L7	Thu, 11 Feb	13:15 – 15:00	Design Principles (Maintainability, Modifiability)	Truong Ho
7	L8	Mon, 15 Feb	10:15 – 12:00	Performance – Analysis & Tactics	Truong Ho
7	S4	Wed, 17 Feb	13:15 – 15:00	<< Supervision/Assignment>>	TAs
7	L9	Thu, 18 Feb	10:15 – 12:00	Tactics: Reliability, Availability, Fault Tolerance	TBD
8	L10	Mon, 22 Feb	13:15 – 15:00	Guest Lecture 1	TBD
8	S5	Wed, 24 Feb	13:15 – 15:00	<< Supervision/Assignment>>	TAs
8	L11	Thu, 25 Feb	10:15 – 12:00	Guest Lecture 2	TBD
9	L12	Mon, 1 Mar	13:15 – 15:00	Reverse Engineering & Correspondence	Truong Ho
9	S6	Wed, 3 Mar	10:15 – 12:00	<< Supervision/Assignment>>	TAs
9	L13	Thu, 4 Mar	13:15 – 15:00	To be determined (exam practice?)	Truong Ho
9		Fri, 5 Mar	Whole day	Group presentation of Assignment (TBD)	Teachers
11	Exam				

Here we are

Schedule

Week		Date	Time	Lecture	Note
3	L1	Wed, 20 Jan	10:15 – 12:00	Introduction & Organization	Truong Ho
3	L2	Thu, 21 Jan	13:15 – 15:00	Architecting Process & Views	Truong Ho
4		Tue, 26 Jan	10:15 – 12:00	Skip	
4	S1	Wed, 27 Jan	10:15 – 12:00	<< Supervision: Launch Assignment 1>>	TAs
4	L3	Thu, 28 Jan	13:15 - 15:00	Roles/Responsibilities & Functional Decomposition	Truong Ho
5	L4	Mon, 1 Feb	10:15 – 12:00	Architectural Styles P1	Truong Ho
5	S2	Wed, 3 Jan	10:15 – 12:00	<< Supervision/Assignment>>	TAs
5	L5	Thu, 4 Jan	13:15 – 15:00	Architectural Styles P2	Truong Ho
6	L6	Mon, 8 Feb	10:15 – 12:00	Architectural Styles P3	Sam Jobara
6	S3	Wed, 10 Feb	13:15 – 15:00	<< Supervision/Assignment>>	TAs
6	L7	Thu, 11 Feb	13:15 – 15:00	Design Principles (Maintainability, Modifiability)	Truong Ho
7	L8	Mon, 15 Feb	10:15 – 12:00	Performance – Analysis & Tactics	Truong Ho
7	S4	Wed, 17 Feb	13:15 – 15:00	<< Supervision/Assignment>>	TAs
7	L9	Thu, 18 Feb	10:15 – 12:00	Tactics: Reliability, Availability, Fault Tolerance	TBD
8	L10			Guest Lecture 1	TBD
8	S5			<< Supervision/Assignment>>	TAs
8	L11			Guest Lecture 2	TBD
9	L12	Mon, 1 Mar	13:15 – 15:00	Reverse Engineering & Correspondence	Truong Ho
9	S6	Wed, 3 Mar	10:15 – 12:00	<< Supervision/Assignment>>	TAs
9	L13	Thu, 4 Mar	13:15 – 15:00	To be determined (exam practice?)	Truong Ho
9		Fri, 5 Mar	Whole day	Group presentation of Assignment (TBD)	Teachers
11	Exam				

Industrial
lectures

Evaluation

- Assignments parallel to course
- Final ‘remote’ written exam (‘proctoring’ TBD)
- I will try to test that
 - You *know* a few things (concepts, facts, UML, ...)
 - You *can do* a few things
 - *THINK* – *analyze, structure, reason logically*
 - *IMPLEMENT*
- There are a few things from this course that are very important
I will emphasize them and expect you notice

Two Assignments

- **Assignment steps:**

- Understand requirements (apply knowledge from RE course)
- Identify stakeholders & architectural drivers
- Design Architecture

Iterate:

- Design & model structure components & behaviours
- Use patterns and tactics to achieve quality properties
- Analyse / Evaluate

- **Assignment 1:**

Hand in architecture design; peer-review; **hand in implementation & updated SAD**; { update after review }

- **Assignment 2:**

Hand in architecture design ; { update after review }

Group Formation

- Group of 5 students
- Group formation strategy:
 - You find group yourself
 - When your group is formed (with agreement from all group members), you send a proposal to us (via email)
 - If you cannot find a group to join and need our help, let's send an email to us by . We shall randomly assign a group to you.
- Once groups are formed, they are stable during the course. Any change needs to be discussed and approved by teachers.
- Deadline to send group proposal: 23:59 on **Monday, Jan. 25, 2021**.
- All groups will be formed and assigned to a supervisor on Jan. 26.

Group Working

- Create Trello board
- Create GitLab project
- Join this slack workspace for discussions
https://join.slack.com/t/dat220ht21adv-we38068/shared_invite/zt-lekekjn4-mMg2s9ytY~PfMZWnplie9g
- Make sure to give access to TA's

Supervision session

- Supervision sessions are used to coordinate group efforts in specific assignments.
- Each group has a supervisor.
- Supervisors are the main contact points if you have specific questions regarding assignments
- Duration of each supervision session: Maximum 30 – 40 mins (can be shorter)
- Supervision time is agreed between supervisors and the corresponding groups
- Teachers will try to join as many supervision as possible
- <<TBD>> Way to handle Zoom meetings will be decided later
 - Alt1: Supervisors to organize zoom call
 - Alt2: All supervision sessions will be organized as a break-out room in a centrally-managed zoom call.

Hand in

- Software Architecture Document
 - Names & email
 - Group number (!)
 - Template will be uploaded
 - UML models of your architecture design
 - Explanation of it (!)
 - Explain mapping of architecture onto implementation
- (access to) Source code
 - Understandable to non-group members
 - Clear method names & variable names
 - Clearly commented

Examination

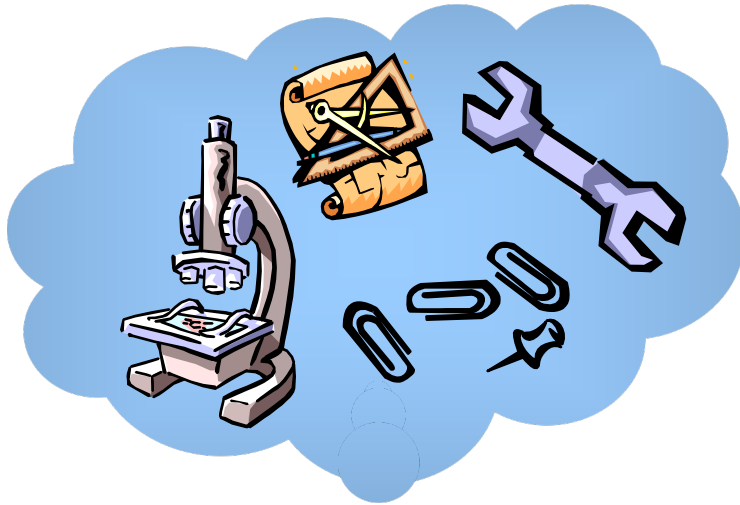
In order to pass the course, you have to

- i) Pass the written exam (individual)
- ii) Pass all assignments (group)
- iii) Contribute an adequate amount in group work (via group report)
- iv) Hand in all peer-evaluations (individual)

Criteria for VG

- VG for both assignments and written exam
- For assignments:
 - Hand-in's are graded on a 10-point scale
 - If (average) score ≥ 7.5 you get a VG for the assignment
- For the software architecture document:
 - Good use of UML, Good use of views
 - Clear mapping between implementation and architecture models
 - Good explanation
- For the implementation
 - Satisfies sanity-rules for code (see earlier)
 - Clearly identifiable implementation of at least one architecture tactic

Teaching Philosophy



Tools

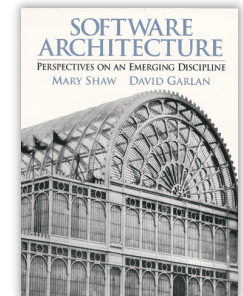


Directions

If you have a hammer, then this doesn't mean all problems are a nail!

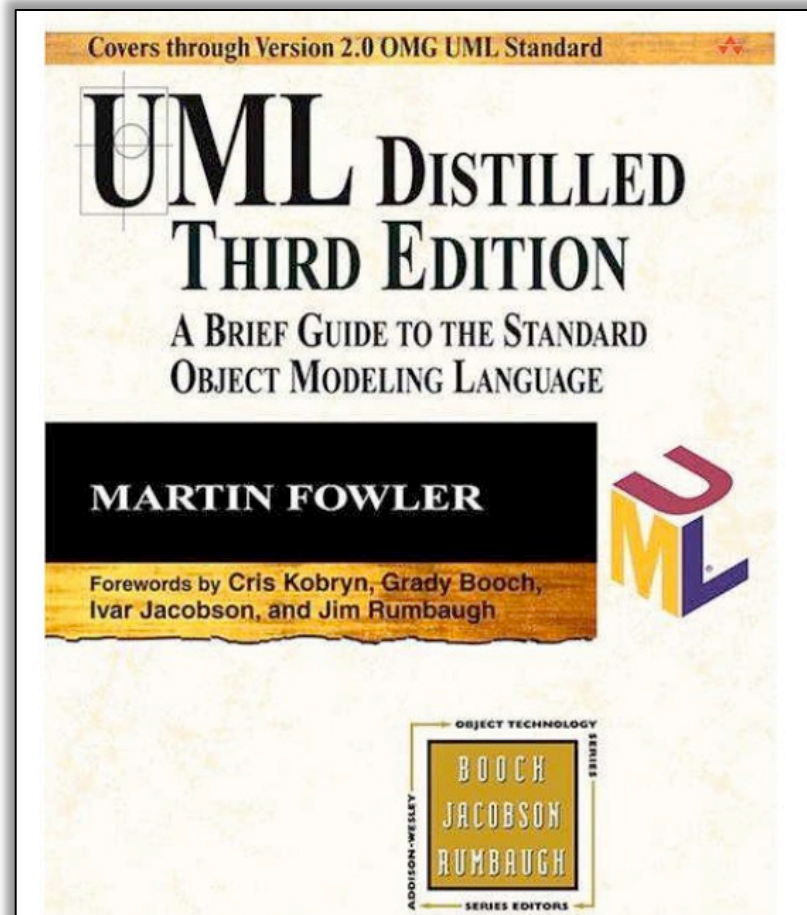
Software Architecture Books

- Software Architecture in Practice, **3rd Edition**,
L. Bass, P. Clements, R. Kazman,
SEI Series in Software Engineering,
Addison-Wesley, 2003
- Software Architecture: Perspectives on an
Emerging Discipline, Mary Shaw, David Garlan,
242 pages, 1996, Prentice Hall
- ISO/IEC/IEEE Standard
 - 1471-2000 - Recommended Practice for Architectural Description
 - 42010-2011 - Systems and software engineering — Architecture description



UML book

- UML Distilled
4th or 3rd edition



Software Architecture Books

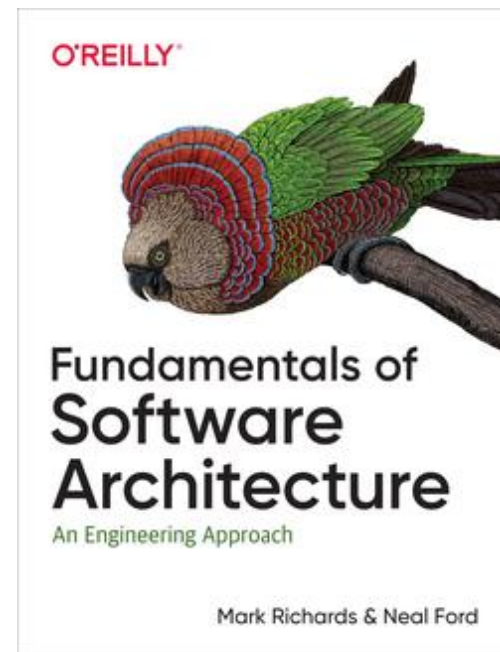
- Fundamentals of Software Architecture

by Mark Richards, Neal Ford

Released January 2020

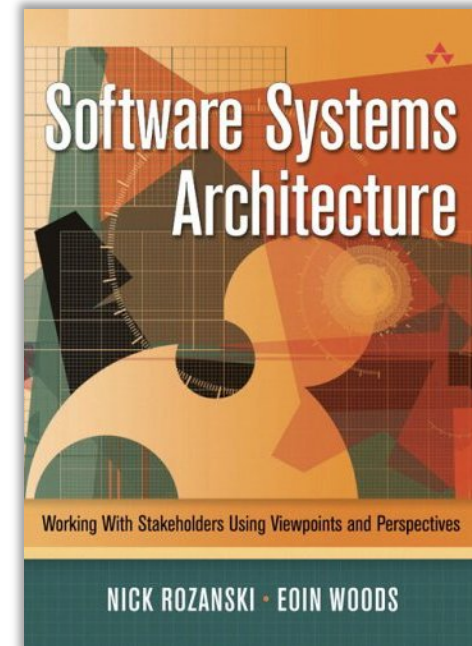
Publisher(s): O'Reilly Media, Inc.

ISBN: 9781492043454



Software Architecture Books

- Software Systems Architecture – With Stakeholders Using Views and Perspectives by Nick Rozanski and Eoin Woods, Addison-Wesley, 2005



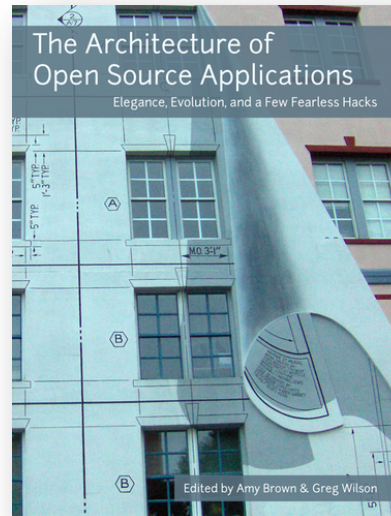
Software Architecture Books

- The Architecture of Open Source Applications (Volumn I & II)

Edited by Amy Brown & Greg Wilson

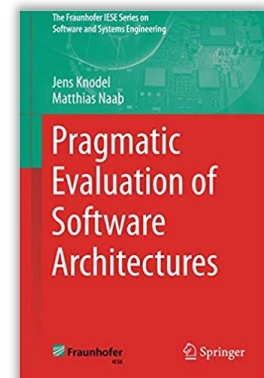
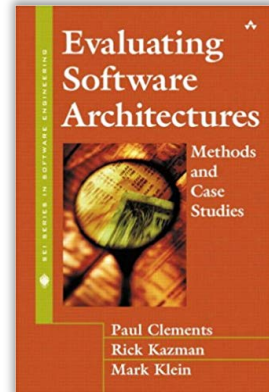
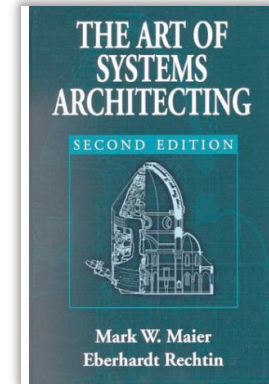
<http://aosabook.org/blog/>

Twitter at [@aosabook](https://twitter.com/aosabook) ([#aosabook](https://twitter.com/aosabook) hashtag)



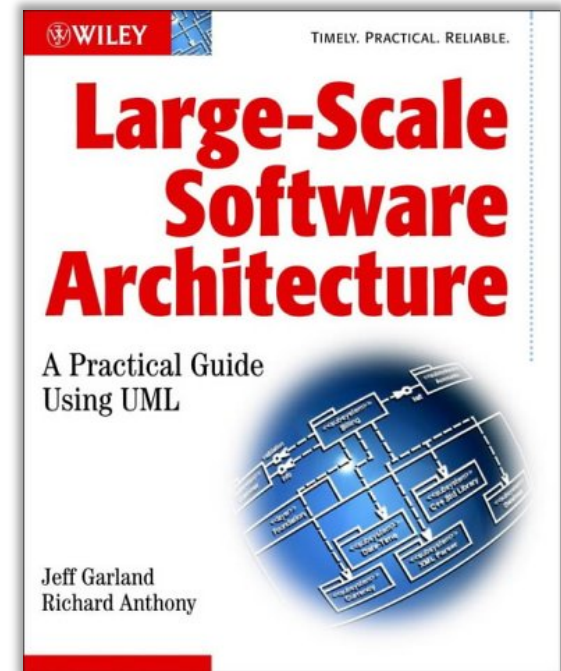
Software Architecture Books

- The Art of Systems Architecting,
Second Edition (Hardcover)
by [Mark W. Maier](#), [Eberhardt Rechtin](#)
Publisher: CRC Press, 2000
- Evaluating Software Architecture
Paul Clements, Rick Kazman, Marc Klein,
2002, Addison-Wesley, ISBN 020170482X
- Pragmatic Evaluation of Software
Architectures, Jens Knodel, Matthias Naab,
Springer, 2016



Software Architecture Books

- *Large Scale Software Architecture*
A Practical Guide Using UML
by Jeff Garland & Richard Anthony,
Wiley, 2002.



Structure of the Course (1)

- Introduction:
 - why, what, when, whom, ...
 - course 'logistics'
 - a bit on software architecture notion
- Roles & Responsibilities

Structure of the Course (2)

- Architecture Modelling
 - view paradigms (IEEE1471, 4+1, SHN, Zachmann)
 - ADL's (AADL, UniCon, Splice, Darwin/Koala)
- Architectural Styles (3 lectures)
 - pipe-and-filter, blackboard, pub/sub, microservices
 - strenghts and weaknessess of styles
- Industrial lecture series (2 lectures)
 - To be confirmed

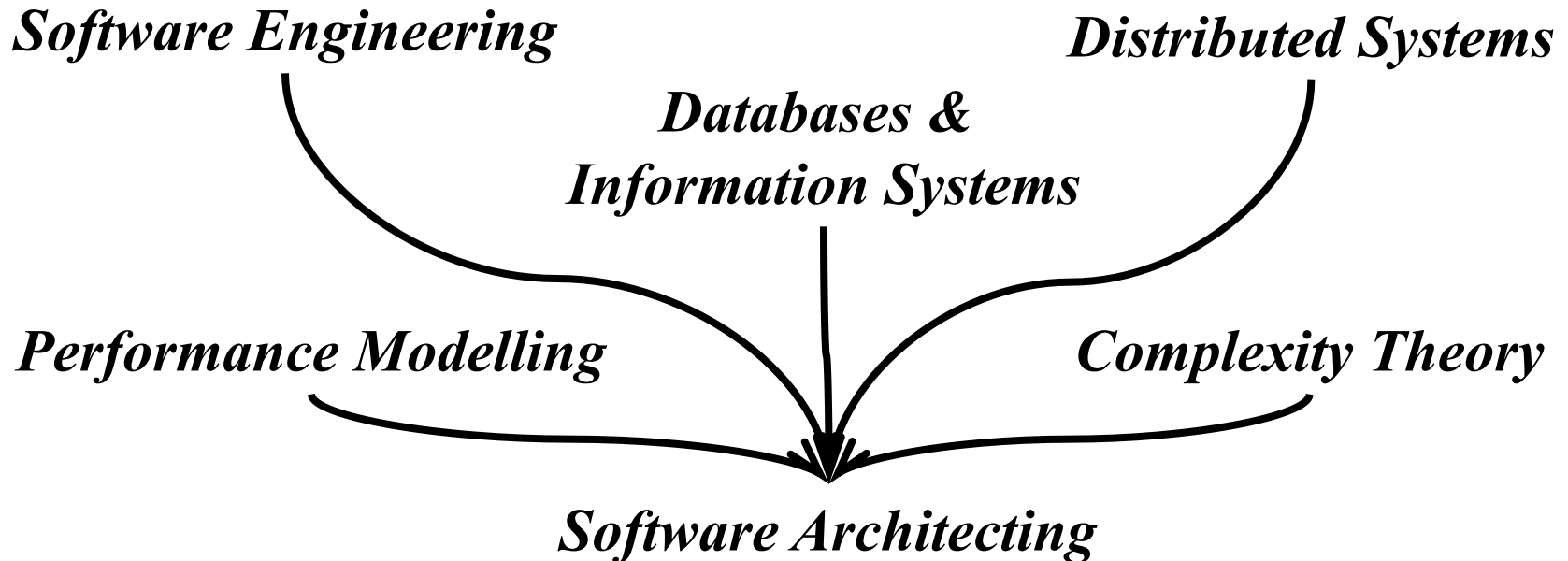
Structure of the Course (3)

- Architecture Evaluation
 - Qualitative:
 - Scenario-based: ATAM
 - Quantitative Techniques for Analysing Architectures
 - Reliability Block Diagrams, Rate Monotonic Analysis
 - Queuing Networks
 - Risk, Cost, Feasibility
- Architecture Reconstruction (reverse engineering)
- Last lecture: to be confirmed

What I find most important:

- You learn to design architectures
 - you apply design principles
decomposition & layering
- You learn to document & communicate your architecture design
 - using UML & views
 - systematic, consistent, layout
- You can reason at conceptual level
 - separate from, yet linked to implementation

Relation to Other CS subjects



Central question:
How to design, document and assess software architectures.

Preview / Recap: What is Software Architecture?

A definition:

Software Architecture is the global organization of a software system, including

- the division of software into subsystems/components,
- policies according to which these subsystems interact,
- the definition of their interfaces.

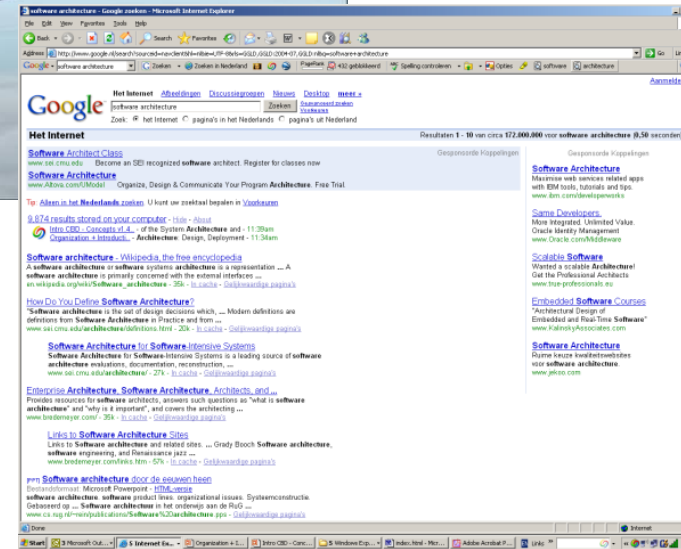
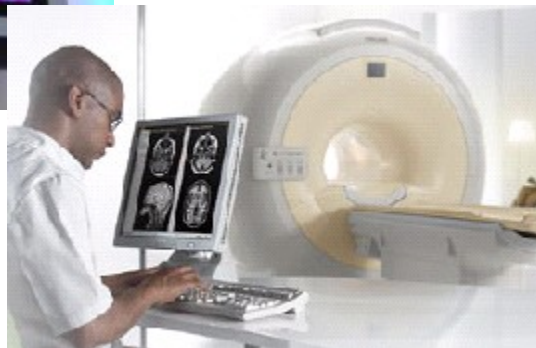
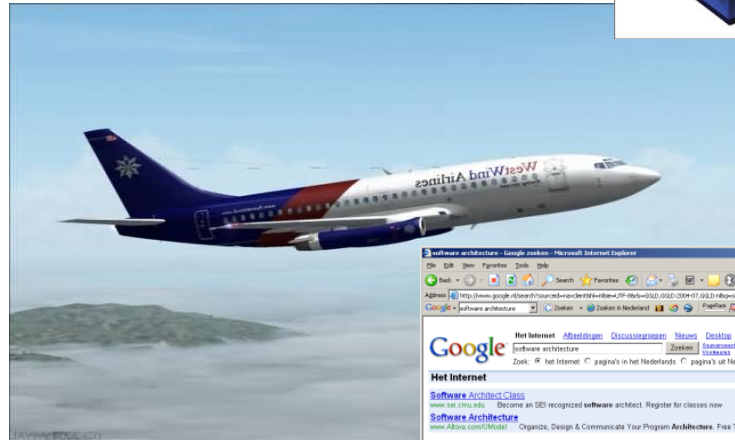
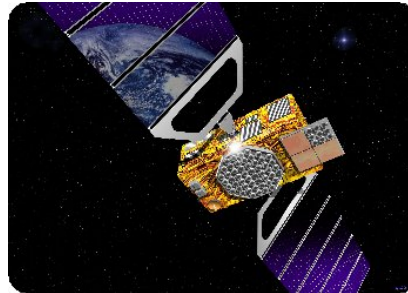
(free after) Object Oriented Software Engineering

T. C. Lethbridge & R. Laganière

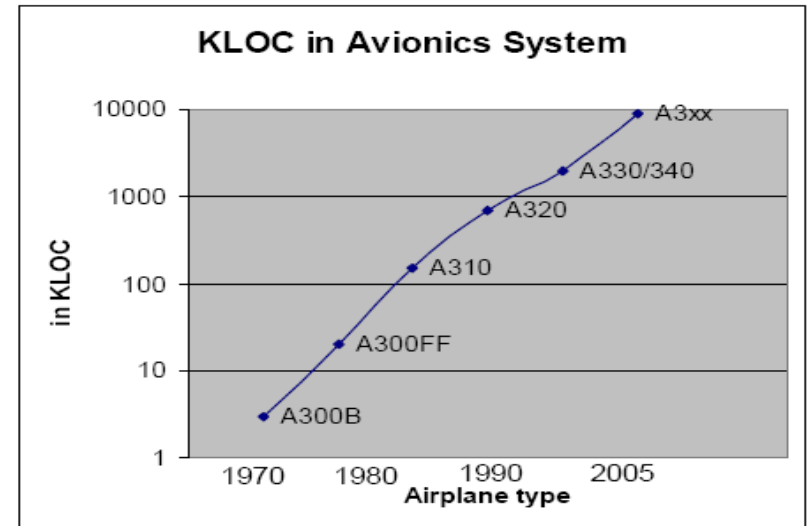
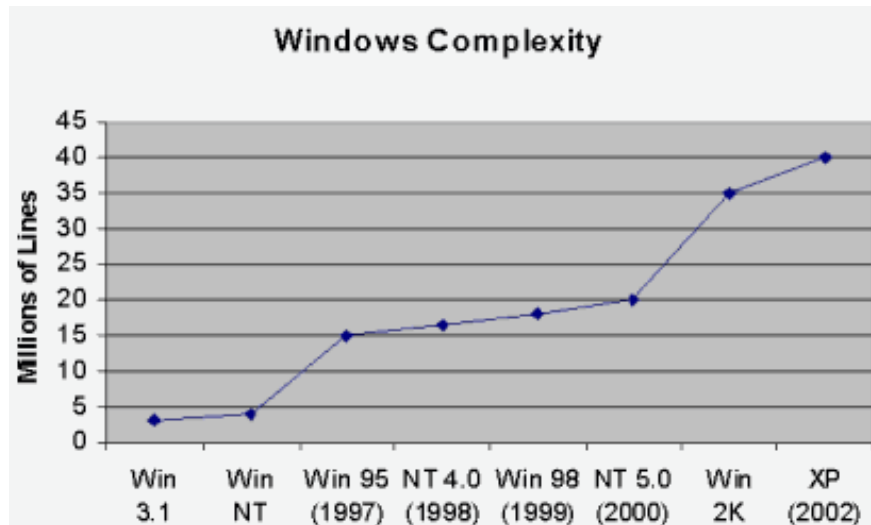
McGraw Hill, 2001

Outline

- Organization of the lectures
- Importance and use of architecture
- What is Software Architecture?
- Concluding Remarks & References

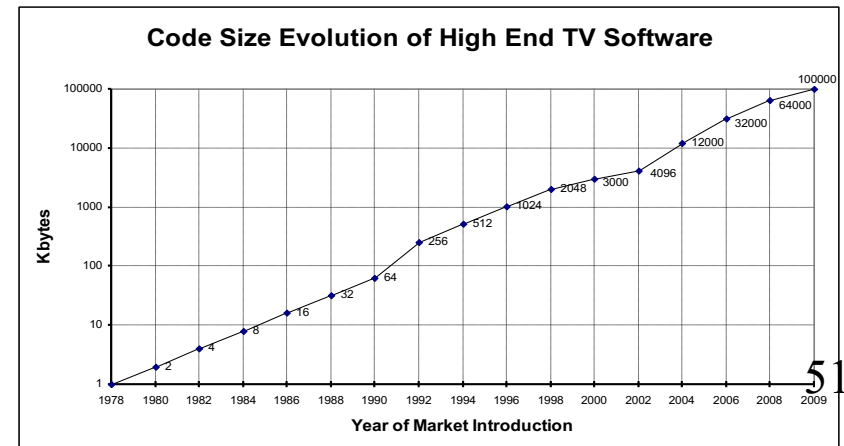


Increasing amount of software in systems

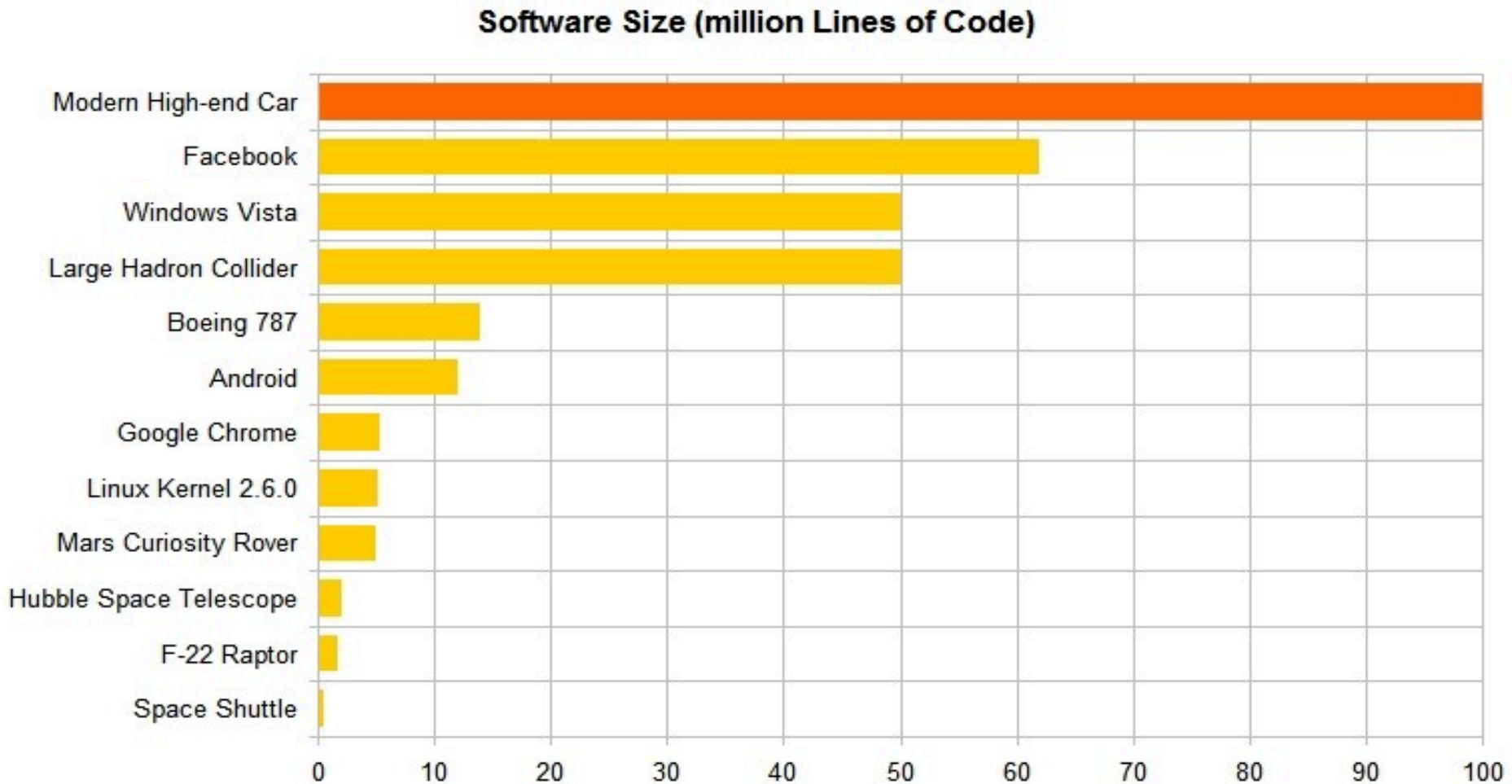


Nb: logarithmic scale

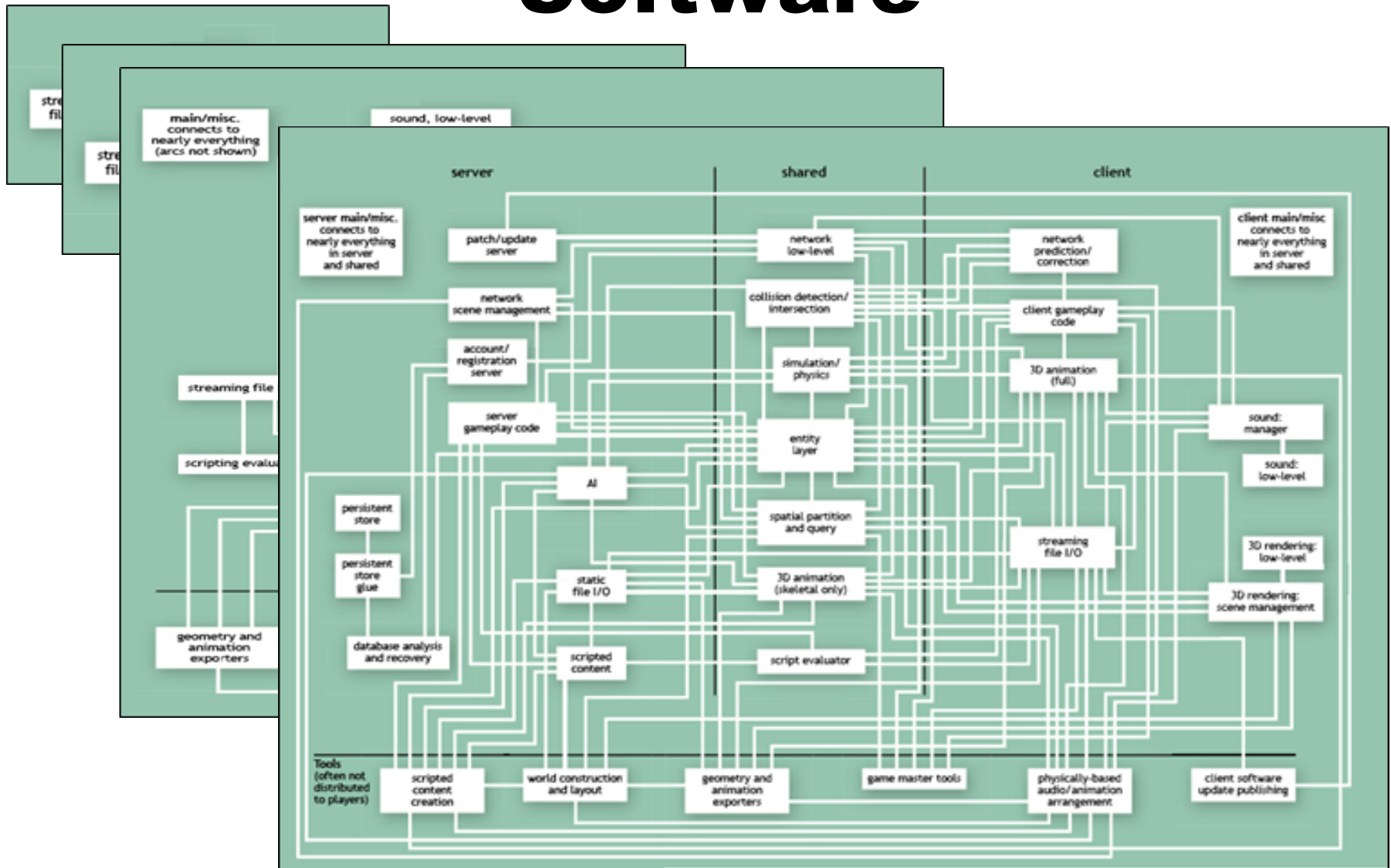
The amount of software increases 10 fold every 10 years



Software size grows enormously



Increasing Complexity of Software



Complexity

The Vacheron Constantin Reference 57260



is a mechanical pocket watch has 57 features.

The company claims that it is the most complicated mechanical pocket watch ever created. *A masterpiece.* The Reference 57260 took eight years to assemble, and has 2826 parts and 31 hands.

(Wikipedia JAN 2020)

it costs several million dollars.

Solitaire App for iOS



- About 50 functions
(e.g. start, settings, move,)
- How much time to build?
- How much does it costs?

Exercise

Analyze a case description

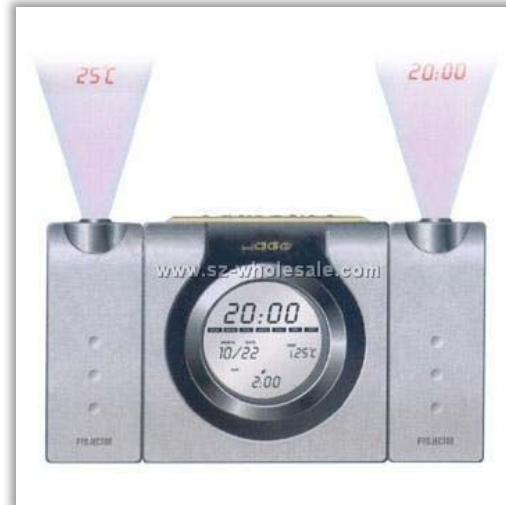
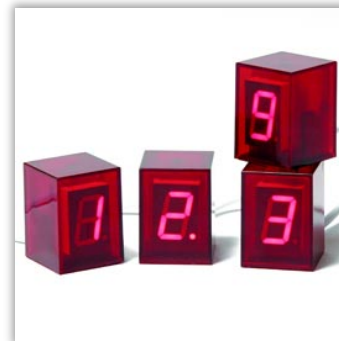
- Radio-Alarm-clock

Case Description

Anyone should be able to use this machine. You can set a time and when the time has arrived, then it turns on the radio. This device shows the time. You can interact with this machine by pressing its buttons.

Radio Alarm Clock

What parts/subsystems does a Radio-Alarm Clock have?



Assignment

- Write/draw a design on paper/a text editor
- Identify which subsystems the radio-alarm-clock must have (at least 3)
- Identify stakeholders
- Describe the responsibility of each of the subsystems.
 - one single sentence per component
 - in a general / generic ('abstract') way

Share your text/photos at in Canvas Discussion topic:

https://chalmers.instructure.com/courses/12514/discussion_topics/52717

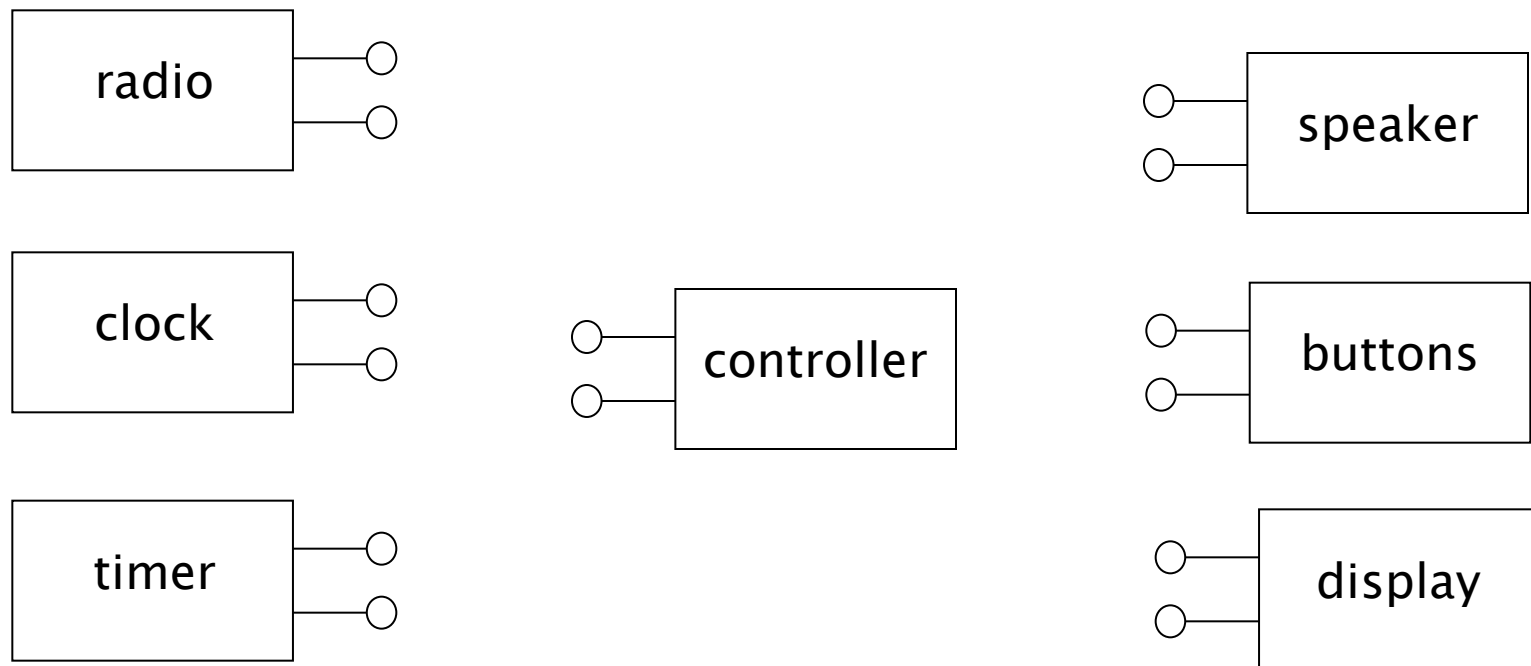
Optional:

- Can you think of any behaviour?
- Can you think of external interfaces?

Radio Alarm Clock

What should be the responsibility of each component?

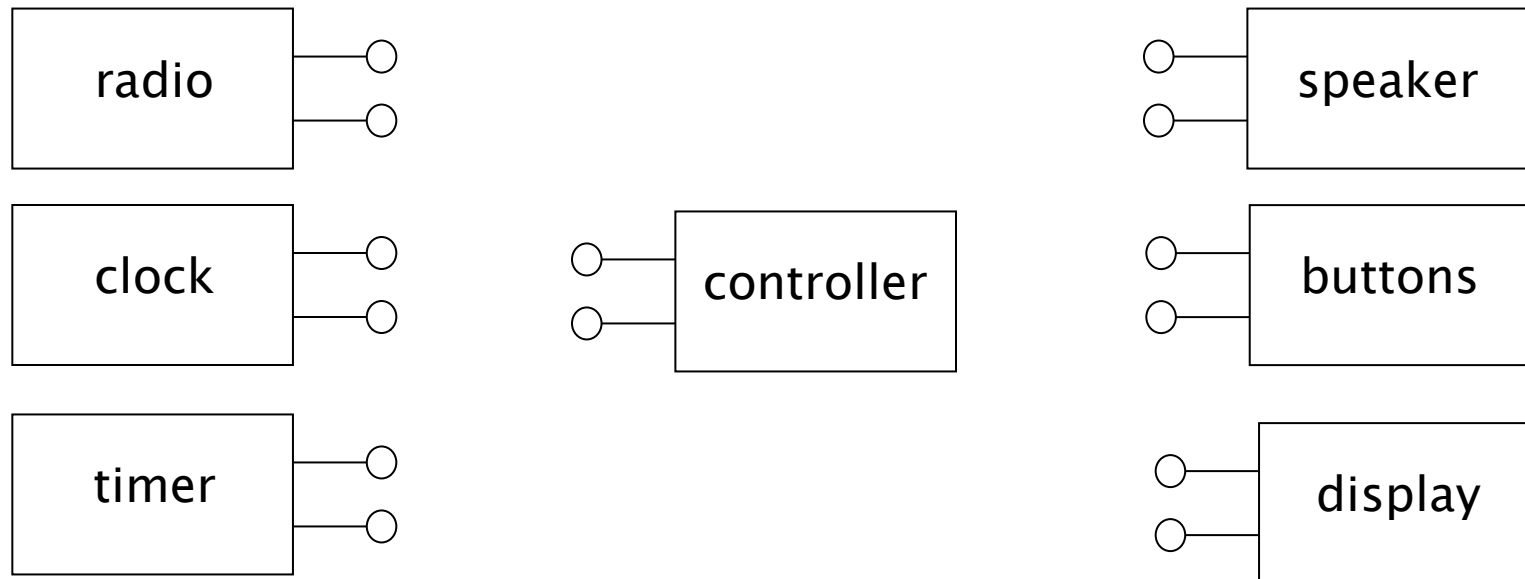
What should be the interface of each component?



Radio Alarm Clock

What should be the responsibility of each component?

What should be the interface of each component?



How about:

- Powersupply? (plug or battery)
- Memory?

Concluding Remarks 1

*Experience is the hardest kind of teacher.
It gives the test first and the lesson afterward.*
Susan Ruth, 1993

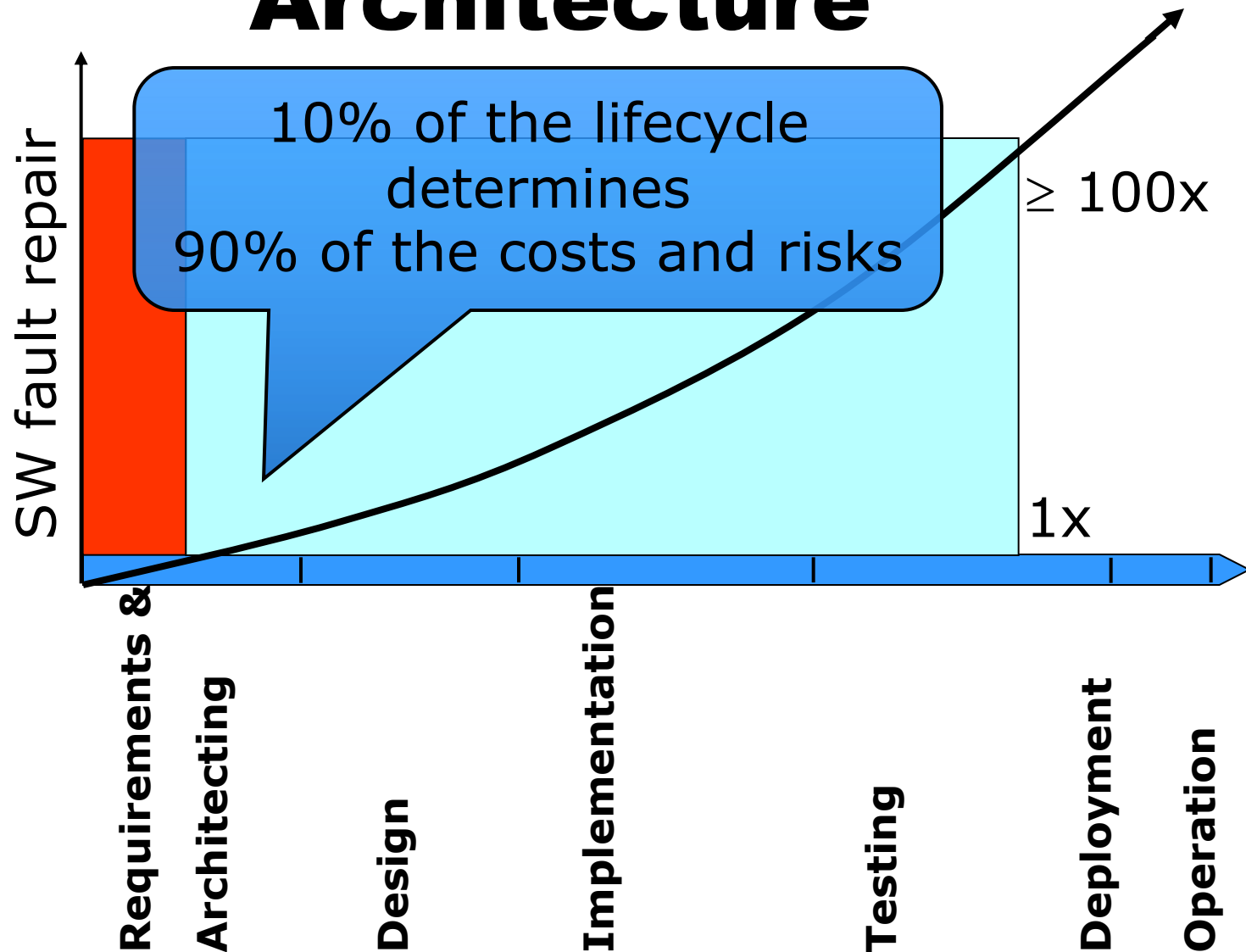
- Software Architecture is a critical aspect in the design and development of software
- We discussed definitions and objectives of Sw.Arch.
- Understanding of basic principles of architecture design, analysis, documentation, and process are necessary, but experience is hard to beat.

The Importance of Architecture

“A correct architecture has the largest single impact on cost and quality of the product.”

Maranzano, ATT, 1995

The Importance of Architecture



Business Objectives of Sw. Arch.

Reduce time-to-market

Through enabling reuse and gradual evolution

Reduce development cost

Through improved communication between developers and earlier assessment of design alternatives and assessment of system risks

Reduce maintenance cost

Through incorporation of foreseeable changes

Improve product quality

Increase fitness for use through stakeholder involvement; reduce errors through enforcement of conceptual integrity

Multiple Purposes of Architecture

Understanding + Analyzing
+ Communicating + Constructing

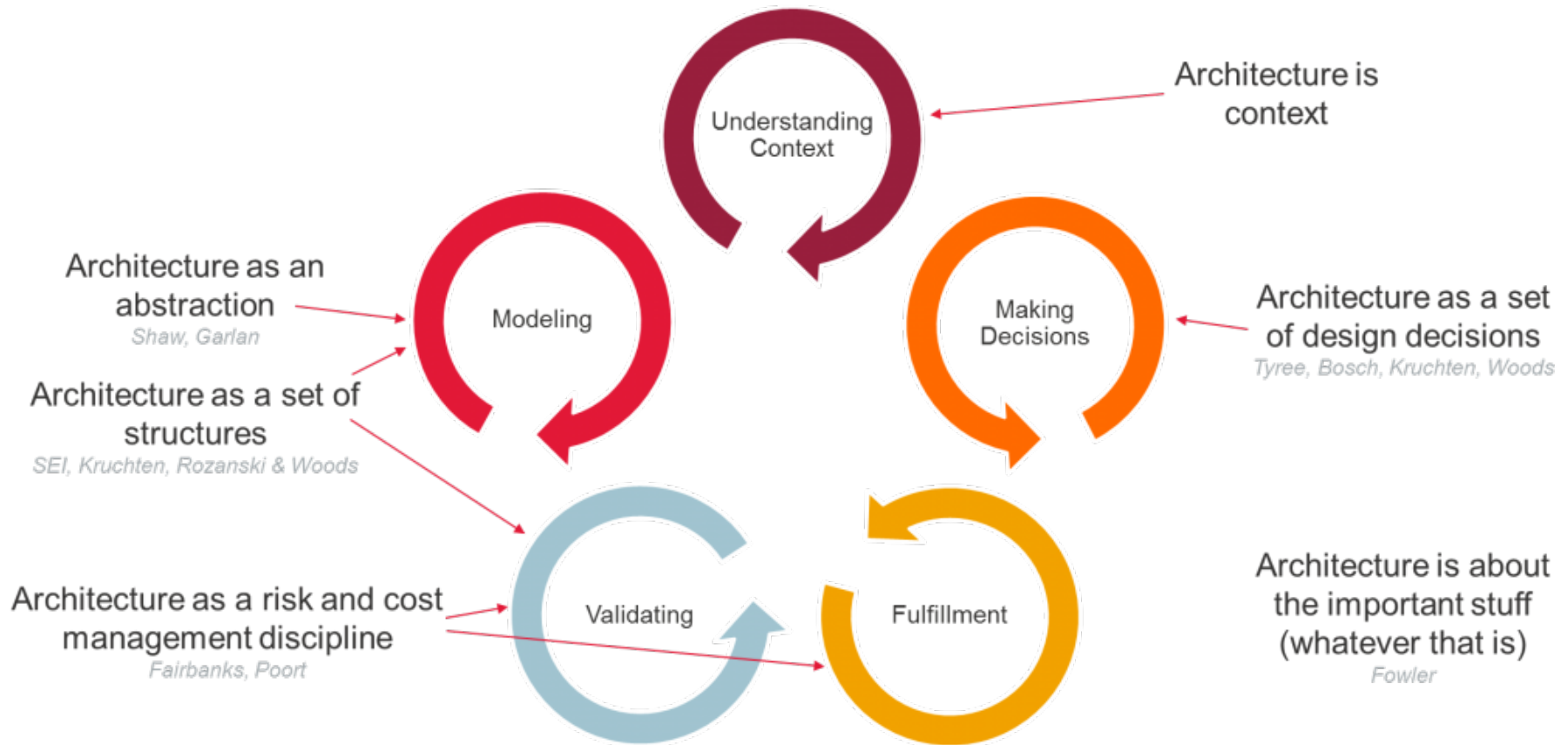


Why is the system needed? What constraints apply?
Understanding the requirements

What are the important design decisions
What functions does the system provide?
What properties does the design have?

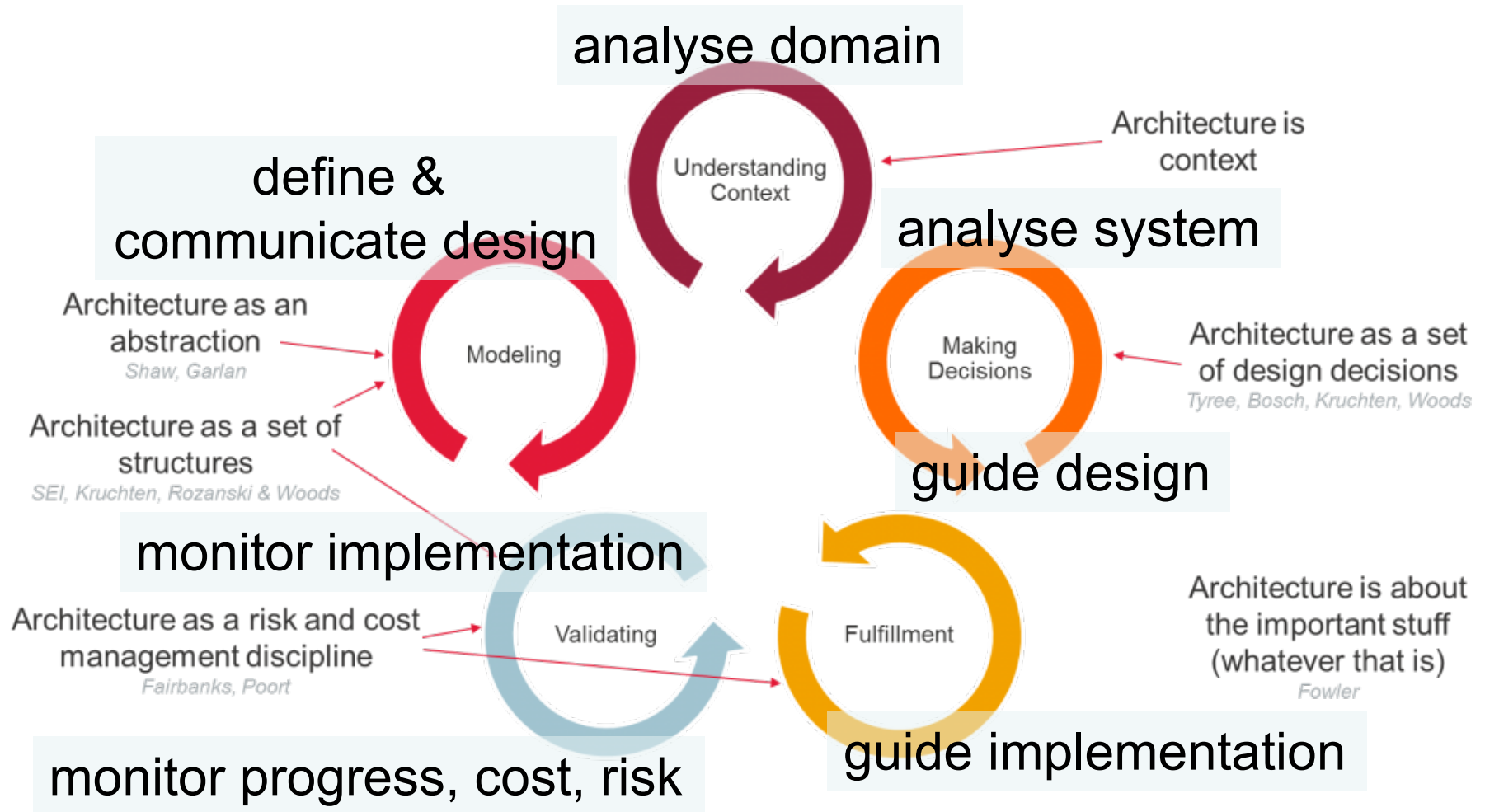
How can the system be built?

Multiple Purposes of Architecture



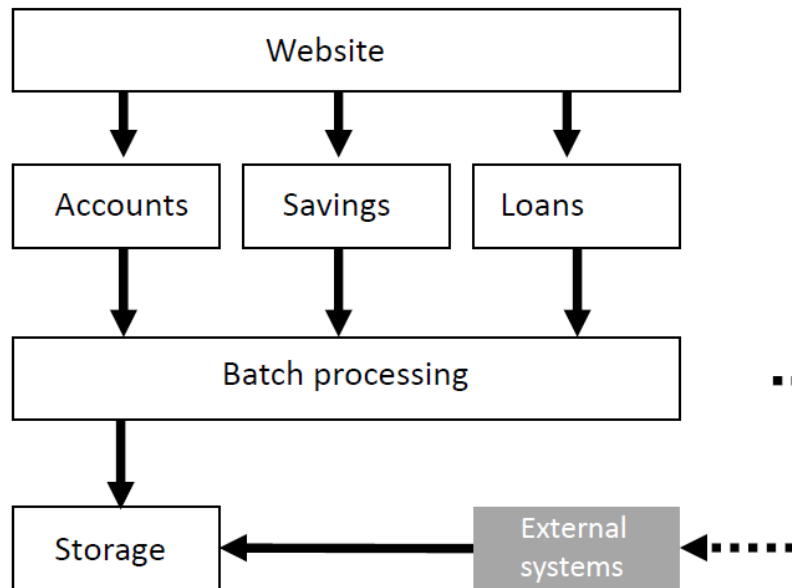
Picture from Eltjo Poort's Architecture Blog - <https://eltjopoort.nl/>

Multiple Purposes of Architecture

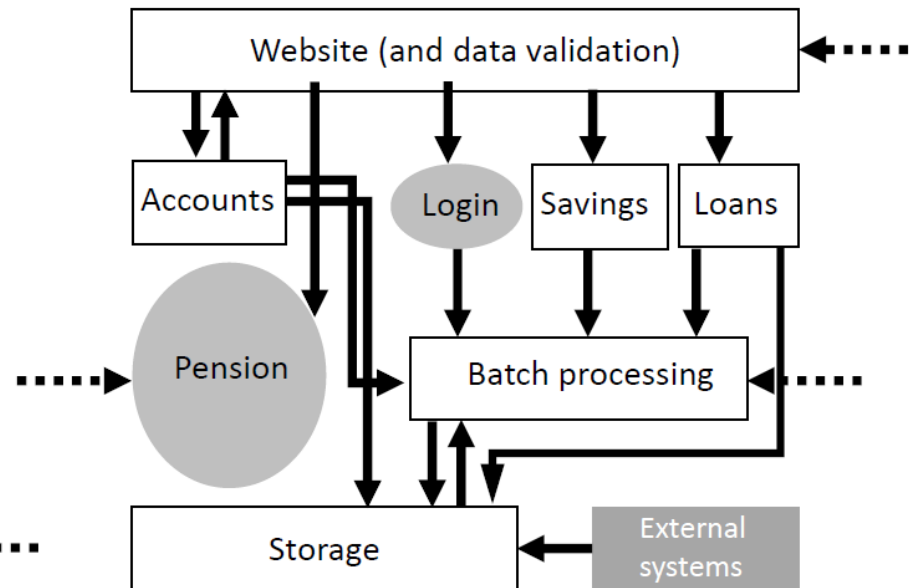


Monitor the Implementation

Plan versus Reality



The design



The implementation

Outline

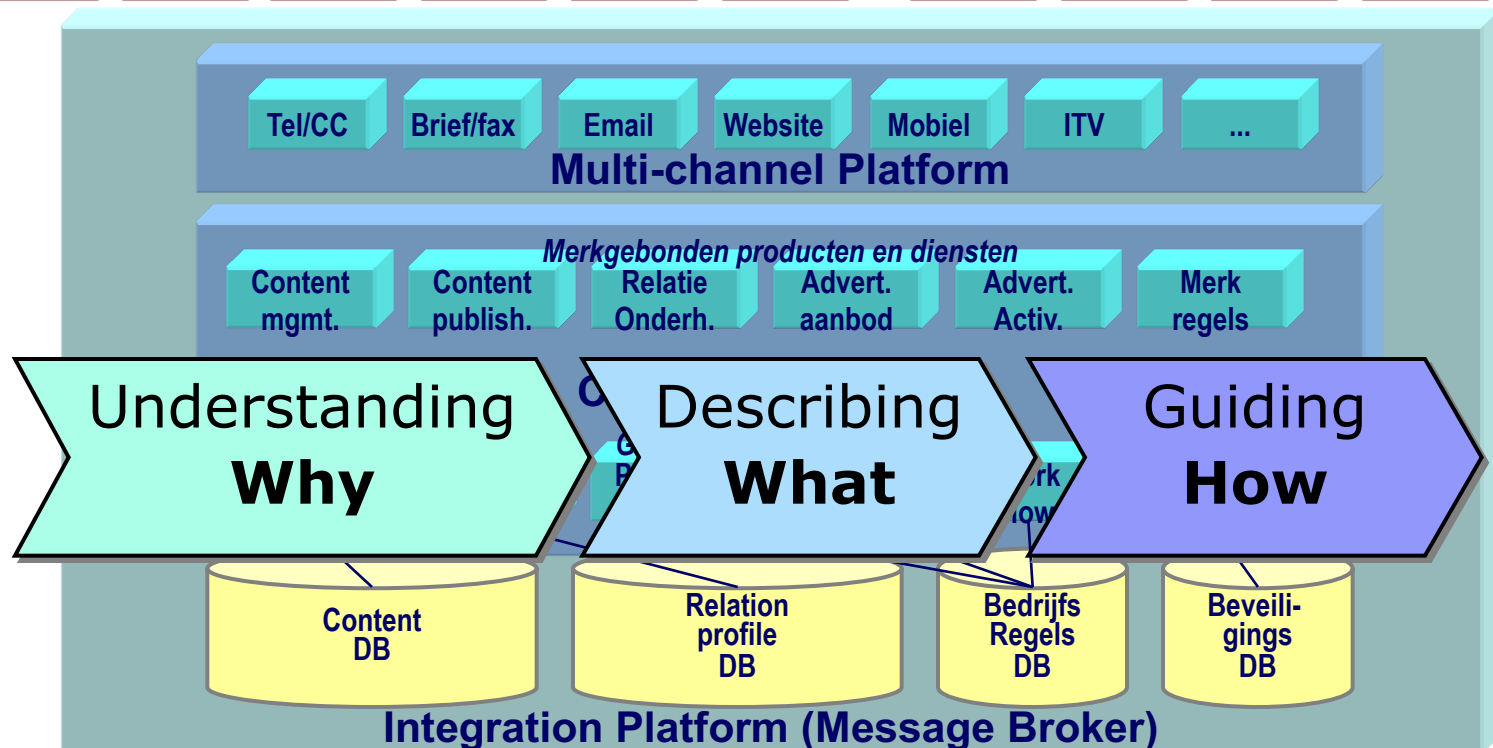
- Organization of the lectures
- Importance and use of architecture
- What is Software Architecture?
- Concluding Remarks & References

Let's Look at some Examples

eCommerce Architecture Blueprint

VK Par. Trw RD AD NRC DtRG Ad.g. Ad.k. Graf. Meul. DocA DocR MRe Losv Distr Ab.A Ab.R

Journ. Auteurs Nieuws-dnsten Advert. (PR-bur) *Extranet* Losse verk.pt. Vervoer-ders Distrib., bezorg. Busin. partners Bedrijv. Instell. *Internet* Advert. (part.) Abon-nees Bezoek-ers *Intranet* PCM medewr.



Software Architecture

Web-enabled “common look and feel” across all WIN-T applications (including those using COTS) plus support for different user device types (current and future)

Presentation

Win-T business logic. Component-based design to support the packaging of applications to the nodes/servers to which that functionality is needed without the need to redesign

Application

Provides mobile inter-node and disparate system/COTS integration services. Includes facilities to decouple Presentation/Application layer components from underlying data storage mechanisms

Integration

File System, Directory,
and Relational DBMS

Data

Provides common component-based services
Most services are COTS based.

Infrastructure

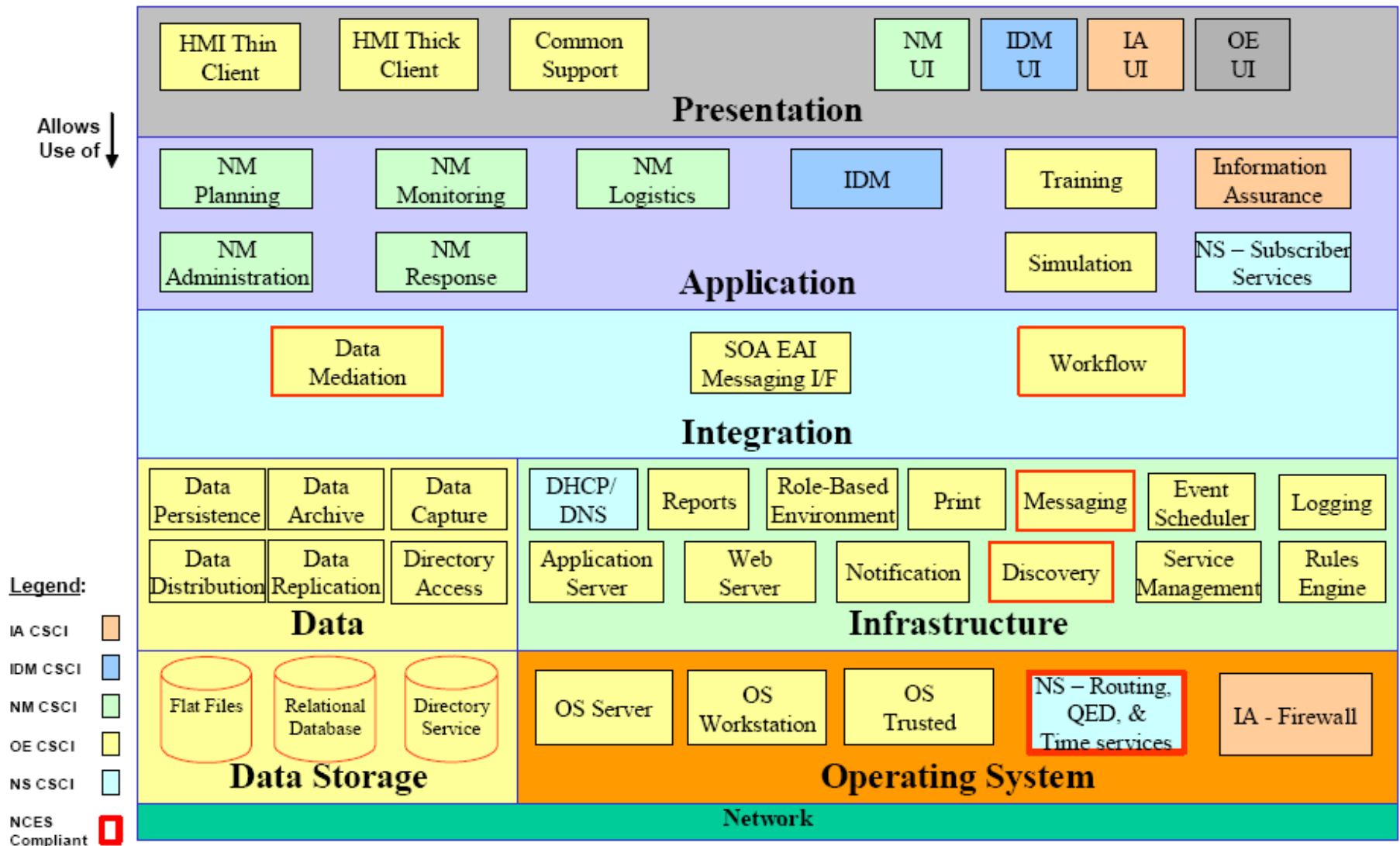
Physical Storage
Mechanisms

Data Storage

Windows operating environment

Operating System

Refinement of previous slide



Software Architecture of K9 e-mail app (on Android)

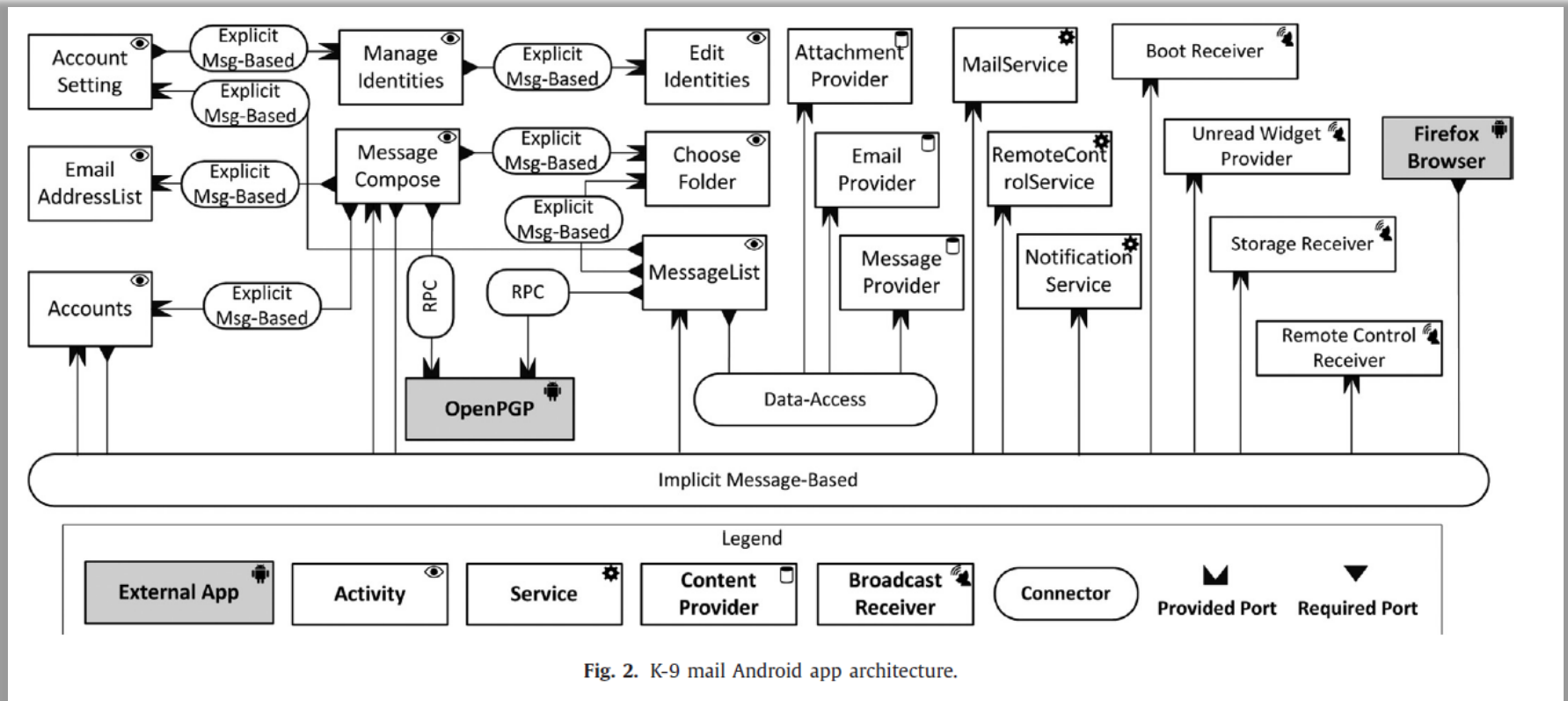
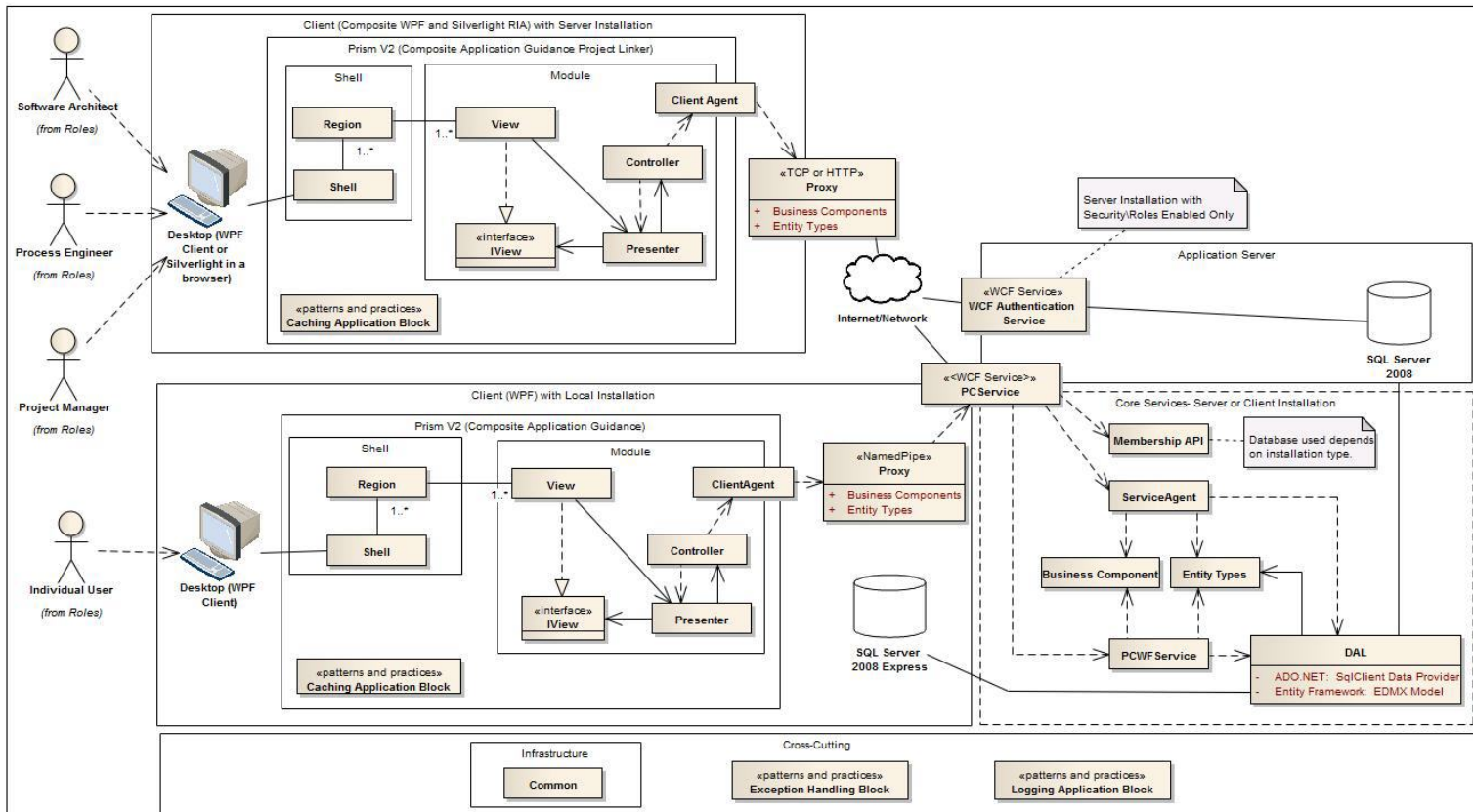
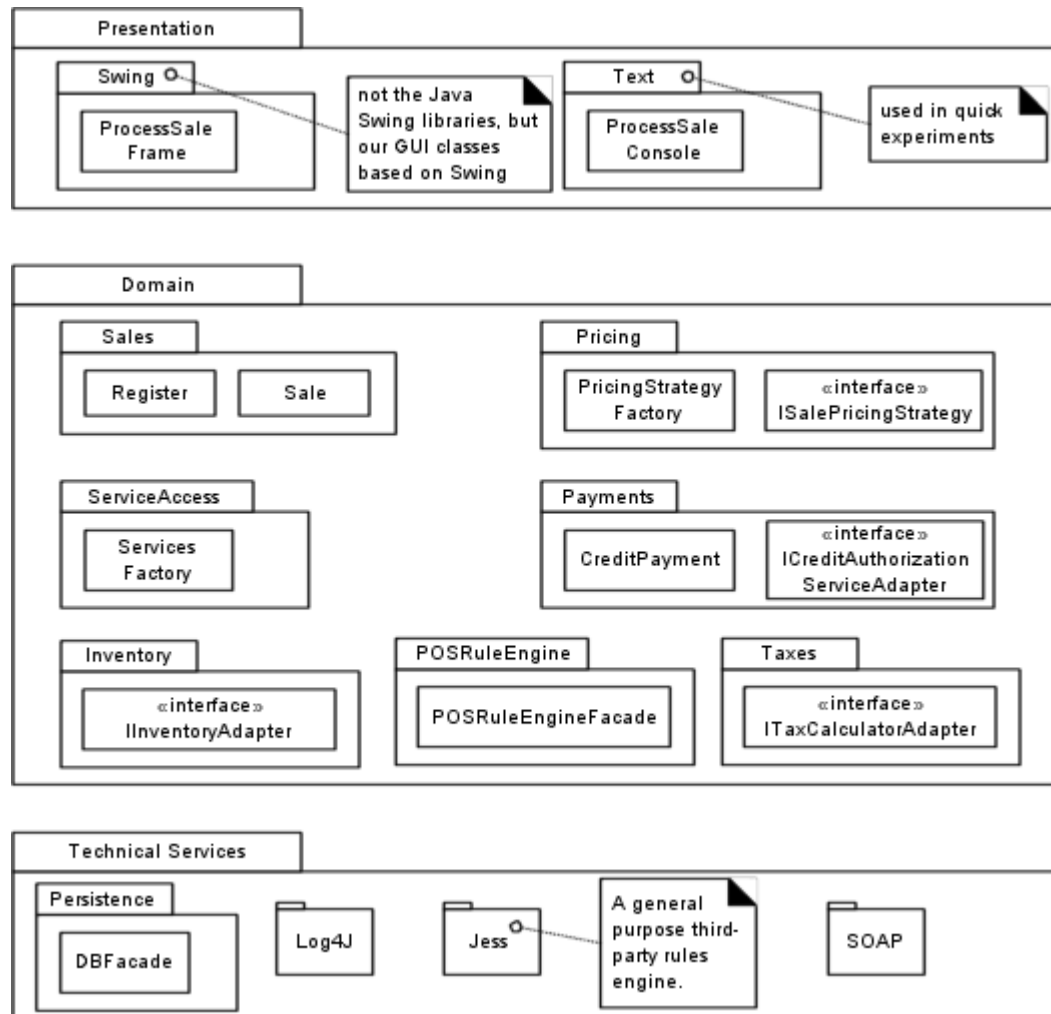


Fig. 2. K-9 mail Android app architecture.

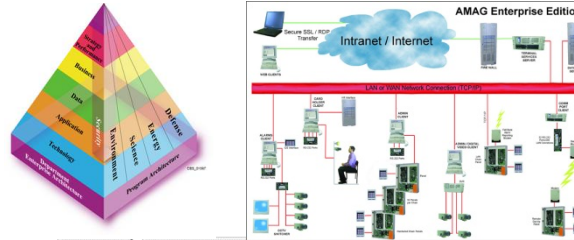
custom Free Form Overview



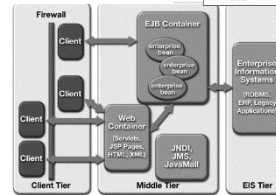


Levels of architecture*

Enterprise architecture

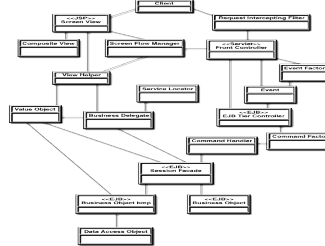


System architecture



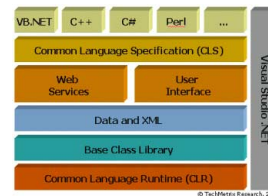
Subsystem

Application architecture



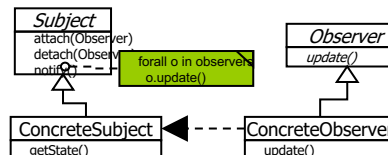
Application

Macro-architecture



Frameworks

Micro-architecture



Design patterns

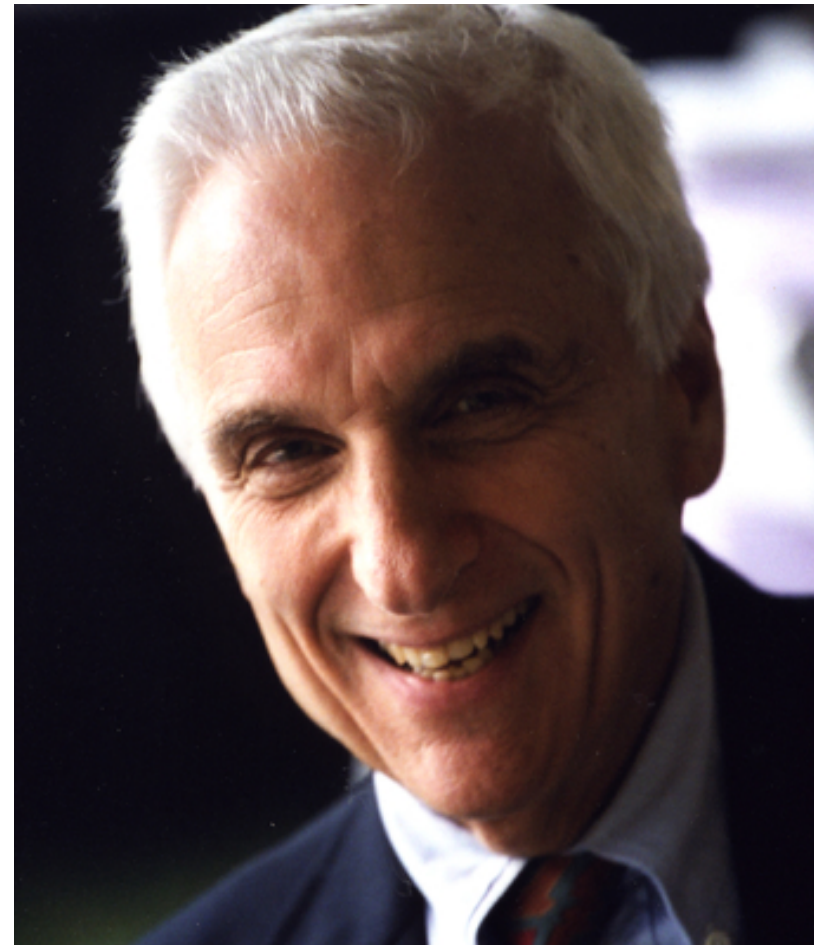
80

* Mowbray and Malveau

Conway's law

Any organization that designs a system (defined broadly) will produce a design whose structure is a copy of the organization's communication structure.

— *Melvin E. Conway*



Banking Architecture (Physical 3D model)



Rabobank's 3D model (Image source: Hans Tesselaar, BIAN)

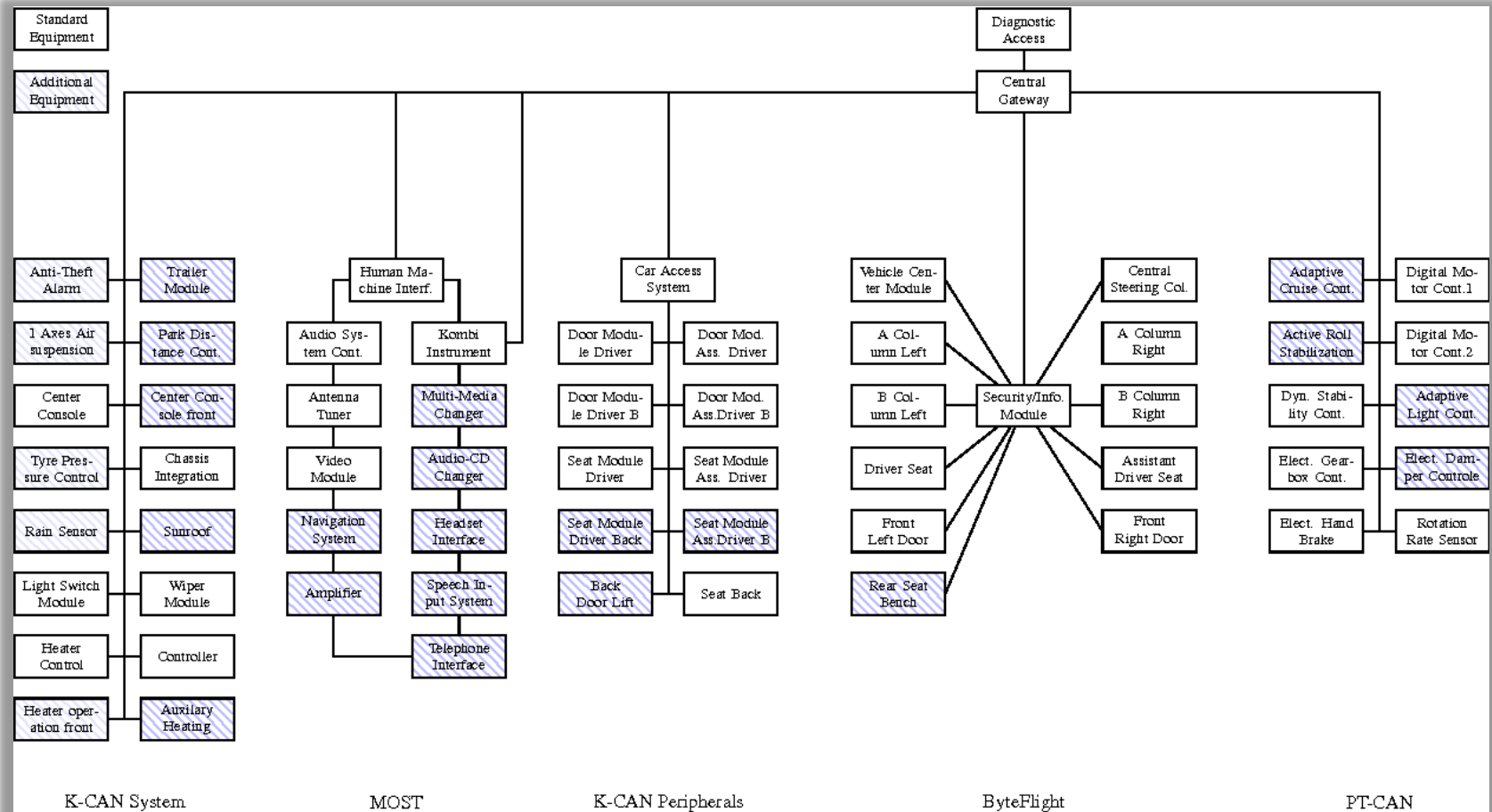
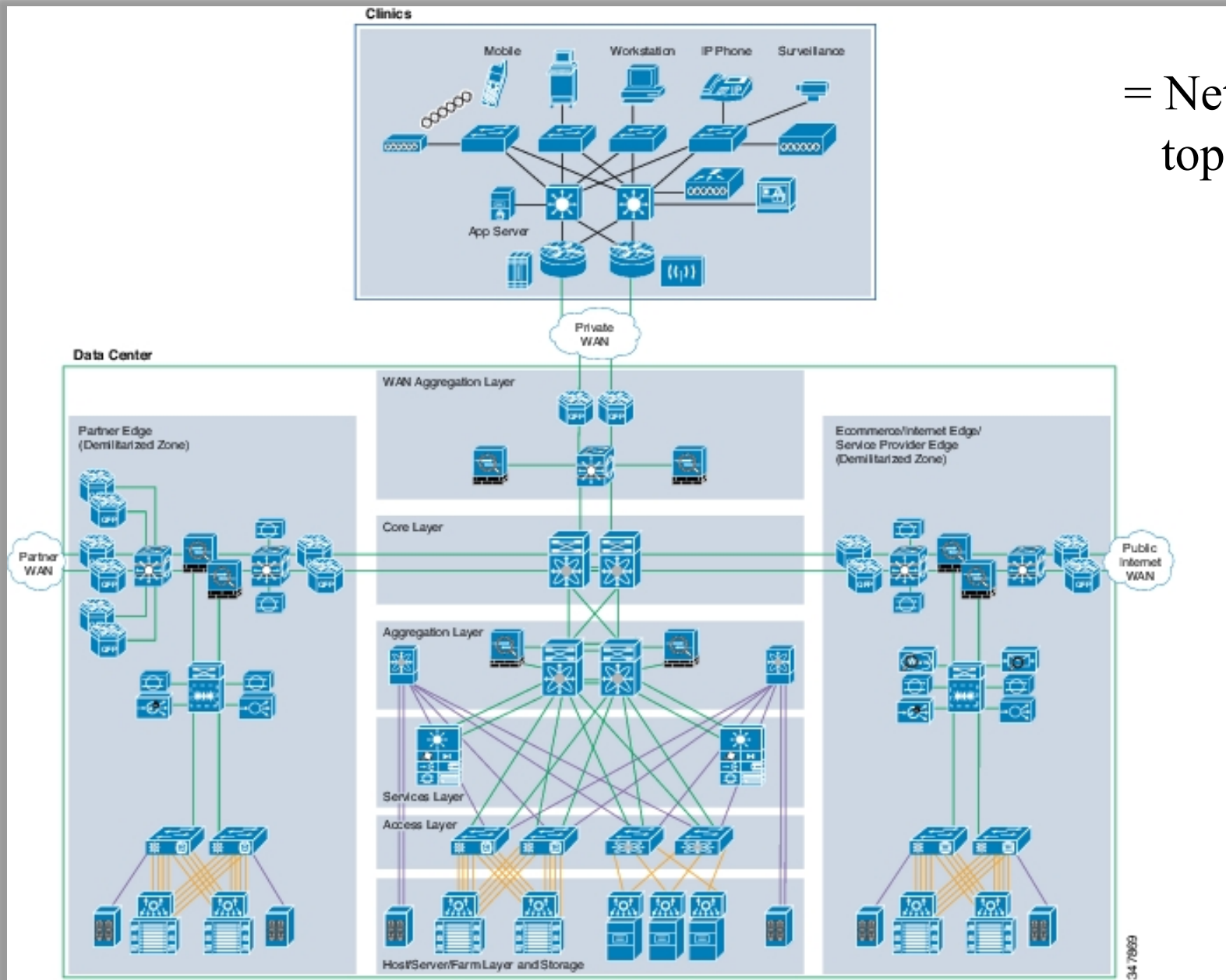


Figure 1: Decentralized ECUs of a example vehicle electrical system architecture (Barthels et al. 2012) (Michel et al. 2012)

= Network topology



What is Software Architecture?

Classic Definitions 1

An architecture is the **set of significant decisions** about

- the organization of a software system,
- the selection of the **structural elements** and their **interfaces** by which the system is composed, together with their **behaviour** as specified in the collaborations among those elements,
- the **composition** of these structural and behavioural elements into progressively larger subsystems,
- the **architectural style** that guides this organization

The UML Modeling Language User Guide, Addison–Wesley, 1999
Booch, Rumbaugh, and Jacobson

What is Software Architecture?

Definition 2

The fundamental organization of a system embodied by its components, their relationships to each other **and to the environment** and the principles guiding its design and evolution

IEEE Standard P1471 Recommended Practice for
Architectural Description of Software-Intensive Systems

What is Software Architecture?

Definition 3

IT-Architecture is the collection of principles, guidelines and modelling standards that are used to guide the ***development, maintenance and use of IT-resources*** within the entire organization.

M. Sikkema (banking industry)

Translation by M.R.V. Chaudron

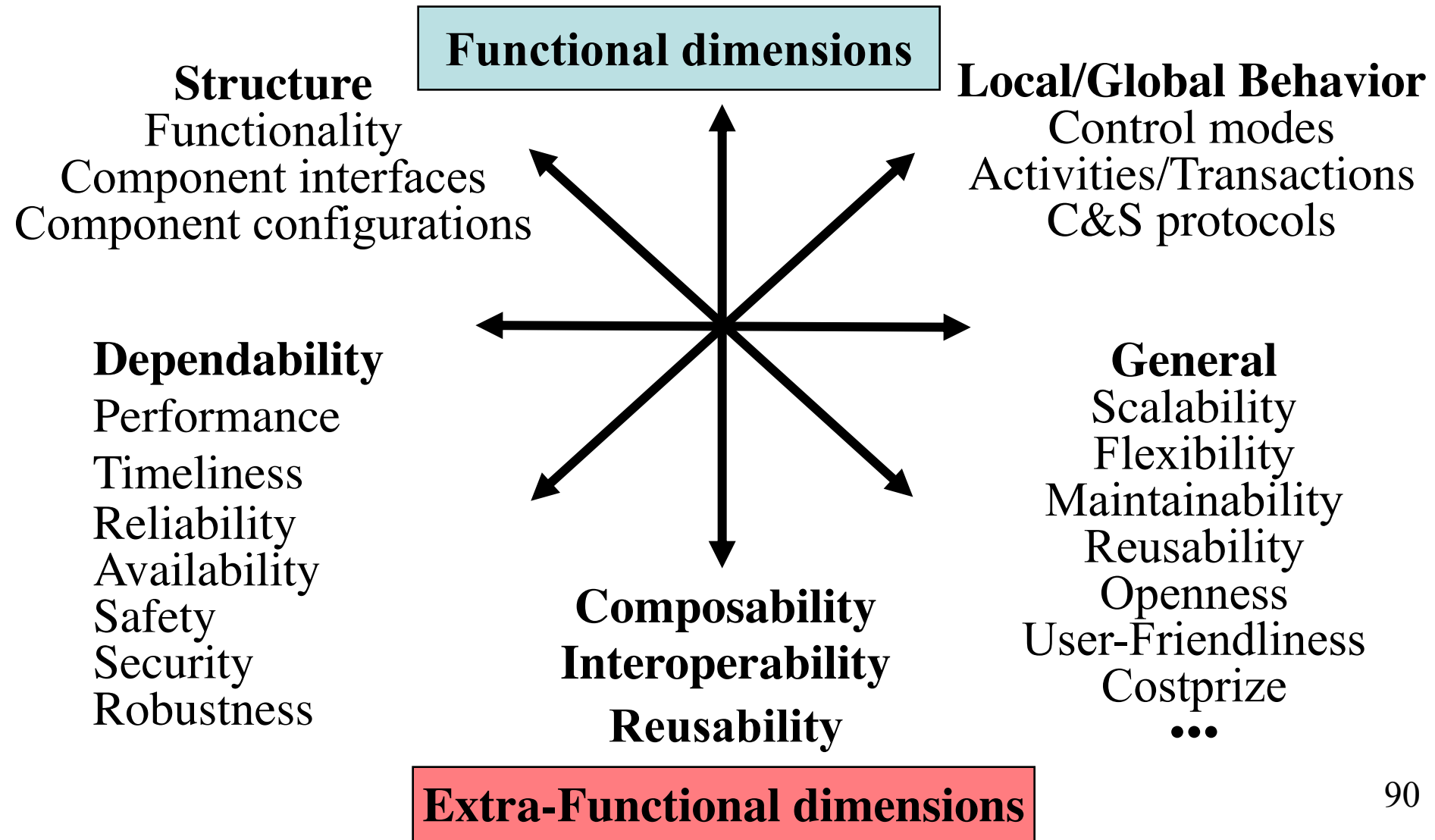
Ontwikkelen Onder Architectuur, Informatie, juni 2000

Oriented towards enterprise systems.

Software Architecture & Quality

- The notion of quality is central in software architecting: a software architecture is devised to gain insight in the qualities of a system at the earliest possible stage.
- Some qualities are observable via execution: performance, security, availability, functionality, usability
- And some are **not** observable via execution, but in the development process: modifiability, portability, reusability, integrability, testability

Architecting = Balancing Objectives



Outline

- Organization of the lectures
- Importance and use of architecture
- What is Software Architecture?
- Concluding Remarks & References

Concluding Remarks

*Experience is the hardest kind of teacher.
It gives the test first and the lesson afterward.*
Susan Ruth, 1993

- Software Architecture is a critical aspect in the design and development of software
- We discussed definitions and objectives of Sw.Arch.
- Understanding of basic principles of architecture design, analysis, documentation, and process are necessary, but experience is hard to beat.