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

Enterprise IT Architectures

IT Architecture Standards, TOGAF and OMG in more Detail, Key Architecture Work Products

IT Architecture Standards

Recap: Architecture

- Architecture is a term that lots of people try to define
  - There is not just one way to state a system’s architecture
- Architecture provides:
  - Managing complexity
  - Layers of abstraction
  - Separation of Concerns
- Architecture covers:
  - 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

Architectures can be implied, apparent, or explicitly planned

- Implied architecture
  - of abstract things such as music or mathematics
- Apparent architecture
  - of natural things, such as geological formations or the structure of biological cells
- Explicitly planned architecture
  - of human-made things such as software, computers, enterprises, and databases, in addition to buildings.

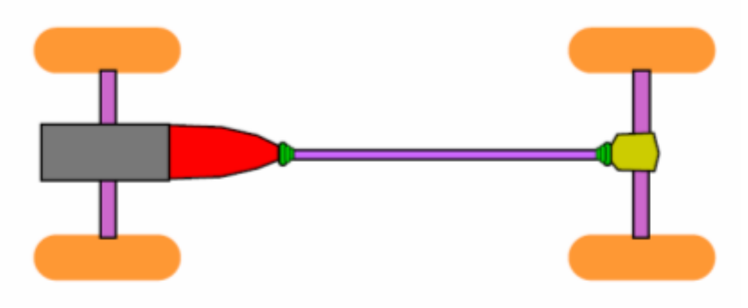
- Etymology:
  - Latin: architectus
  - Greek: arkhitekton (αρχιτεκτων) = master builder

    From arkhi (αρχι) = chief + tekton (τεκτων) = builder, carpenter

    archon: one of the nine chief magistrates of ancient Athens, 1659, from Gk. arkhon "ruler"

In every usage, an architecture, whether implied, apparent or explicitly planned, may be seen as:

- **A *subjective* mapping from one of many possible human perspectives**
  - **to the elements or components of some kind of structure or system,**
  - **which preserves the relationships among the elements or components.**



Models and Semantics

- Addressing *Content* and *Visualization* of a model or a view; first refers to concepts, second to visualisations
- A *Symbolic Model* expresses properties of architectures of systems by means of symbols that refer to reality
- A *Semantic Model* is an interpretation of a symbolic model, expressing the meaning of the symbols in that model

Source: Lankhorst, 2009, Section 3.3

Overview: Standards and Frameworks (Selection) – all with Enterprise View

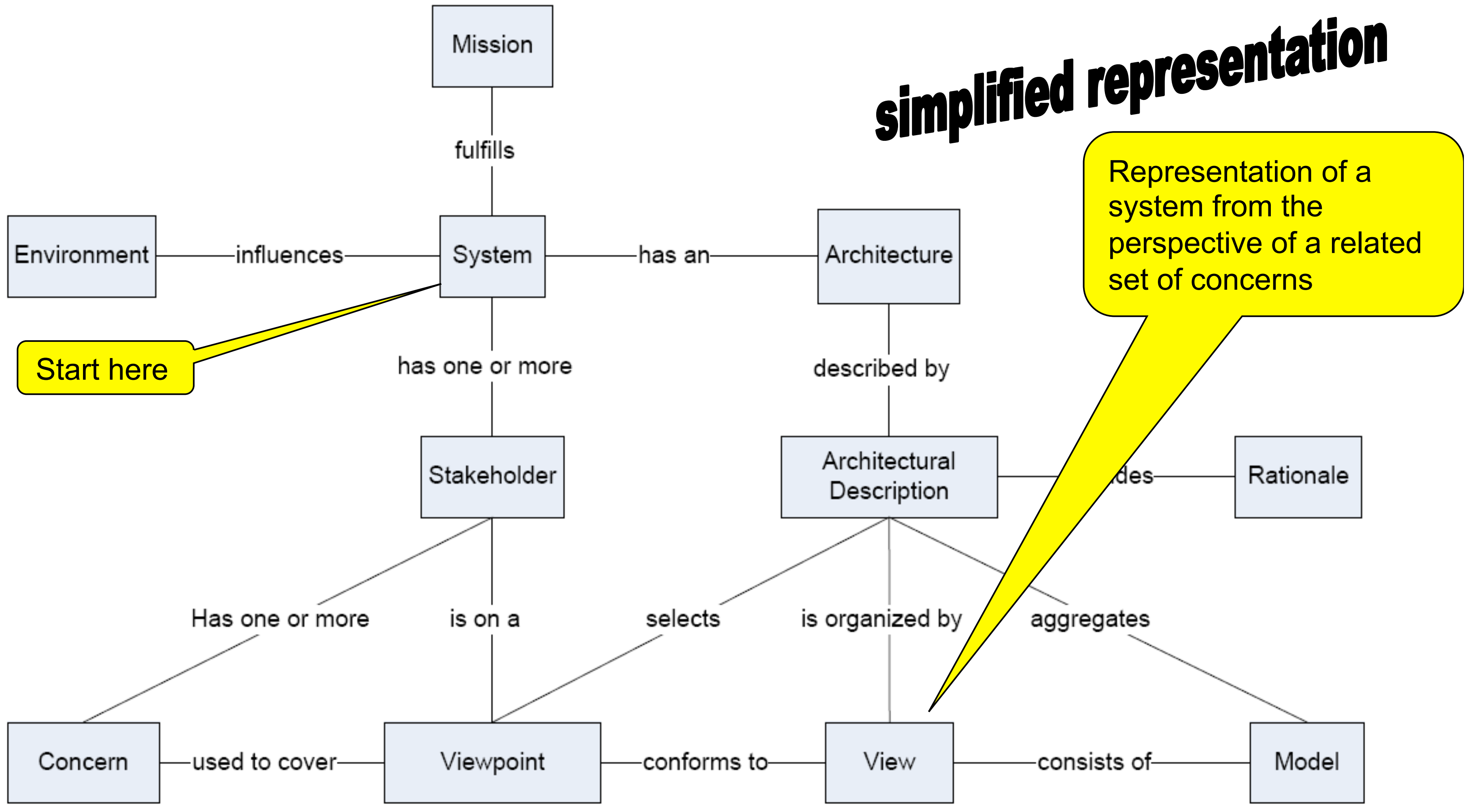
- IEEE
  - Definition and Metamodel
  - Uses UML notation
  - Provides a number of scenarios
- Zachman
  - First comprehensive framework
- TOGAF – by The Open Group
  - Will be our reference
- MDA (Model Driven Architecture) – by the OMG

IEEE 1471: Standard for Defining Architecture

- IEEE
  - "Recommended Practice for Architectural Description of Software-Intensive Systems"
  - Approved by IEEE in 2000, adopted by ISO 2007

- Standard:
  - It provides definitions and a meta-model for the description of architecture
  - It states that an architecture should address a system's stakeholders concerns
  - It asserts that architecture descriptions are inherently multi-view, no single view adequately captures all stakeholder concerns

IEEE 1471: Standard for Defining Architecture



Zachman Framework (see https://www.zachman.com)

- Zachman Framework
  - Is an ontology and a meta-model
  - Is not a methodology

Row 1 – Scope
External Requirements and Drivers
Business Function Modeling

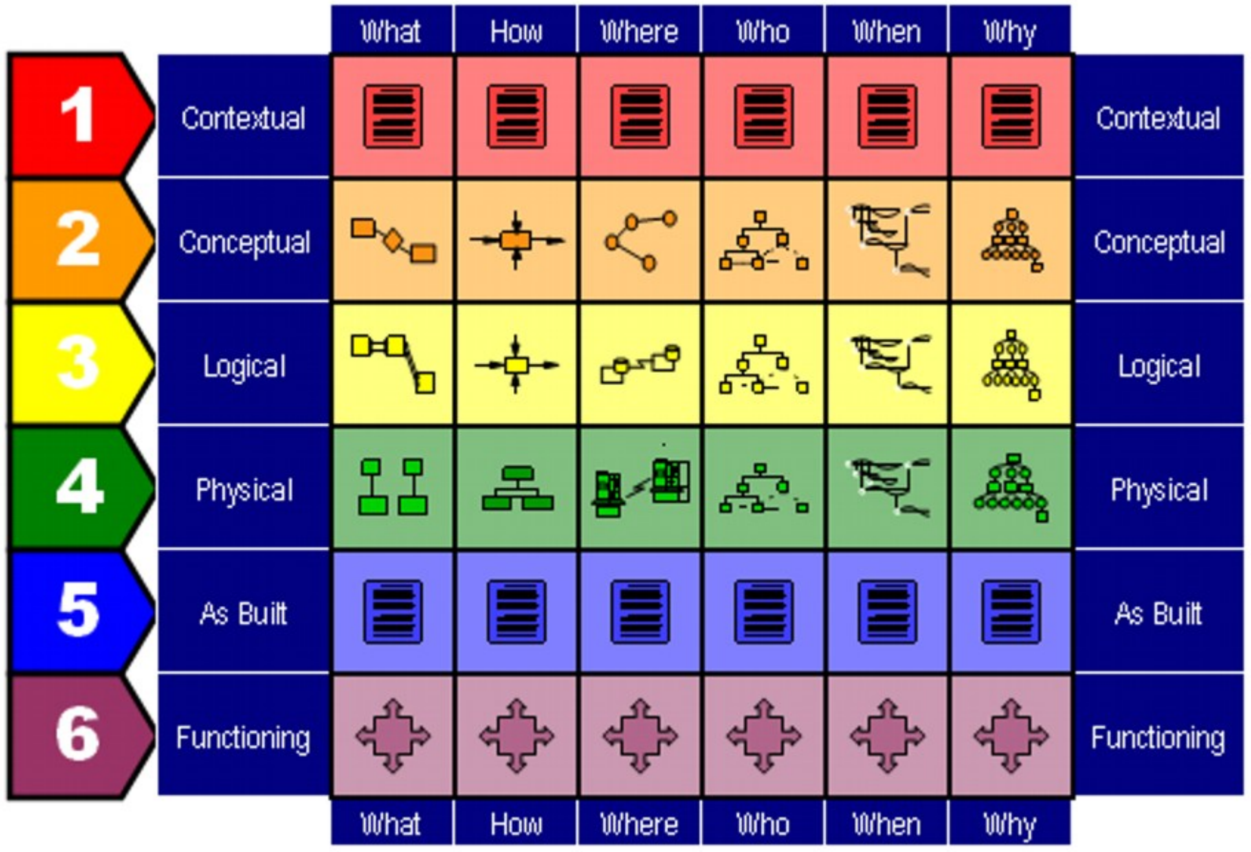
Row 2 – Enterprise Model
Business Process Models

Row 3 – System Model
Logical Models
Requirements Definition

Row 4 – Technology Model
Physical Models
Solution Definition and Development

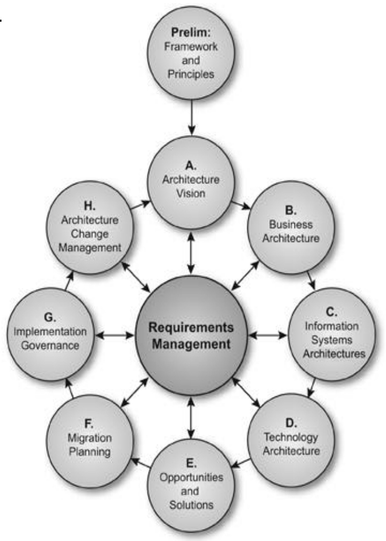
Row 5 – As Built
As Built
Deployment

Row 6 – Functioning Enterprise
Functioning Enterprise
Evaluation



TOGAF
(The Open Group Architecture Framework)

What is TOGAF


Prelim: Framework and Principles
A. Architecture Vision
B. Business Architecture
C. Information Systems Architectures
D. Technology Architecture
E. Opportunities and Solutions
F. Migration Planning
G. Implementation Governance
H. Architecture Change Management
Requirements Management

- **TOGAF consists of the following :**
  - **Architecture Development Method (ADM)**
  - **Enterprise Continuum**
  - **Resource Base**

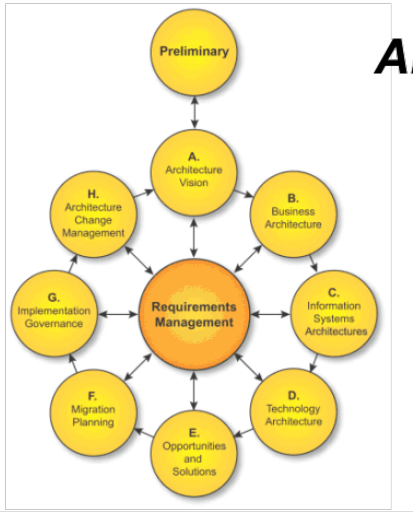
- **The ADM is depicted as the 'crop-circle' and represents the core of the TOGAF specification. It is a method for deriving a specific enterprise architecture.**

- **The Enterprise Continuum is a model for structuring a 'virtual repository' of architectural assets such as patterns, models, & architecture descriptions.**

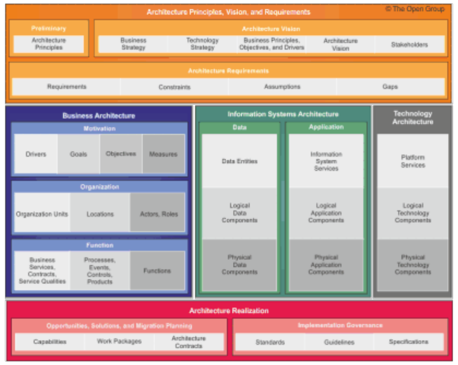
- **The Resource Base is a set of 'good practice' resources such as guidelines, checklists and templates provided to assist the architect when using TOGAF ADM.**

TOGAF Components (Version 9.1)

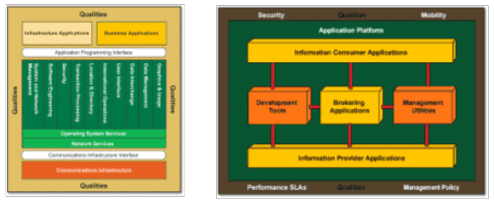
ADM


Preliminary
A. Architecture Vision
B. Business Architecture
C. Information Systems Architectures
D. Technology Architecture
E. Opportunities and Solutions
F. Migration Planning
G. Implementation Governance
H. Architecture Change Management
Requirements Management

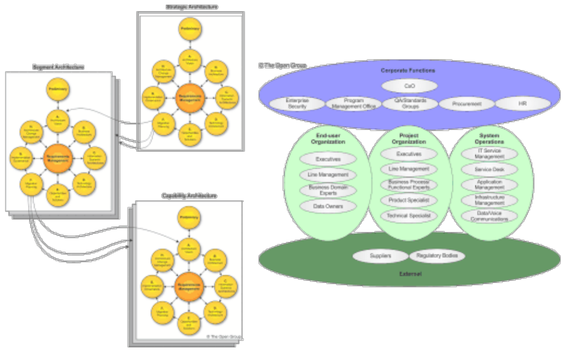
Architecture Content Framework



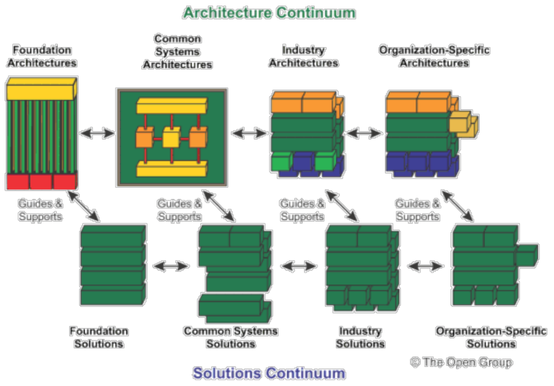
Reference Models


Qualities
Infrastructure Applications
Business Applications
Application Programming Interface
Operating System Services
Network Services
Communications Infrastructure Interface
Communications Infrastructure
Security
Mobility
Application Platform
Information Consumer Applications
Development Tools
Brokering Applications
Management Utilities
Information Provider Applications
Performance SLAs
Management Policy

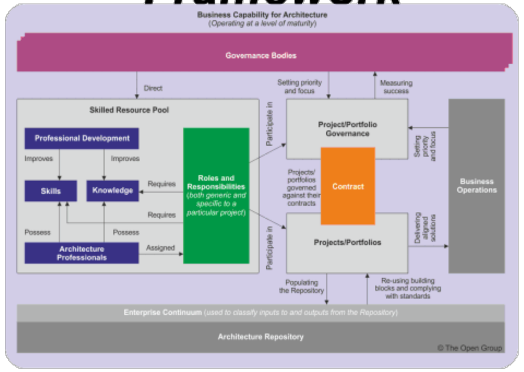
ADM Guidelines & Techniques



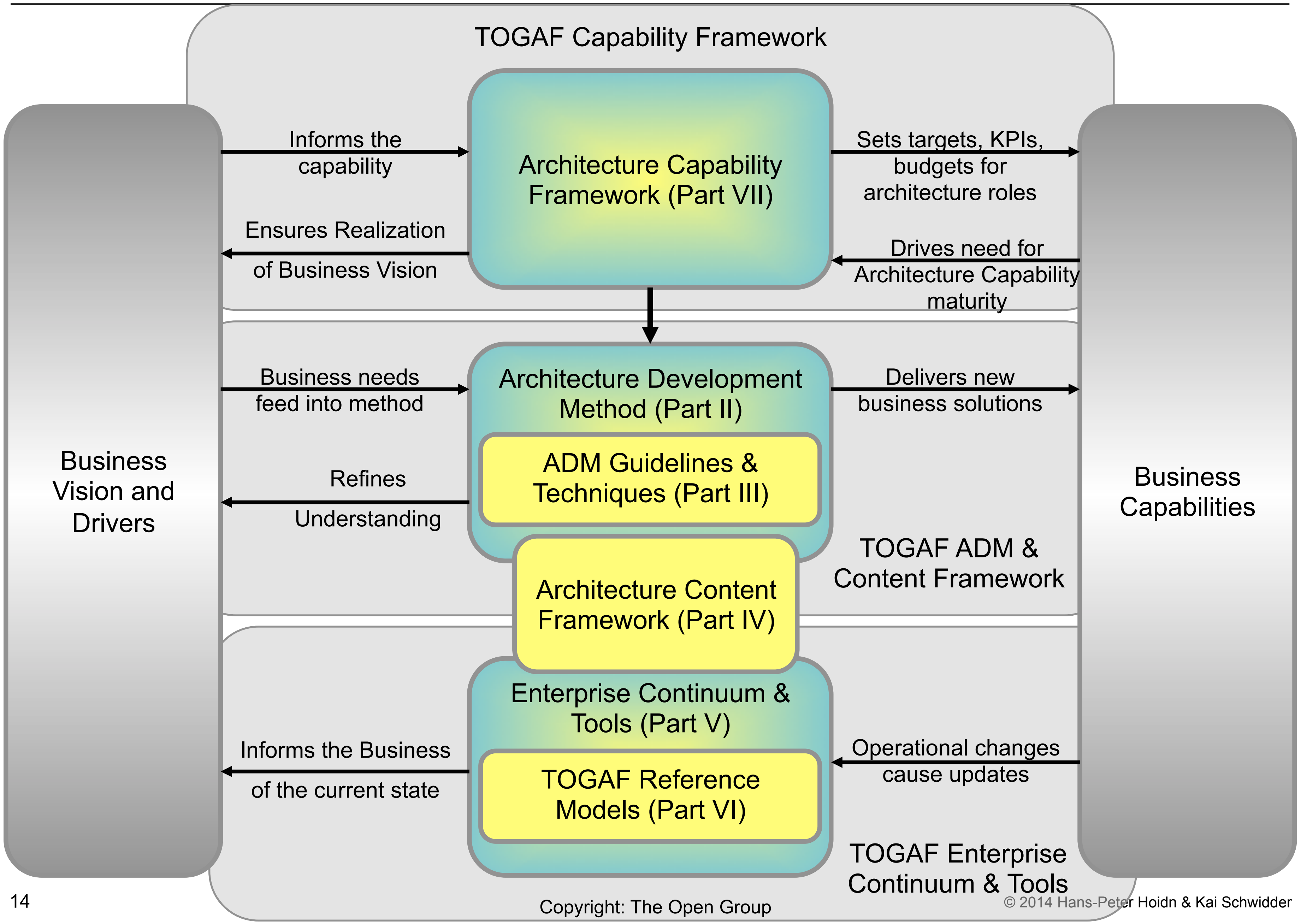
Enterprise Continuum



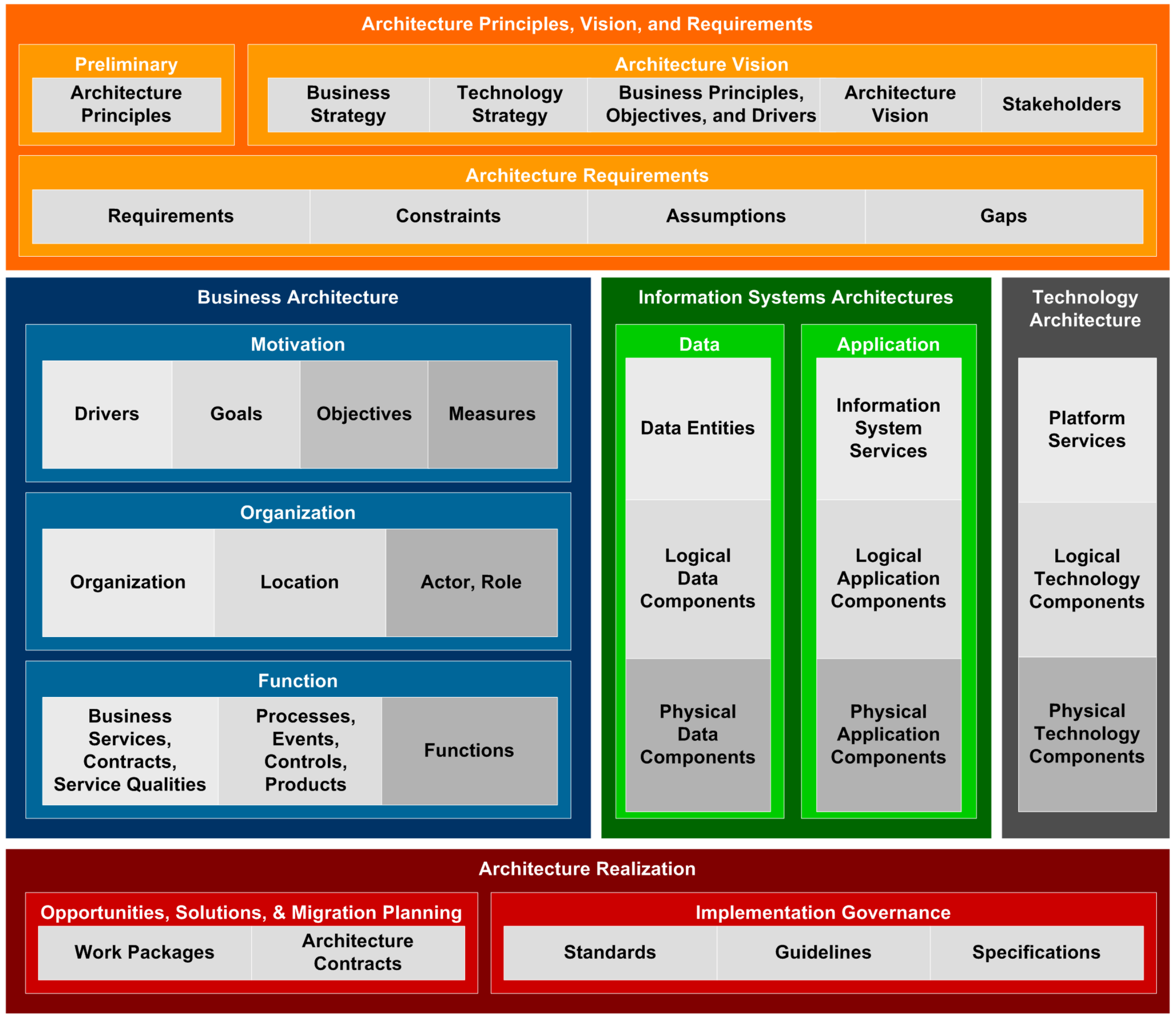
Architecture Capability Framework


Business Capability for Architecture
(Operating at a level of maturity)
Governance Bodies
Direct
Setting priority and focus
Measuring success
Skilled Resource Pool
Professional Development
Improves
Improves
Skills
Knowledge
Requires
Requires
Possess
Possess
Assigned
Architecture Professionals
Roles and Responsibilities (both generic and specific to a particular project)
Participate in
Project/Portfolio Governance
Projects/portfolios governed against their contracts
Contract
Setting priority and focus
Business Operations
Projects/Portfolios
Delivering aligned solutions
Populating the Repository
Re-using building blocks and complying with standards
Enterprise Continuum (used to classify inputs to and outputs from the Repository)
Architecture Repository
© The Open Group

Copyright: The Open Group


TOGAF Capability Framework
Business Vision and Drivers
Informs the capability
Ensures Realization of Business Vision
Architecture Capability Framework (Part VII)
Sets targets, KPIs, budgets for architecture roles
Drives need for Architecture Capability maturity
Business needs feed into method
Refines Understanding
Architecture Development Method (Part II)
ADM Guidelines & Techniques (Part III)
Delivers new business solutions
TOGAF ADM & Content Framework
Architecture Content Framework (Part IV)
Enterprise Continuum & Tools (Part V)
TOGAF Reference Models (Part VI)
Informs the Business of the current state
Operational changes cause updates
TOGAF Enterprise Continuum & Tools
Business Capabilities
Copyright: The Open Group

Content Metamodel

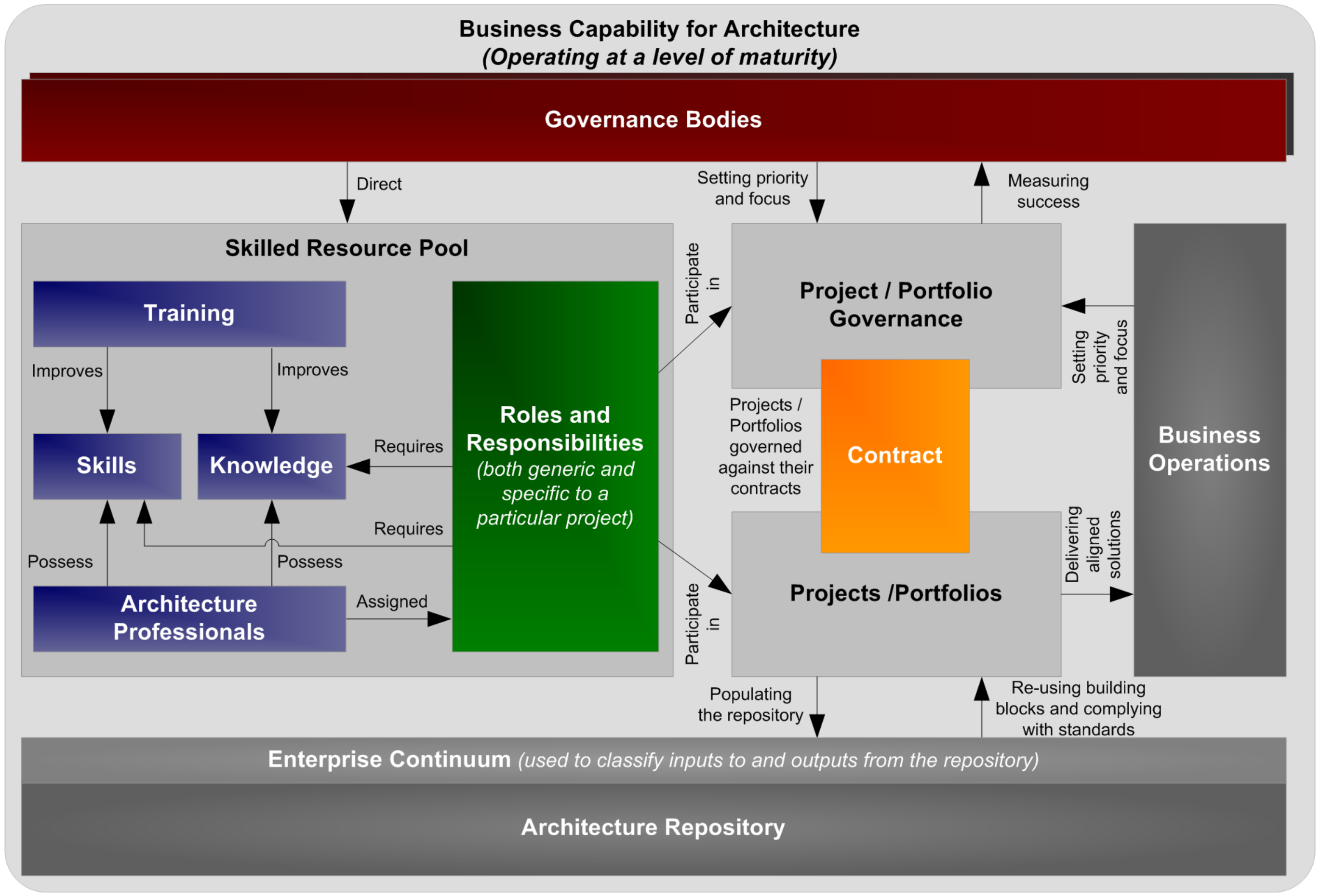


Copyright: The Open Group

Content Metamodel

- Building blocks continue to be the basic elements of the architecture within TOGAF
- The content framework provides a structured model of building block types, relationships and attributes
- The content framework can be used informally, or as the basis for configuration of an Enterprise Architecture modelling tool
- The content framework features a core and extension concept, with optional building block types, in order to support lightweight and detailed architectures

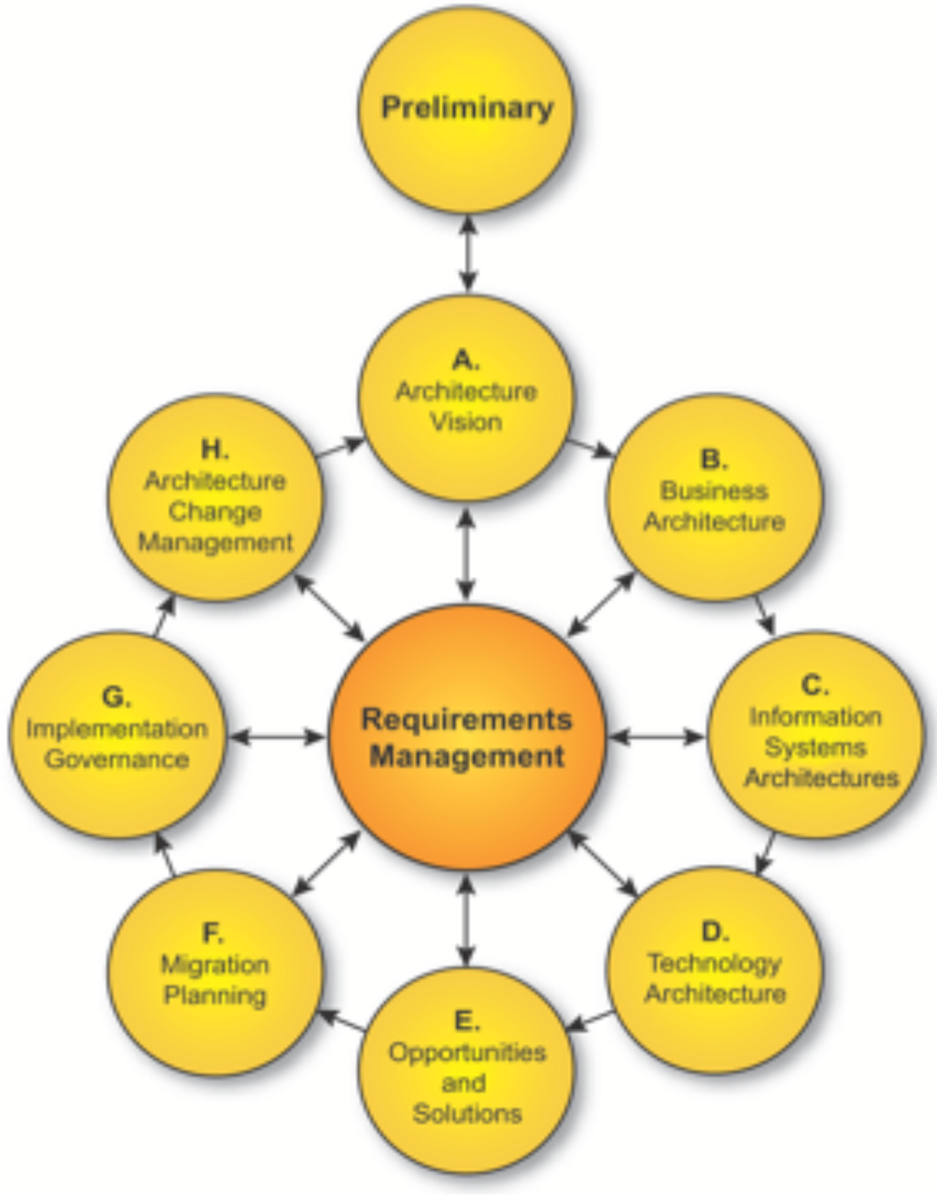
Capability Framework


Business Capability for Architecture
(Operating at a level of maturity)
Governance Bodies
Direct
Setting priority and focus
Measuring success
Skilled Resource Pool
Training
Improves
Improves
Skills
Knowledge
Requires
Requires
Possess
Possess
Architecture Professionals
Assigned
Roles and Responsibilities (both generic and specific to a particular project)
Participate in
Participate in
Project / Portfolio Governance
Contract
Projects / Portfolios governed against their contracts
Projects /Portfolios
Setting priority and focus
Delivering aligned solutions
Business Operations
Populating the repository
Re-using building blocks and complying with standards
Enterprise Continuum (used to classify inputs to and outputs from the repository)
Architecture Repository

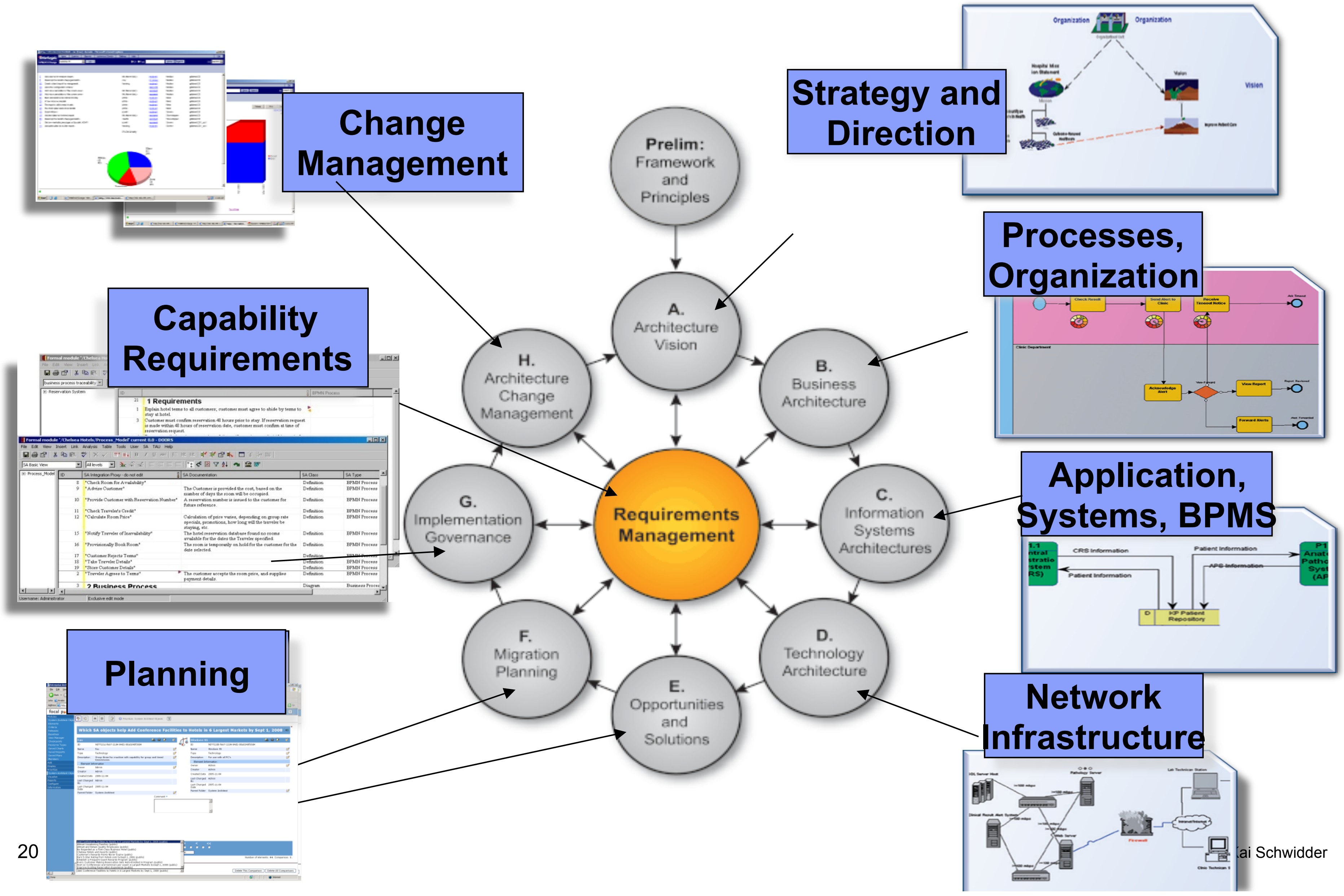
Capability Framework

- A structured definition of the organizations, skills, roles and responsibilities to establish and operate an Enterprise Architecture, including:
  - Terms of Reference for an Architecture Board
  - Guidance on measuring levels of Architecture Compliance against Architecture contracts
  - Processes and organization structures required to operate Architecture Governance
  - Techniques for assessing Architecture Maturity
  - An overview of the Skills required by practicing architects

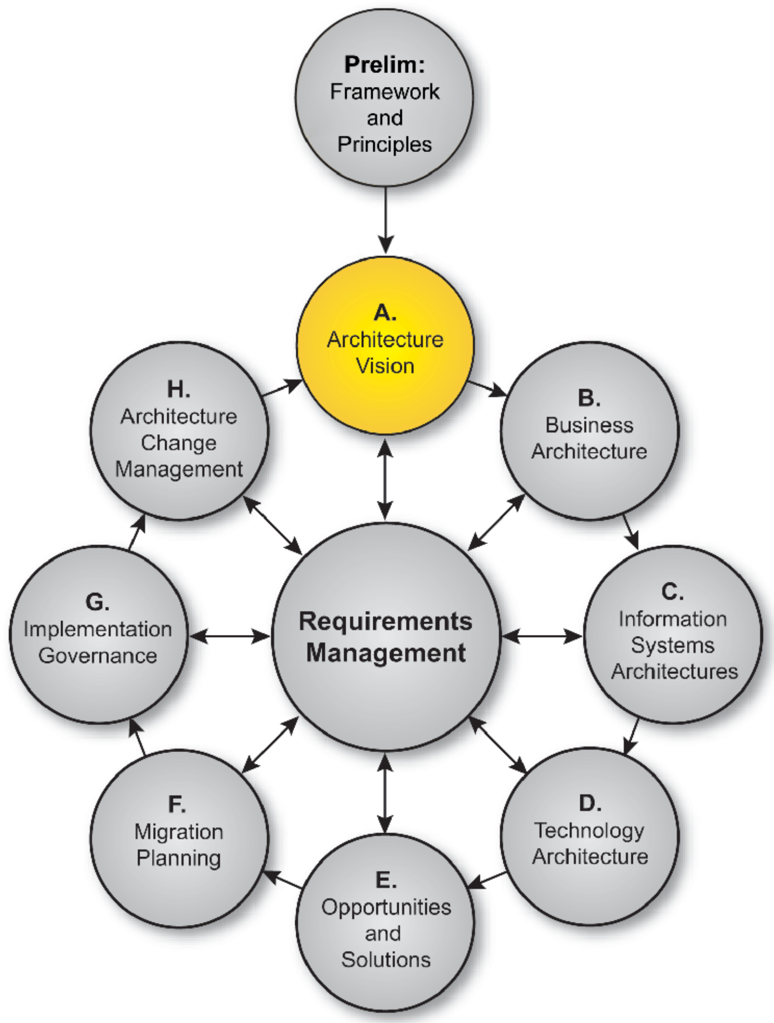
TOGAF ADM (Architecture Development Method)


Preliminary
A. Architecture Vision
B. Business Architecture
C. Information Systems Architectures
D. Technology Architecture
E. Opportunities and Solutions
F. Migration Planning
G. Implementation Governance
H. Architecture Change Management
Requirements Management

ADM Detail Overview



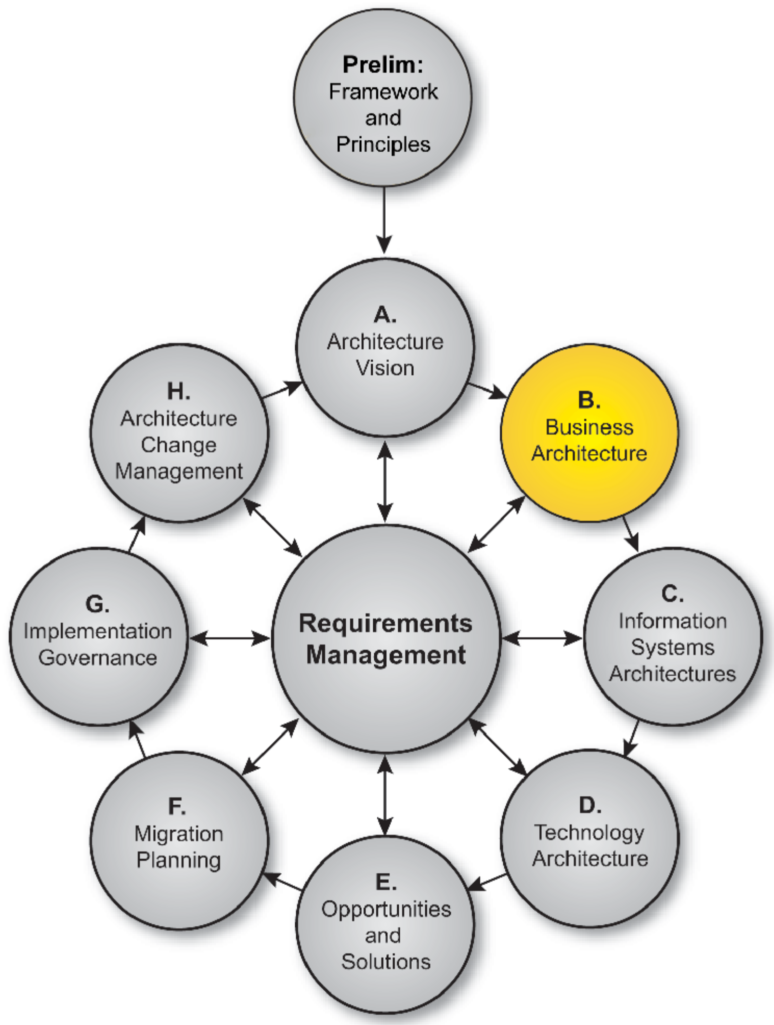
In more Detail: Capabilities Content
(Architecture Vision according to TOGAF - augmented)


Prelim: Framework and Principles
A. Architecture Vision
B. Business Architecture
C. Information Systems Architectures
D. Technology Architecture
E. Opportunities and Solutions
F. Migration Planning
G. Implementation Governance
H. Architecture Change Management
Requirements Management

- Initiates one iteration of the architecture process
  - Sets scope, constraints, expectations
  - Required at the start of every architecture cycle
- Validates business context
- Creates Statement of Architecture work

- Guiding Principles (according to IBM EA Methodology)
- Architectural Decisions (according to IBM EA Methodology)

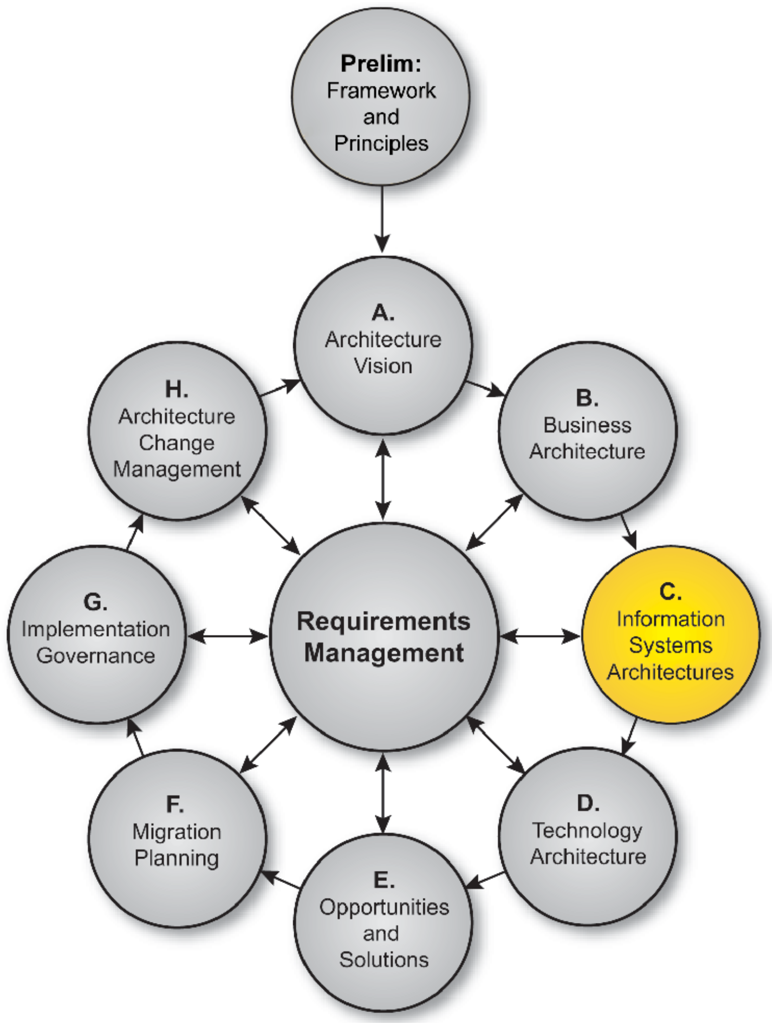
In more Detail: Business Architecture Content
(according to TOGAF augmented)


Prelim: Framework and Principles
A. Architecture Vision
B. Business Architecture
C. Information Systems Architectures
D. Technology Architecture
E. Opportunities and Solutions
F. Migration Planning
G. Implementation Governance
H. Architecture Change Management
Requirements Management

- Organization structure
- Business Goals and Objectives
- Business Functions
- Business Services
- Business Processes
- Business Roles
- Correlation of organization and functions.

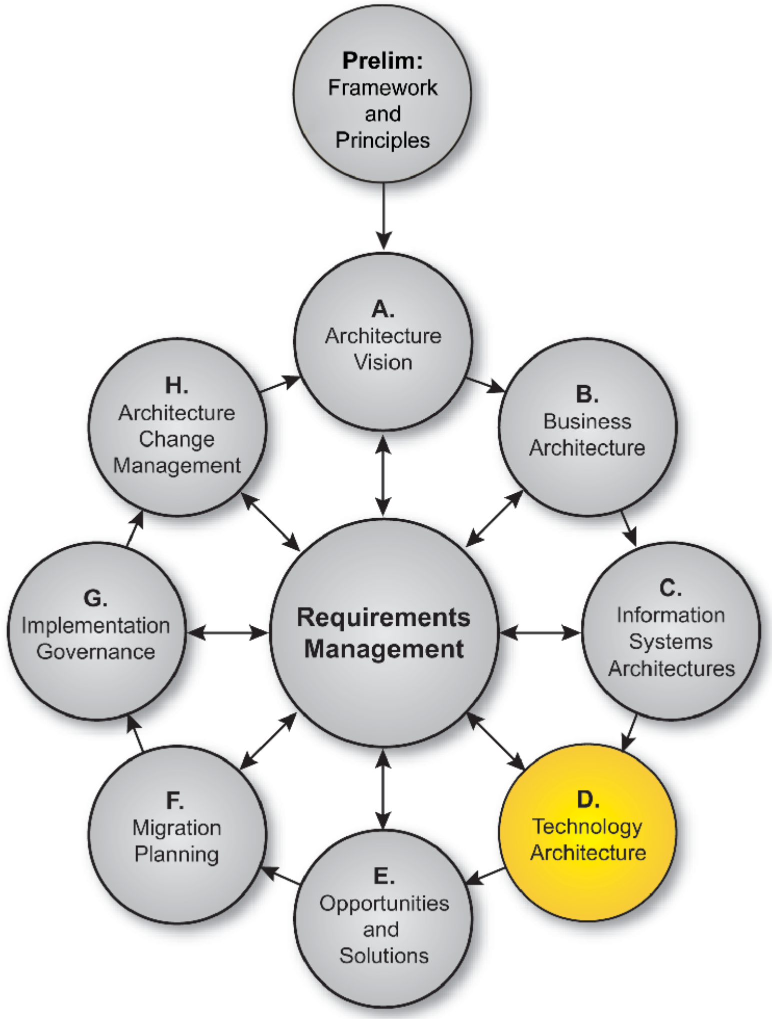
- Enterprise Information Model (according to IBM EA Methodology)

In more Detail: IS Architecture Content
(according to TOGAF)


Prelim: Framework and Principles
A. Architecture Vision
B. Business Architecture
C. Information Systems Architectures
D. Technology Architecture
E. Opportunities and Solutions
F. Migration Planning
G. Implementation Governance
H. Architecture Change Management
Requirements Management

- The fundamental organization of an IT system, embodied in
  - relationships to each other and the environment, and the principles governing its design and evolution
- Shows how the IT systems meets the business goals of the enterprise

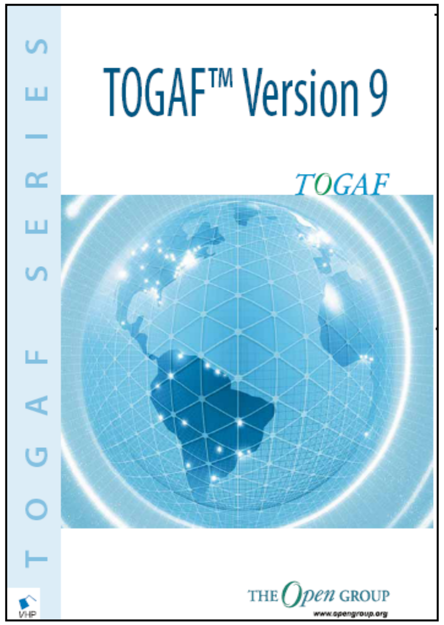
In more Detail: Technology Architecture Content (according to TOGAF)

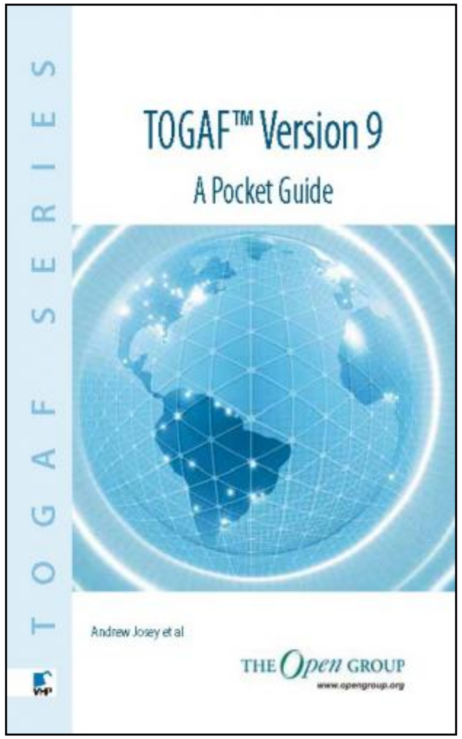

Prelim: Framework and Principles
A. Architecture Vision
B. Business Architecture
C. Information Systems Architectures
D. Technology Architecture
E. Opportunities and Solutions
F. Migration Planning
G. Implementation Governance
H. Architecture Change Management
Requirements Management

- The fundamental organization of an IT system, embodied in
  - its hardware, software and communications technology
  - their relationships to each other and the environment,
  - and the principles governing its design and evolution

Further Information on TOGAF

- Detailed information on TOGAF 9 including downloads of the specification, links to white papers, information sheets, reference cards, etc is available at:
  - http://www.opengroup.org/togaf/
  - http://www.togaf.info



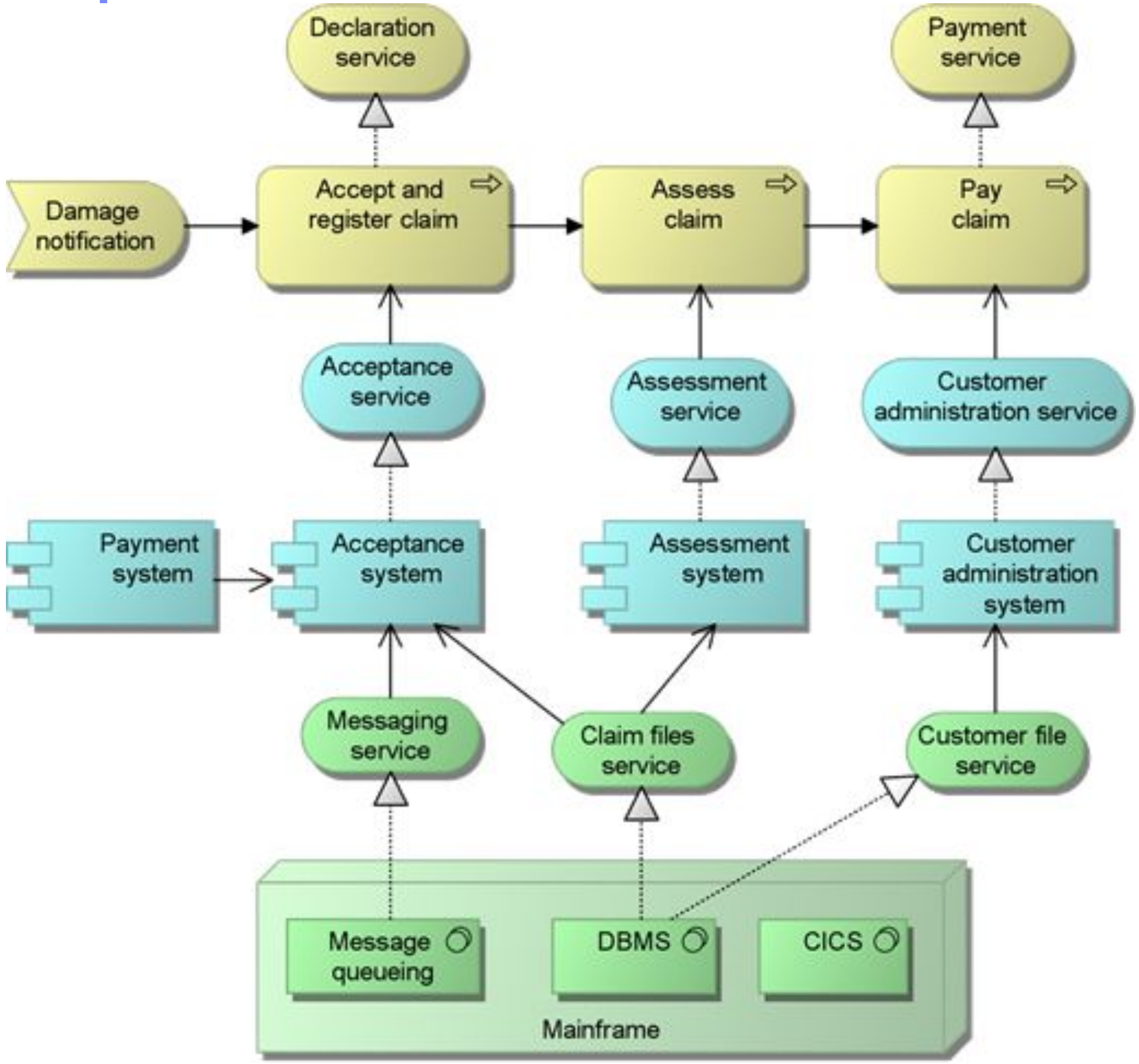


ArchiMate

- ArchiMate is a modeling technique ("language") for describing enterprise architectures
  - ArchiMate is partly based on the IEEE 1471 standard. It was developed in the Netherlands by a project team from the Telematica Instituut in cooperation with several Dutch partners
  - Now Open Group Standard – see http://www.opengroup.org/subjectareas/enterprise/archimate
- ArchiMate offers a common language for describing the construction and operation of business processes, organizational structures, information flows
  - Layering: Business, Applications, Infrastructure
  - Dimensions: Passive Structure, Behavior, Active Structure

Source: Wikipedia

ArchiMate Example



Source: Wikipedia

TOGAF – HPH’s Personal Remarks

- **TOGAF provides solid and agreed definitions of terms (backed by a large membership of companies)**
- **ArchiMate provides a modeling language for architecture, however we may do some work redundantly, e.g. we will later model business processes on its own**
- **Includes not much about deliverables to support architecture work (“work products” / “artifacts”)**
- **Does NOT emphasize iterative work – however IMHO (In My Humble Opinion) ALL work should be done in iterations**

Models – and OMG (Object Management Group)

OMG (Object Management Group)

- "owner" of CORBA, UML, MOF, MDA, CWM, BPMN, …
- Only organizations (ca. 300 now) can be members, work done by the representatives of the members
- OMG's Role
  - Build *Consensus* for Interoperability over heterogeneous networks
  - and, through *Modeling*, support the design, implementation, and maintenance of the interoperable systems that run on them

OMG Major Successes

Unified Modeling Language

- UML® remains the world's only standardized modeling language

Business Process Modeling Notation

- BPMN™ provides businesses with the capability of understanding their internal business procedures

Systems Modeling Language

- SysML™ supports the specification, analysis, design, and verification and validation of a broad range of complex systems.

Data Distribution Service

- DDS™, Real-time, data-centric, publish-subscribe OMG specification for data distribution

Meta-Object Facility

- MOF™, the repository standard

XML Metadata Interchange

- XMI®, the XML-UML standard

Common Object Request Broker Architecture

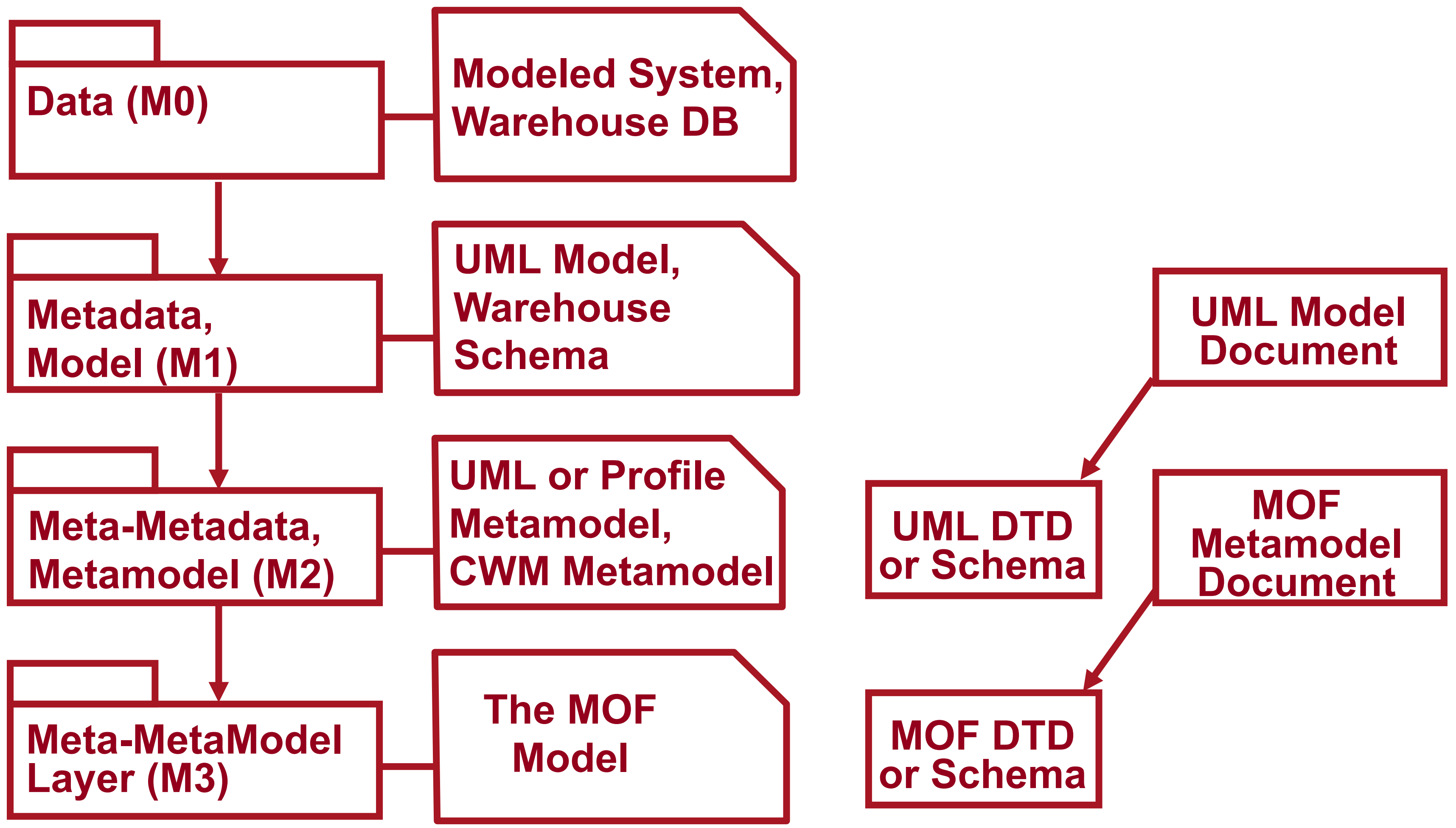
- CORBA® remains the only language- and platform-neutral interoperability standard

MOF (Meta-Object Facility)

- Is a meta-meta-model OR a model that defines the constructs for meta-models (a meta-model defines the structure of models, e.g. the meta-model for UML Class diagrams, Use cases, …)

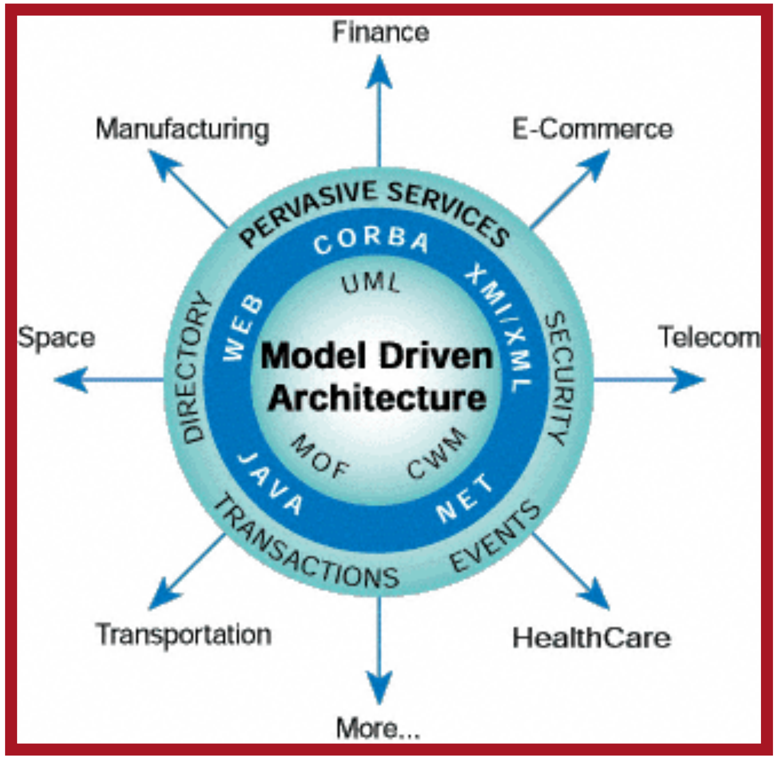
- Advantages (quote from OMG):
  - MOF standardizes the format of Metadata for compliant Languages and Models
  - If you know the Meta-Model for your Modeling Language, you can specify a Transformation
  - And a Tool can apply the Transformation to your Input Model, and produce an Output Model
  - Based on MOF, all diverse model elements can share repositories and interchange models among compliant tools

Interchanging Metadata


Data (M0)
Modeled System, Warehouse DB
Metadata, Model (M1)
UML Model, Warehouse Schema
Meta-Metadata, Metamodel (M2)
UML or Profile Metamodel, CWM Metamodel
Meta-MetaModel Layer (M3)
The MOF Model
UML Model Document
UML DTD or Schema
MOF Metamodel Document
MOF DTD or Schema

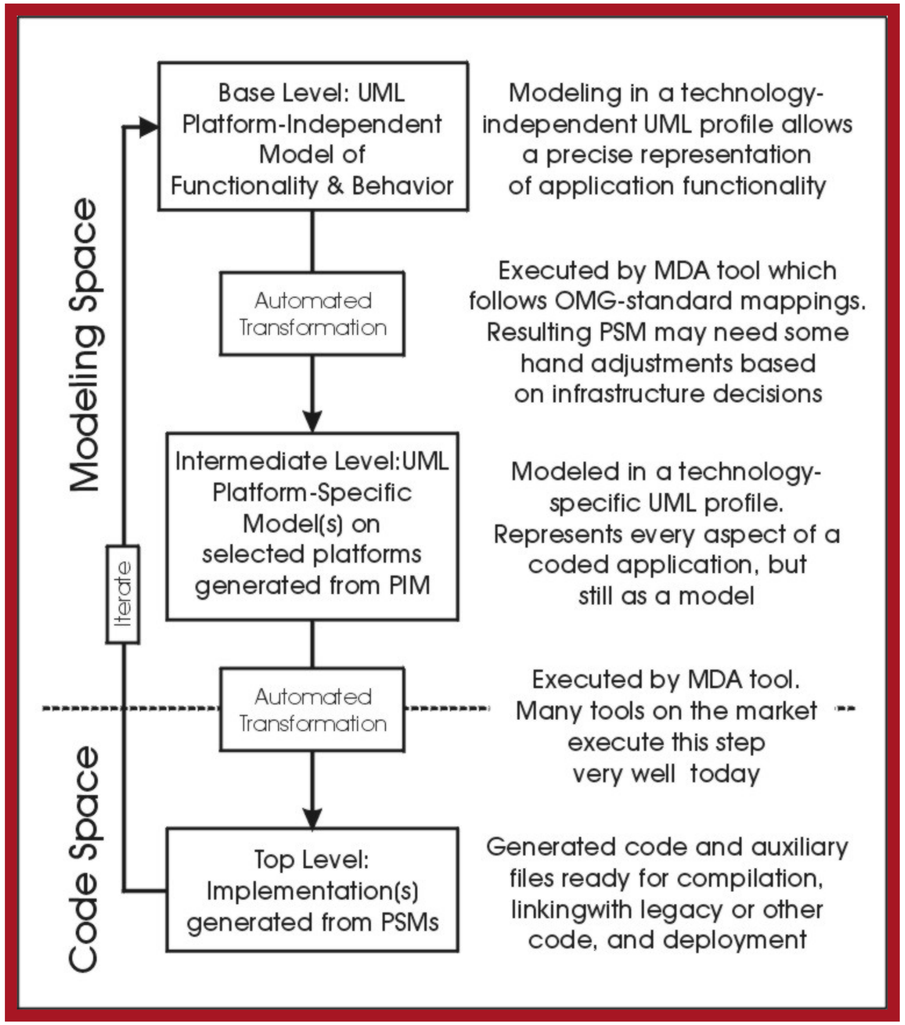
MDA (Model Driven Architecture)

- A Way to Specify and Build Systems
  - Based on Modeling and UML
  - Supports full lifecycle: A&D, implementation, deployment, maintenance, and evolution
  - Builds in Interoperability and Portability
- Reference:
  - MDA Guide rev. 2.0 OMG Document ormsc/2014-06-01
  - See http://www.omg.org/mda/
- For
  - Unifies modeling and implementation into a synergistic environment

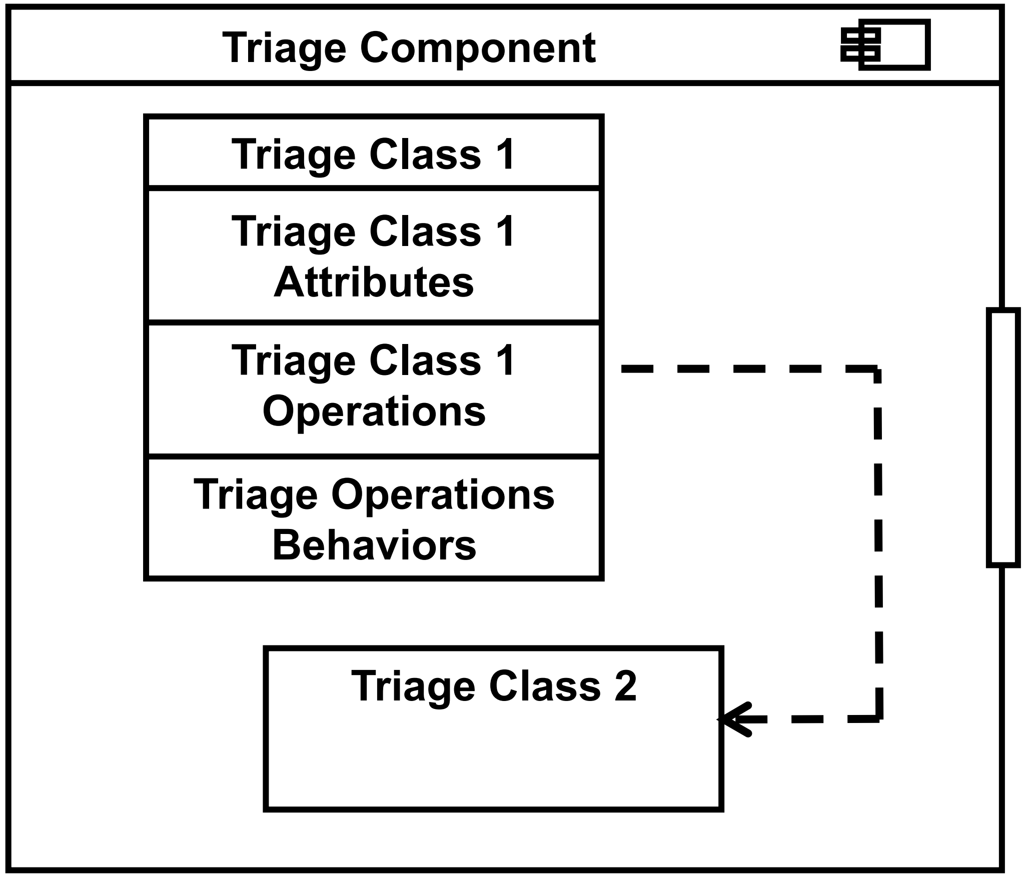


Models designed for Business – MDA Model Transformation and Execution

- Structure is a Spectrum progressing from Modeling at the Top to Code development at the bottom
- Input and Investment concentrate at the business zone at the top
- Automated tools take over coding IT infrastructure towards the bottom


Modeling Space
Code Space
Iterate
Base Level: UML Platform-Independent Model of Functionality & Behavior
Modeling in a technology-independent UML profile allows a precise representation of application functionality
Automated Transformation
Executed by MDA tool which follows OMG-standard mappings. Resulting PSM may need some hand adjustments based on infrastructure decisions
Intermediate Level:UML Platform-Specific Model(s) on selected platforms generated from PIM
Modeled in a technology-specific UML profile. Represents every aspect of a coded application, but still as a model
Automated Transformation
Executed by MDA tool. Many tools on the market execute this step very well today
Top Level: Implementation(s) generated from PSMs
Generated code and auxiliary files ready for compilation, linkingwith legacy or other code, and deployment

Component Model and MDA


Triage Component
Triage Class 1
Triage Class 1 Attributes
Triage Class 1 Operations
Triage Operations Behaviors
Triage Class 2

Then MDA Generates the application and its connectivity from this detailed model

So you know that the application conforms to the model, connectivity works, and changes to any level model work in the real world

OMG – HPH's Personal Remarks

- OMG is heavily supporting the modeling approach. Modeling is a very strong approach because of the semantics that allow to verify models as well as use the model for execution (as done by BMPN)
- The standardization process is slow and sometimes interests of member companies drive too much the development (as it happens with UML 2)
- Models are suitable for IT professionals – not always consumable for stakeholders (thus the tendency to replace models by ad hoc visualizations)
- HPH's bias: Served as primary representative for UBS and PwC at the OMG 1999 – 2002 (including alternate board member 2001)

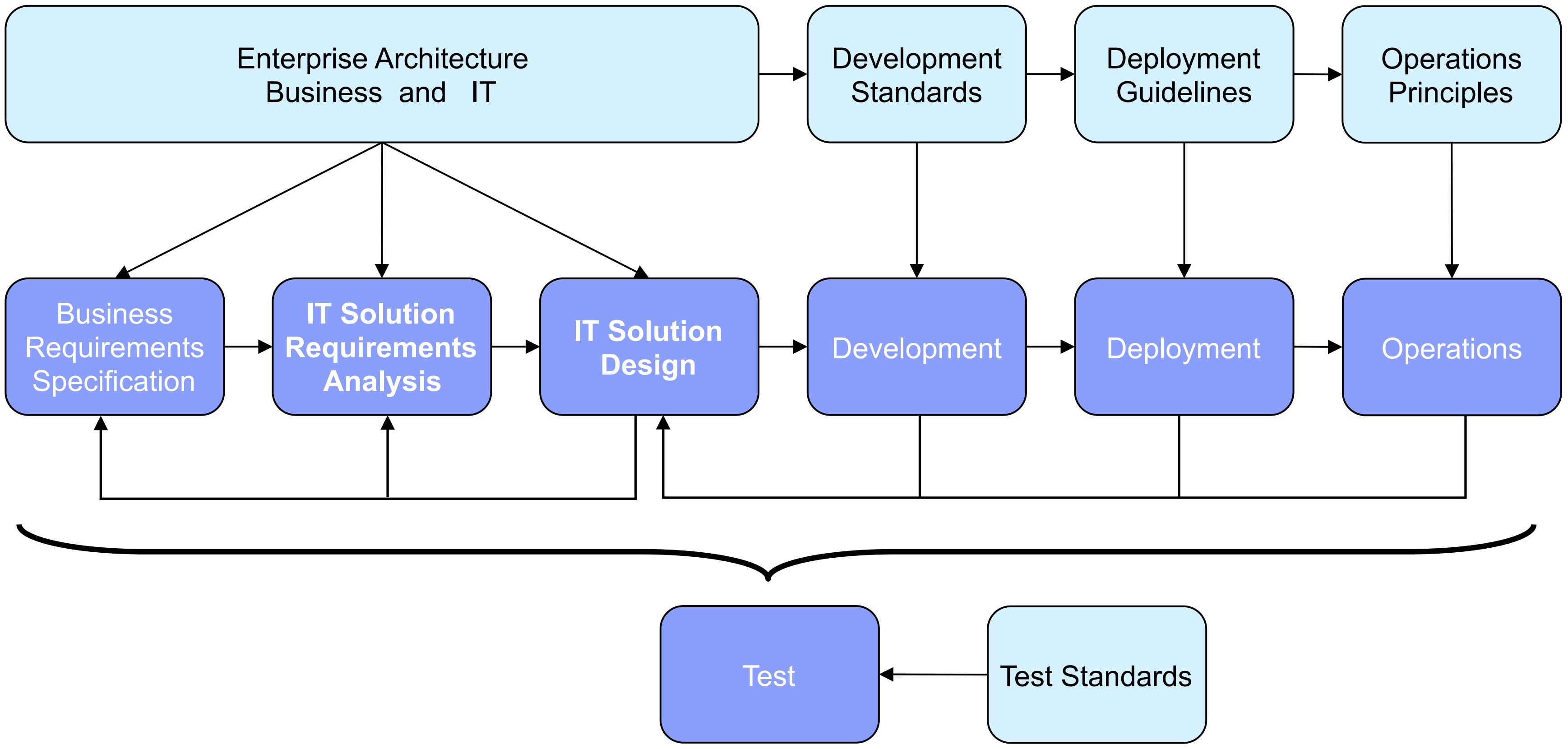
Questions

“Work Products” following (a bit) IBM’s Methodology

Techniques and Methods

- **Architectural Thinking**
  - **Business – Information Systems – Technology**
  - **Taking decisions and documenting them ("what is not written down does not exist")**
- **Assessments and Evolutions**
  - **Assessment of Current IT Environment**
  - **Existing goals, functional and non-functional requirements**
- **Major artifacts (work products):**
  - **Architecture Overview**
  - **Component Model**
  - **Operational Model**

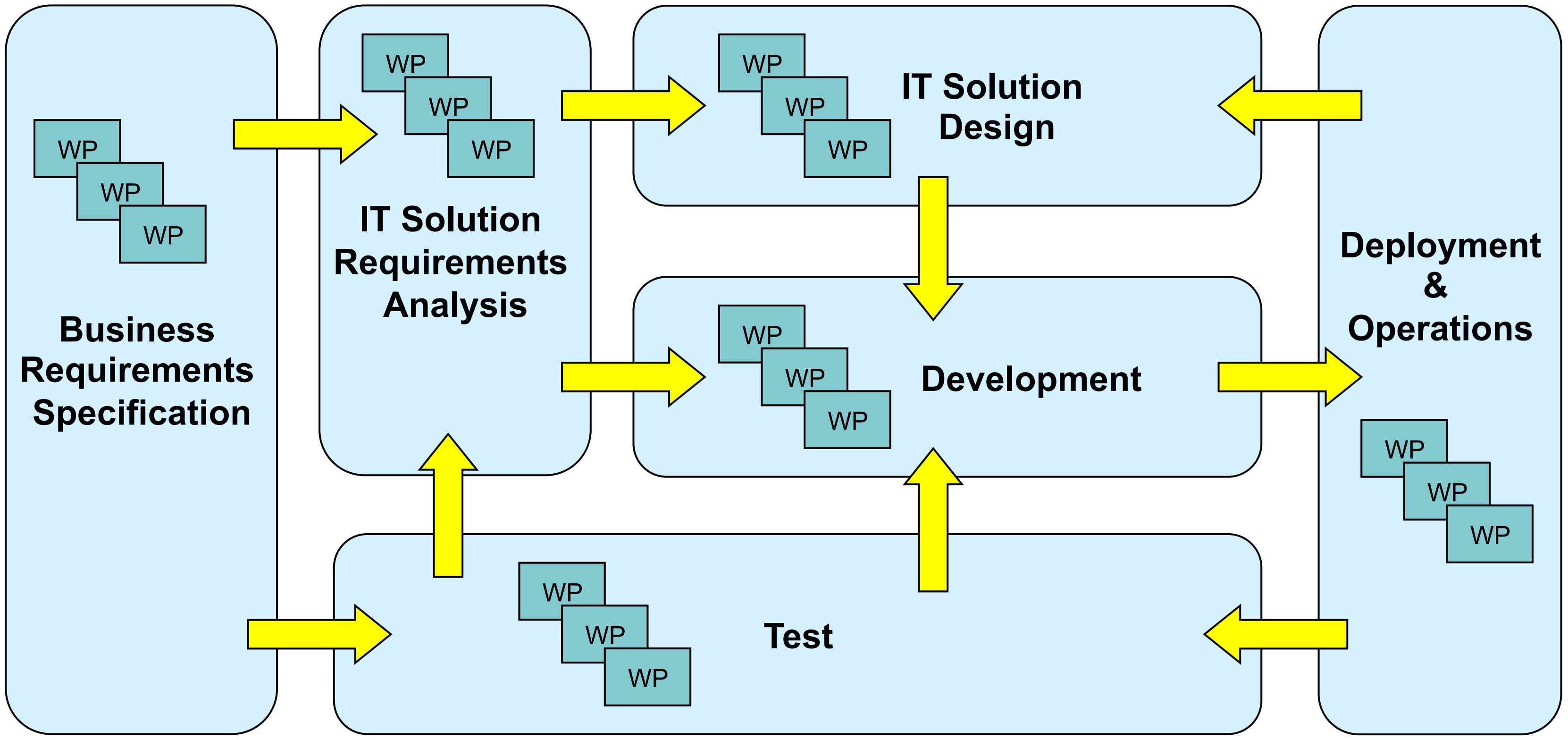
An IT Architect does not work alone – their work is intimately interconnected with that of others in both the business & IT organisations…



Source: IBM Architectural Thinking

…with a strong set of inter-dependencies between all areas

Illustrative dependencies only! Most arrows are two way!


WP
Business Requirements Specification
IT Solution Requirements Analysis
IT Solution Design
Development
Test
Deployment & Operations

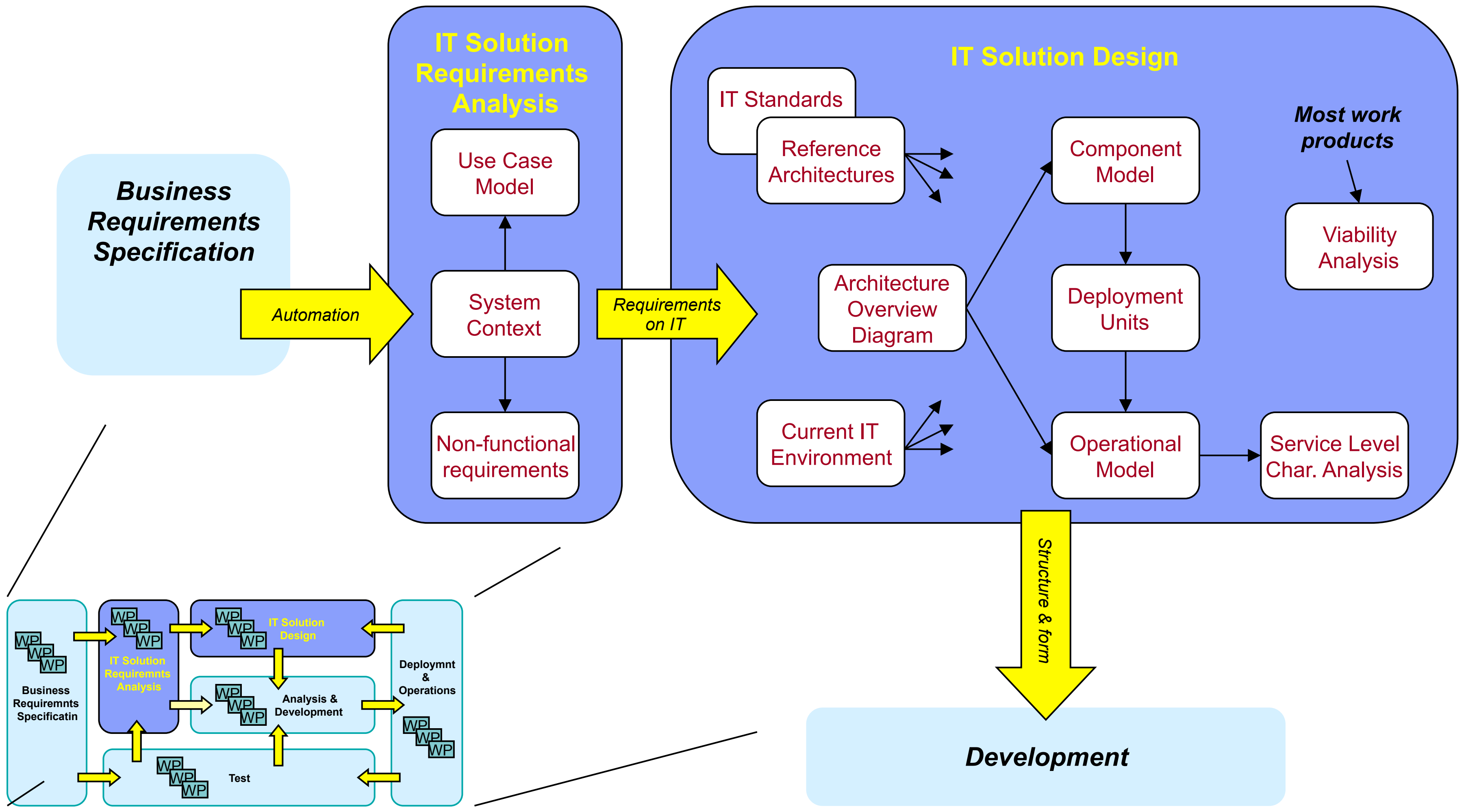
How do we manage this division of responsibilities, while accommodating interdependencies across the life-cycle of a project, without adopting "waterfall development"?

A work product is (usually) a document that defines some specific aspect of the solution and "lives" through the full life of the project

Source: IBM Architectural Thinking

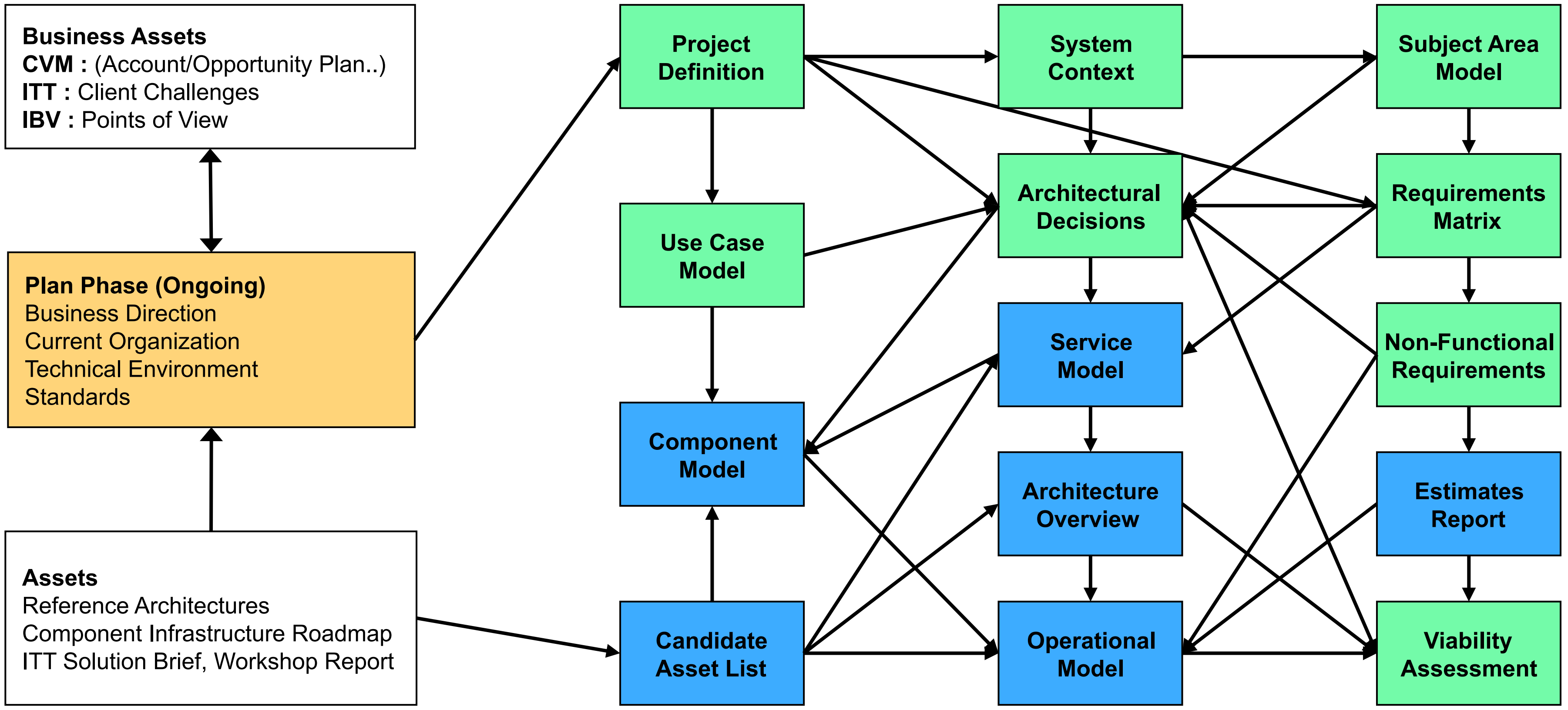
Defining and documenting the various aspects of the IT solution's requirements and design is achieved by using a set of IT Architecture work products, each focused on a specific view of the IT system

Separation of concerns



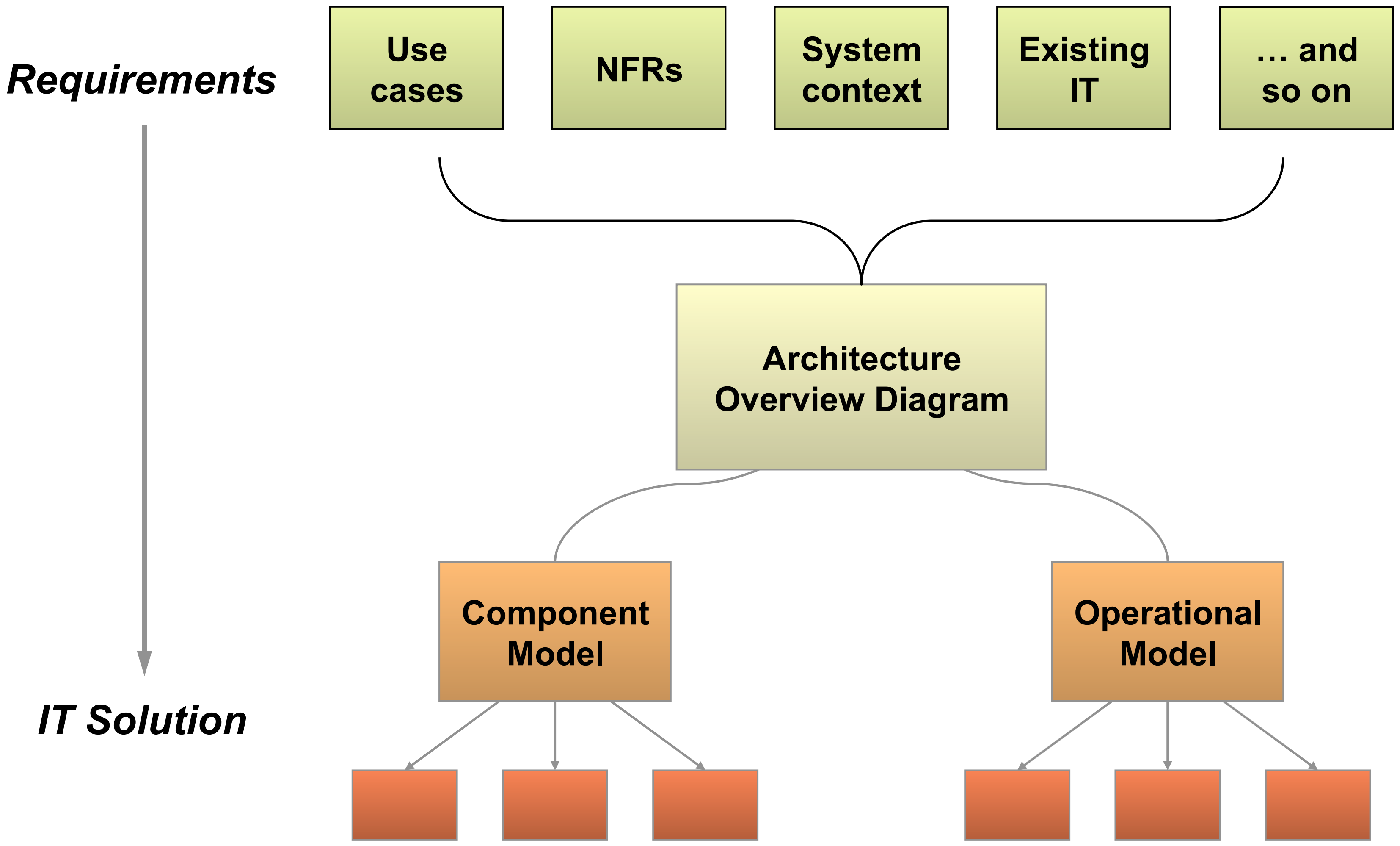
Source: IBM Architectural Thinking

IT Architect's Tool Box: Work Product Dependency Diagram – Work Products are the Artifacts of Architecture Work



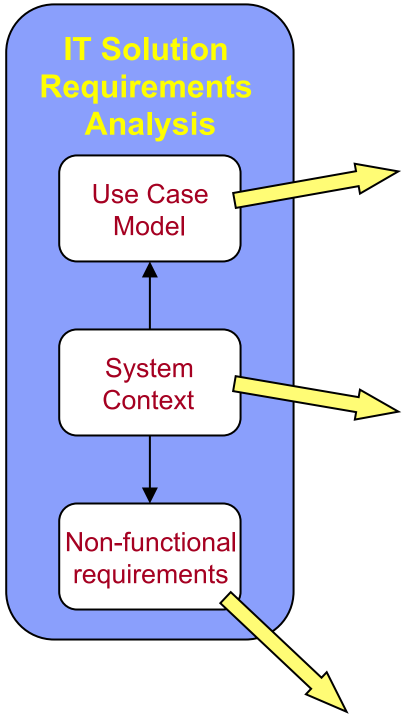
Source: IBM Architectural Thinking

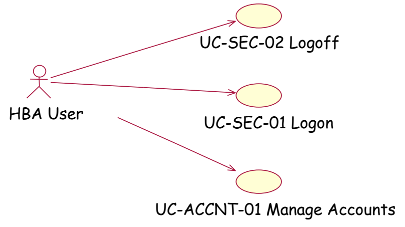
Overview Work Products – From Requirements to IT Solution



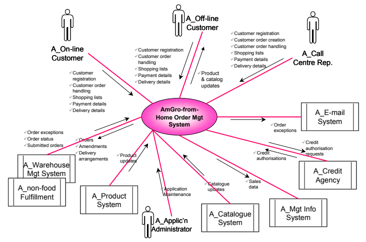
Source: IBM Architectural Thinking

The IT architect uses three core work products to document the business requirements their IT System will support…

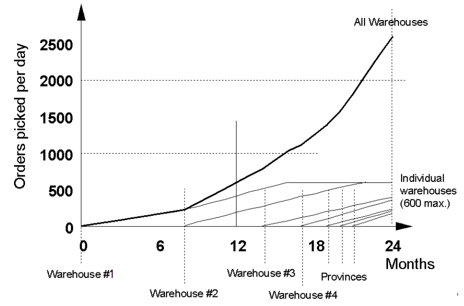

IT Solution Requirements Analysis
Use Case Model
System Context
Non-functional requirements



"**<u>Use Case Model</u>**" defines what the IT system must do for all its business (and other) users – it describes the system behaviour at the "automation boundary"



"**<u>System Context</u>**" defines the boundaries of the to be designed system, documenting the external users and other IT systems with which this system must interact



"**<u>Non functional requirements</u>**" describe the characteristics of the IT systems functionality, such as transaction response times and workload volumes

Source: IBM Architectural Thinking

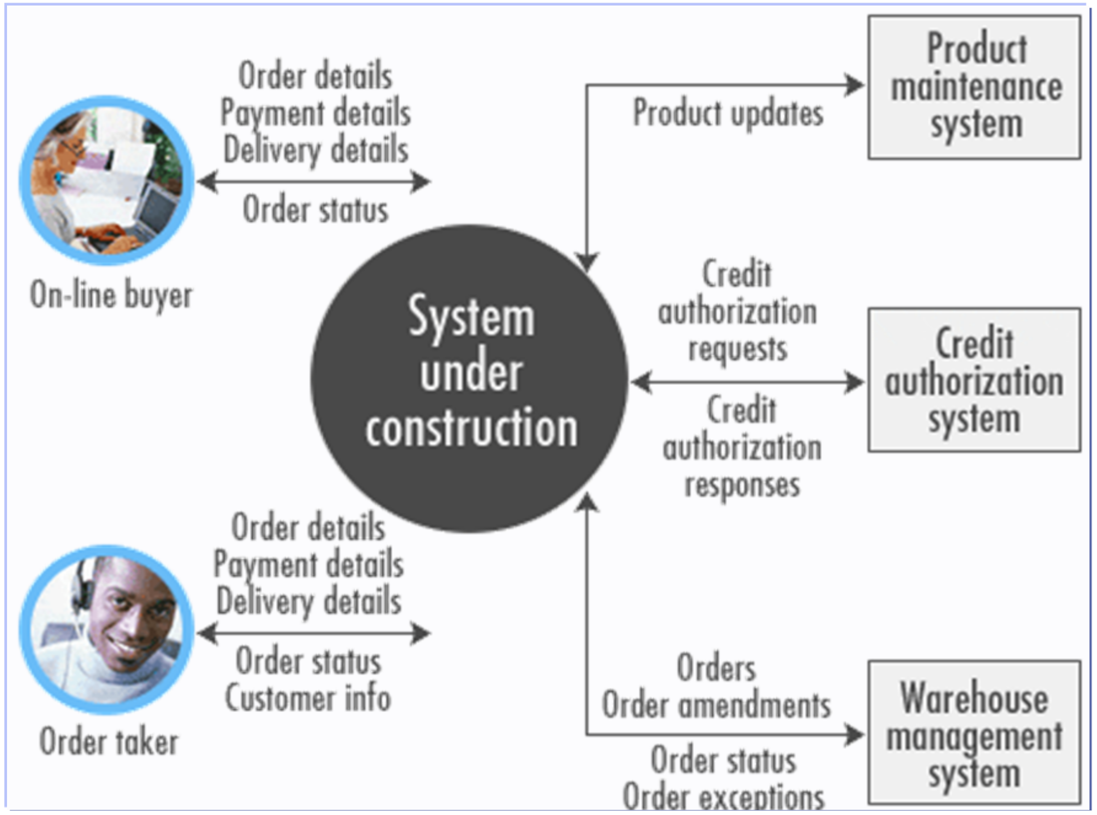
First Work Products (aka Artifacts)

- The three “C’s”
  - Context
  - Common Sense
  - Communication

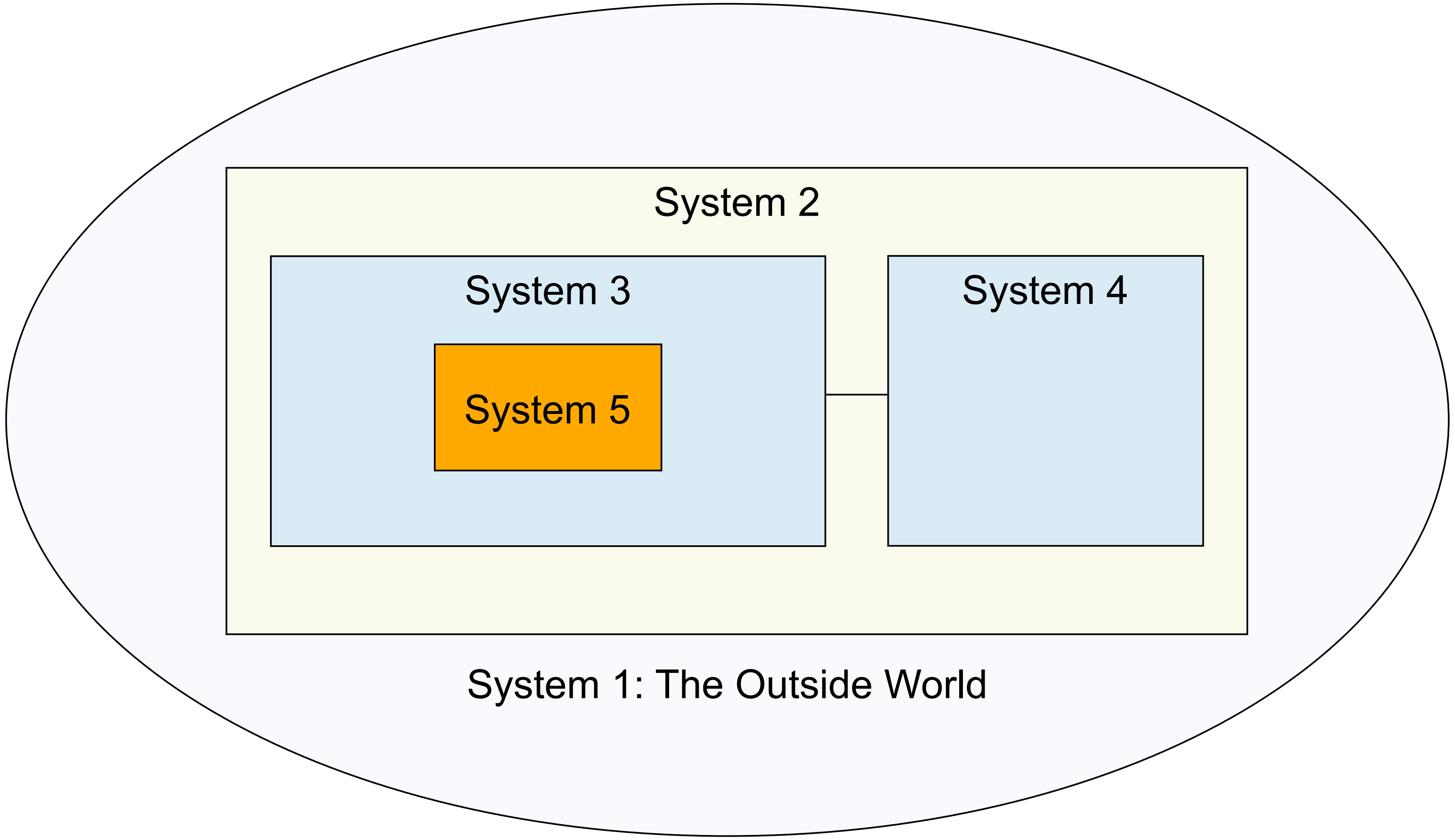
- Context Diagram
  - External view showing the surroundings of the area to be addressed by the new solution
  - Helps to clarify the scope and thus is an important part of a project definition
  - Addresses various stakeholders, can be used as Business Context as well as System Context

The (System) Context is essential to capturing the scope of the project

- The System Context helps to:
  – Clarify the environment in which the system has to operate
  – Put bounds on the system
  – Identify external interfaces (users or systems)



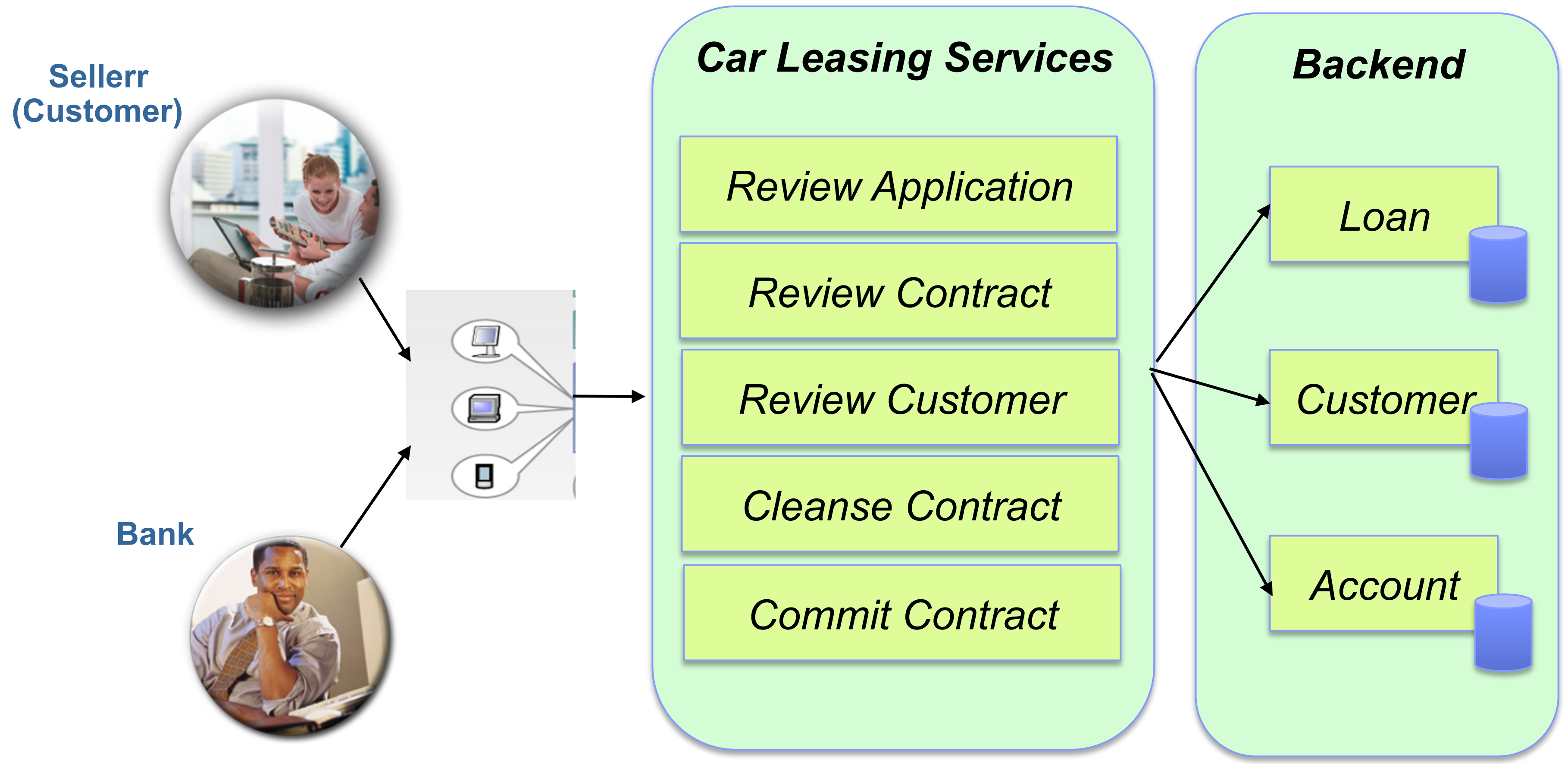
Finding the system boundaries



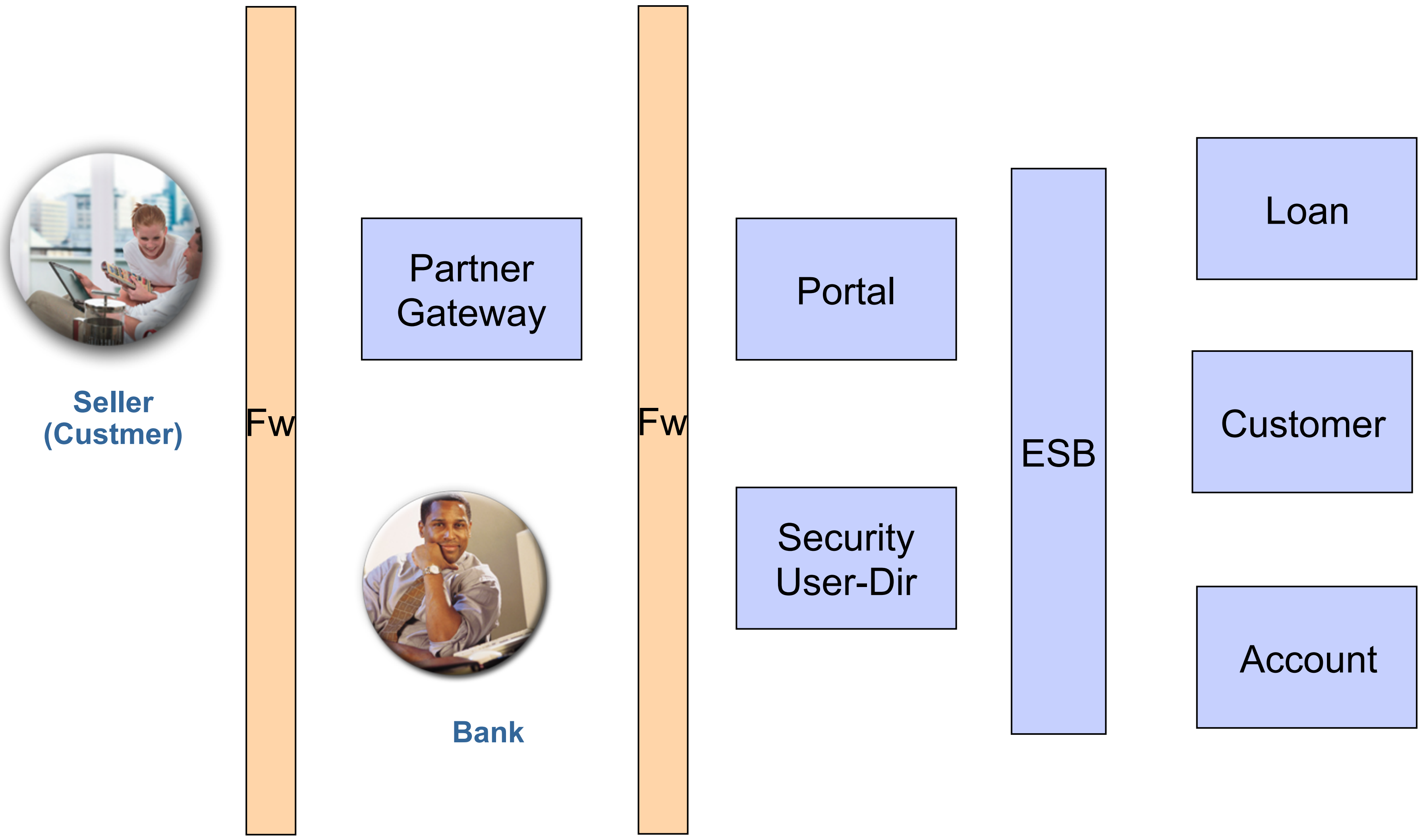
Architecture Overview Diagram (AOD) – Purpose

- To communicate to the sponsor and external stakeholders a conceptual understanding of the (intended) IT system (thus is a schematic diagram – not a model)
- To provide a high-level shared vision of the architecture and scope of the proposed IT system for the development teams (both functional and operational concepts)
- To explore and evaluate alternative architectural options
- To enable early recognition and validation of the implications of the architectural approach
- To facilitate effective communication between different communities of stakeholders and developers
- To facilitate orientation for new people who join the project

Example Car Leasing: Business-Oriented Architecture Overview



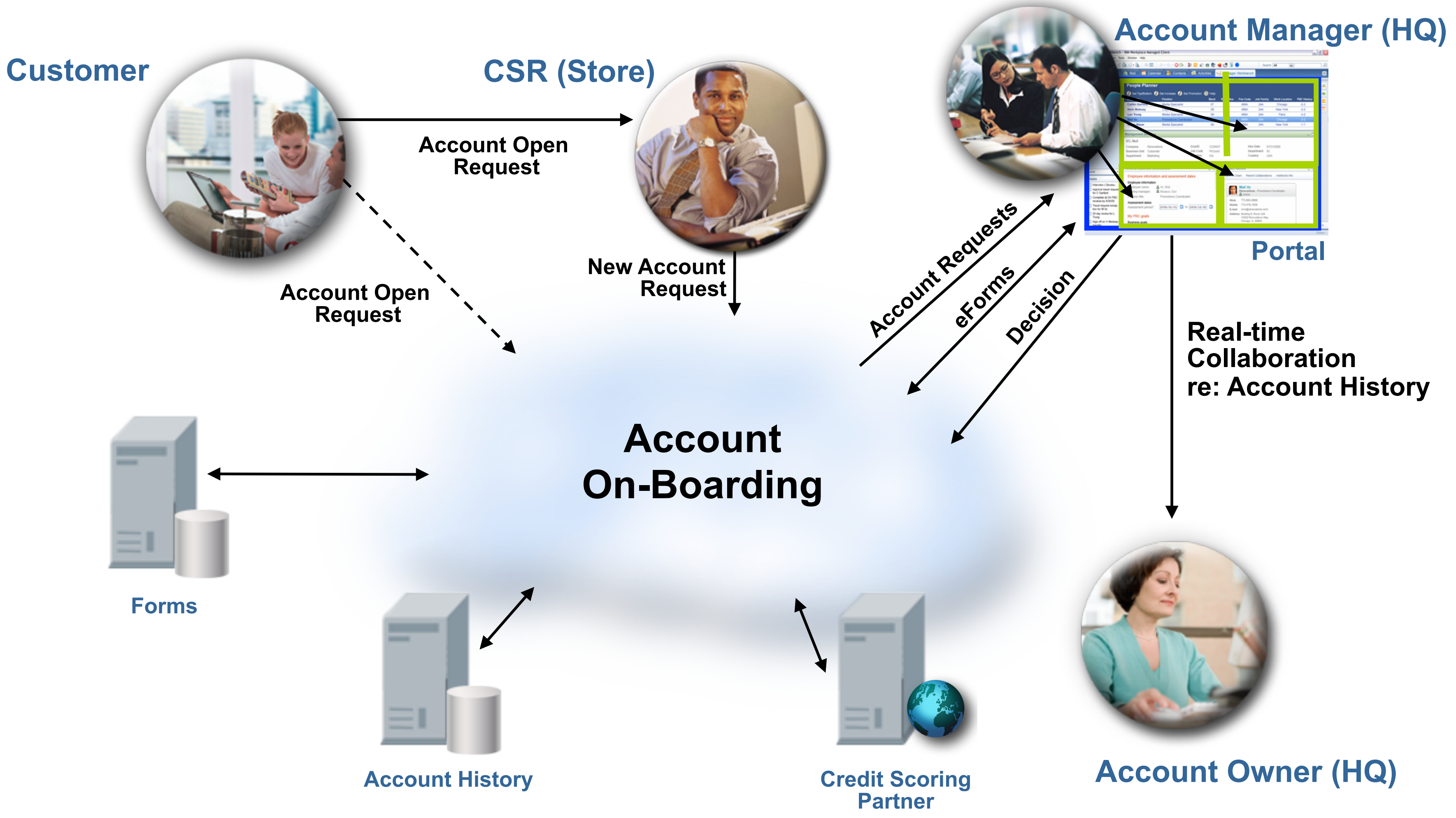
Example Car Leasing: IT-Oriented Architecture Overview



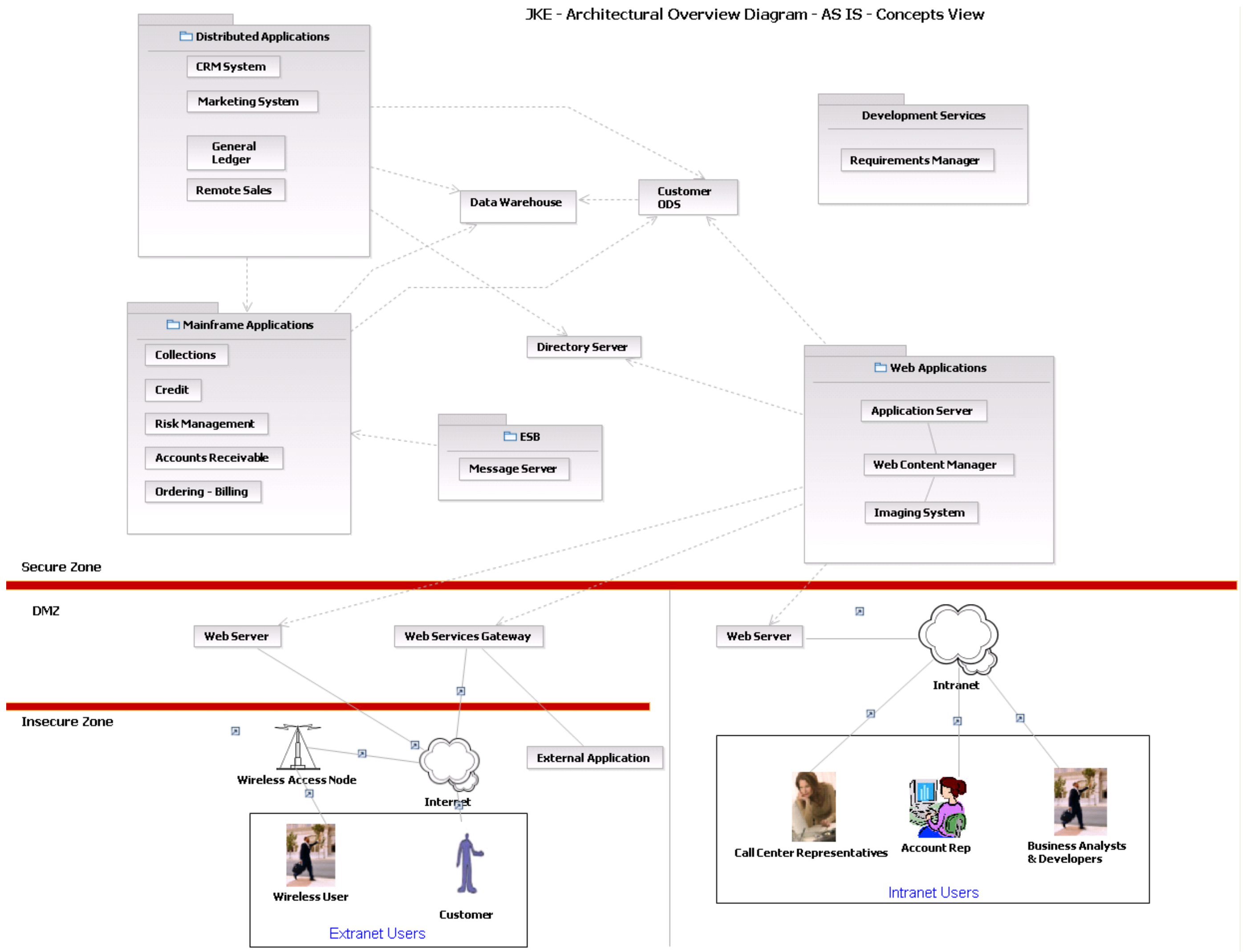
Example JKE: Account Opening – Fiction Case

- Situation:
  - A retail company wants to improve the “onboarding” of customers to their fidelity program and the possibilities to buy on credit
  - The company has many stores in various locations (as in Switzerland Jelmoli, Globus, …)
- Business Pain:
  - Loosing customers because the processing of a care leasing request takes 14 days (or even more)
  - Too much manual work e.g. for getting information about the eligibility of the customer

Example JKE: Context Diagram for Business Process "Open Account" (Solution Viewpoint)



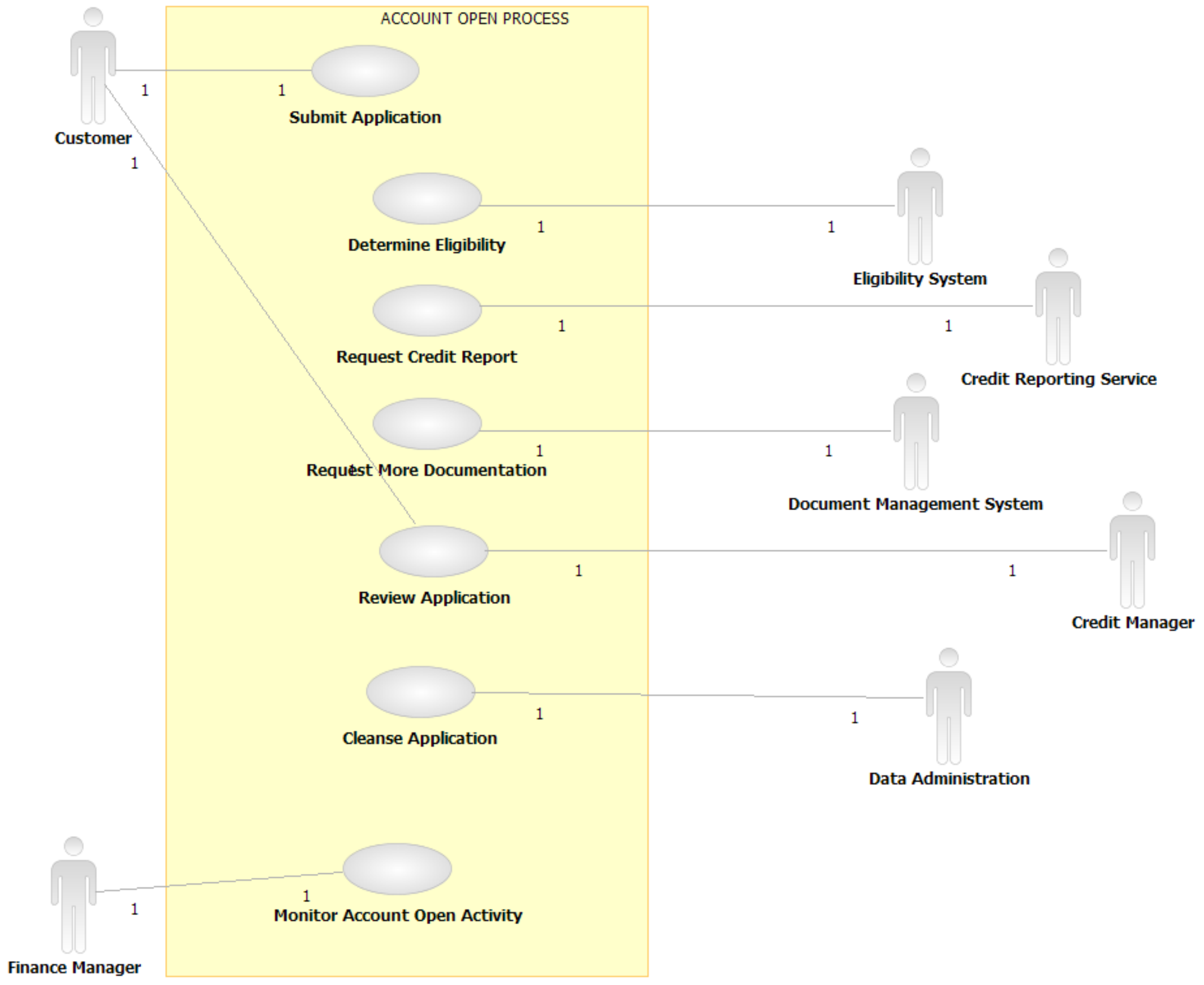
Example JKE: As-Is Architectural Overview Diagram



Functional Requirements

- Functional Requirements describe the business functions required, they can be derived form the Business Architecture
  - Requirements should be traced back to Business Goals
- Functional Requirements:
  - Are capabilities needed by users to fulfill their job
  - Answers the question of "what" does the customer want (but not "how" it is achieved)
- Note: Requirements Engineering is a discipline for Software Engineering

Example JKE: Use Case for JKE's "Open Account" Specifying Functional Requirements



Non-Functional Requirements

- Qualities:
  - Define the expectations and characteristics that the system should support
  - Might be runtime (for example, performance or availability) or non-runtime (for example, scalability or maintainability)

- Constraints:
  - Givens, those things that cannot be changed within the scope and lifetime of the project
  - Other factors, such as mandated technologies, available skills, and budget

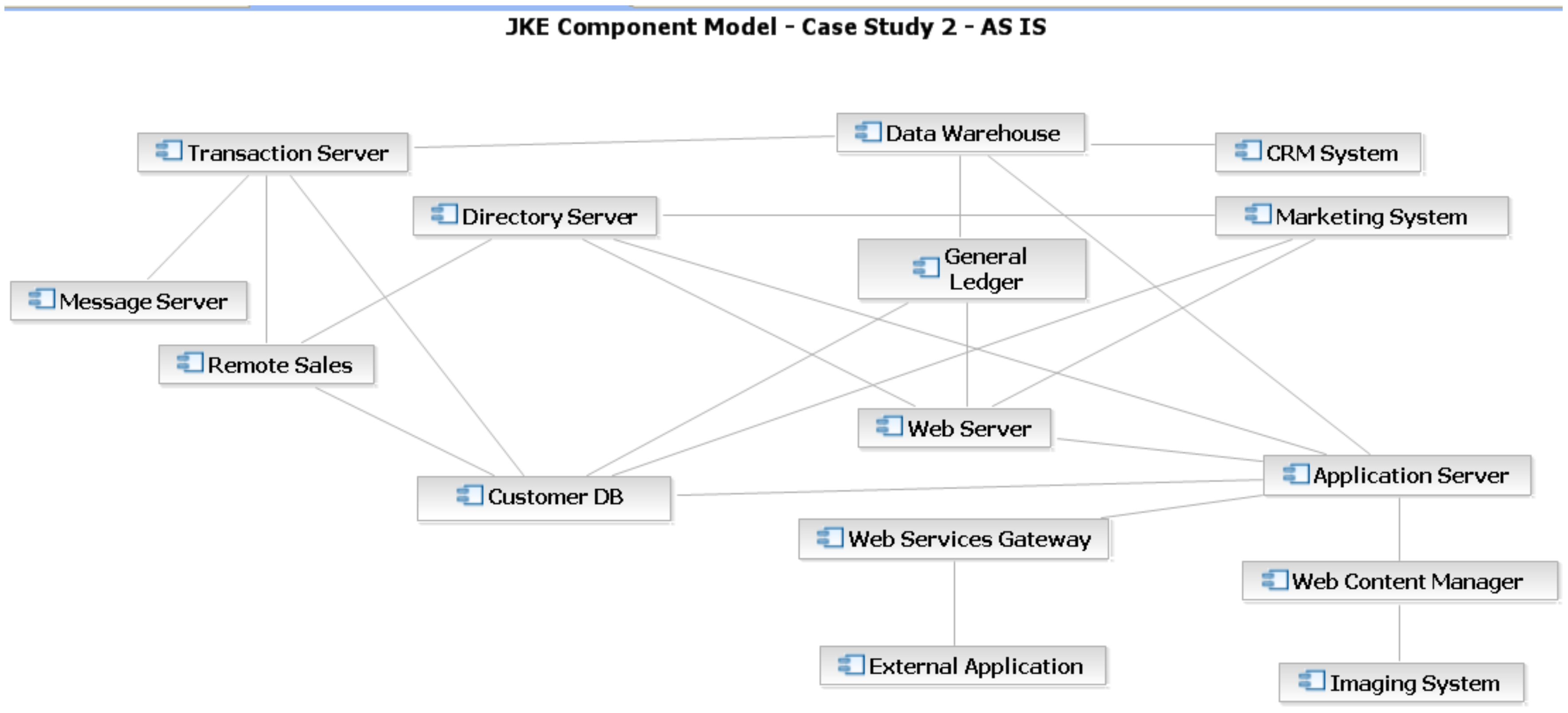
- Qualities and Constraints are sometimes referred to as “non-functional requirements”

Component Model

- **Components are defined with appropriate interfaces**

- **Dependencies**
  - **Relationships (can be shown in a relationship diagram)**
  - **Interaction**
  - **Collaboration**

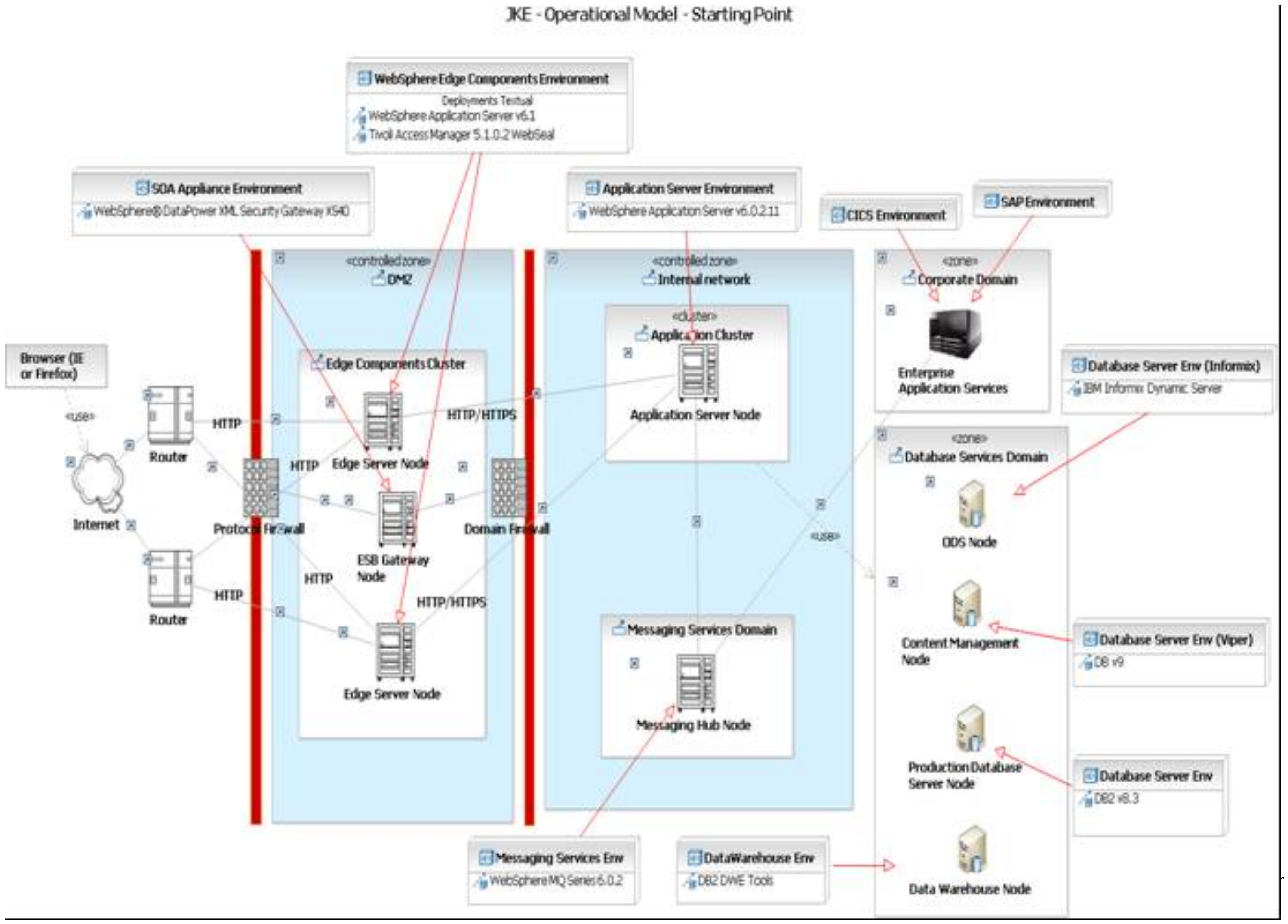
Example JKE: As-Is Component Model (JKE)



Operational Model

- Focus on how particular systems works
- Concerned with the systems that will run in production
- Includes guidelines for the physical design and types of hardware
  - Nodes
- Addresses non-functional requirements
  - Performance
  - Availability
  - Security (e.g. firewalls, zones)

Example JKE: As-Is Operational Model (JKE)



For all of those: Using Patterns and Reference Architectures (e.g. SOA Reference Architecture)

- **A Pattern is a reusable generalization (or abstraction) that can be used as the starting point in future solutions.**

- **The benefits of Patterns are that they:**
  - **Provide a mechanism to capture knowledge and experience**
  - **Provide a common vocabulary among architects and designers**
  - **Facilitate reuse of approaches that have been successful elsewhere; thus, contributing towards the following aspects of a project by:**
    - **Reducing risk**
    - **Increasing quality**
    - **Improving delivery time**

References

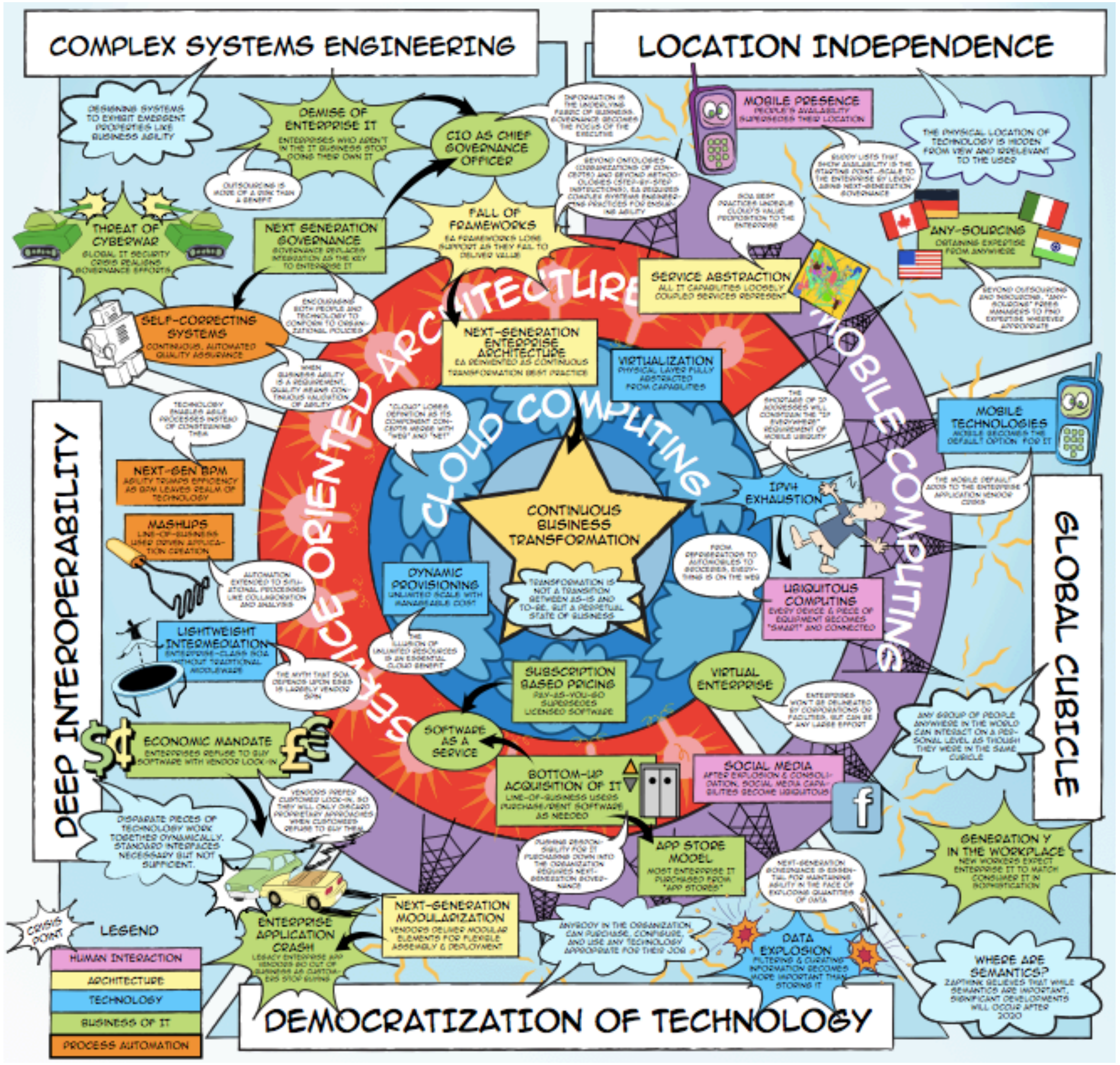
Overview Architecture Literature

- Plenty of books about
  - Architecture,
  - Enterprise Architecture,
- Typical problems:
  - they are very generic, introducing architecture in general (and the technology specific books – e.g. about SOA – are that much “architecture minded”)
  - they do not provide concrete means (e.g. advice which diagram to use) because some of them are pattern-specific
  - They are very lengthy
- Thus: I’m writing my own book (in German)

References – Books

- **Jason Bloomberg, The Agile Architecture Revolution, 2013, Wiley CIO Series, 280pp**
  - **Connects Service-Oriented Architecture and enterprise as a complex system**
  - **Considers Cloud Computing (does not work without architecture !) and REST (Representational State Transfer) for Service implementation**
  - **Advertises the ZapThink Vision 2020**
- **Marc Lankhorst et al, Enterprise Architecture at Work, 2nd edition, Springer, 2009, 350pp**
  - **Provides a good overview of frameworks**
  - **Is based on TOGAF and ARCHIMATE**
- **Wolfgang Keller, IT-Unternehmensarchitektur, dpunkt.verlag, 2012, (German) – emphasizes Governance, 400 pp**

ZapThink's Vision for Enterprise IT in 2020


COMPLEX SYSTEMS ENGINEERING
LOCATION INDEPENDENCE
DEEP INTEROPERABILITY
GLOBAL CUBICLE
DEMOCRATIZATION OF TECHNOLOGY
SERVICE ORIENTED ARCHITECTURE
CLOUD COMPUTING
MOBILE COMPUTING
CONTINUOUS BUSINESS TRANSFORMATION
TRANSFORMATION IS NOT A TRANSITION BETWEEN AS-IS AND TO-BE, BUT A PERPETUAL STATE OF BUSINESS
DESIGNING SYSTEMS TO EXHIBIT EMERGENT PROPERTIES LIKE BUSINESS AGILITY
DEMISE OF ENTERPRISE IT ENTERPRISES WHO AREN'T IN THE IT BUSINESS STOP DOING THEIR OWN IT
OUTSOURCING IS MORE OF A RISK THAN A BENEFIT
CIO AS CHIEF GOVERNANCE OFFICER
INFORMATION IS THE UNDERLYING FABRIC OF BUSINESS. GOVERNANCE BECOMES THE FOCUS OF THE EXECUTIVE
BEYOND ONTOLOGIES (ORGANIZATIONS OF CONCEPTS) AND BEYOND METHODOLOGIES (STEP-BY-STEP INSTRUCTIONS), EA REQUIRES COMPLEX SYSTEMS ENGINEERING PRACTICES FOR ENSURING AGILITY
THREAT OF CYBERWAR GLOBAL IT SECURITY CRISIS REALIGNS GOVERNANCE EFFORTS
NEXT GENERATION GOVERNANCE GOVERNANCE REPLACES INTEGRATION AS THE KEY TO ENTERPRISE IT
FALL OF FRAMEWORKS EA FRAMEWORKS LOSE SUPPORT AS THEY FAIL TO DELIVER VALUE
SELF-CORRECTING SYSTEMS CONTINUOUS, AUTOMATED QUALITY ASSURANCE
ENCOURAGING BOTH PEOPLE AND TECHNOLOGY TO CONFORM TO ORGANIZATIONAL POLICIES
NEXT-GENERATION ENTERPRISE ARCHITECTURE EA REINVENTED AS CONTINUOUS TRANSFORMATION BEST PRACTICE
WHEN BUSINESS AGILITY IS A REQUIREMENT, QUALITY MEANS CONTINUOUS VALIDATION OF AGILITY
TECHNOLOGY ENABLES AGILE PROCESSES INSTEAD OF CONSTRAINING THEM
"CLOUD" LOSES DEFINITION AS ITS COMPONENT CONCEPTS MERGE WITH "WEB" AND "NET"
NEXT-GEN BPM AGILITY TRUMPS EFFICIENCY AS BPM LEAVES REALM OF TECHNOLOGY
MASHUPS LINE-OF-BUSINESS USER DRIVEN APPLICATION CREATION
AUTOMATION EXTENDED TO SITUATIONAL PROCESSES LIKE COLLABORATION AND ANALYSIS
LIGHTWEIGHT INTERMEDIATION ENTERPRISE-CLASS SOA WITHOUT TRADITIONAL MIDDLEWARE
THE MYTH THAT SOA DEPENDS UPON ESBS IS LARGELY VENDOR SPIN
DYNAMIC PROVISIONING UNLIMITED SCALE WITH MANAGEABLE COST
THE ILLUSION OF UNLIMITED RESOURCES IS AN ESSENTIAL CLOUD BENEFIT
SUBSCRIPTION BASED PRICING PAY-AS-YOU-GO SUPERSEDES LICENSED SOFTWARE
SOFTWARE AS A SERVICE
ECONOMIC MANDATE ENTERPRISES REFUSE TO BUY SOFTWARE WITH VENDOR LOCK-IN
VENDORS PREFER CUSTOMER LOCK-IN, SO THEY WILL ONLY DISCARD PROPRIETARY APPROACHES WHEN CUSTOMERS REFUSE TO BUY THEM
DISPARATE PIECES OF TECHNOLOGY WORK TOGETHER DYNAMICALLY. STANDARD INTERFACES NECESSARY BUT NOT SUFFICIENT.
BOTTOM-UP ACQUISITION OF IT LINE-OF-BUSINESS USERS PURCHASE/RENT SOFTWARE AS NEEDED
PUSHING RESPONSIBILITY FOR IT PURCHASING DOWN INTO THE ORGANIZATION REQUIRES NEXT-GENERATION GOVERNANCE
APP STORE MODEL MOST ENTERPRISE IT PURCHASED FROM "APP STORES"
ENTERPRISE APPLICATION CRASH LEGACY ENTERPRISE APP VENDORS GO OUT OF BUSINESS AS CUSTOMERS STOP BUYING
NEXT-GENERATION MODULARIZATION VENDORS DELIVER MODULAR ELEMENTS FOR FLEXIBLE ASSEMBLY & DEPLOYMENT
ANYBODY IN THE ORGANIZATION CAN PURCHASE, CONFIGURE, AND USE ANY TECHNOLOGY APPROPRIATE FOR THEIR JOB
DATA EXPLOSION FILTERING & CURATING INFORMATION BECOMES MORE IMPORTANT THAN STORING IT
NEXT-GENERATION GOVERNANCE IS ESSENTIAL FOR MAINTAINING AGILITY IN THE FACE OF EXPLODING QUANTITIES OF DATA
WHERE ARE SEMANTICS? ZAPTHINK BELIEVES THAT WHILE SEMANTICS ARE IMPORTANT, SIGNIFICANT DEVELOPMENTS WILL OCCUR AFTER 2020
GENERATION Y IN THE WORKPLACE NEW WORKERS EXPECT ENTERPRISE IT TO MATCH CONSUMER IT IN SOPHISTICATION
SOCIAL MEDIA AFTER EXPLOSION & CONSOLIDATION, SOCIAL MEDIA CAPABILITIES BECOME UBIQUITOUS
VIRTUAL ENTERPRISE
ENTERPRISES WON'T BE DELINEATED BY CORPORATIONS OR FACILITIES, BUT CAN BE ANY LARGE EFFORT
ANY GROUP OF PEOPLE ANYWHERE IN THE WORLD CAN INTERACT ON A PERSONAL LEVEL AS THOUGH THEY WERE IN THE SAME CUBICLE
UBIQUITOUS COMPUTING EVERY DEVICE & PIECE OF EQUIPMENT BECOMES "SMART" AND CONNECTED
FROM REFRIGERATORS TO AUTOMOBILES TO BICYCLES, EVERYTHING IS ON THE WEB
IPV4 EXHAUSTION
THE SHORTAGE OF IP ADDRESSES WILL CONSTRAIN THE "IP EVERYWHERE" REQUIREMENT OF MOBILE UBIQUITY
VIRTUALIZATION PHYSICAL LAYER FULLY ABSTRACTED FROM CAPABILITIES
SERVICE ABSTRACTION ALL IT CAPABILITIES LOOSELY COUPLED SERVICES REPRESENT
SOA BEST PRACTICES UNDERLIE CLOUD'S VALUE PROPOSITION TO THE ENTERPRISE
MOBILE PRESENCE PEOPLE'S AVAILABILITY SUPERSEDES THEIR LOCATION
BUDDY LISTS THAT SHOW AVAILABILITY IS THE STARTING POINT...SCALE TO THE ENTERPRISE BY LEVERAGING NEXT-GENERATION GOVERNANCE
THE PHYSICAL LOCATION OF TECHNOLOGY IS HIDDEN FROM VIEW AND IRRELEVANT TO THE USER
ANY-SOURCING OBTAINING EXPERTISE FROM ANYWHERE
BEYOND OUTSOURCING AND INSOURCING, "ANY-SOURCING" FREES MANAGERS TO FIND EXPERTISE WHEREVER APPROPRIATE
MOBILE TECHNOLOGIES MOBILE BECOMES THE DEFAULT OPTION FOR IT
THE MOBILE DEFAULT ADDS TO THE ENTERPRISE APPLICATION VENDOR CRISIS
CRISIS POINT
LEGEND
HUMAN INTERACTION
ARCHITECTURE
TECHNOLOGY
BUSINESS OF IT
PROCESS AUTOMATION

References – Actual
(ZIP-File – will be available for Download)

- What do IT Architects do all day? IBM, 2006
- TOGAF 9.1 Reference Card, 2011, 2pp
- Object Management Group, Model Driven Architecture (MDA), MDA Guide rev. 2.0 OMG Document ormsc/2014-06-01, 15pp
- ZapThink's Vision for Enterprise IT 2020

References – History (ZIP-File – will be available for Download)

- J.A. Zachman, A framework for information systems architecture, IBM Systems Journal, 1987
- M. Maier, D. Emery, R. Hilliard: Software Architecture - Introducing IEEE Standard 1471
- IEEE Standard 1471-2000, Recommended Practice for Architectural Description of Software-Intensive Systems, 2000
- Rechtin, Appendix A: Heuristics for systems level architecting
- R. Youngs et al., A standard for architecture description, IBM Systems Journal, 1999
- M. R. McBride, The software architect, Comm. ACM, May 2007

Just remember – the future might bring more than you think

"I think there is a world market
for maybe five computers."
Thomas Watson, chairman of IBM, 1943

"Computers in the future may weigh
no more than 1.5 tons. "
Popular Mechanics, 1949

"There is no reason anyone would
want a computer in their home. "
Ken Olsen, founder of DEC,1977

"Prediction is difficult, especially
about the future"
Niels Bohr, 1957

"640K ought to be enough
for anybody. "
Bill Gates, 1981

Questions