Software Architecture DIT344

Truong Ho-Quang
truongh@chalmers.se
Software Engineering Division
Chalmers | GU

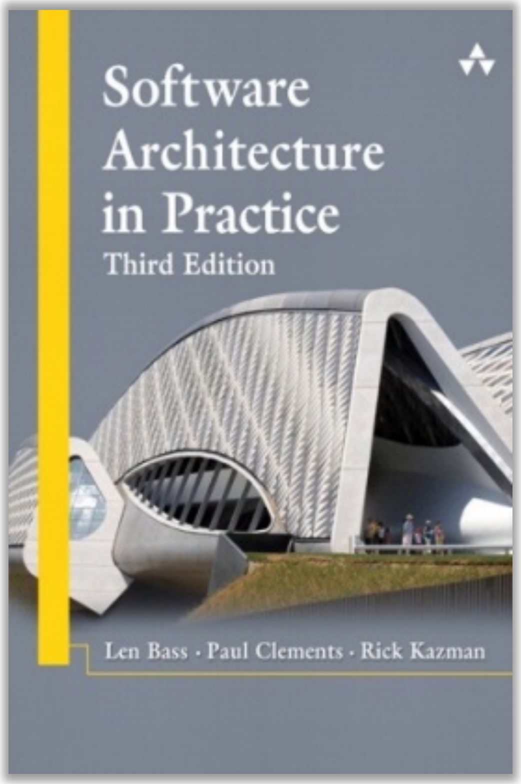
Software Architec***ting***

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

Group supervisors

Al-Amir	Adegbuji-Onikoyi	gusadegal@student.gu.se
Katalin	Ferenc	gusfereka@student.gu.se
Talha	Hussain	gustalhhu@student.gu.se
Stanko	Jankovic	gusjankost@student.gu.se
Adelric	Wong	guswonad@student.gu.se

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

Your Hopes and Expectations

- Who are you?
- What do you expect / want to learn?

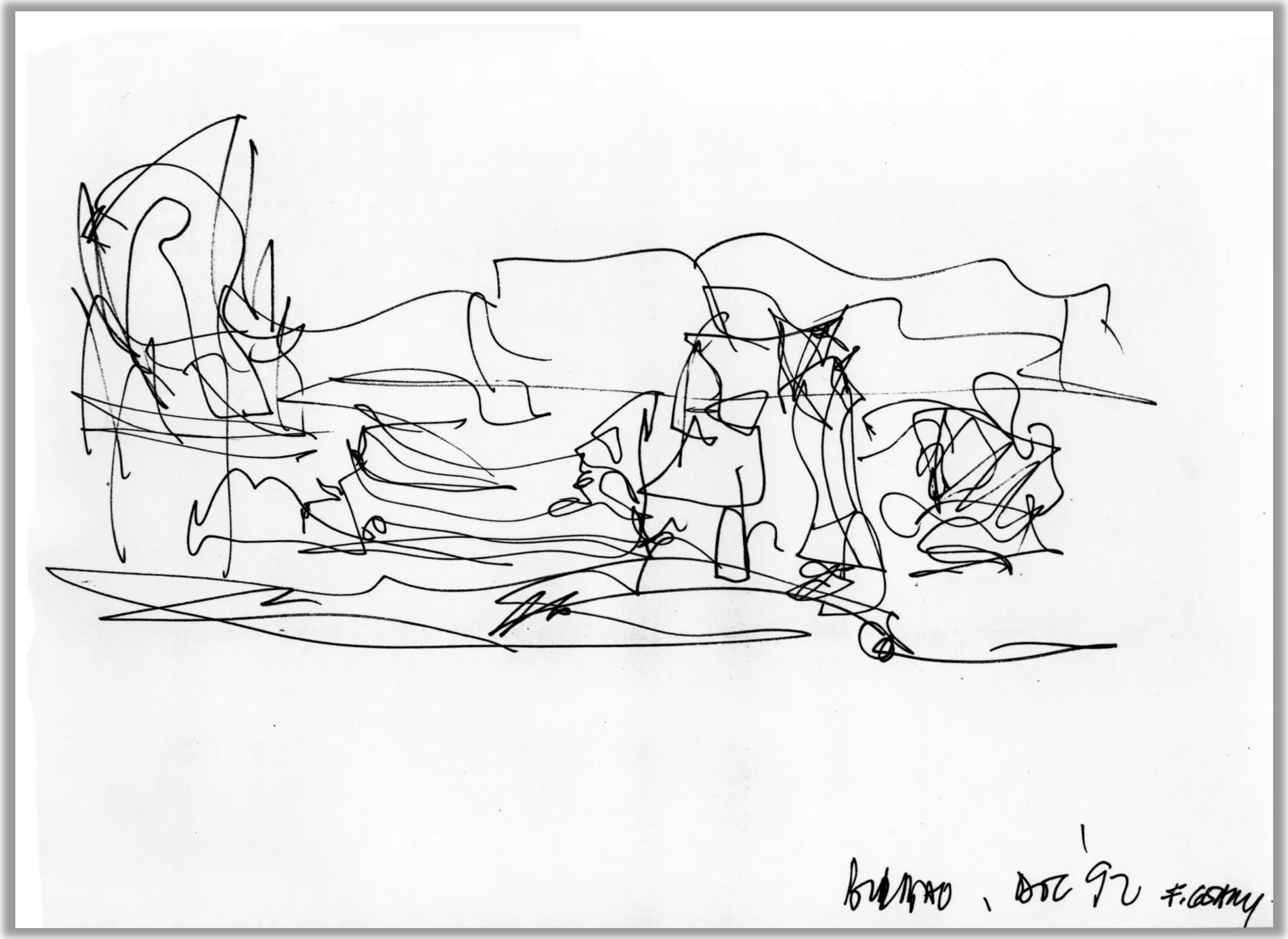
Learning/Teaching in Corona times

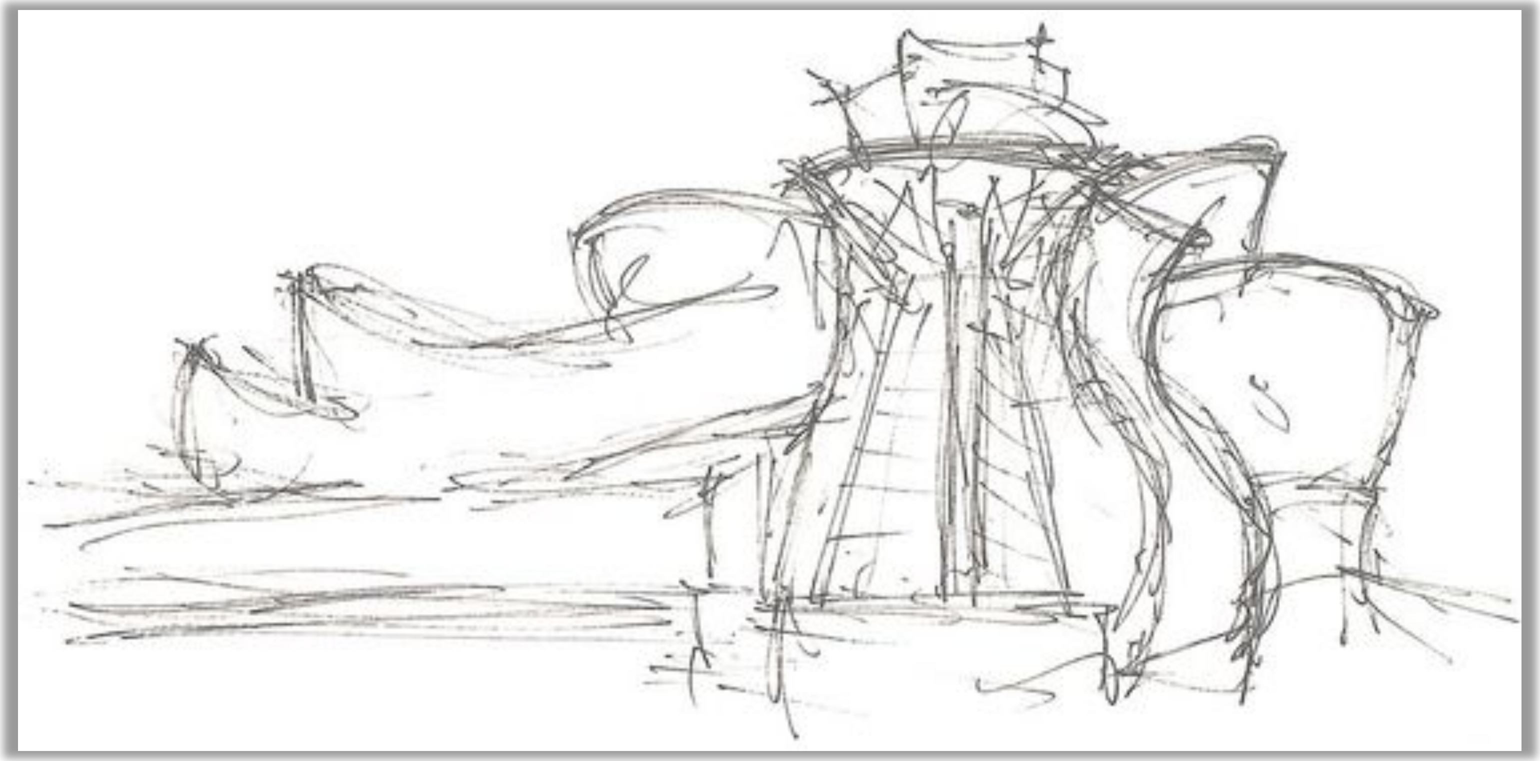
- 100% remote (No physical meetings)
  - Live lectures (‘Swedish’ timezone - UTC+2)
  - Lectures will be recorded (do NOT distribute the recordings to outside of the class)
  - Group activities will be done online
  - Remote written-exam (with ‘proctoring’)
- Course is administrative in Canvas
  - Be 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

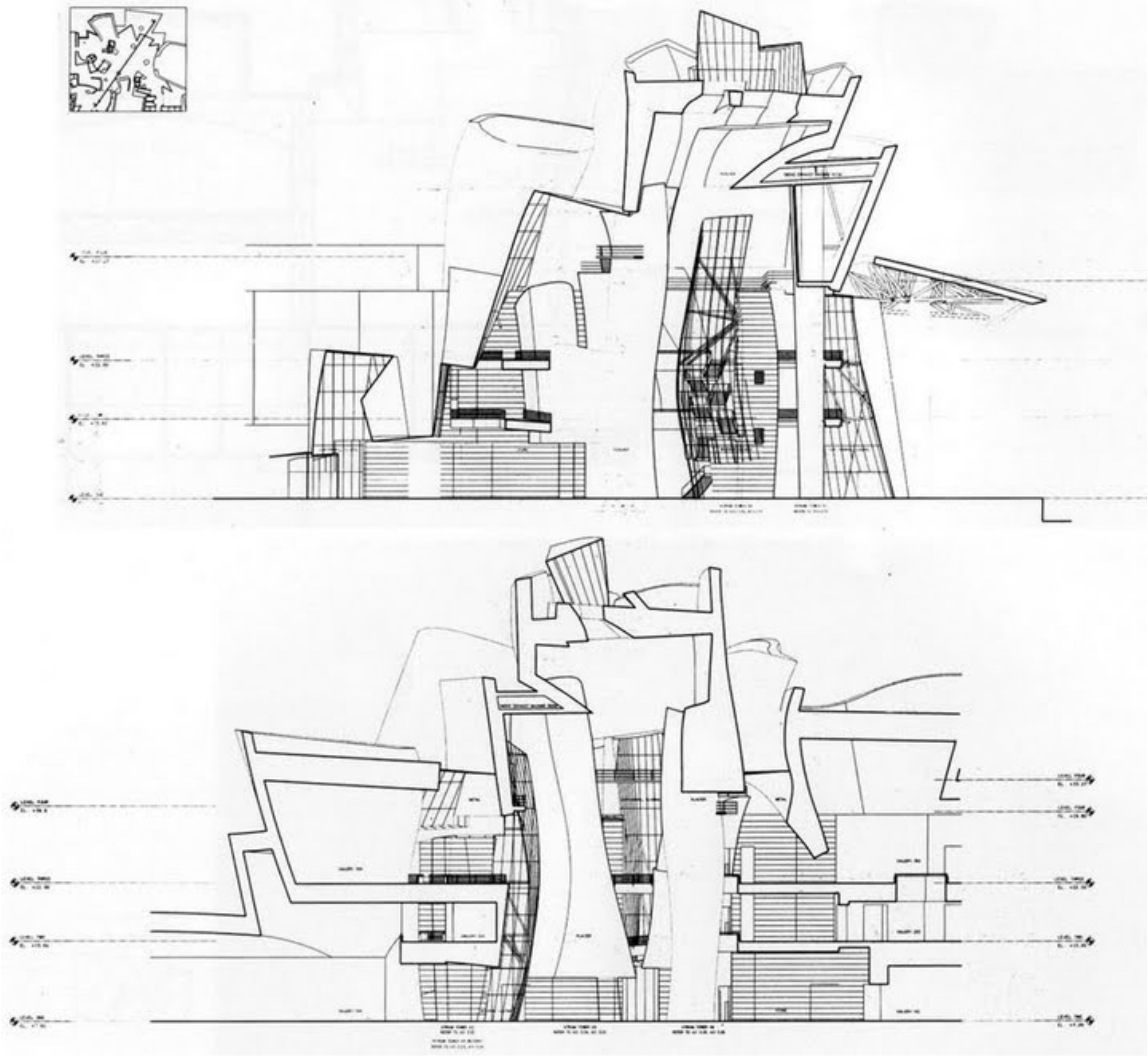
What is this? What is it for?



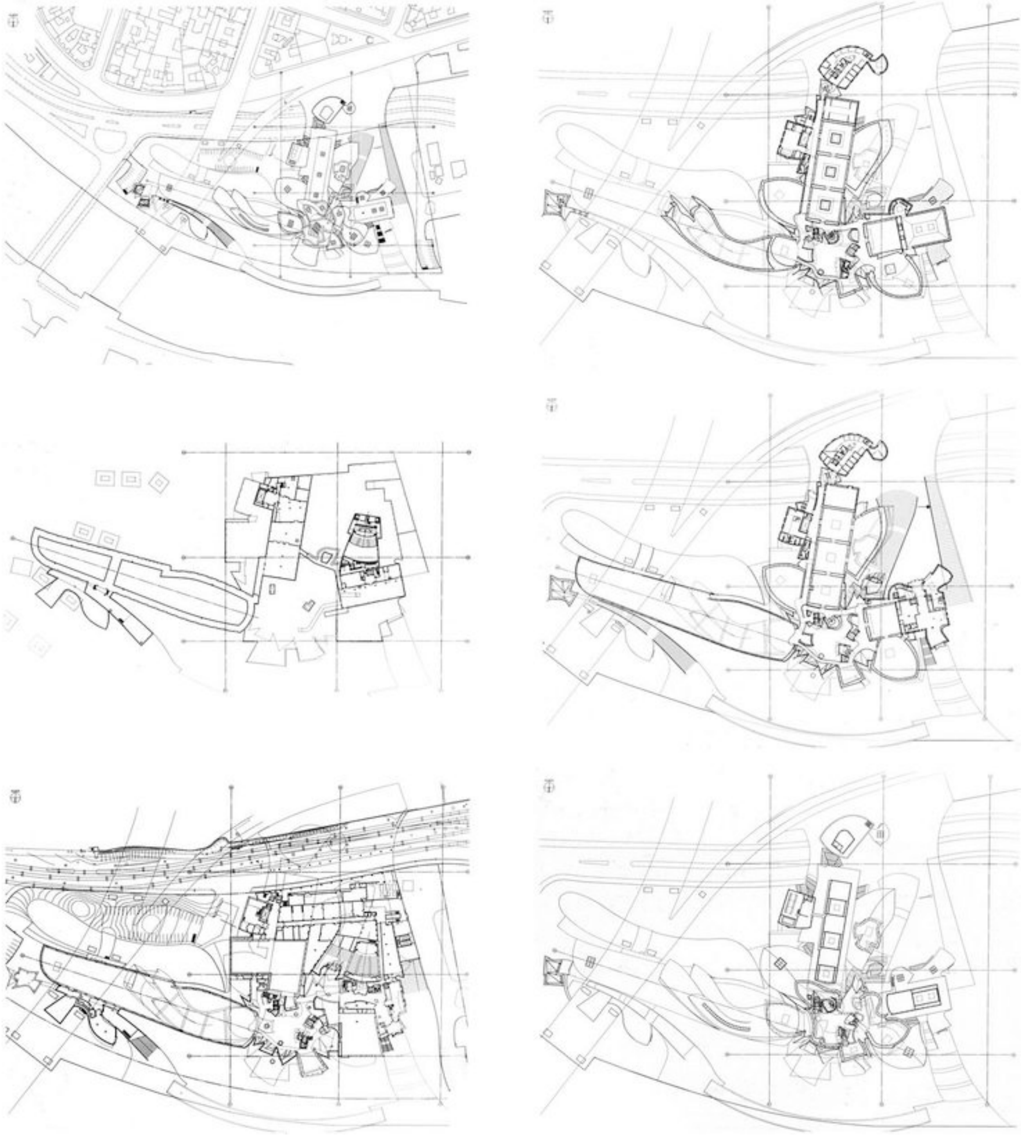


Form & Function

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
36	L1	Wed, 2 Sept	13:15 – 15:00	Introduction & Organization	Truong Ho
37	L2	Wed, 9 Sept	13:15 – 15:00	Architecting Process & Views	Truong Ho
37	S1	Thu, 10 Sept	10:15 – 12:00	<< Supervision/Assignment>>	TAs
38	L3	Wed, 16 Sept	13:15 - 15:00	Requirements & Quality Attributes	Sam Jobara
38	S2	Thu, 17 Sept	10:15 – 12:00	<< Supervision/Assignment>>	TAs
38	L4	Fri, 18 Sept	13:15 – 15:00	Architectural Tactics & Roles and Responsibilities	Truong Ho
39	S3	Wed, 23 Sept	13:15 – 15:00	<< Supervision/Assignment>>	TAs
39	L5	Thu, 24 Sept	10:15 – 12:00	Functional Decomposition & Architectural Styles P1	Truong Ho
39	L6	Fri, 25 Sept	13:15 – 15:00	Architectural Styles P2	Sam Jobara
40	S4	Wed, 30 Sept	13:15 – 15:00	<< Supervision/Assignment>>	TAs
40	L7	Thu, 1 Oct	10:15 – 12:00	Architectural Styles P3	Truong Ho
40	L8	Fri, 2 Oct	13:00 – 15:00	Guest Lecture: Scaling DevOps – GitHub's Journey from 500+ to 1500+ People	Johannes Nicolai
41	S5	Wed, 7 Oct	13:15 – 15:00	<< Supervision/Assignment>>	TAs
41	L9	Thu, 8 Oct	10:15 – 12:00	Current Industrial SW Architecture Issues: Software Architectures of Blockchain with Case Study	Sam Jobara
42	L10	Wed, 14 Oct	13:15 – 15:00	Design Principles	Truong Ho
42	S6	Thu, 15 Oct	10:15 – 12:00	<< Supervision/Assignment>>	TAs
42	L11	Fri, 16 Oct	13:15 – 15:00	Guest Lecture: Architecture changes at Volvo Truck's Application System (TAS)	Anders Magnusson
43	L12	Wed, 21 Oct	13:15 – 15:00	Architecture Evaluation	Truong Ho
43	L13	Thu, 22 Oct	10:15 – 12:00	Reverse Engineering & Correspondence	Truong Ho
43		Fri, 23 Oct	13:00 – 15:00	To be determined (exam practice?)	Teachers
44	Exam	30 Oct	8:30 – 12:30		

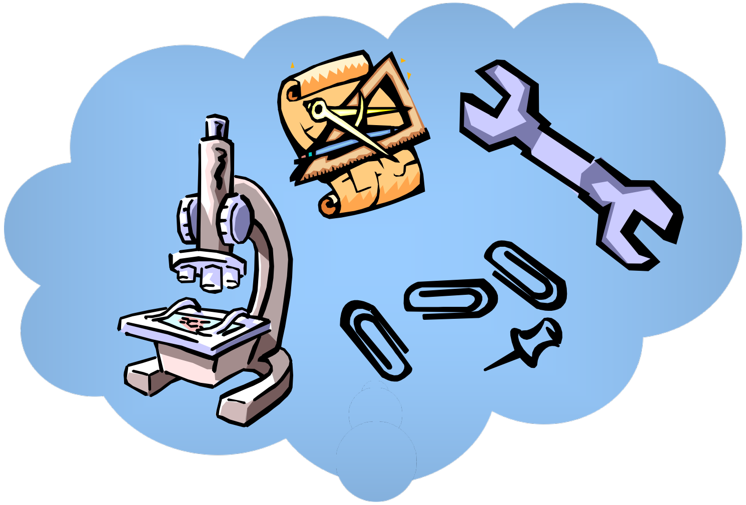
Here we are

Schedule

Week		Date	Time	Lecture	Note
36	L1	Wed, 2 Sept	13:15 – 15:00	Introduction & Organization	Truong Ho
37	L2	Wed, 9 Sept	13:15 – 15:00	Architecting Process & Views	Truong Ho
37	S1	Thu, 10 Sept	10:15 – 12:00	<< Supervision/Assignment>>	TAs
38	L3	Wed, 16 Sept	13:15 - 15:00	Requirements & Quality Attributes	Sam Jobara
38	S2	Thu, 17 Sept	10:15 – 12:00	<< Supervision/Assignment>>	TAs
38	L4	Fri, 18 Sept	13:15 – 15:00	Architectural Tactics & Roles and Responsibilities	Truong Ho
39	S3	Wed, 23 Sept	13:15 – 15:00	<< Supervision/Assignment>>	TAs
39	L5	Thu, 24 Sept	10:15 – 12:00	Functional Decomposition & Architectural Styles P1	Truong Ho
39	L6	Fri, 25 Sept	13:15 – 15:00	Architectural Styles P2	Sam Jobara
40	S4	Wed, 30 Sept	13:15 – 15:00	<< Supervision/Assignment>>	TAs
40	L7	Thu, 1 Oct	10:15 – 12:00	Architectural Styles P3	Truong Ho
40	L8	Fri, 2 Oct	13:00 – 15:00	Guest Lecture: Scaling DevOps – GitHub's Journey from 500+ to 1500+ People	Johannes Nicolai
41	S5		- 15:00	<< Supervision/Assignment>>	TAs
41	L9		- 12:00	Current Industrial SW Architecture Issues: Software Architectures of Blockchain with Case Study	Sam Jobara
42	L10		- 15:00	Design Principles	Truong Ho
42	S6		– 12:00	<< Supervision/Assignment>>	TAs
42	L11	Fri, 16 Oct	13:15 – 15:00	Guest Lecture: Architecture changes at Volvo Truck's Application System (TAS)	Anders Magnusson
43	L12	Wed, 21 Oct	13:15 – 15:00	Architecture Evaluation	Truong Ho
43	L13	Thu, 22 Oct	10:15 – 12:00	Reverse Engineering & Correspondence	Truong Ho
43		Fri, 23 Oct	13:00 – 15:00	To be determined (exam practice?)	Teachers
44	Exam	30 Oct	8:30 – 12:30		

Industrial lecture series

Teaching Philosophy



+

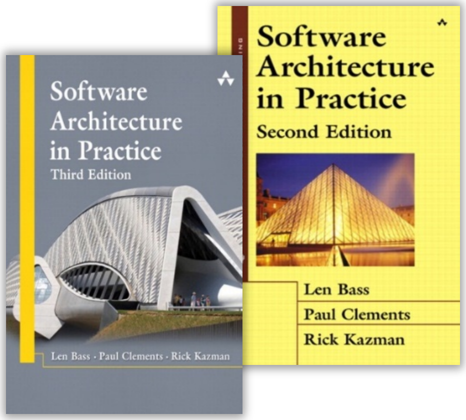
Tools

Directions

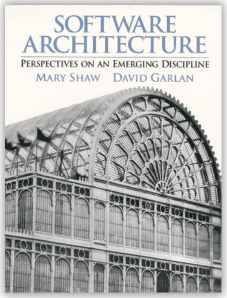
If you have a hammer, then this doesn't mean all problems are a nail!

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 in Practice
Third Edition
Len Bass · Paul Clements · Rick Kazman
Software Architecture in Practice
Second Edition
Len Bass
Paul Clements
Rick Kazman

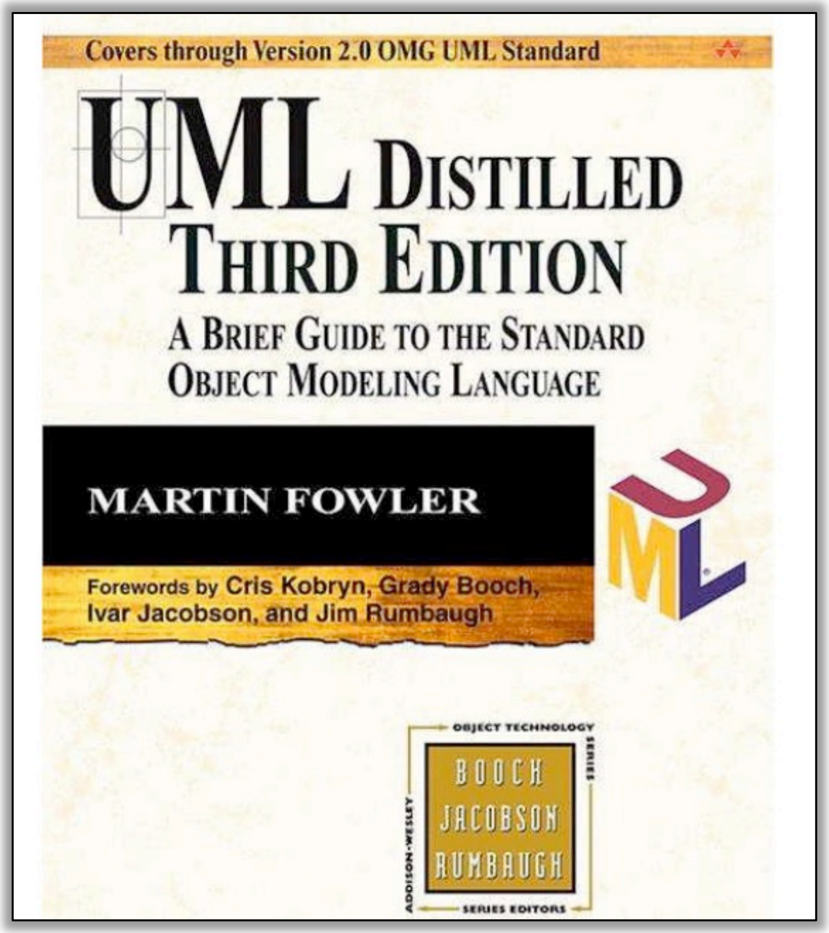
- Software Architecture: Perspectives on an Emerging Discipline, Mary Shaw, David Garlan, 242 pages, 1996, Prentice Hall



- 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



Covers through Version 2.0 OMG UML Standard

UML DISTILLED
THIRD EDITION

A BRIEF GUIDE TO THE STANDARD
OBJECT MODELING LANGUAGE

MARTIN FOWLER

Forewords by Cris Kobryn, Grady Booch,
Ivar Jacobson, and Jim Rumbaugh

OBJECT TECHNOLOGY SERIES
BOOCH
JACOBSON
RUMBAUGH
SERIES EDITORS
ADDISON-WESLEY

Software Architecture Books

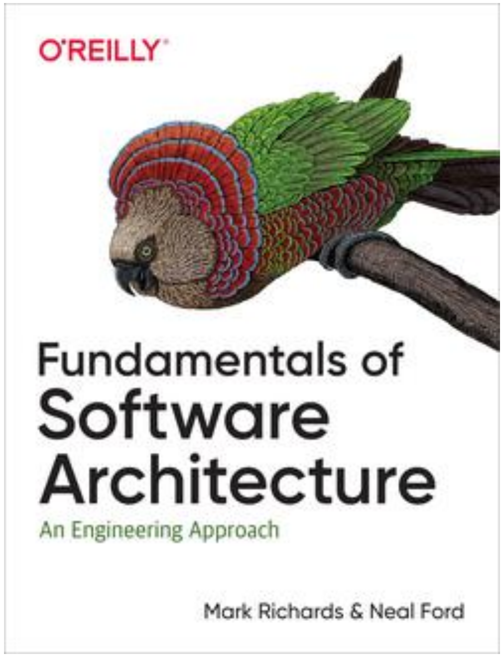
- Fundamentals of Software Architecture

  by Mark Richards, Neal Ford

  Released January 2020

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

  ISBN: 9781492043454



O'REILLY

Fundamentals of Software Architecture

An Engineering Approach

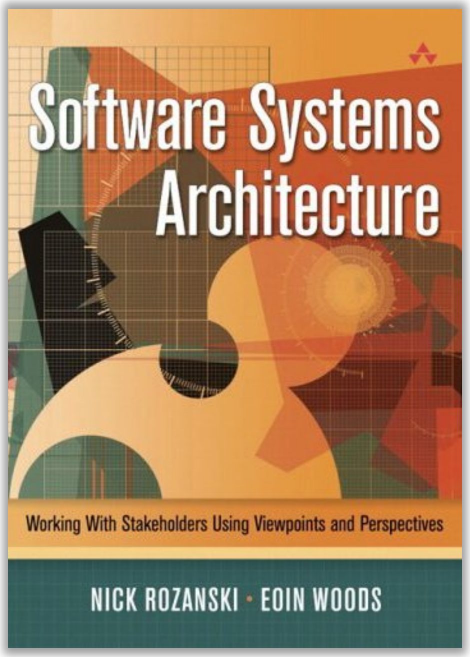
Mark Richards & Neal Ford

Software Architecture Books

- Software Systems Architecture – With Stakeholders Using Views and Perspectives

  by Nick Rozanski and Eoin Woods,

  Addison-Wesley, 2005


Software Systems Architecture
Working With Stakeholders Using Viewpoints and Perspectives
NICK ROZANSKI · EOIN WOODS

Software Architecture Books

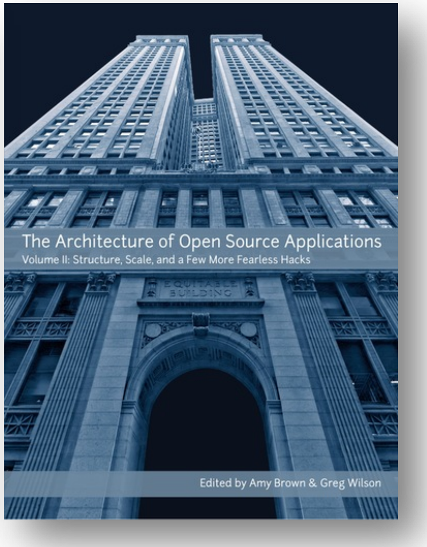
- The Architecture of Open Source Applications (Volumn I & II)

  Edited by Amy Brown & Greg Wilson

  http://aosabook.org/blog/

  Twitter at @aosabook (#aosa hashtag)

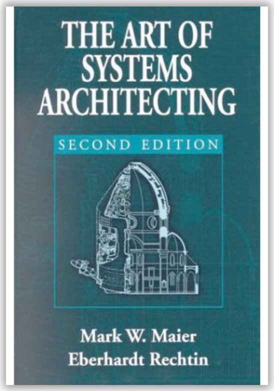
The Architecture of
Open Source Applications
Elegance, Evolution, and a Few Fearless Hacks
Edited by Amy Brown & Greg Wilson


The Architecture of Open Source Applications
Volume II: Structure, Scale, and a Few More Fearless Hacks
Edited by Amy Brown & Greg Wilson

Software Architecture Books

- The Art of Systems Architecting, Second Edition (Hardcover) by Mark W. Maier, Eberhardt Rechtin

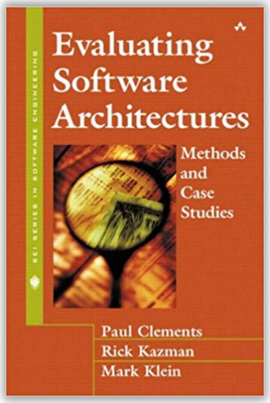
  Publisher: CRC Press, 2000



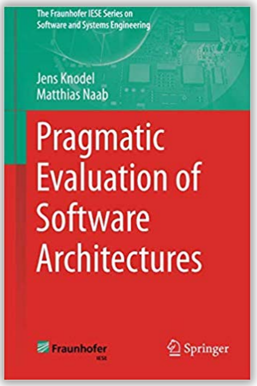
- Evaluating Software Architecture

  Paul Clements, Rick Kazman, Marc Klein,

  2002, Addison-Wesley, ISBN 020170482X

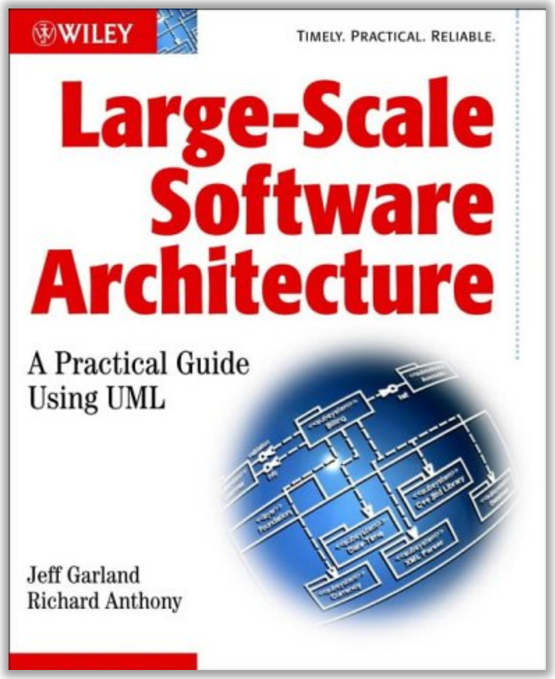

Evaluating Software Architectures
Methods and Case Studies
SEI SERIES IN SOFTWARE ENGINEERING
Paul Clements
Rick Kazman
Mark Klein

- Pragmatic Evaluation of Software Architectures, Jens Knodel, Matthias Naab, Springer, 2016


The Fraunhofer IESE Series on Software and Systems Engineering
Jens Knodel
Matthias Naab
Pragmatic Evaluation of Software Architectures
Fraunhofer
Springer

Software Architecture Books

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


WILEY
TIMELY. PRACTICAL. RELIABLE.
Large-Scale Software Architecture
A Practical Guide Using UML
Jeff Garland
Richard Anthony

Evaluation

- Assignments parallel to course
- Final ‘remote’ written exam (with ‘proctoring’)
- 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*

- There are a few things from this course that are very important
  I will emphasize them and expect you notice

Assignment

- To be announced
- Start of term
    - UML
- Assignment follows typical steps:
    - Understand requirements (apply knowledge from RE course)
    - Identify stakeholders
    - Identify architectural drivers
    - Iterate:
        - Identify components & behaviours
        - Use patterns and tactics to achieve quality properties
    - Hand in: RE doc & SW ARCH DESIGN doc
- Peer-evaluation for assessing individual contribution to assignments

Group Formation

- Group of 4 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, Sept. 7, 2020**.
- All groups will be formed and assigned to a supervisor on Sept. 8.

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.

Structure of the Course (1)

- Introduction:
  - why, what, when, whom, …
  - course ‘logistics’
  - a bit on software architecture notion
- Requirements, Architecture Modelling & Documentation
  - bit more about requirements
  - using UML for modelling
  - documentation guidelines
- 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 (3 lectures)
    - Architectural changes to support DevOps practices
    - Architecture of Blockchain + Case study
    - Evolution of SW Architecture in Automotive domain

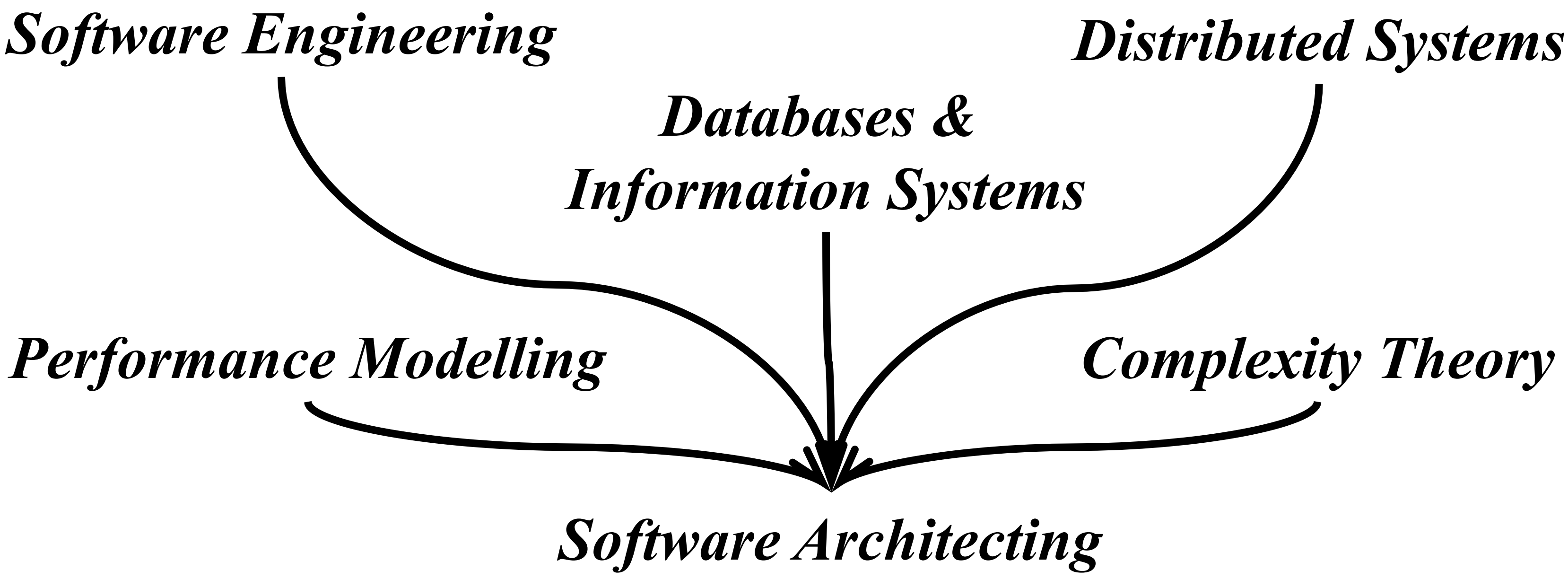
Structure of the Course (3)

- Architecture Evaluation
  - Qualitative:
    - Scenario-based: ATAM
  - Quantitative Techniques for Analysing Architectures
    - Reliability Block Diagrams, Rate Monotonic Analysis
    - Queuing Networks
    - Risk, Cost, Feasability

- Architecture Reconstruction (reverse engineering)

- Last lecture: to be confirmed

Relation to Other CS subjects



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

Preview / Recap: What is Software Architecture?

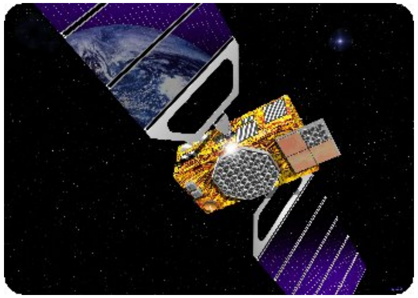
A definition:

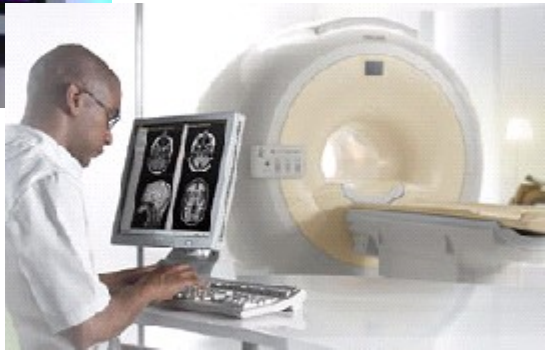
Software Architecture is the global organization of a software system, including
- the division of software into subsystems/components,
- policies according to which these subsystems interact,
- the definition of their interfaces.

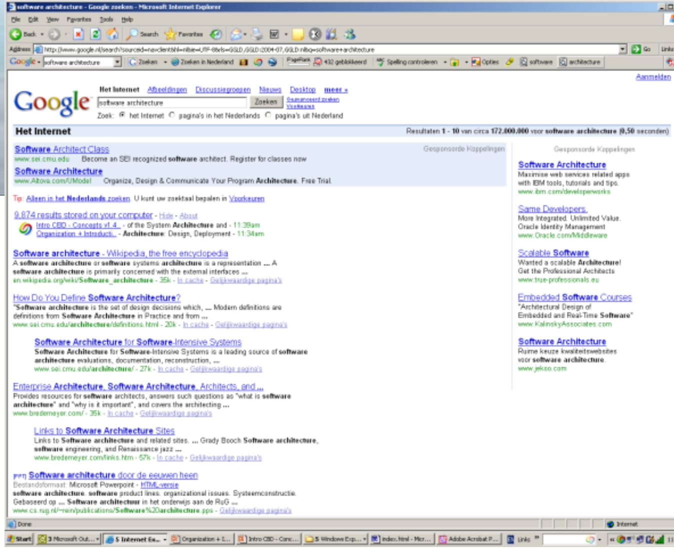
(free after) Object Oriented Software Engineering
T. C. Lethbridge & R. Laganière
McGraw Hill, 2001

Outline

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







Het Internet

Resultaten 1 - 10 van circa 172.000.000 voor software architecture (0,50 seconden)

Gesponsorde Koppelingen

Software Architect Class
www.sei.cmu.edu Become an SEI recognized software architect. Register for classes now

Software Architecture
www.Altova.com/UModel Organize, Design & Communicate Your Program Architecture. Free Trial.

Tip: Alleen in het Nederlands zoeken. U kunt uw zoektaal bepalen in Voorkeuren

9.874 results stored on your computer - Hide - About
Intro CBD - Concepts v1.4... - of the System Architecture and - 11:39am
Organization + Introducti... - Architecture: Design, Deployment - 11:34am

Software architecture - Wikipedia, the free encyclopedia
A software architecture or software systems architecture is a representation .... A software architecture is primarily concerned with the external interfaces ...
en.wikipedia.org/wiki/Software_architecture - 35k - In cache - Gelijkwaardige pagina's

How Do You Define Software Architecture?
"Software architecture is the set of design decisions which, ... Modern definitions are definitions from Software Architecture in Practice and from ...
www.sei.cmu.edu/architecture/definitions.html - 20k - In cache - Gelijkwaardige pagina's

Software Architecture for Software-Intensive Systems
Software Architecture for Software-Intensive Systems is a leading source of software architecture evaluations, documentation, reconstruction, ...
www.sei.cmu.edu/architecture/ - 27k - In cache - Gelijkwaardige pagina's

Enterprise Architecture, Software Architecture, Architects, and ...
Provides resources for software architects, answers such questions as "what is software architecture" and "why is it important", and covers the architecting ...
www.bredemeyer.com/ - 35k - In cache - Gelijkwaardige pagina's

Links to Software Architecture Sites
Links to Software architecture and related sites. ... Grady Booch Software architecture, software engineering, and Renaissance jazz ...
www.bredemeyer.com/links.htm - 57k - In cache - Gelijkwaardige pagina's

[PPT] Software architecture door de eeuwen heen
Bestandsformaat: Microsoft Powerpoint - HTML-versie
software architecture. software product lines. organizational issues. Systeemconstructie. Gebaseerd op ... Software architectuur in het onderwijs aan de RuG ...
www.cs.rug.nl/~rein/publications/Software%20architecture.pps - Gelijkwaardige pagina's

Gesponsorde Koppelingen

Software Architecture
Maximise web services related apps with IBM tools, tutorials and tips.
www.ibm.com/developerworks

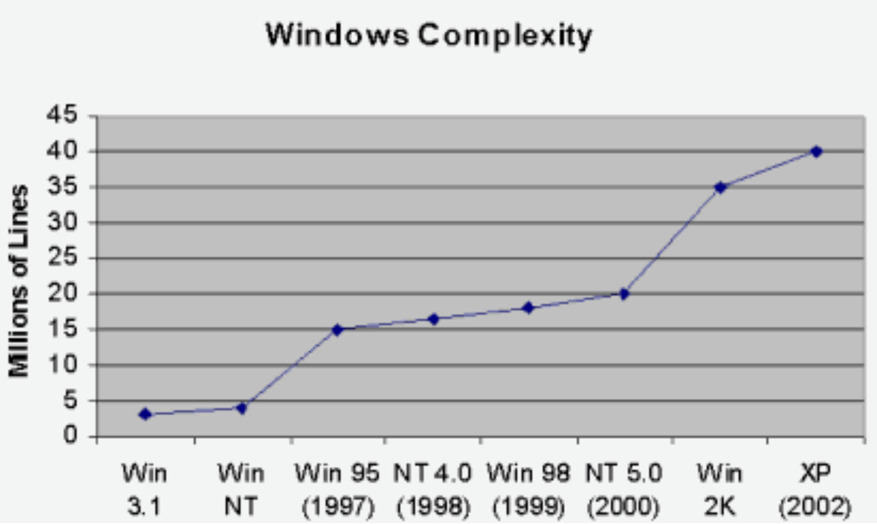
Same Developers.
More Integrated. Unlimited Value. Oracle Identity Management
www.Oracle.com/Middleware

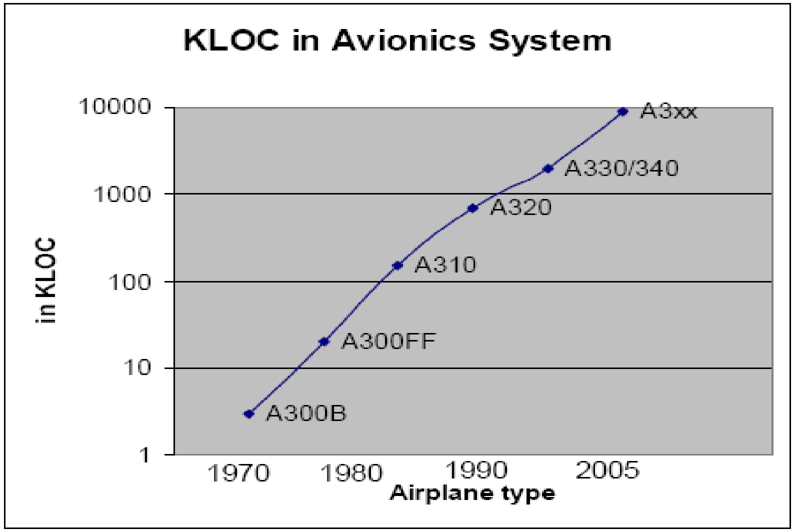
Scalable Software
Wanted a scalable Architecture! Get the Professional Architects
www.true-professionals.eu

Embedded Software Courses
"Architectural Design of Embedded and Real-Time Software"
www.KalinskyAssociates.com

Software Architecture
Ruime keuze kwaliteitswebsites voor software architecture.
www.jekoo.com

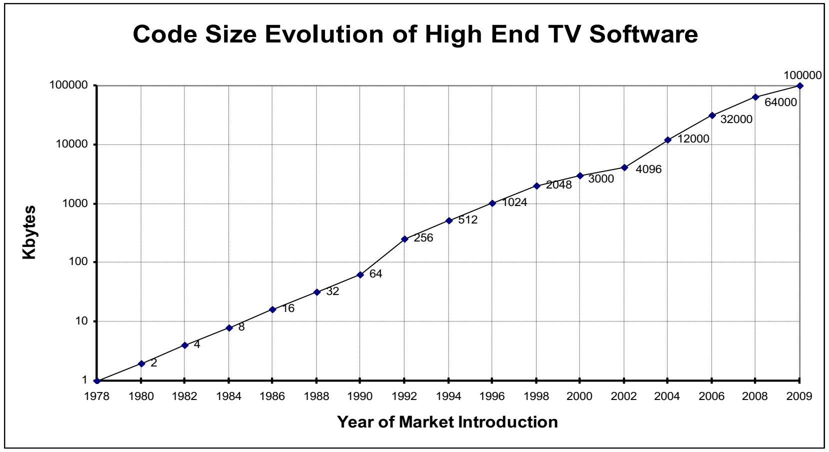
Increasing amount of software in systems



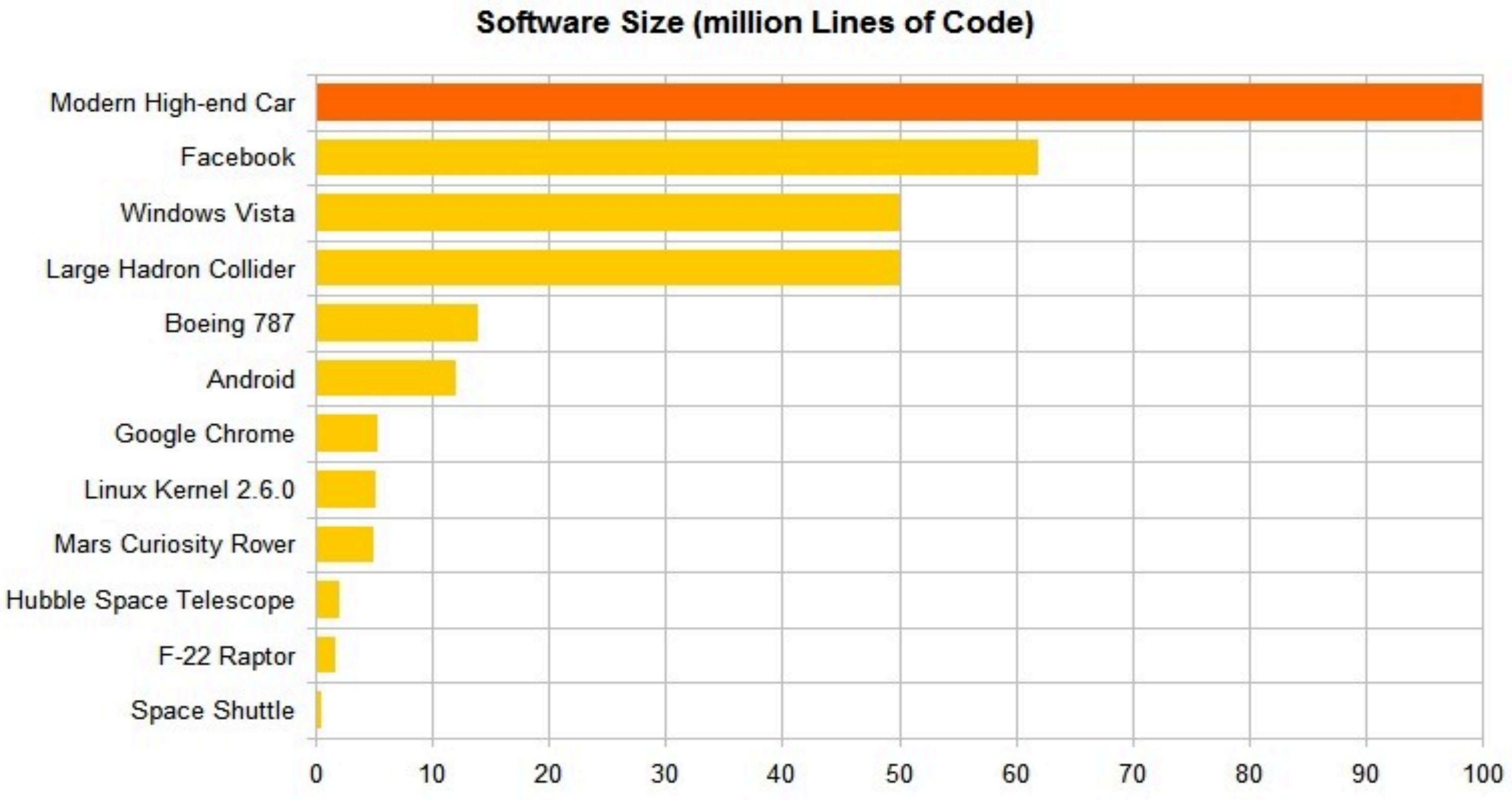


Nb: logarithmic scale

The amount of software increases 10 fold every 10 years

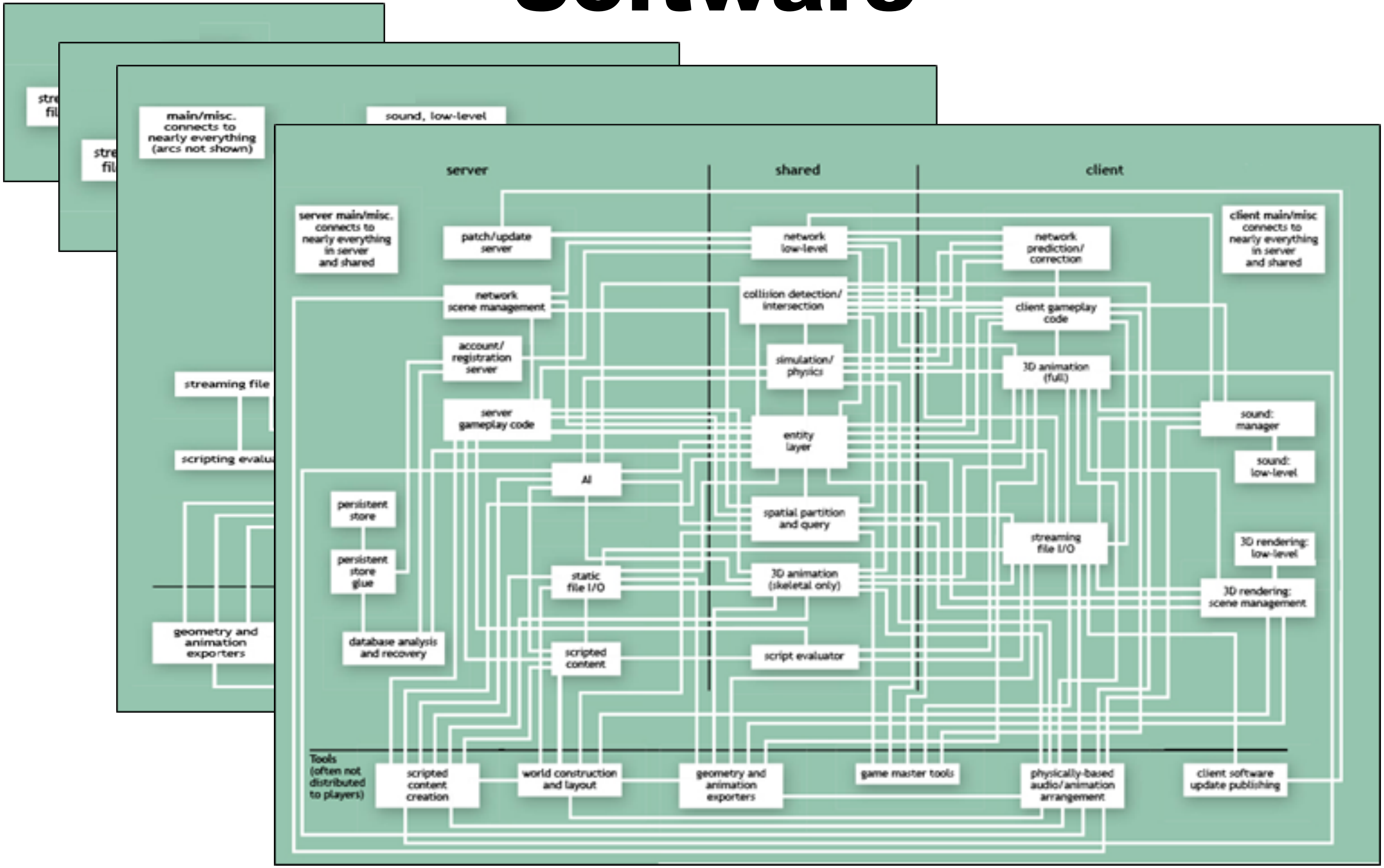


Software size grows enormously



Source: https://www.informationisbeautiful.net/visualizations/million-lines-of-code/

Increasing Complexity of Software


stre fil
stre fil
main/misc. connects to nearly everything (arcs not shown)
sound, low-level
streaming file
scripting evalu
geometry and animation exporters
server
shared
client
server main/misc. connects to nearly everything in server and shared
patch/update server
network scene management
account/ registration server
server gameplay code
AI
persistent store
persistent store glue
static file I/O
database analysis and recovery
scripted content
network low-level
collision detection/ intersection
simulation/ physics
entity layer
spatial partition and query
3D animation (skeletal only)
script evaluator
network prediction/ correction
client gameplay code
3D animation (full)
streaming file I/O
client main/misc connects to nearly everything in server and shared
sound: manager
sound: low-level
3D rendering: low-level
3D rendering: scene management
Tools (often not distributed to players)
scripted content creation
world construction and layout
geometry and animation exporters
game master tools
physically-based audio/animation arrangement
client software update publishing

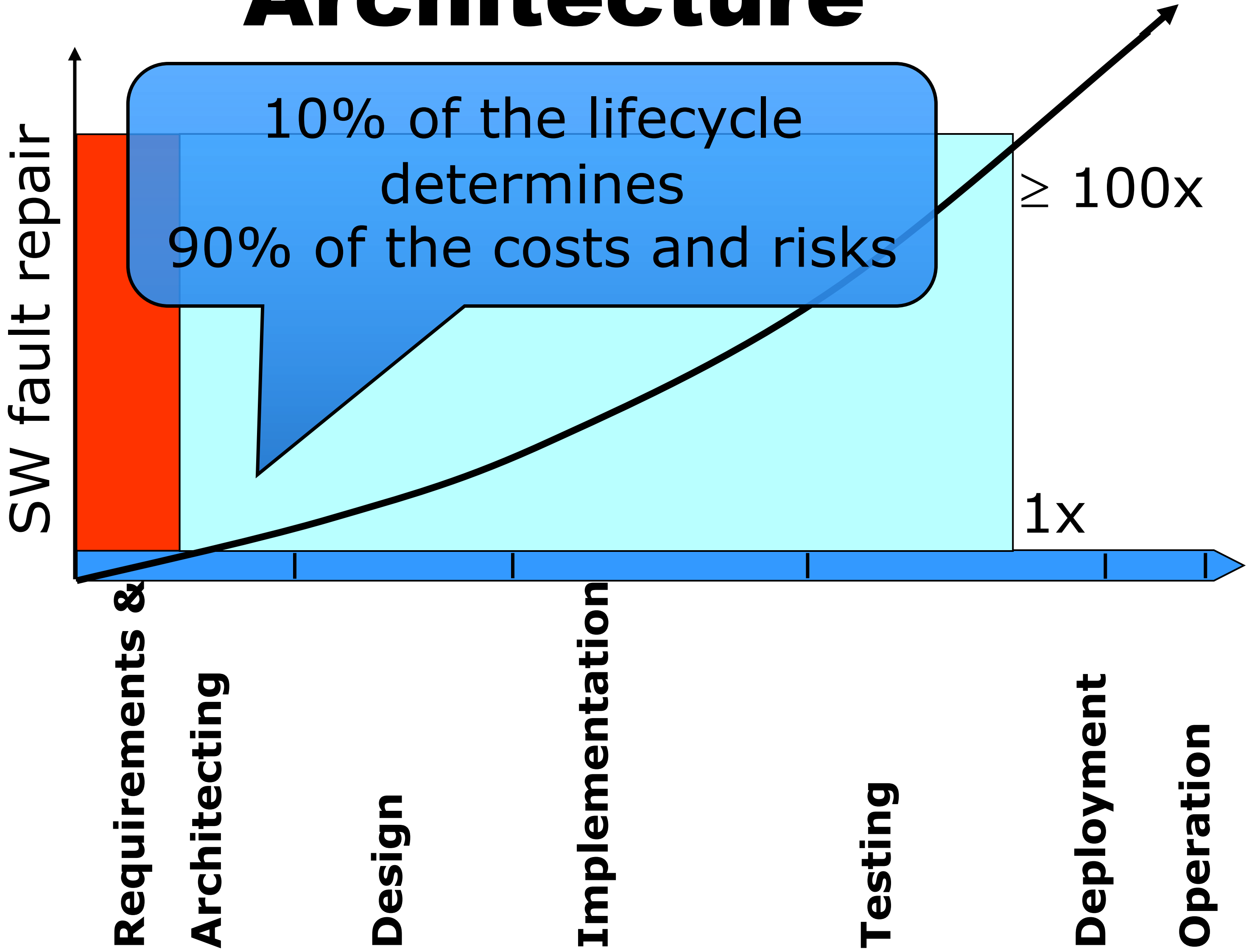
Blow, J. "Game Development: Harder Than You Think." *ACM Queue*

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



10% of the lifecycle determines
90% of the costs and risks

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

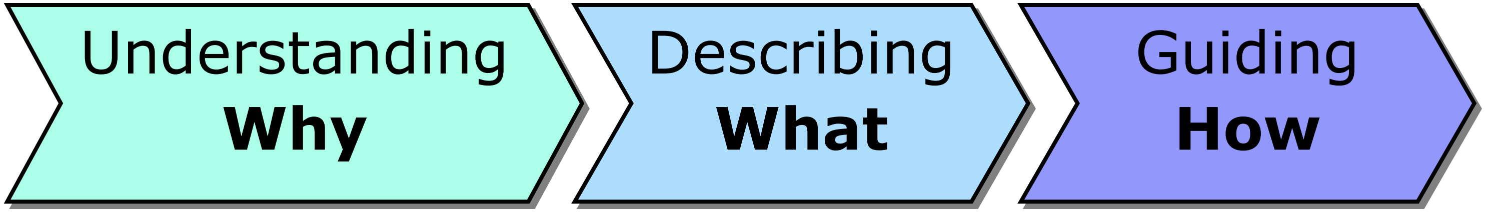
Through incorporation of foreseeable changes

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

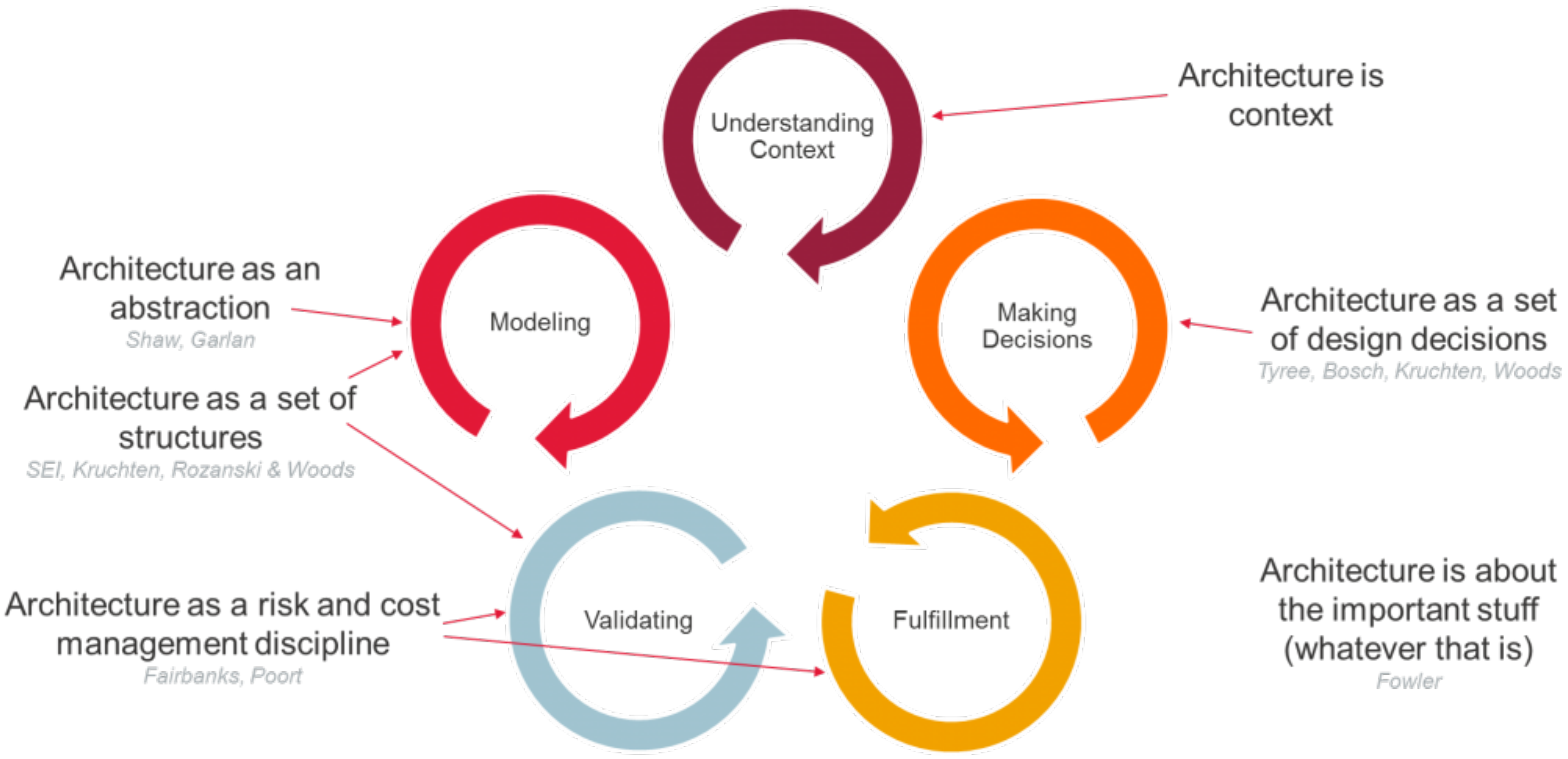
What functions does the system provide?

What properties does the design have?

How can the system be built?

Figure from Gerrit Muller, How to Create a Manageable Platform Architecture

Multiple Purposes of Architecture



Understanding Context

Modeling

Making Decisions

Validating

Fulfillment

Architecture is context

Architecture as an abstraction
Shaw, Garlan

Architecture as a set of structures
SEI, Kruchten, Rozanski & Woods

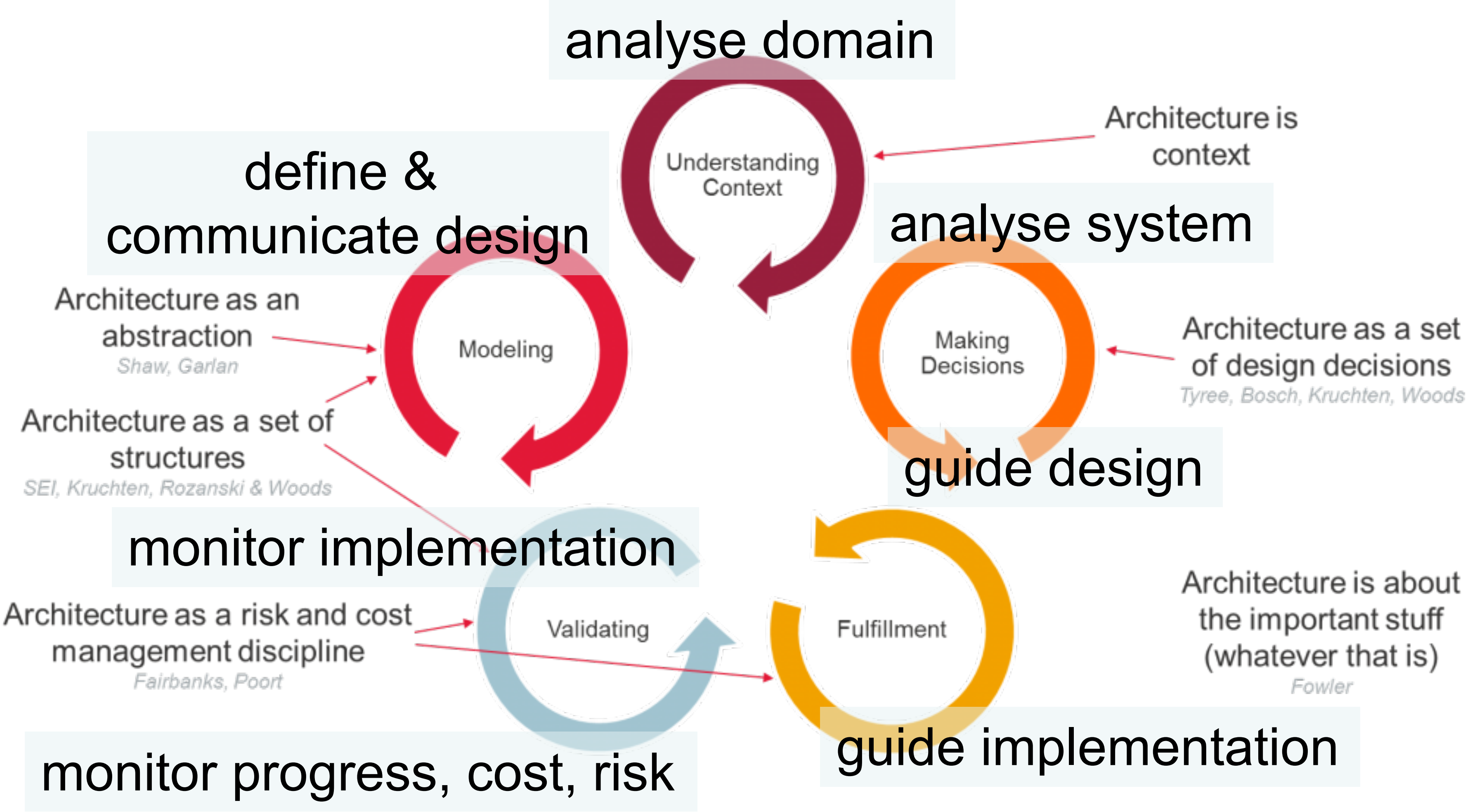
Architecture as a risk and cost management discipline
Fairbanks, Poort

Architecture as a set of design decisions
Tyree, Bosch, Kruchten, Woods

Architecture is about the important stuff (whatever that is)
Fowler

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

Multiple Purposes of Architecture



analyse domain

define & communicate design

analyse system

guide design

monitor implementation

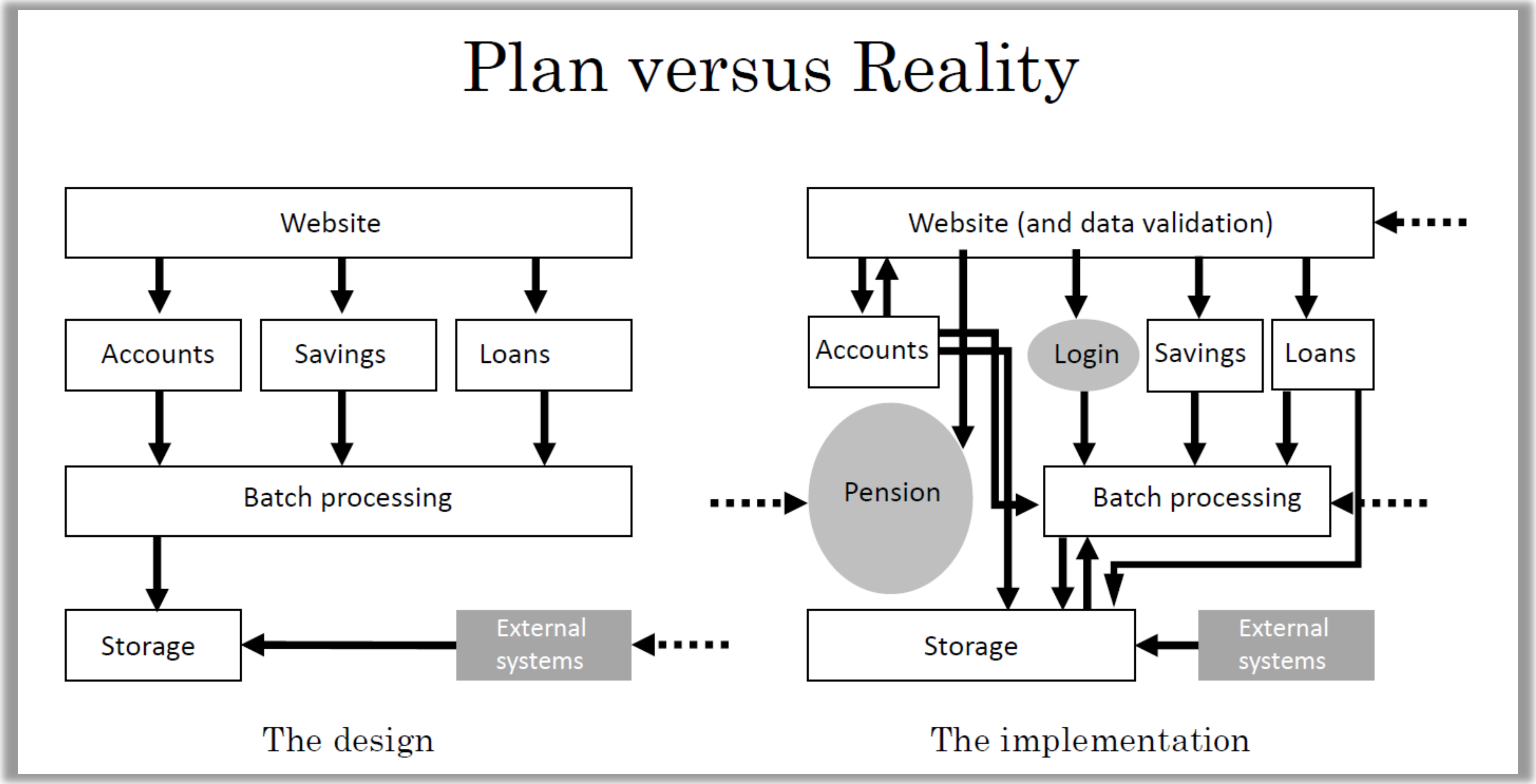
monitor progress, cost, risk

guide implementation

Understanding Context
Architecture is context
Modeling
Making Decisions
Architecture as an abstraction
Shaw, Garlan
Architecture as a set of design decisions
Tyree, Bosch, Kruchten, Woods
Architecture as a set of structures
SEI, Kruchten, Rozanski & Woods
Validating
Fulfillment
Architecture as a risk and cost management discipline
Fairbanks, Poort
Architecture is about the important stuff (whatever that is)
Fowler

Picture from Eltjo Poort's Architecture Blog (2019) - https://eltjopoort.nl/

Monitor the Implementation



Plan versus Reality

The design

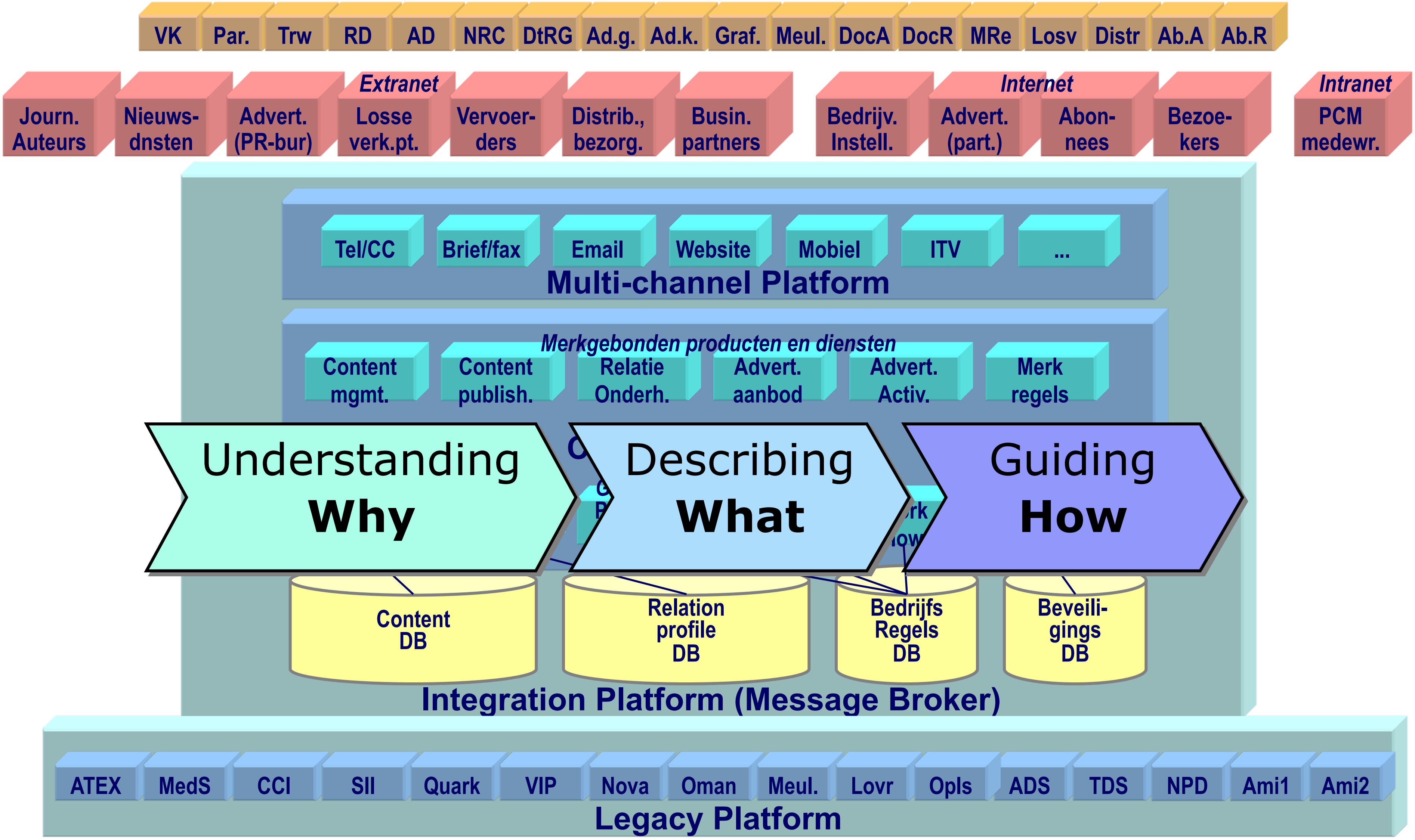
The implementation

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

Extranet

Journ. Auteurs | Nieuws-dnsten | Advert. (PR-bur) | Losse verk.pt. | Vervoer-ders | Distrib., bezorg. | Busin. partners

Internet

Bedrijv. Instell. | Advert. (part.) | Abon-nees | Bezoe-kers

Intranet

PCM medewr.

Tel/CC | Brief/fax | Email | Website | Mobiel | ITV | ...

Multi-channel Platform

Merkgebonden producten en diensten

Content mgmt. | Content publish. | Relatie Onderh. | Advert. aanbod | Advert. Activ. | Merk regels

Understanding **Why**

Describing **What**

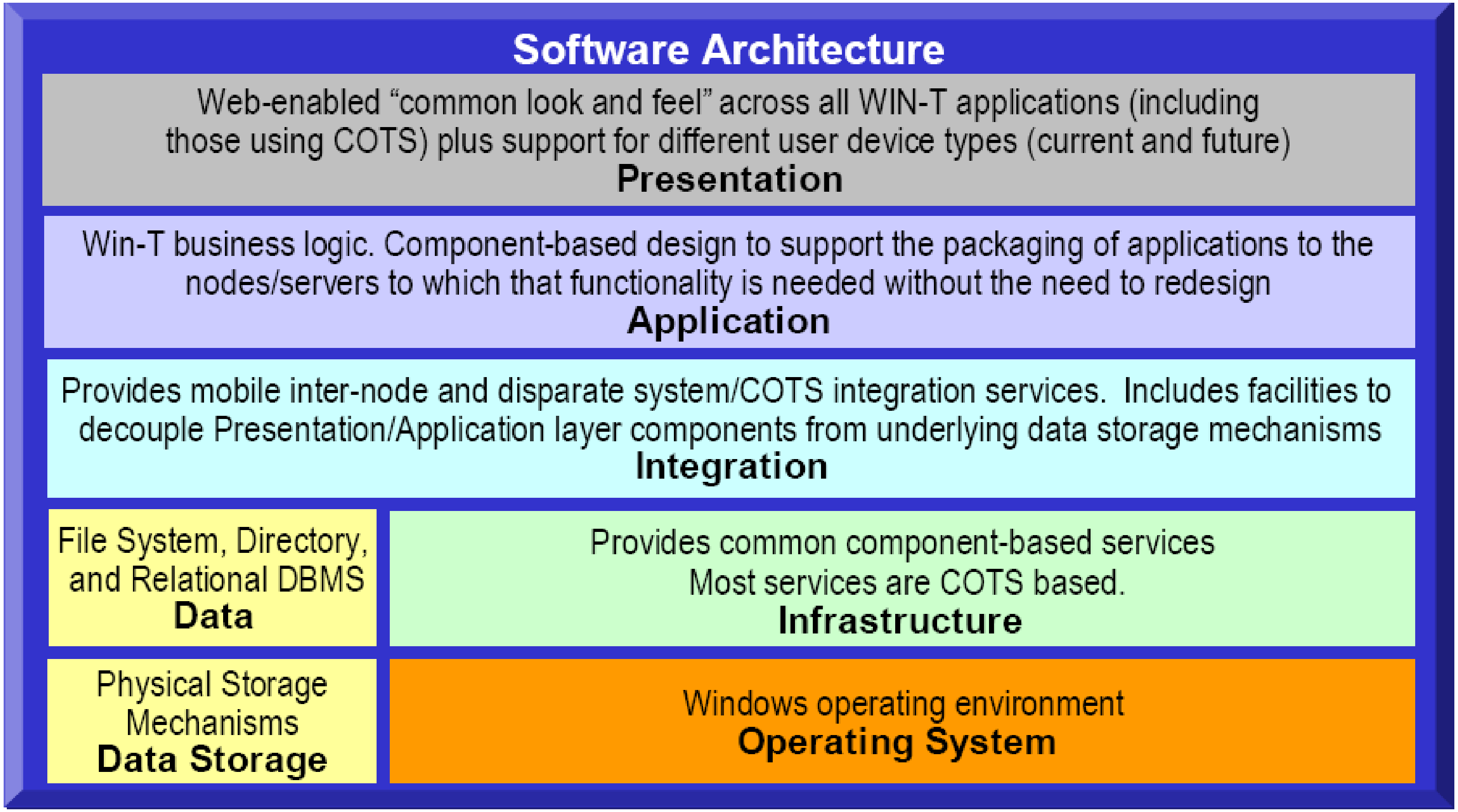
Guiding **How**

Content DB | Relation profile DB | Bedrijfs Regels DB | Beveili-gings DB

Integration Platform (Message Broker)

ATEX MedS CCI SII Quark VIP Nova Oman Meul. Lovr Opls ADS TDS NPD Ami1 Ami2

Legacy Platform



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

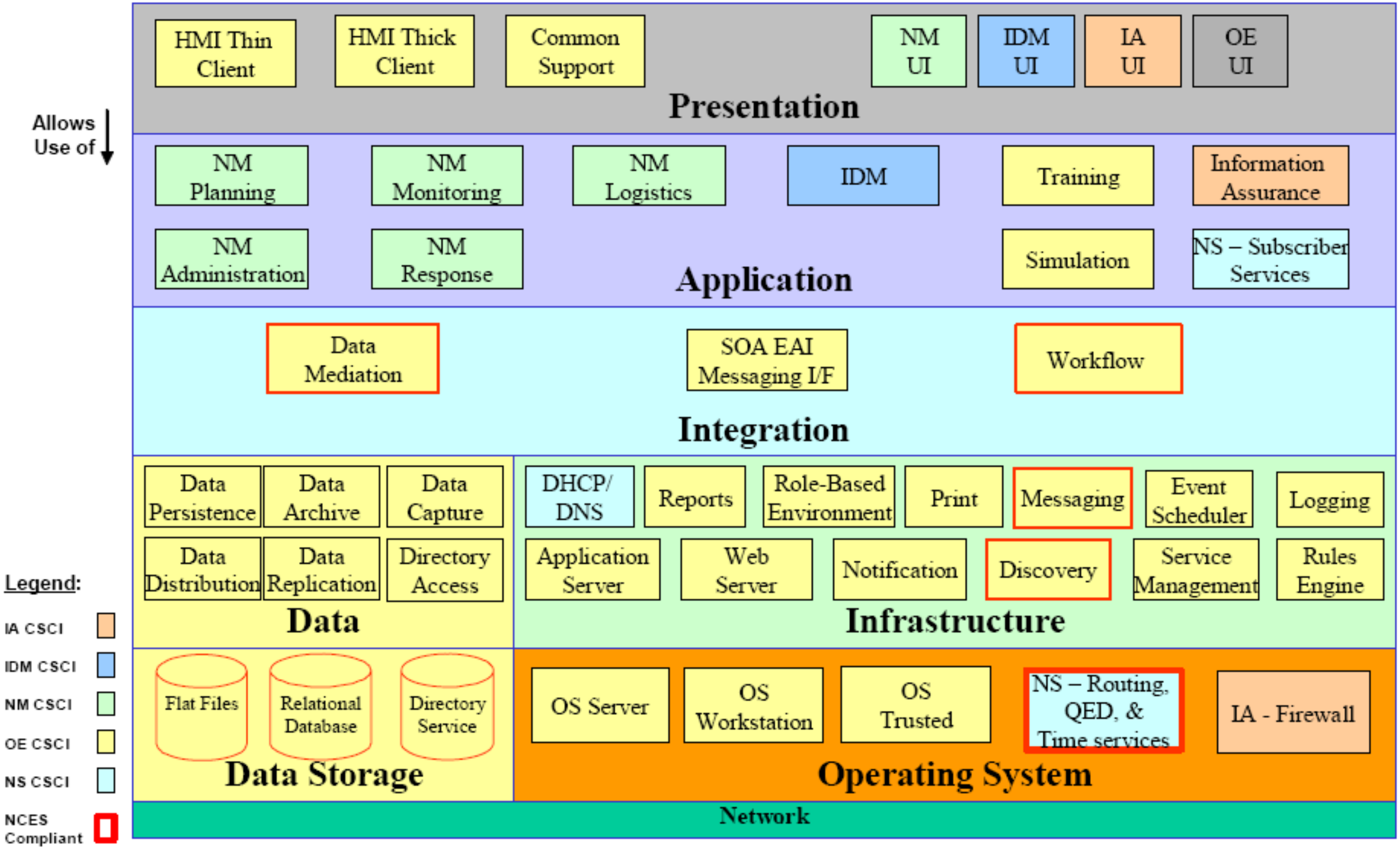
Provides common component-based services
Most services are COTS based.
Infrastructure

Physical Storage Mechanisms
Data Storage

Windows operating environment
Operating System

From: Using the SEI Architecture Tradeoff Analysis Method to Evaluate WIN-T: A Case Study
Technical Note CMU/SEI-2005-TN-027

Refinement of previous slide



Allows Use of

Presentation
- HMI Thin Client
- HMI Thick Client
- Common Support
- NM UI
- IDM UI
- IA UI
- OE UI

Application
- NM Planning
- NM Monitoring
- NM Logistics
- IDM
- Training
- Information Assurance
- NM Administration
- NM Response
- Simulation
- NS – Subscriber Services

Integration
- Data Mediation
- SOA EAI Messaging I/F
- Workflow

Data
- Data Persistence
- Data Archive
- Data Capture
- Data Distribution
- Data Replication
- Directory Access

Infrastructure
- DHCP/ DNS
- Reports
- Role-Based Environment
- Print
- Messaging
- Event Scheduler
- Logging
- Application Server
- Web Server
- Notification
- Discovery
- Service Management
- Rules Engine

Data Storage
- Flat Files
- Relational Database
- Directory Service

Operating System
- OS Server
- OS Workstation
- OS Trusted
- NS – Routing, QED, & Time services
- IA - Firewall

Network

Legend:

IA CSCI	🟧
IDM CSCI	🟦
NM CSCI	🟩
OE CSCI	🟨
NS CSCI	⬜
NCES Compliant	🟥

From: Using the SEI Architecture Tradeoff Analysis Method to Evaluate WIN-T: A Case Study
Technical Note CMU/SEI-2005-TN-027

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

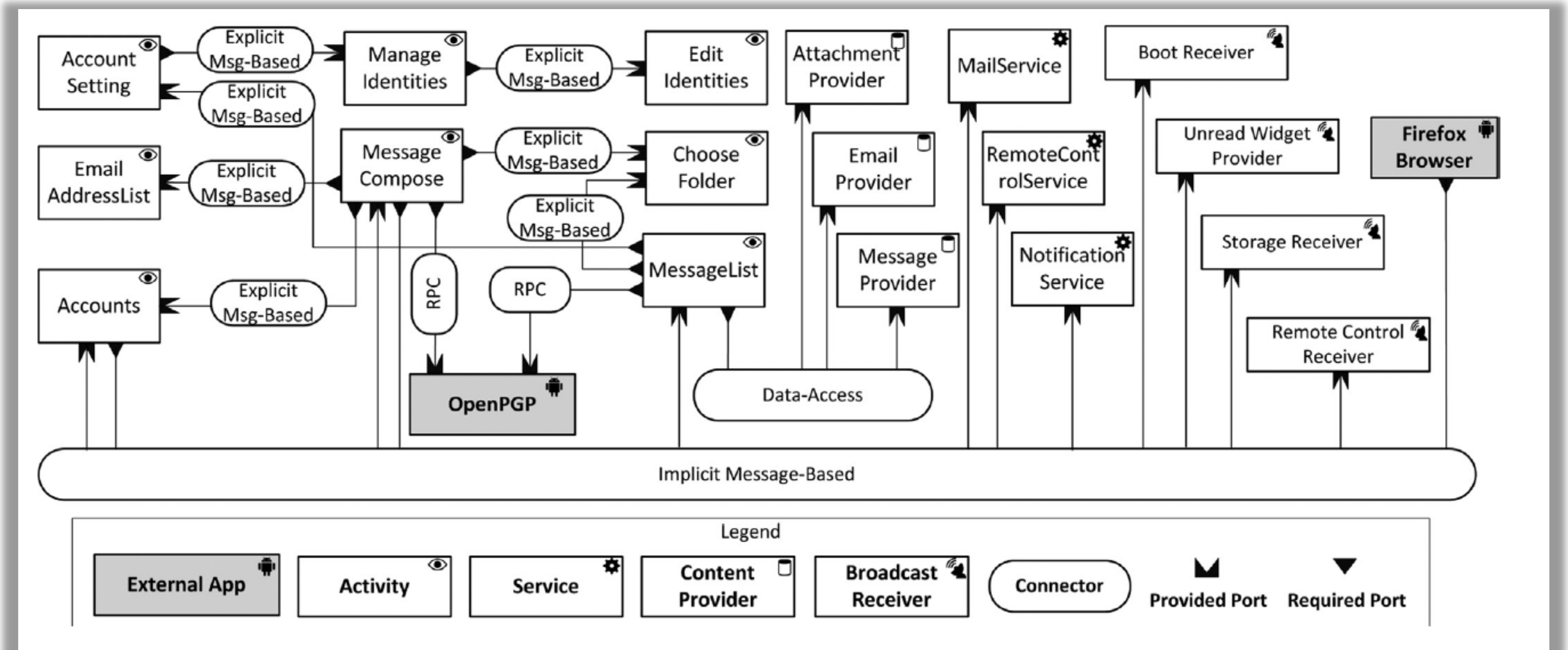
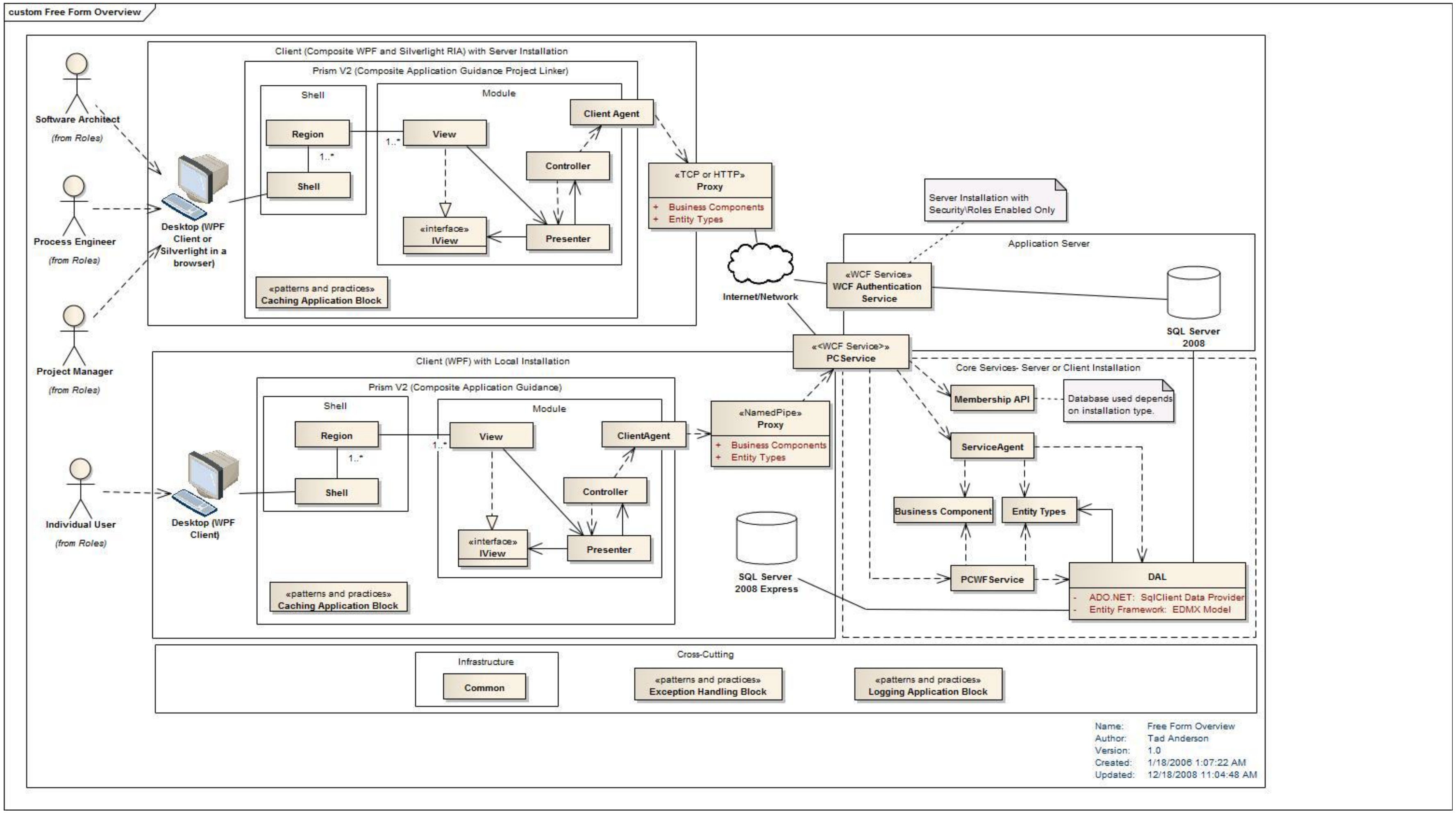


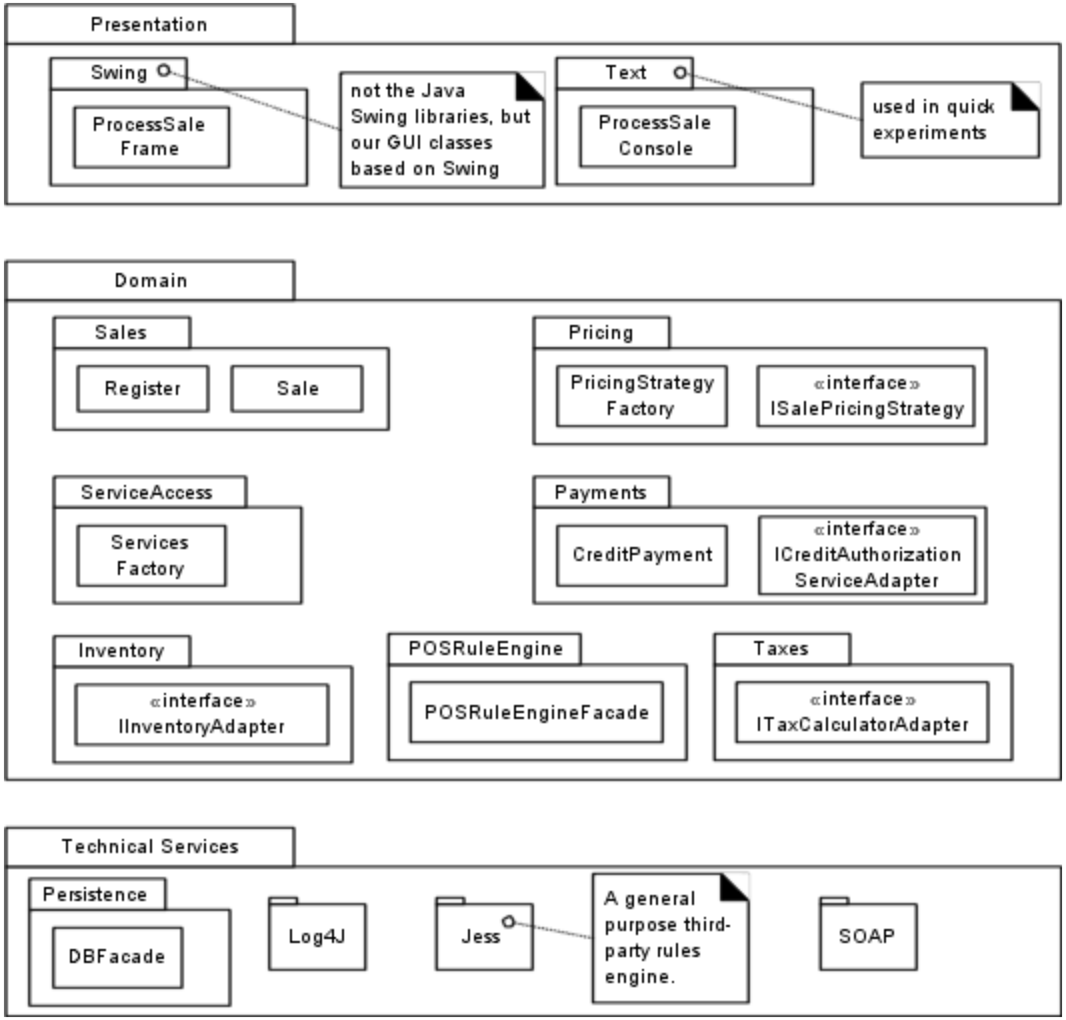
Fig. 2. K-9 mail Android app architecture.

Software architectural principles in contemporary mobile software: from conception to practice

Hamid Bagheri[a,*], Joshua Garcia[a], Alireza Sadeghi[a], Sam Malek[a], Nenad Medvidovic[b]

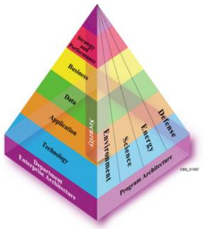


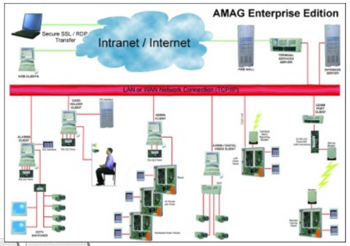
Name:	Free Form Overview
Author:	Tad Anderson
Version:	1.0
Created:	1/18/2006 1:07:22 AM
Updated:	12/18/2008 11:04:48 AM


Presentation
Swing
ProcessSale Frame
not the Java Swing libraries, but our GUI classes based on Swing
Text
ProcessSale Console
used in quick experiments
Domain
Sales
Register
Sale
Pricing
PricingStrategy Factory
«interface» ISalePricingStrategy
ServiceAccess
Services Factory
Payments
CreditPayment
«interface» ICreditAuthorization ServiceAdapter
Inventory
«interface» IInventoryAdapter
POSRuleEngine
POSRuleEngineFacade
Taxes
«interface» ITaxCalculatorAdapter
Technical Services
Persistence
DBFacade
Log4J
Jess
A general purpose third-party rules engine.
SOAP

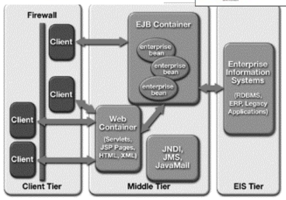
Levels of architecture*

Enterprise architecture



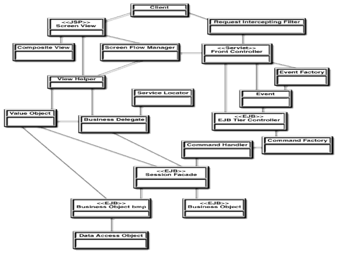

AMAG Enterprise Edition
Secure SSL / RDP Transfer
Intranet / Internet
LAN or WAN Network Connection (TCP/IP)

System architecture



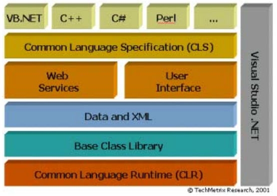
Subsystem

Application architecture



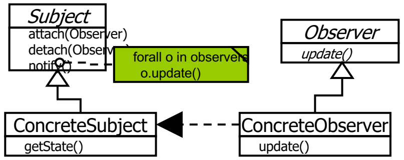
Application

Macro-architecture



Frameworks

Micro-architecture



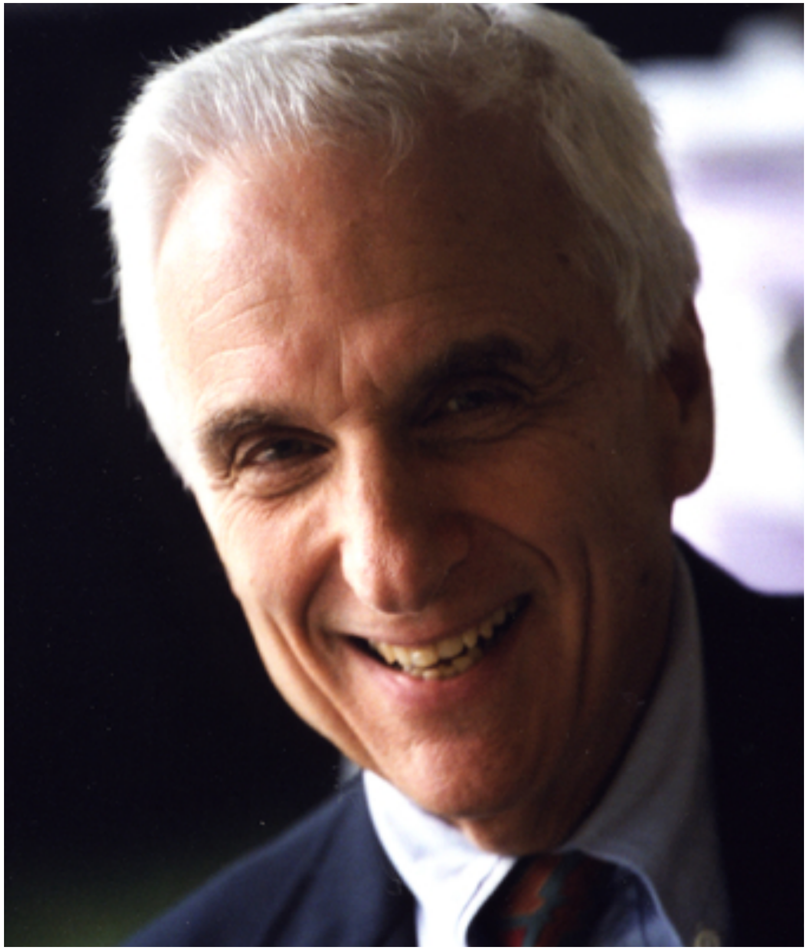
Design patterns

* 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)

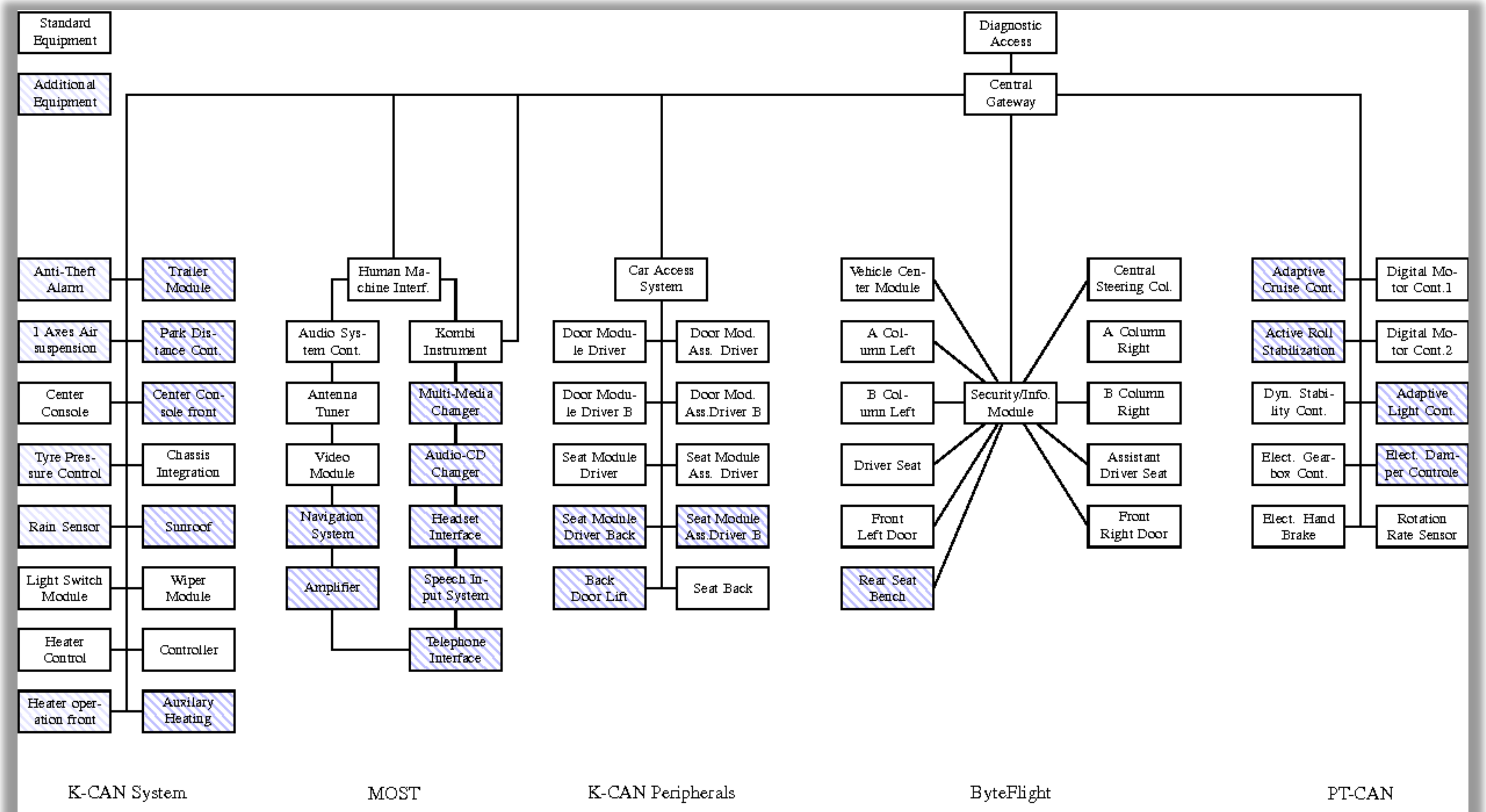
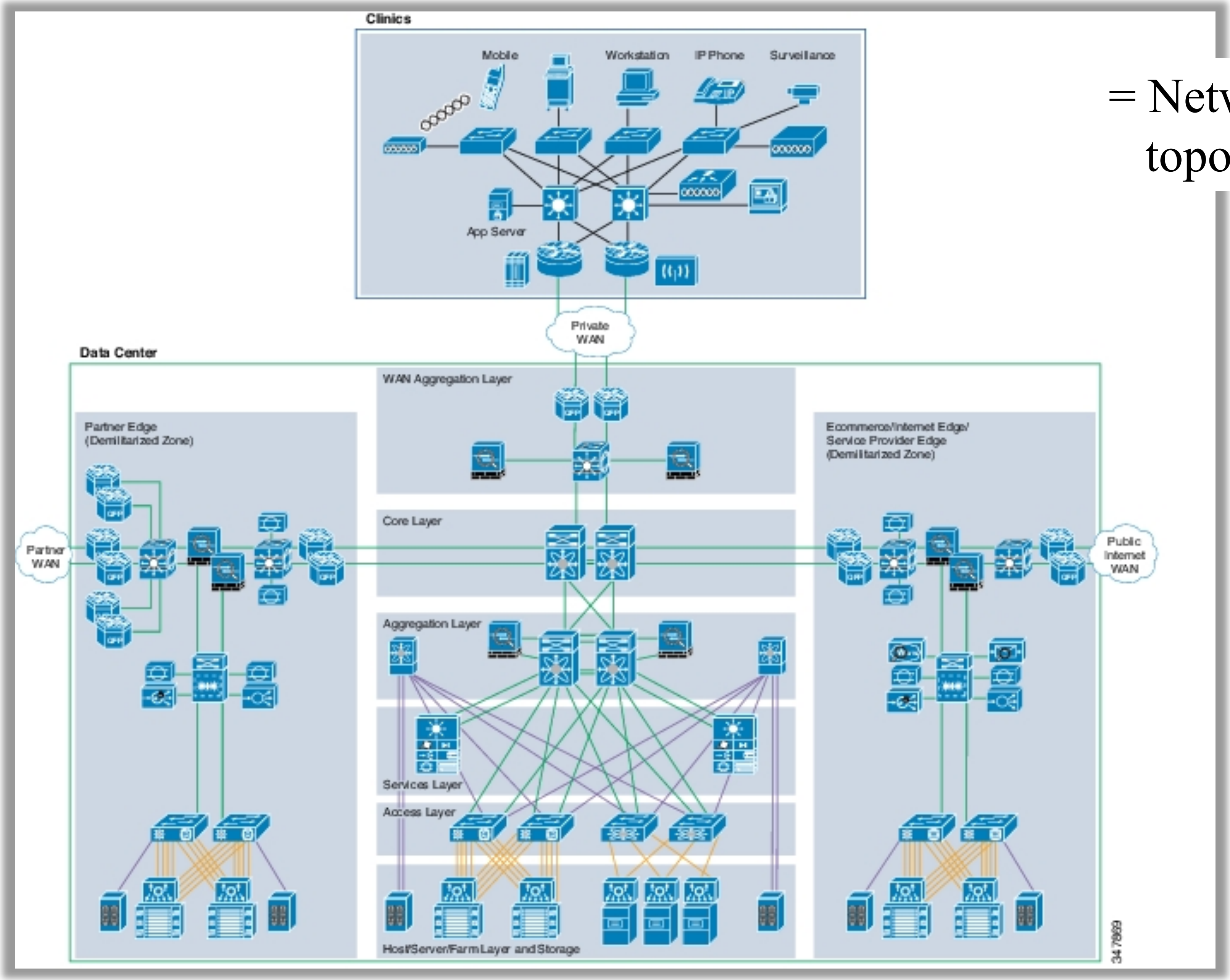


Figure 1: Decentralized ECUs of a example vehicle electrical system architecture (Barthels et al., 2012), (Michel et al., 2012)



= 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.

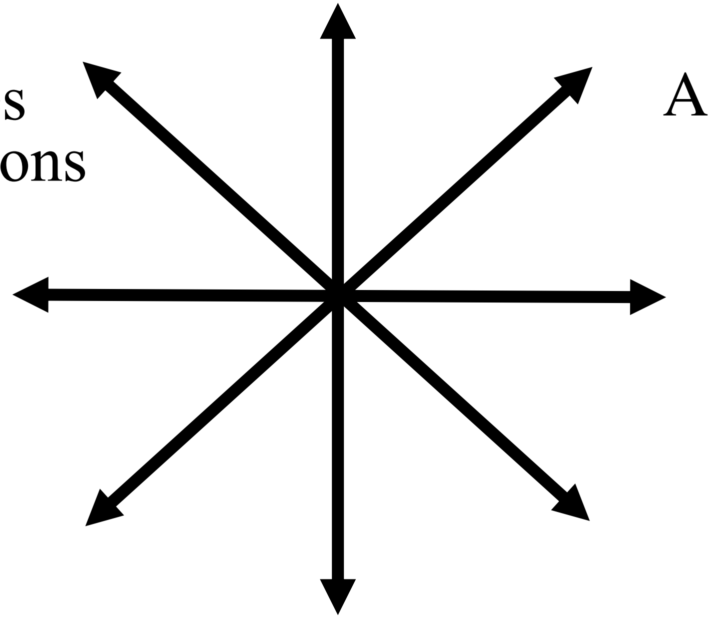
Software Architecture & Quality

- 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



Functional dimensions

Structure
Functionality
Component interfaces
Component configurations

Local/Global Behavior
Control modes
Activities/Transactions
C&S protocols

Dependability
Performance
Timeliness
Reliability
Availability
Safety
Security
Robustness

General
Scalability
Flexibility
Maintainability
Reusability
Openness
User-Friendliness
Costprize
•••

Composability
Interoperability
Reusability

Extra-Functional dimensions

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.