Dr. Hans-Peter Hoidn
*Distinguished IT Architect (Open Group certified)*

# *Enterprise IT Architectures*

# Lecture Overview & Introduction

© 2014 Hans-Peter Hoidn & Kai Schwidder

# Introduction to the Course


© 2014 Hans-Peter Hoidn & Kai Schwidder

# Aim of the course

- The course is providing an introduction to the work of an IT *Architect* in projects as it is done in the industry. The goal is to enable YOU to work in such a role and to judge and develop solution scenarios
- *Architectural Methods* are used to structure, describe, and specify architectures based on *business goals* as well as *functional* and *non-functional requirements*
- *Enterprise Architecture* as well as *Governance* concepts provide a scope beyond projects on an enterprise level
- The course introduces into current important *Technology Concepts* like *SOA* (Service Oriented Architecture) and *BPM* (Business Process Management)


© 2014 Hans-Peter Hoidn & Kai Schwidder

# Organization of the course

- Lectures with some exercises and a *Term Paper* working out the architecture of a project including a proposal (to a fictitious CEO and CIO), which will be
  - Worked out in a group
  - Presented of the group
  - Group presentations will be either on November 28 or on December 5, every team member has to present a significant part (therefore must be present at this day)

- The lecture is in BIN 2.A.10 on Fridays from September 19 to December 12 at 12:15 to 13:45 including a break of approximately 5 minutes

- December 19 is reserved for the written exam if necessary


© 2014 Hans-Peter Hoidn & Kai Schwidder

# Requirements for Obtaining the Credits

- Contribution to the project presentation and presenting a part of this work in the class room lecture; attendance for the group's presentation is mandatory and the group must pass (counts 1/3 to the final evaluation)

- The (updated) presentation must be sent to hans-peter@hoidn.ch until December 12, 23:59

- Passing the exam (counts 2/3 to the final evaluation)
  - Either written (December 19)
  - Or 20 to 30 minutes oral exam for everybody (in December or January)
  - Preference is an oral exam


© 2014 Hans-Peter Hoidn & Kai Schwidder

# Overview of the Course

- *Part 1* – Introduction, Role Play, What is IT Architecture, Key Aspects and Deliverables of Solution's IT Architecture (19.9. & 26.9. – Hans-Peter Hoidn)
- *Part 2* – Introduction to the Term Paper ("Semesterarbeit"), Content of the Term Paper (03.10.2014 – Kai Schwidder)
- *Part 3* – Information Systems Architecture and SOA (Service Oriented Architecture), Technical Architecture, Business Architecture and BPM (Business Process Management) (10.10 & 17.10. – Hans-Peter Hoidn)
- *Part 4* – Key Aspects and Deliverables of Solution's IT Architecture in more detail, Functional and Non-Functional Requirements (24.10 to 14.11. – Kai Schwidder)
- *Part 5* – EA (Enterprise Architecture) (21.11. – Hans-Peter Hoidn)
- Presentation of projects by Students (28.11 & 5.12.)
- *Part 6* – Architecture Management and Remaining Selected Topics – on student's request (12.12 – Hans-Peter Hoidn)
- Exam (19.12. if written)


© 2014 Hans-Peter Hoidn & Kai Schwidder

# Lecturers


- **Dr. Hans-Peter Hoidn**
  **retired from IBM**
  **Distinguished IT Architect (Opengroup)**
  **hans-peter@hoidn.ch**

- **Kai Schwidder**
  **on Sabbatical**
  **Distinguished IT Architect**
  **kai@schwidder.com**


- **Together about 70 years IT experience**


© 2014 Hans-Peter Hoidn & Kai Schwidder

# Hans-Peter Hoidn

- Retired from IBM, Coach for Architecture and Education, Consultant for IT Architecture and IT Strategy
- Open Group certified Distinguished IT Architect
- Executive Architect in Global Business Integration “Tiger” Team until end of September 2012
- 11 years IBM and PwC
- Previously with UBS, Digital Equipment Corp, etc.
- Architect since about 25 years
- 40+ years in IT – started programming 1971
- Dr. sc. math ETH

three daughters (39, 36, 34 years old); five grandchildren (born January 2008, August 2009, June 2010, January 2012, April 2012)

© 2014 Hans-Peter Hoidn & Kai Schwidder

# Q & A – Next Steps

- Q & A – Questions & Answers

- Next Steps
  - Let us know when you intend to attend
  - Let us know the groups working together

- Contact:
  - hans-peter@hoidn.ch
  - +41 79 500 94 33


© 2014 Hans-Peter Hoidn & Kai Schwidder

© 2014 Hans-Peter Hoidn & Kai Schwidder

# Role Play
# YOUR New Job: IT Architect


© 2014 Hans-Peter Hoidn & Kai Schwidder

# The Course simulates some aspects of “real life”

- You are now THE architect for a new project – the course is a pre-run of an architect’s work
- Beginning with an Introduction on main aspects of IT Architecture and what IT Architects do all day
- You will receive an RFP (Request for Proposal) in order to deliver a proposal for a project
- Then you will get to work with a real project using the methodology in a mentored environment
- You will present your results and provide the “term paper” (the presentation) to your “customer”


© 2014 Hans-Peter Hoidn & Kai Schwidder

# Learning the Architecture Methodology

- What is Architecture
- Key aspects of Architect's Work
- How to handle Requirements (Qualities and Constraints)
- What are the main "Work Products"
- Working through a "Case Study"


© 2014 Hans-Peter Hoidn & Kai Schwidder

# Applying the Architecture Methodology

- YOU will be the IT Architect for a new solution, which is the problem to be solved and presented – in the Term Paper (Semesterarbeit)
- YOU will address business goals and handle functional and non-functional requirements
- YOU will model the solution, investigate boundary conditions and risks
- YOU will work out the proposal (team work !) and present to the "customer" (CEO and CIO)


© 2014 Hans-Peter Hoidn & Kai Schwidder

# Technology Update

- Service Oriented Architecture (SOA)
  - Services are providing business functionality to the users
  - Connectivity through Middleware
  - SOAP and REST
  - Enterprise Service Bus (ESB)
- Business Process Management (BPM)
  - Modeling processes such that they can be automated
- And additional topics


© 2014 Hans-Peter Hoidn & Kai Schwidder

# Beyond Solution

- Enterprise Architecture overseeing multiple solutions
  - addressing the IT environment of the enterprise

- Architecture Management / Governance
  - Committees
  - Policies, Processes


© 2014 Hans-Peter Hoidn & Kai Schwidder

# In a Nutshell:
# IT Architecture



© 2014 Hans-Peter Hoidn & Kai Schwidder

# Positioning – it is for Solving Business Problems

- Addressing:
  - Customer's Pain Points, Why and What he / she wants to achieve
  - Current situation (as-is) and the areas of action
  - Constraints, Plans, Decisions already taken
  - Sketch future situation (to-be)

- Using Architecture Methodology to
  - DOING THE RIGHT THINGS RIGHT
  - Defining scope and overview
  - Capturing as-is and developing to-be architectures
  - Architecting possible solutions
  - Specifying roadmaps


© 2014 Hans-Peter Hoidn & Kai Schwidder

# IT Architecture – what is it ?

- **Architecture: "*Doing the right things right*"**
- **IT Architecture dealing with the building blocks of a solution (and in addition of the Enterprise IT Environment)**
- **Using some "work products", like Context Diagram, AOD (Architectural Overview Diagram), Service Model, …**
- **Using Reference Architectures like *SOA (Service Oriented Architecture)* implying some Architectural Decisions and making the architecture more specific**
- ***BPM (Business Process Management)* adds new IT Capabilities in order to manage Business Requirements**


© 2014 Hans-Peter Hoidn & Kai Schwidder

# IT Architecture – Definitions (I)

- Following TOGAF there are two aspects:
  - A formal description of a system, or a detailed plan of the system at component level to guide its implementation
  - The structure of components, their inter-relationships, and the principles and guide-lines governing their design and evolution over time

- Common elements of most attempts to define architecture, in the context of IT and other systems:
  - The organization of an IT-System or IT-Systems
  - Breakdown of a system into its parts
  - The relationship between the parts (static and dynamic)
  - Decisions about the design of a system that are hard to change


© 2014 Hans-Peter Hoidn & Kai Schwidder

# IT Architecture – Definitions (II)

- Rechtin, The Art of Systems Architecting
  - The structure (in terms of components, connections, and constraints) of a product, process, or element.
  - Architecture is what architects produce: The set of information that defines a system's value, cost, and risk for the purposes of the systems sponsor.

- Must address
  - Function and quality, including aesthetics for the user (client / customer)
  - Feasibility and cost for the builder


© 2014 Hans-Peter Hoidn & Kai Schwidder

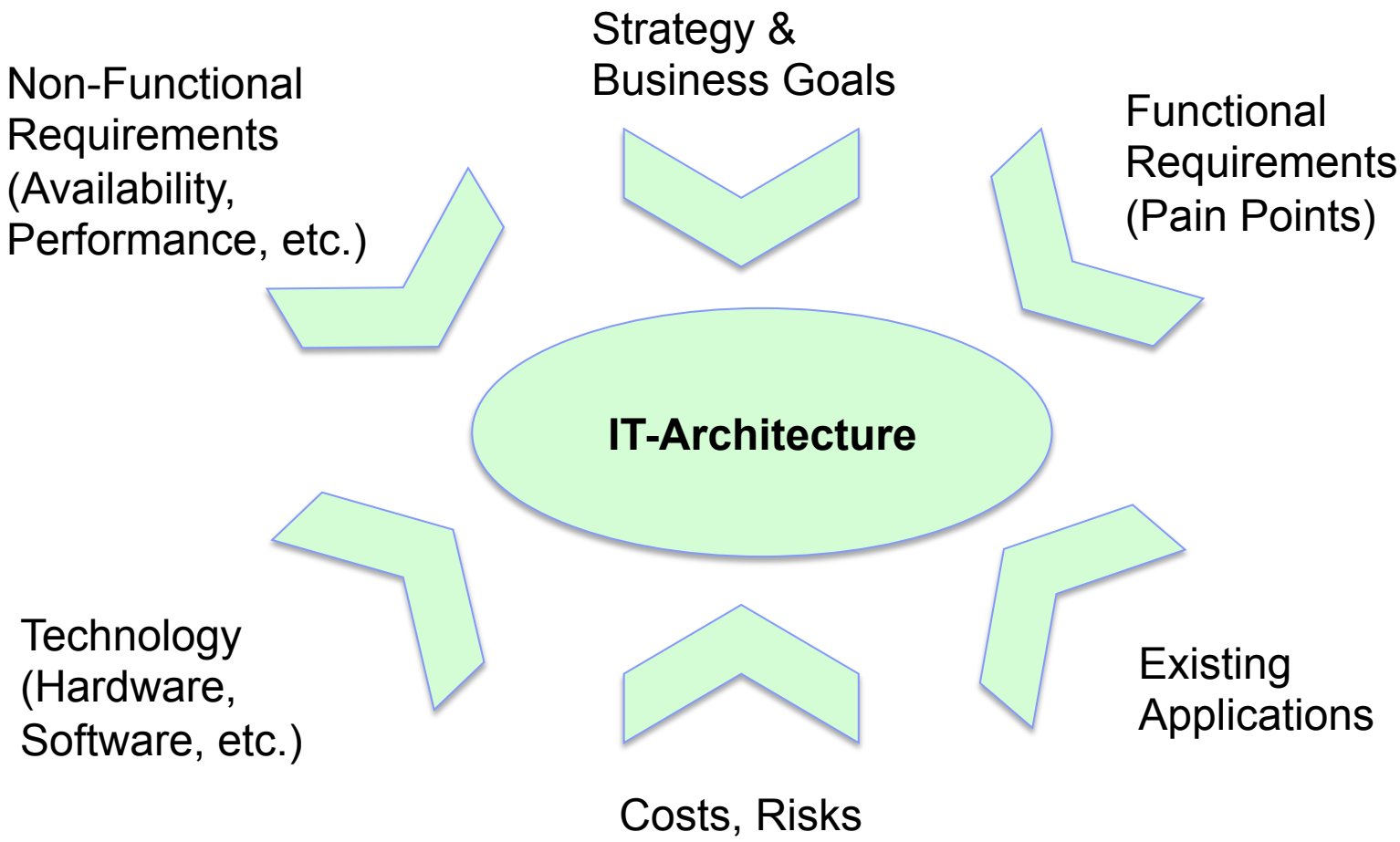
# IT Architecture – Definitions (III)

- Lankhorst, Enterprise Architecture at Work
  - Architecture is the art and science of designing complex structures, an indispensable instrument in controlling the complexity.
  - Is defined as a coherent whole of principles, methods, and models that are used in the design and realization of an enterprise's organizational structure, business processes, information systems, and infrastructure.

- The term "IT Architecture" is defined slightly differently by various authors (in addition, after 5000 years of history "architecture" has not a crisp definition
  - Thus we will use standards
    (to be introduced in Lecture 2 – next week)


© 2014 Hans-Peter Hoidn & Kai Schwidder

# *Drivers of an IT-Architecture*






© 2014 Hans-Peter Hoidn & Kai Schwidder

# Architectural Thinking at the solution architecture level leads to a complete systems architecture that serves multiple purposes.

- **It *breaks down the complexity* of the IT System.**
- **It *analyzes the required functionality* to identify required technical components.**
- **It *provides a basis for the specification* of the physical computer systems.**
- **It *defines the structuring and strategy* for the connection of system elements.**
- **It *provides the rules* of composition and decomposition of system elements.**
- **It *assists in the analysis* of service level requirements to design a means of delivery.**
- **It *provides a decision trail*, which enables the system to evolve over time.**

**It includes qualities (nonfunctional requirements).**

- Performance and Capacity
- Availability
- Manageability
- Security
- Usability
- Portability
- Reliability
- Maintainability
- Scalability
- Safety
- Extensibility

**It utilizes basic architectural principles.**

- Separation of concerns
- Information hiding
- Design by interface
- Separation of interface and implementation
- Partitioning and distributing responsibilities



© 2014 Hans-Peter Hoidn & Kai Schwidder

# IT Architecture – Description

- A collection of Work Products to document an architecture
- Addressed to one or more Stakeholders to answer their Concerns about the system
- Organized into one or more Views of the system
- Each View addresses one or more Concerns of the Stakeholders
- A View is a way of looking at an architecture
- A View is what you see when you look at the architecture from a particular Viewpoint

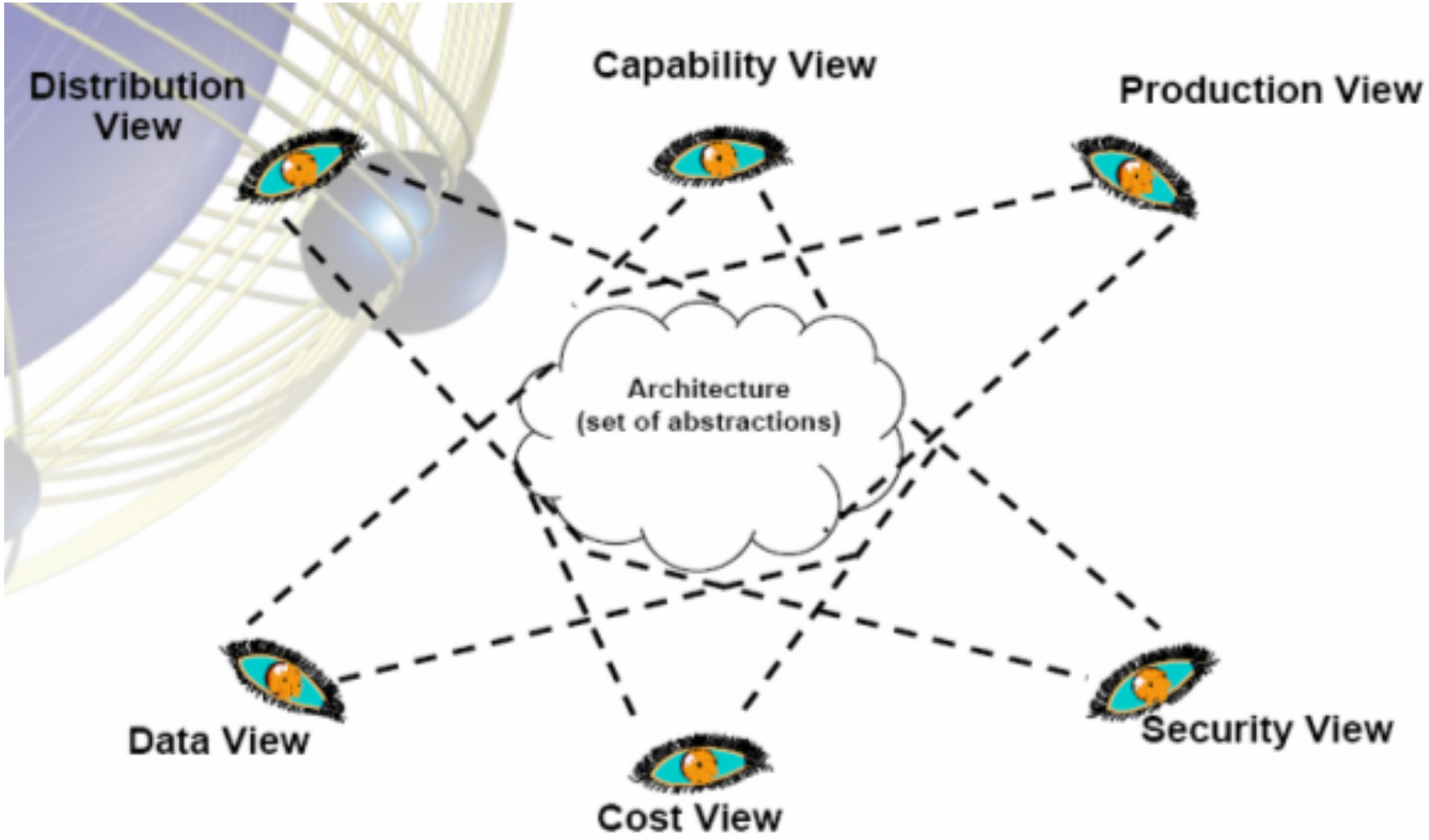

© 2014 Hans-Peter Hoidn & Kai Schwidder

# IT Architecture – Architectural Views

- Enable the architecture to be communicated to, and understood by, all the stakeholders
- Enable stakeholders to verify that the system will address their concerns
- Examples
  - Scope description: Planner's view
  - Model of the business: Owner's view
  - Information system model: Designer's view
  - Technology model: Builder's view
  - Detailed blueprints: Subcontractor's view


© 2014 Hans-Peter Hoidn & Kai Schwidder

# IT Architecture – Sample Views

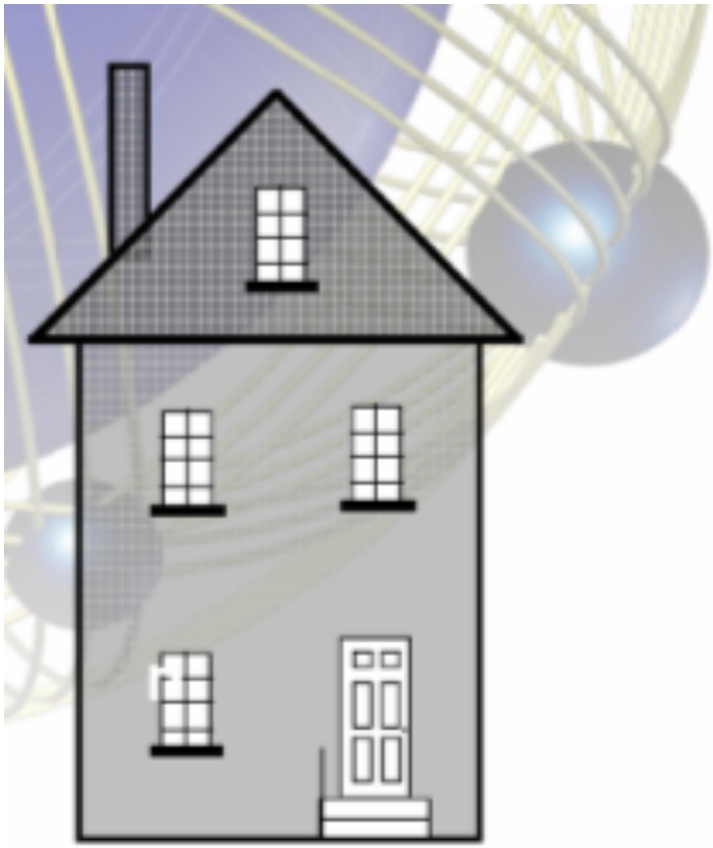


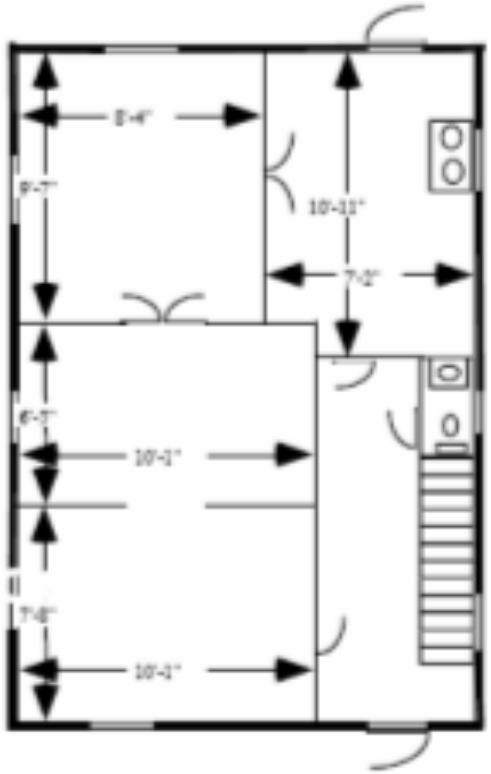

Source: Ira S. Sachs, Recommended Practice for Architectural Description, IEEE Standard P1471



© 2014 Hans-Peter Hoidn & Kai Schwidder

# *Analogy*: Multiple views and models





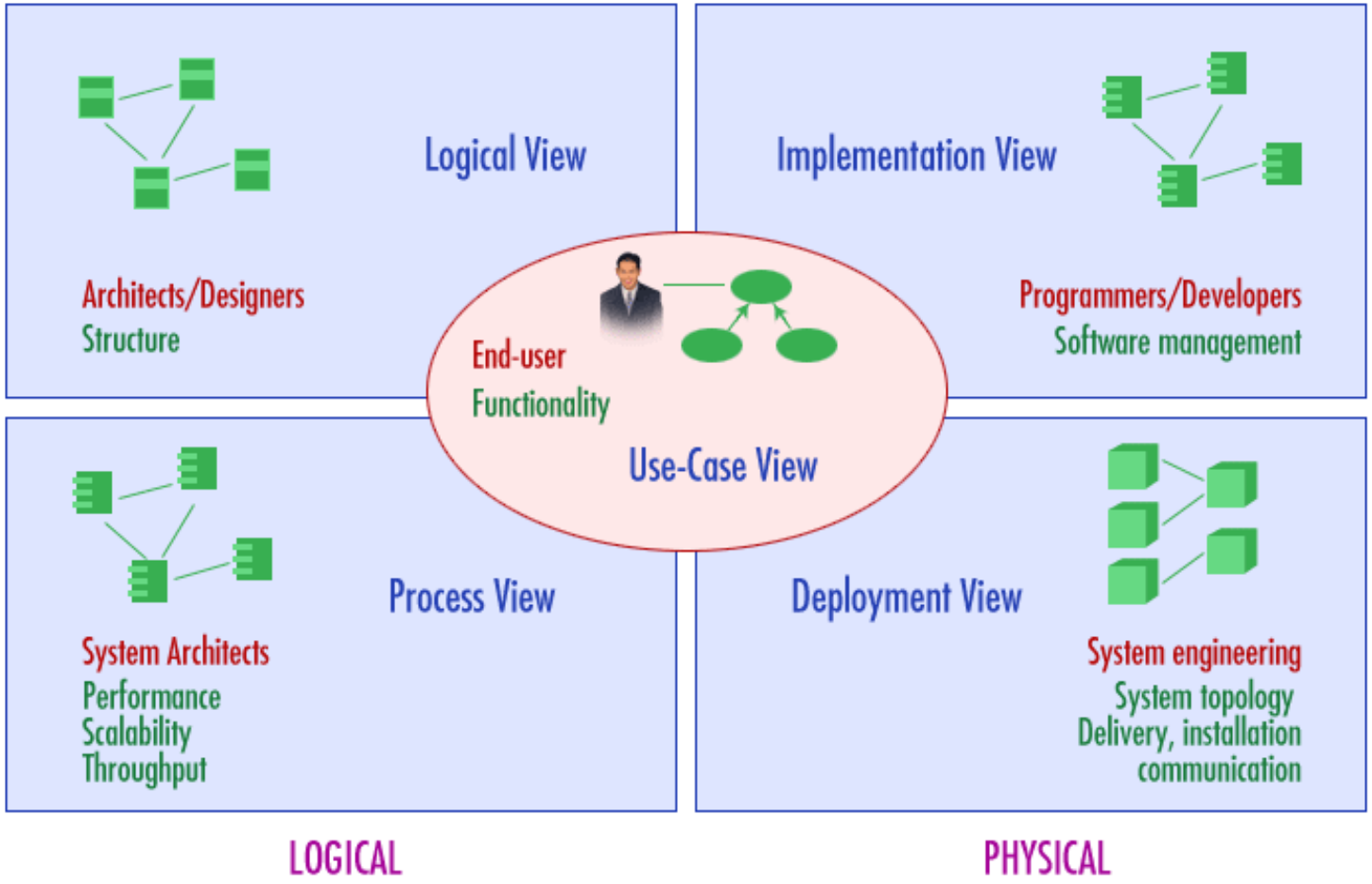
- Bill of Materials
- 2x4x8 250
- 2x6x8 150
- 4x4x10 10
- Siding 1500 sq ft
- Shingles 500 sq ft
- 8d nails 20 lb
- 6d nails 10 lb

Source: Ira S. Sachs, Recommended Practice for Architectural Description, IEEE Standard P1471



© 2014 Hans-Peter Hoidn & Kai Schwidder

# The 4+1 view model of software architecture (Kruchten)

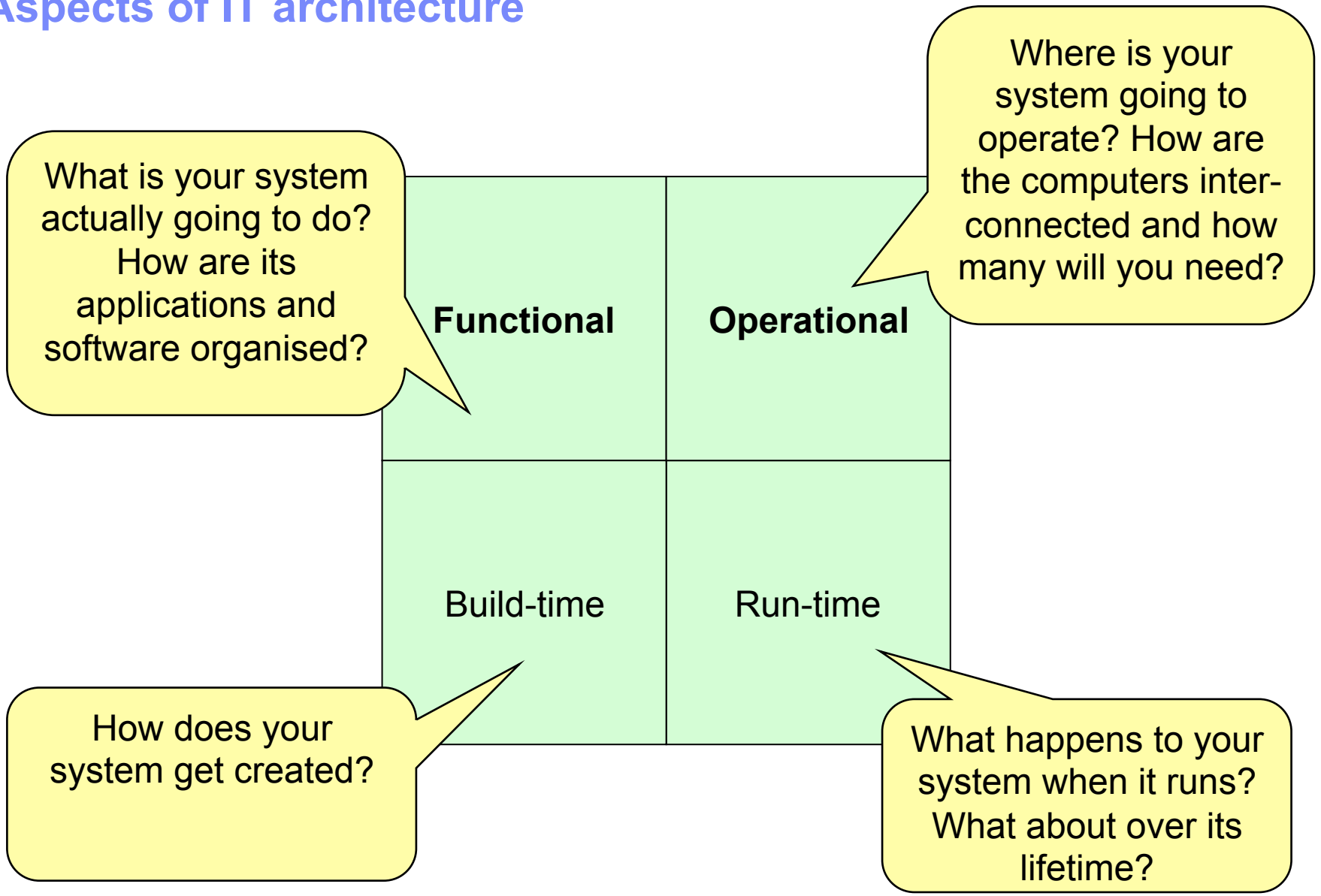





Source: Kruchten 1995

© 2014 Hans-Peter Hoidn & Kai Schwidder

# Aspects of IT architecture

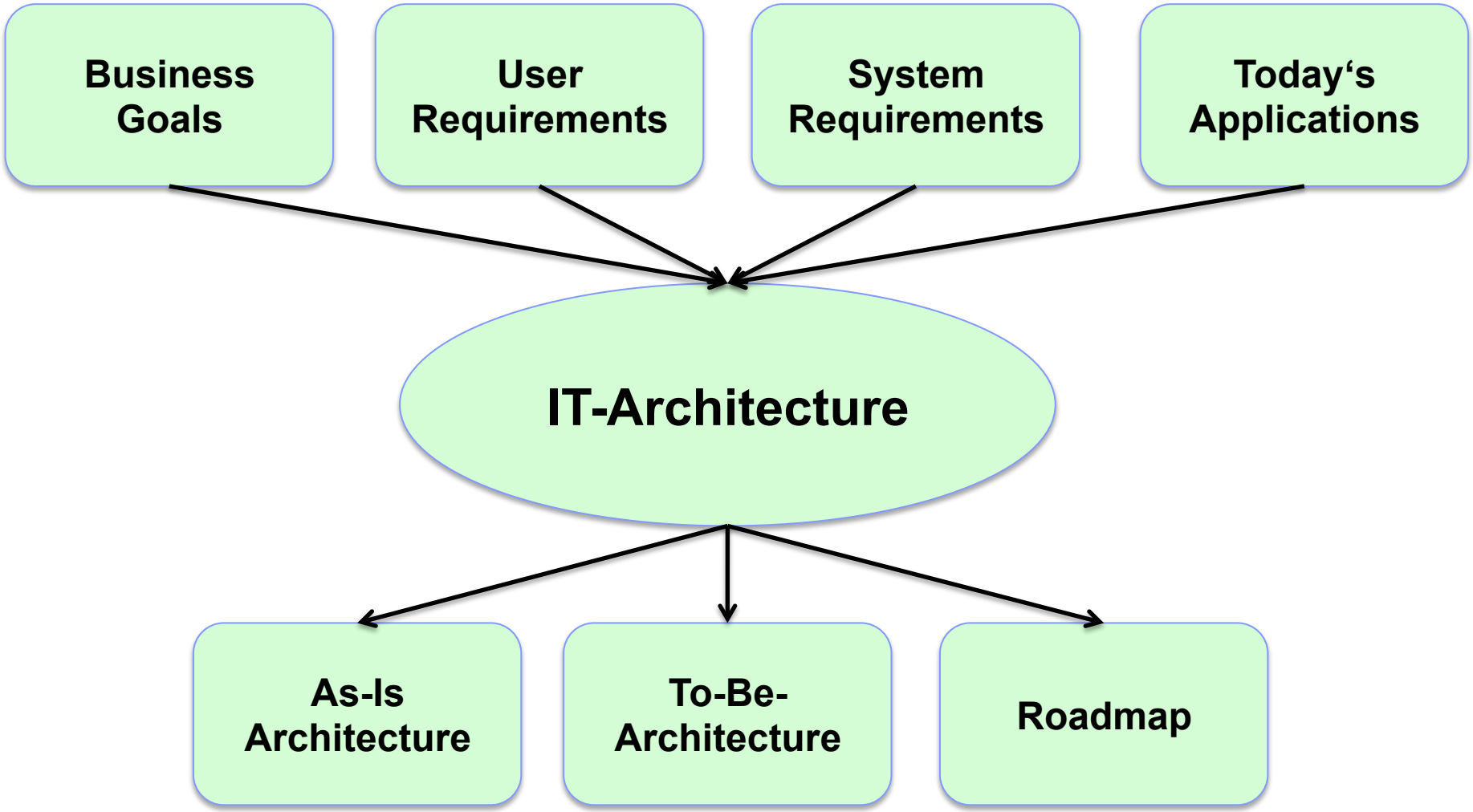



Source: IBM



© 2014 Hans-Peter Hoidn & Kai Schwidder

# Architecture Overview






© 2014 Hans-Peter Hoidn & Kai Schwidder

# In a Nutshell: IT Architect Roles

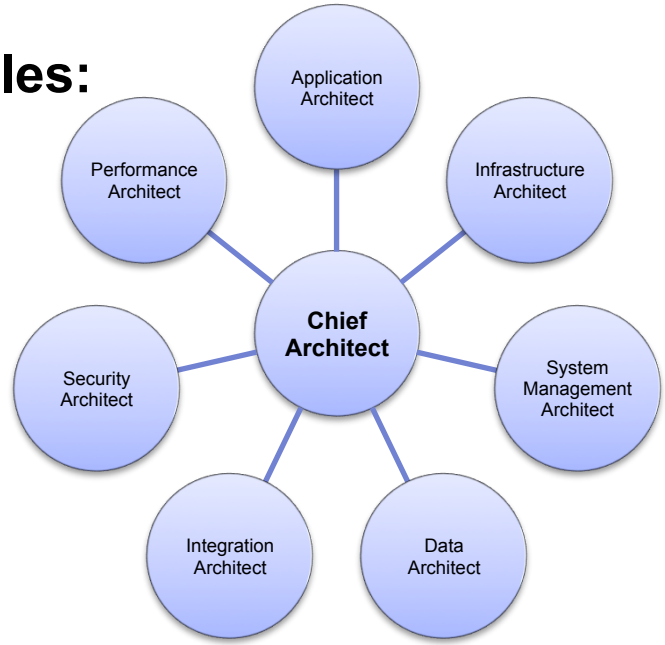


© 2014 Hans-Peter Hoidn & Kai Schwidder

# Architects – Overview

- IT Architects
  - are technically competent system-level thinkers, guiding planned and economically efficient design processes to bring a system into existence
  - Focus on system- and subsystem-level issues to establish a solid foundation for detailed design, particularly for large-scale efforts

- There are different roles:



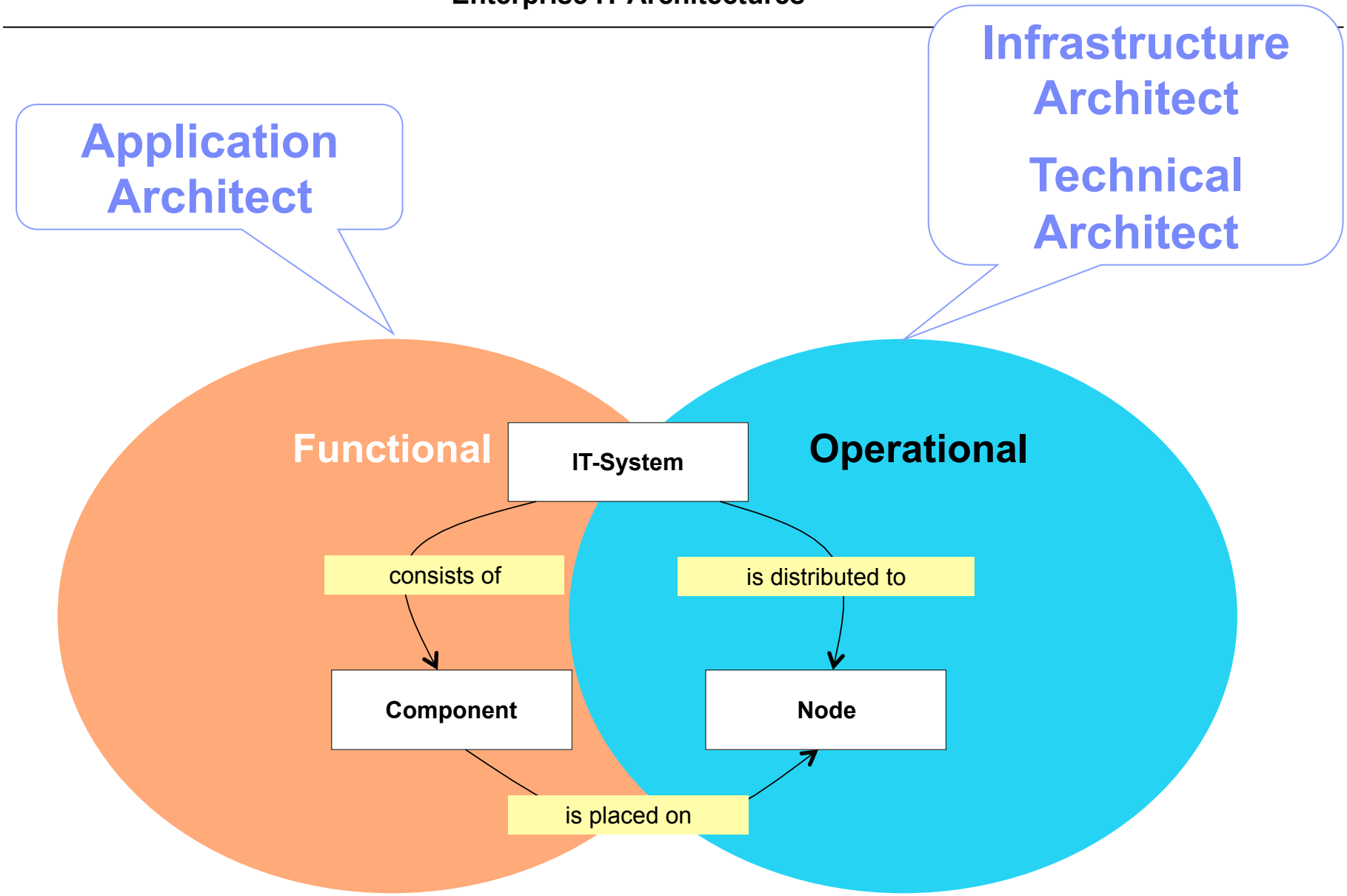



© 2014 Hans-Peter Hoidn & Kai Schwidder

# Chief Architect Roles and Responsibilities

- **Provide the technical leadership necessary to implement or achieve a business strategy through an IT solution**
- **Carry end-to-end technical solution responsibility**
- **Carry the whole scope of the problem to be solved, and the solution in his/her head**
- **Technical management of Requirements, Issues, Risks & Changes**
- **Definition of applicable Architectural Principles**
- **Manage reviews**
  - **Work products and deliverables**
  - **Co-ordinating external reviewers, Quality Assurance**
- **Internal: Advise the program manager and project executive on all aspects of the technical solution**
- **External: Develop relationships with client technical executives**


© 2014 Hans-Peter Hoidn & Kai Schwidder






Source: IBM

© 2014 Hans-Peter Hoidn & Kai Schwidder

# Application Architect

- **Defines** what the solution does
- **Responsible for the** Functional Aspects **of the system**
- **Key responsibilities**
  - **Understands how the business requirements can be met using application software, and defines what** application software packages **and / or** bespoke code **is needed**
  - **Develops and maintains** application architectures **and strategies and to ensure the design integrity of the application subsystem and that it meets the agreed requirements**
  - **Defines** high level data flows **between applications**
  - **Leads any** bespoke application development
  - **Leads the** configuration of the application software


© 2014 Hans-Peter Hoidn & Kai Schwidder

# The **Application Architect** is responsible for the Functional Aspects, which include these key concepts:

- Component
  - **Modular unit of functionality which makes this functionality available through an interface**
- Subsystem
  - **Any grouping of components in IT system**
- Interaction and Collaboration
  - **Collaboration between components**
  - **Sequence of component operations**
  - **Exchanges between two components**
  - **Interface usage contract / protocol**

**Link between Use Cases, and Components**

**Use Case Realizations**

- Data



© 2014 Hans-Peter Hoidn & Kai Schwidder

# Infrastructure (or Technical) Architect

- **Defines the overall system shape**
  - **What the building blocks are from which the solution will be made**
  - **How the data and functionality will be placed**

- **Responsible for the Operational Aspects of the system**

- **Key responsibilities**
  - **Establishes non-functional and technical infrastructure requirements**
  - **Defines the infrastructure solution**

    **Networking, hardware configurations, system software, middleware**
  - **Performance, Capacity, Scalability**
  - **Availability, Recoverability**
  - **Systems Management, Service Levels**

**Non-Functional Requirements**



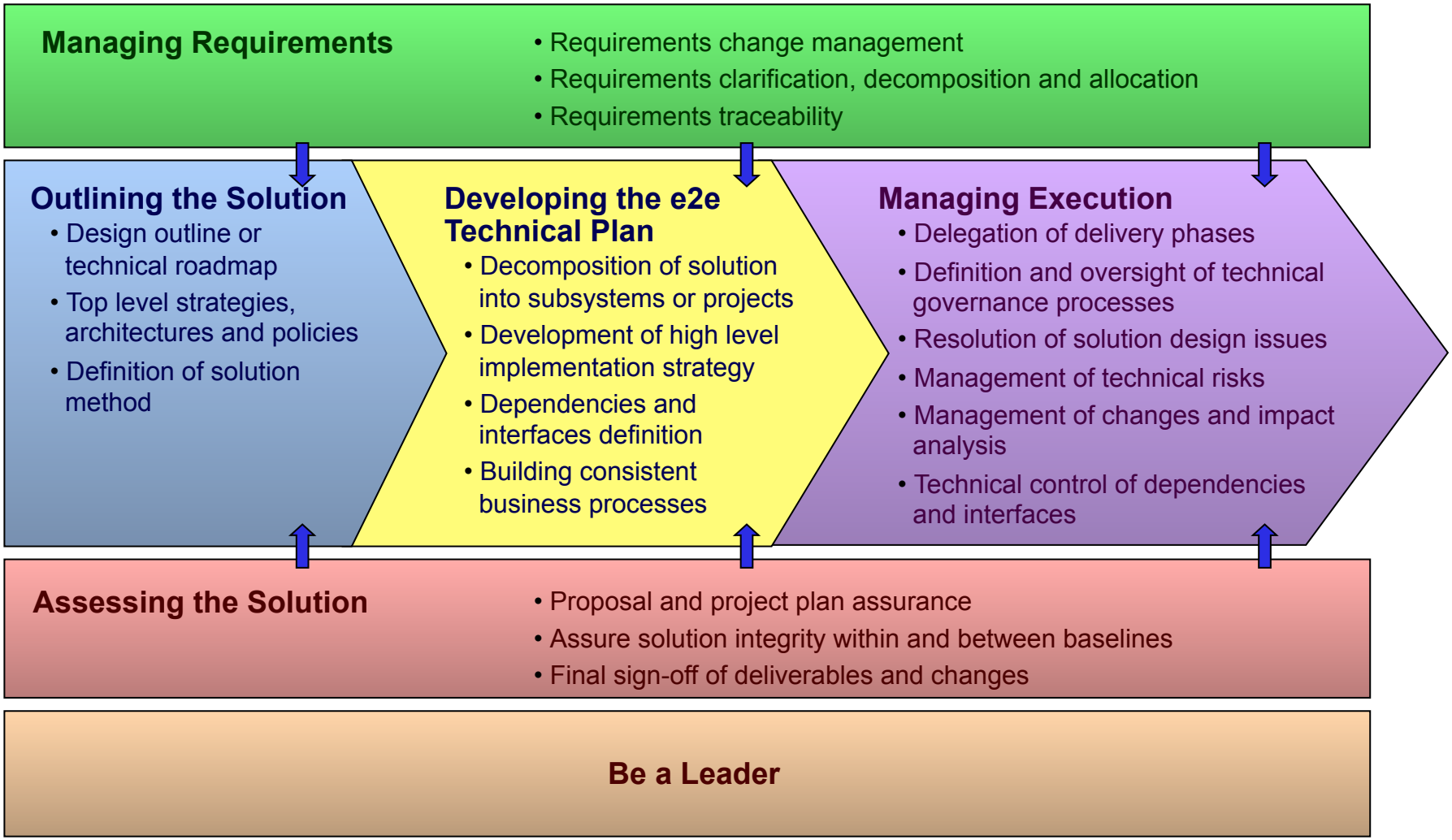
© 2014 Hans-Peter Hoidn & Kai Schwidder

# The Infrastructure Architect is responsible for the Operational Aspects, which include these key concepts:

- Node
  - **platform on which software executes**
- Location
  - **type of geographical area or position**
- Zone
  - **an area for which a common set of non-functional requirements can be defined**
- Connection
  - **physical data path between nodes (LAN, WAN, dial-up etc)**
- Deployment Unit
  - **one or more components placed together on a node**
- Non-functional Requirements (NFRs)
  - **Service Level Requirement (SLR) like performance, availability, etc.**
  - **Constraints: business / geography, IT Standards, current Infrastructure, etc.**
- Walkthrough
  - **description of the flow of a scenario starting from a user all the way through the system and back to the user**


© 2014 Hans-Peter Hoidn & Kai Schwidder

# Architect's Responsibilities across the full life-cycle






© 2014 Hans-Peter Hoidn & Kai Schwidder

# “Architect” as a Profession

- Certification program from Open Group
  - Applying for a certification
  - Providing a package describing your architect work
  - Will be reviewed and possibly approved

- HPH’s Personal Experiences
  - Certified @ IBM as “Senior IT Architect” – once re-certified
  - Senior Certification @ IBM as “Executive Architect”
  - Re-Certification @ Open Group as “Distinguished IT Architect” (April 2013)


© 2014 Hans-Peter Hoidn & Kai Schwidder

# Example of an IT Architect's Work



© 2014 Hans-Peter Hoidn & Kai Schwidder

# *"Car Leasing" of a Bank*: Situation and Customer's Goals

- The "Car Leasing" process is very, very slow and the bank is loosing a lot of the business because clients are moving to the competition
- Major Goal for the IT Architect is the definition of a Pilot Project for better support of "Car Leasing"
- Characteristics: The work has to be done within one week, from the customer side Business People as well as IT people were involved.
- Major Results grouped in
  - Findings / Recommendations,
  - Pilot Project High level scope,
  - Next Steps Phase 1 and 2
  - Phase 1 Project Plan


© 2014 Hans-Peter Hoidn & Kai Schwidder

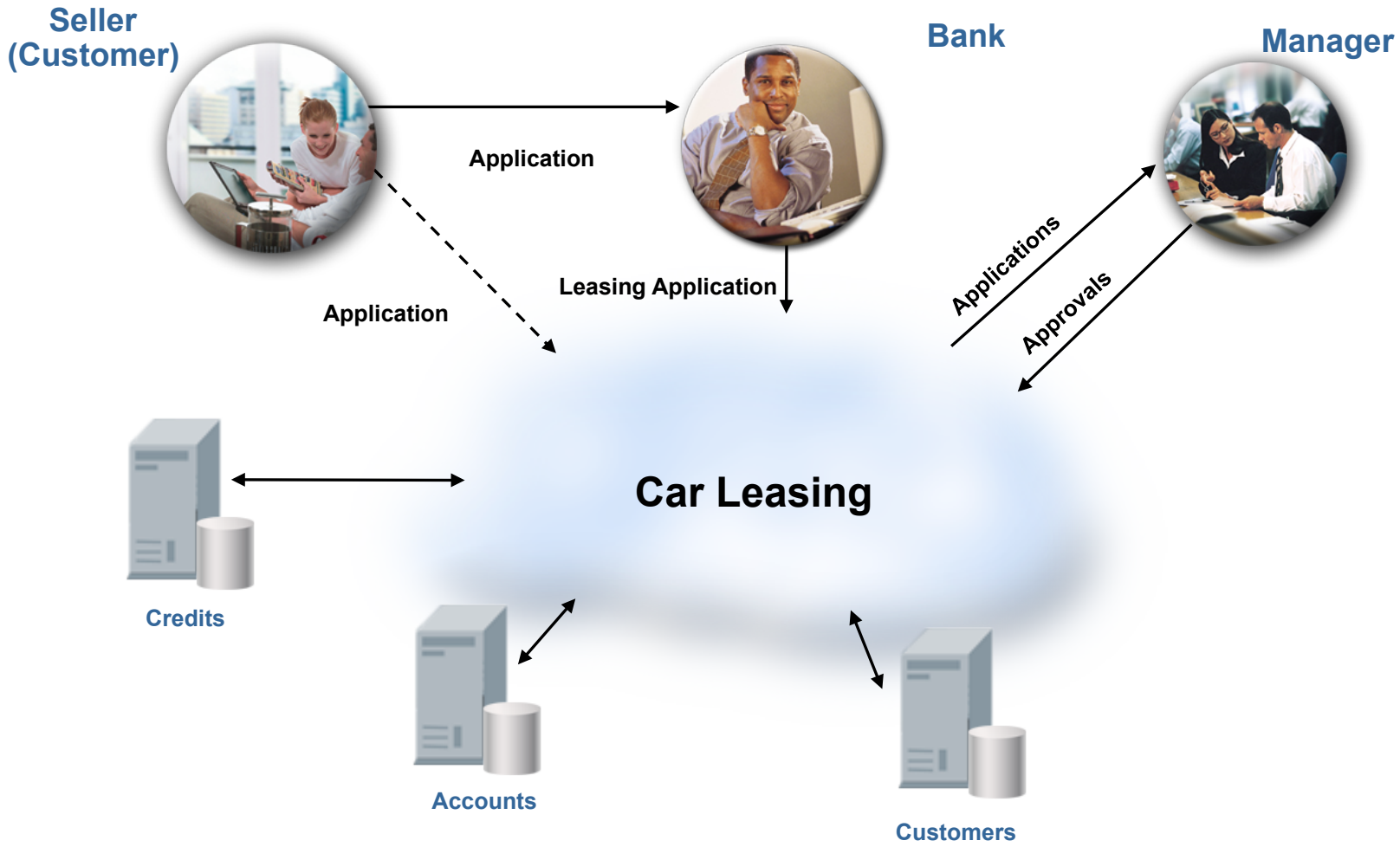
# *Example – an IT Architect at Work*: Workshop Overview

| Day | Agenda | Participants |
|---|---|---|
| Day 1 | Introductions, Met LoB, which included Corporate, Retail & Risk Mgmt | Consultant / Business from Customer |
| Day 2 | Met IT folks and discussed our findings, agreed on Pilot Project. | Consultant / IT from Customer |
| Day 3 | Technical Deep Dive including pilot arch overview | Consultant / IT from Customer |
| Day 4 | Develop initial recommendations | Consultant only – need access to participants |
| Day 5 | Presentation draft of Findings & Recommendations | All Workshop Participants and Key Executive |



© 2014 Hans-Peter Hoidn & Kai Schwidder

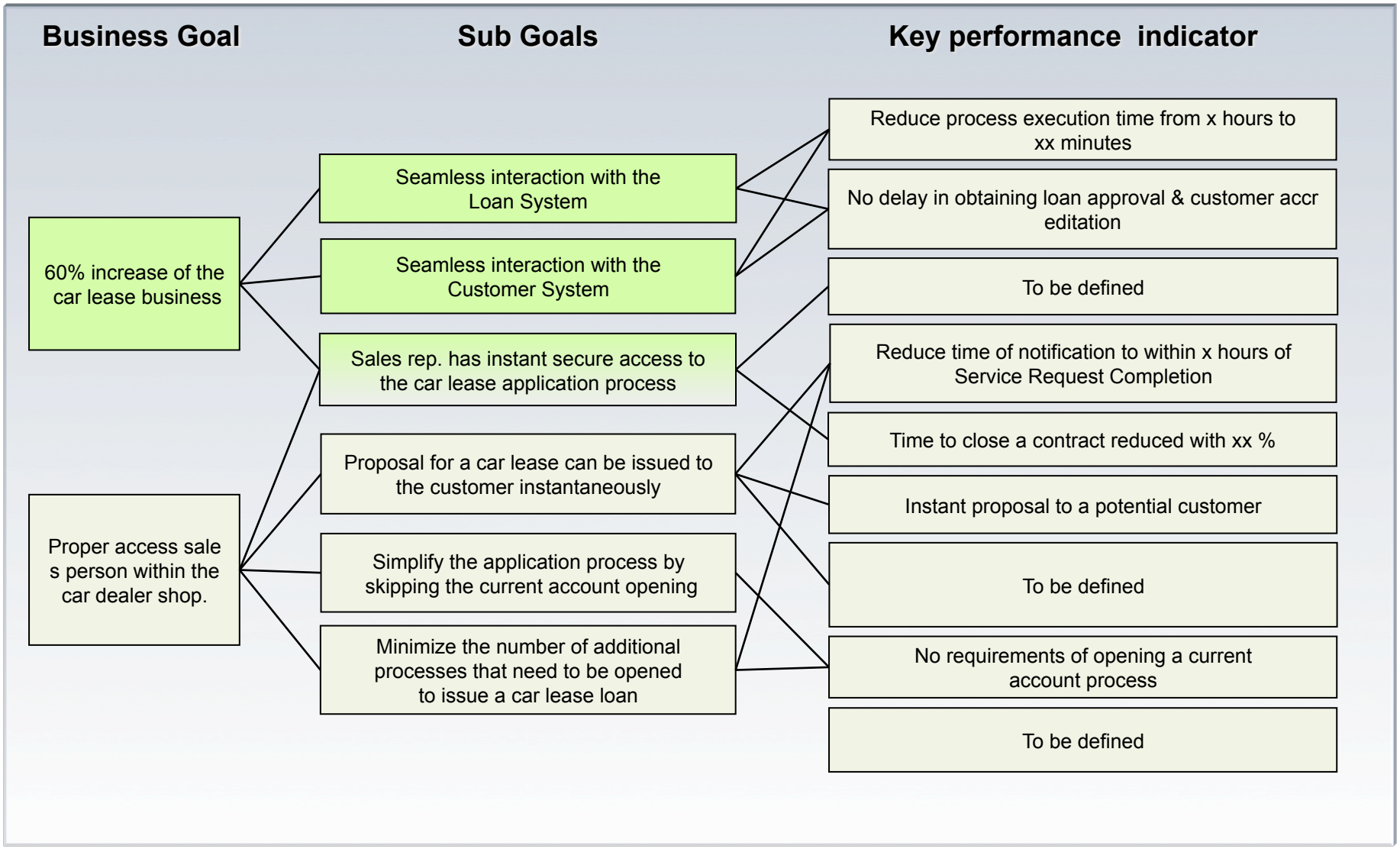
# *Show Case*: Car Leasing






© 2014 Hans-Peter Hoidn & Kai Schwidder

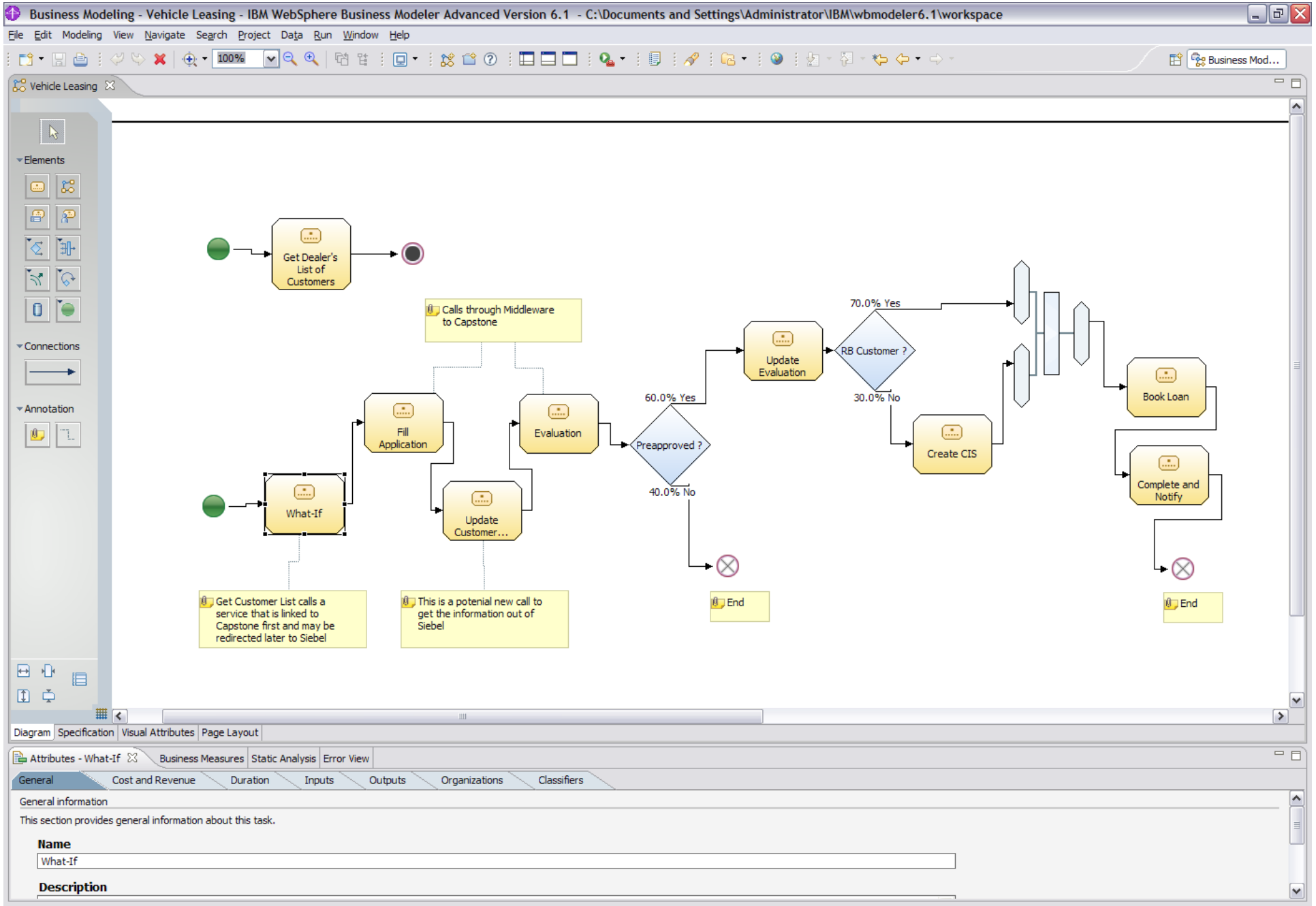
# *Example*: Capturing Business Side



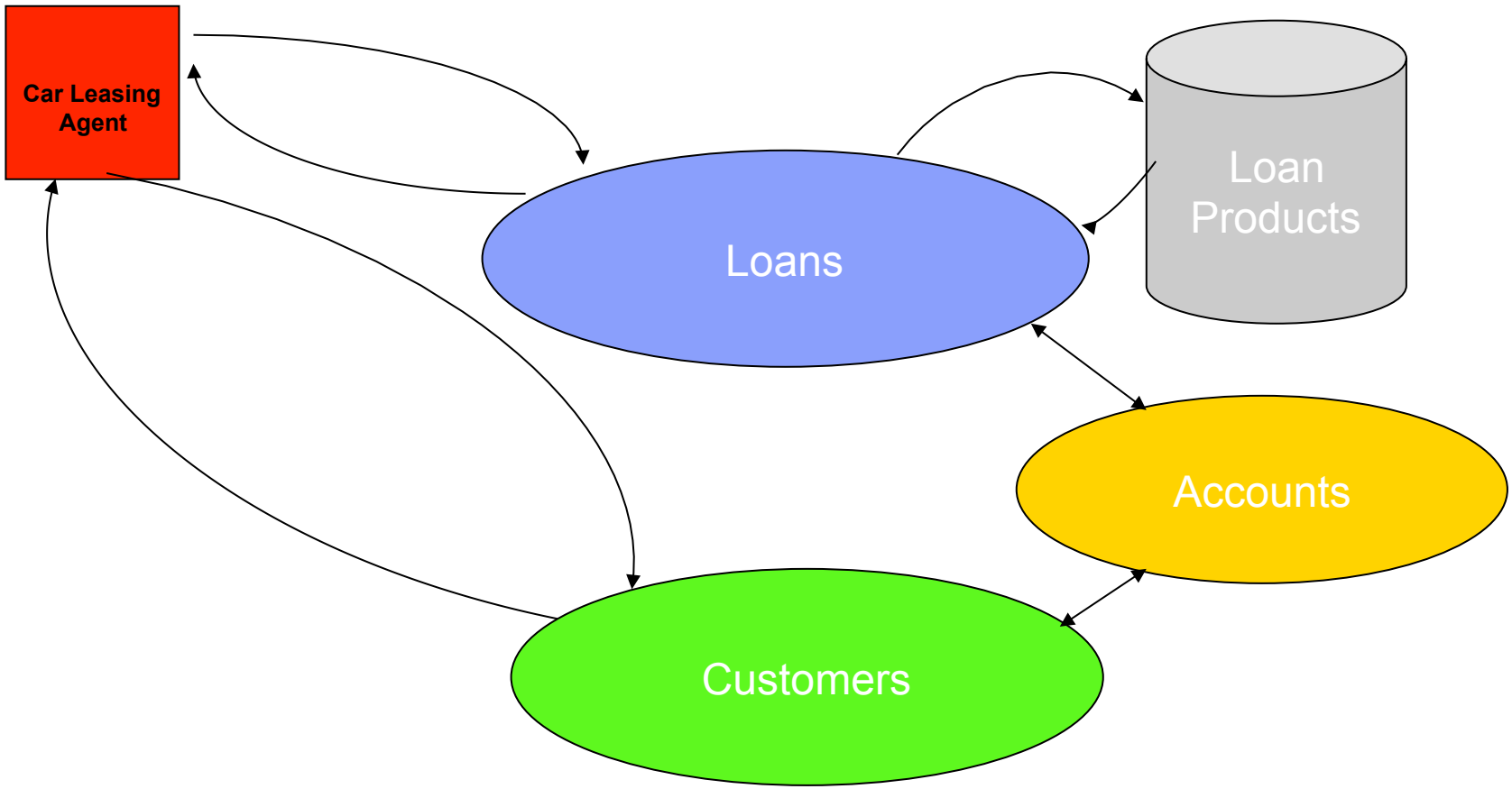



© 2014 Hans-Peter Hoidn & Kai Schwidder

# Enterprise IT Architectures



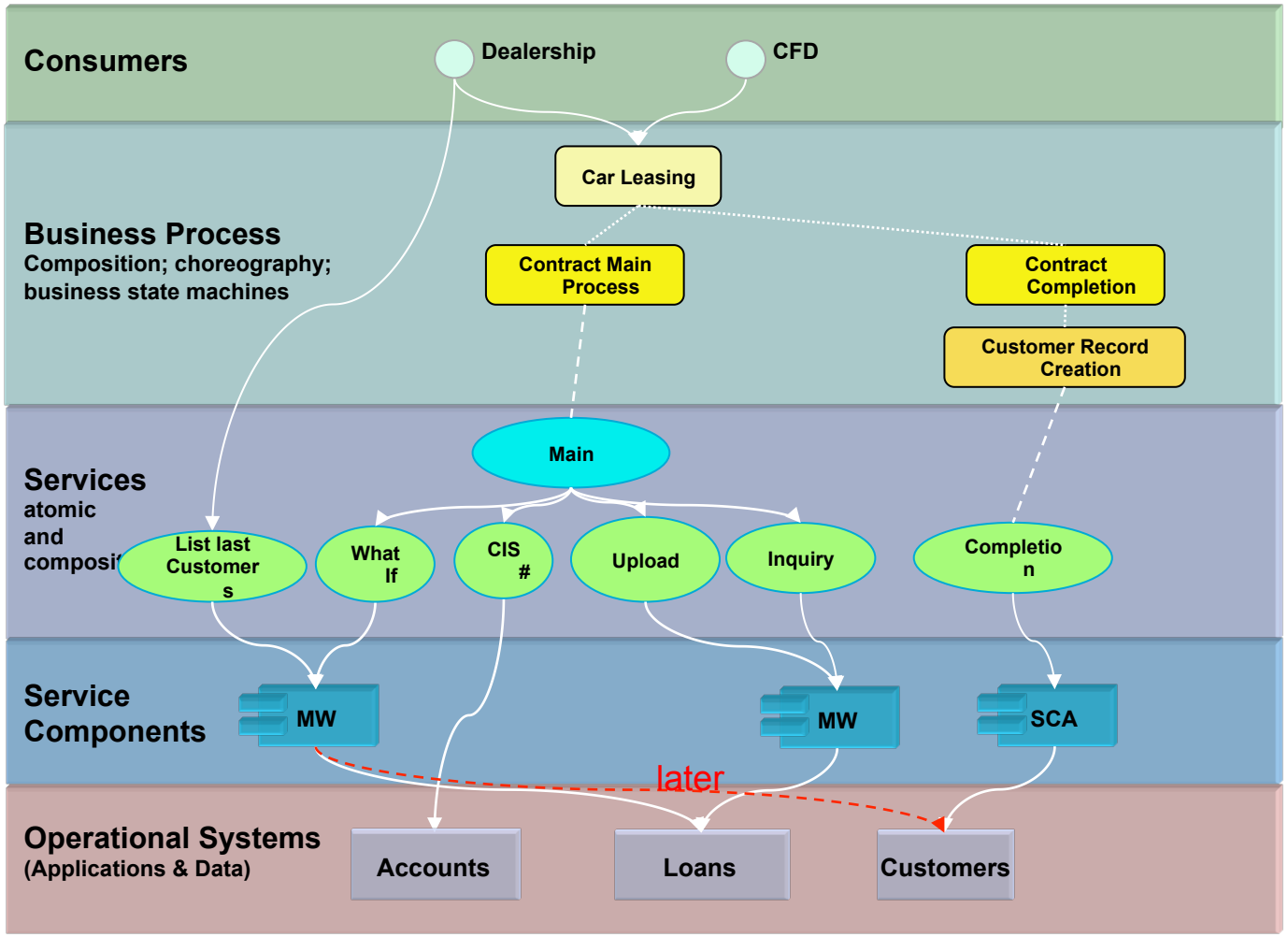
# *Example*: Context Diagram – Car Leasing






© 2014 Hans-Peter Hoidn & Kai Schwidder

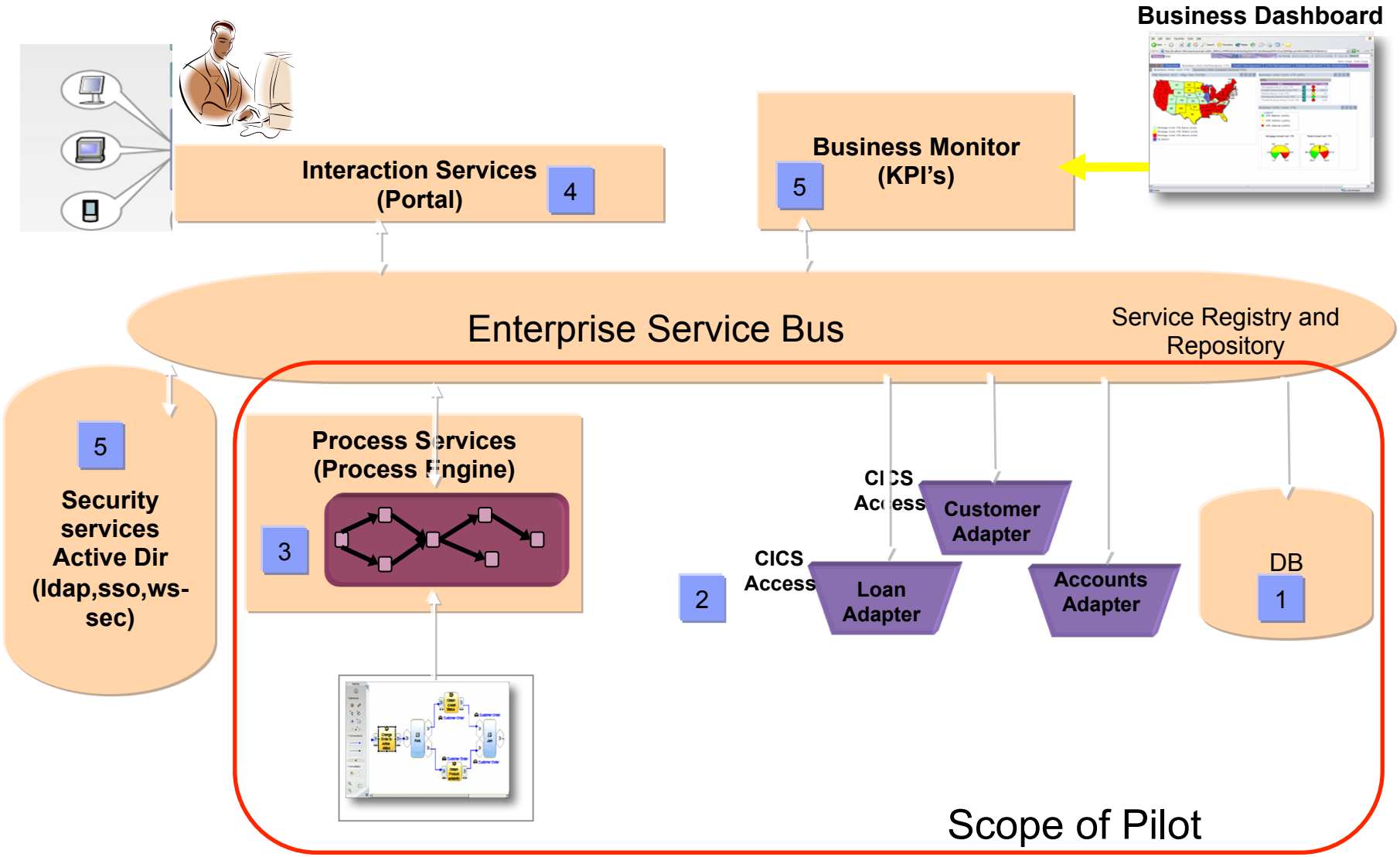
# *Example*: SOA Layered View – Car Leasing






© 2014 Hans-Peter Hoidn & Kai Schwidder

# *Example*: Technical View – Car Leasing

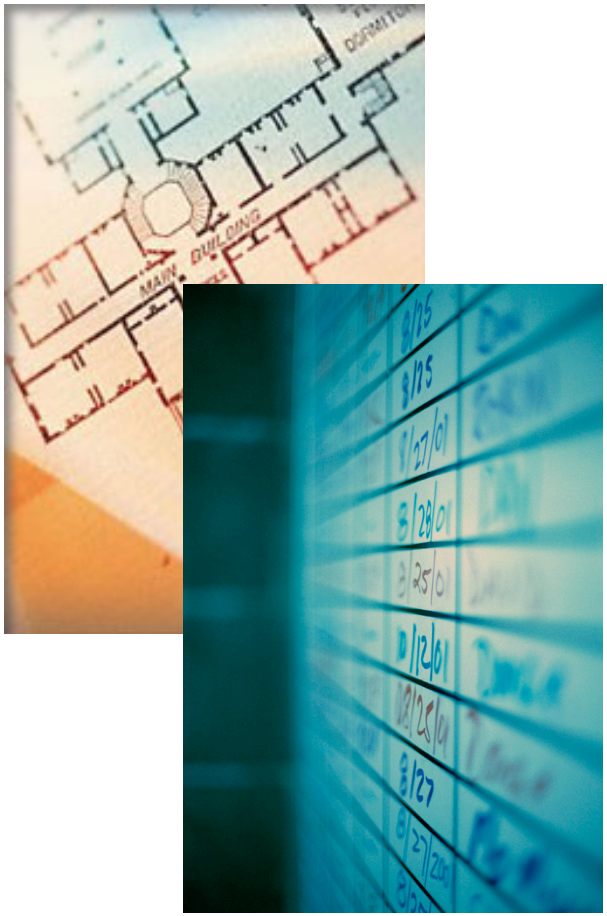





© 2014 Hans-Peter Hoidn & Kai Schwidder

# *Example*: Time Line – Car Leasing

- ***Phase 1*** Pilot project
  - Total duration 10 weeks
  - Estimated start July 2008
  - Estimated completion Mid September 2008
- ***Phase 2*** Pilot projects
  - Estimated total duration 1-4 months per process
  - - Extension and optimization of Car Lease
  - - New Credit Card process
  - - Document Archiving and Retrieval
  - - Consumer Loans

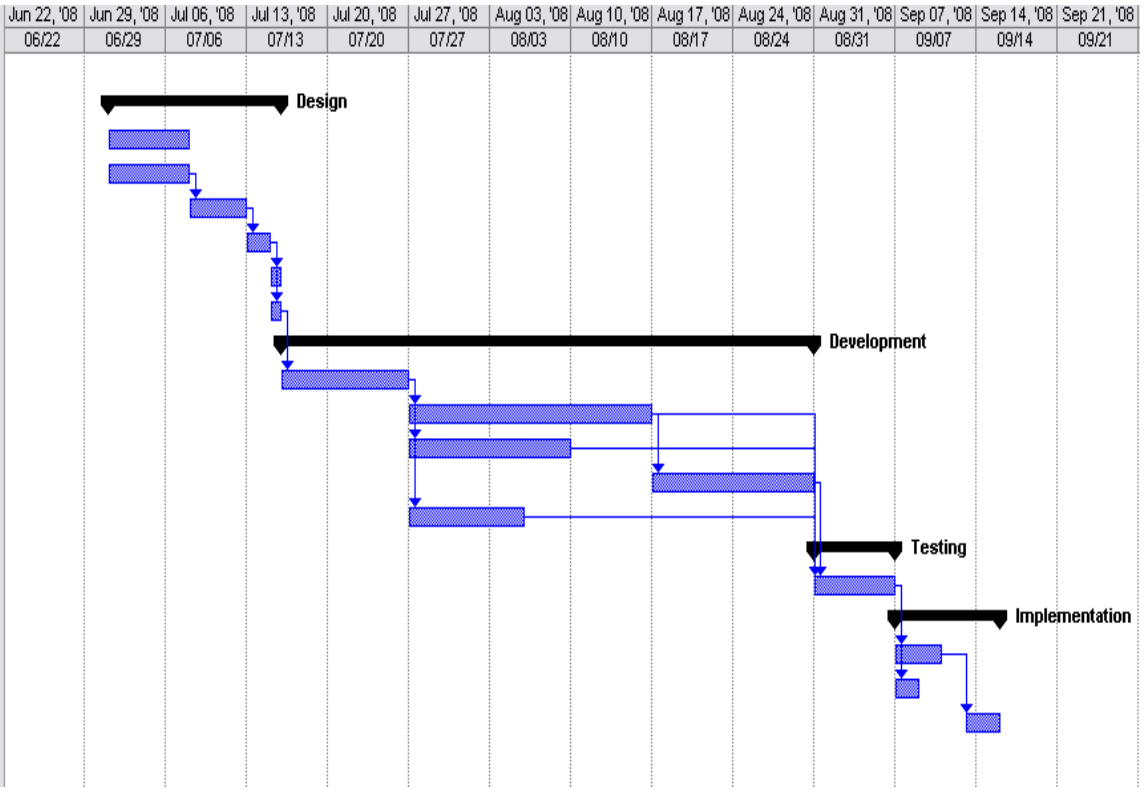




© 2014 Hans-Peter Hoidn & Kai Schwidder

# *Example*: Project Plan – Phase 1 Car Leasing

| | Task Name | Duration | Start | Finish | Predecessors |
|---|---|---|---|---|---|
| 1 | | | | | |
| 2 | **- Design** | **11 days** | **Tue 07/01/08** | **Tue 07/15/08** | |
| 3 | Seibel CSR Architecture | 5 days | Tue 07/01/08 | Mon 07/07/08 | |
| 4 | Security Architetcure | 5 days | Tue 07/01/08 | Mon 07/07/08 | |
| 5 | Access Services for Cor | 3 days | Tue 07/08/08 | Sat 07/12/08 | 4 |
| 6 | User Interface | 2 days | Sun 07/13/08 | Mon 07/14/08 | 5 |
| 7 | Reports | 1 day | Tue 07/15/08 | Tue 07/15/08 | 6 |
| 8 | Test Cases | 1 day | Tue 07/15/08 | Tue 07/15/08 | 6 |
| 9 | **- Development** | **32 days** | **Wed 07/16/08** | **Sat 08/30/08** | |
| 10 | Environment Setup/build | 7 days | Wed 07/16/08 | Sat 07/26/08 | 8 |
| 11 | Seibel Development | 15 days | Sun 07/27/08 | Sat 08/16/08 | 10 |
| 12 | Security Development | 10 days | Sun 07/27/08 | Sat 08/09/08 | 10 |
| 13 | Process Development | 10 days | Sun 08/17/08 | Sat 08/30/08 | 11 |
| 14 | User Interface | 8 days | Sun 07/27/08 | Tue 08/05/08 | 10 |
| 15 | **- Testing** | **5 days** | **Sun 08/31/08** | **Sat 09/06/08** | |
| 16 | Testing | 5 days | Sun 08/31/08 | Sat 09/06/08 | 14,11,12,13 |
| 17 | **- Implementation** | **7 days** | **Sun 09/07/08** | **Mon 09/15/08** | |
| 18 | Rollout | 4 days | Sun 09/07/08 | Wed 09/10/08 | 16 |
| 19 | Documentation | 2 days | Sun 09/07/08 | Mon 09/08/08 | 16 |
| 20 | Training | 3 days | Sat 09/13/08 | Mon 09/15/08 | 18 |






© 2014 Hans-Peter Hoidn & Kai Schwidder

© 2014 Hans-Peter Hoidn & Kai Schwidder