



Enterprise Architecture Frameworks

Prof. Dr. Knut Hinkelmann



ISO/IEC/IEEE 42010 Systems and Software Engineering — Architecture Description

- International standard for architecture descriptions of systems and software.
- The original IEEE 1471 specified requirements on the contents of **architecture descriptions** of systems.
 - ◆ An architecture description (AD) expresses the architecture of a system of interest
- ISO/IEC/IEEE 42010 adds definitions and requirements on **architecture frameworks** and **architecture description languages (ADLs)**

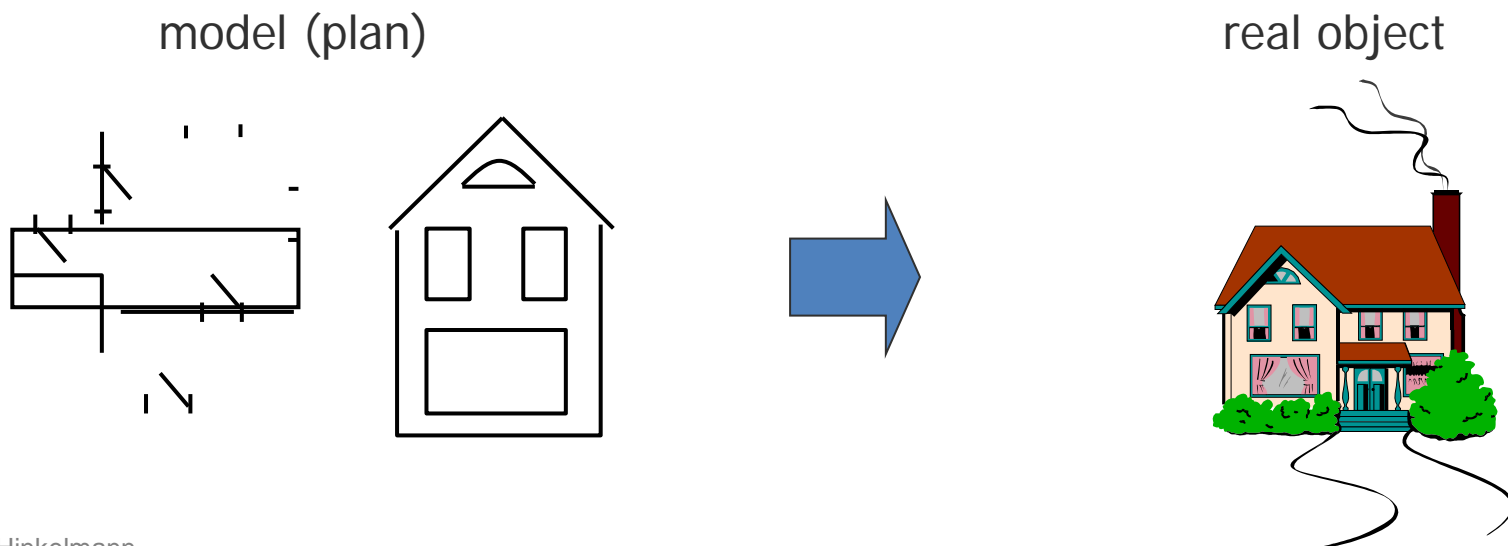
Architecture and Architecture Description

- An architecture is a *conception of a system* – i.e., it is in the human mind. An architecture may exist without ever being written down.
- An *architecture description* (AD) is an artifact that expresses an Architecture to share with others.
 - ◆ An AD is what is written down as a concrete work product. It could be a document, a repository or a collection of artifacts used to define and document an architecture
 - ◆ Architects and other system stakeholders use Architecture Descriptions to understand, analyze and compare Architectures, and often as "blueprints" for planning and construction.

<http://www.iso-architecture.org/ieee-1471/cm/>

Architecture Description and Architecture Models

- An *Architecture Description* consists of one or several *Architecture Models*
- A Model is a reproduction of a *relevant* part of reality which contains the essential aspects to be investigated.
- Relevance depends on stakeholders and their concerns.



Architecture Models and Model Kinds

- An *Architecture Model* represents a specific part of an Architecture, i.e. a *View on the Architecture*
 - ◆ Examples of Models:
 - The model of the order process of the company,
 - the organisation structure of a specific company
 - the model of the customer data,
- A *Model Kind* defines the concepts and relations needed to model Architecture Views.
 - ◆ Examples of Model Kinds:
 - process models
 - organisation models
 - data models

<http://www.iso-architecture.org/ieee-1471/cm/>

Architecture Views and Viewpoints

- Views and Viewpoints are a means to specify which part of an Architecture Description is of relevance
 - ◆ *View*: Part of an architecture description, which is relevant for
 - one or more *Stakeholders*
 - to address specific *Concerns*
 - ◆ *Viewpoint* specifies a view
 - a characterisation of stakeholders and their concerns
 - the viewpoint determines the concepts, relationships, models, and visualizations that should be provided by the view

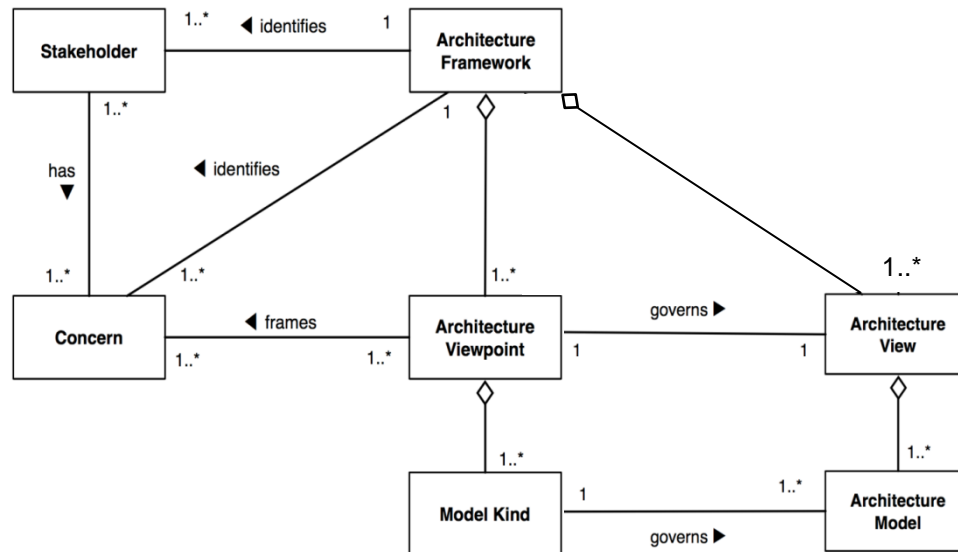
*A view is what you see and
a viewpoint is where you are looking from*

Source: ArchiMate 2.0 Specification, chapter 8, <http://pubs.opengroup.org/architecture/archimate2-doc/chap08.html>



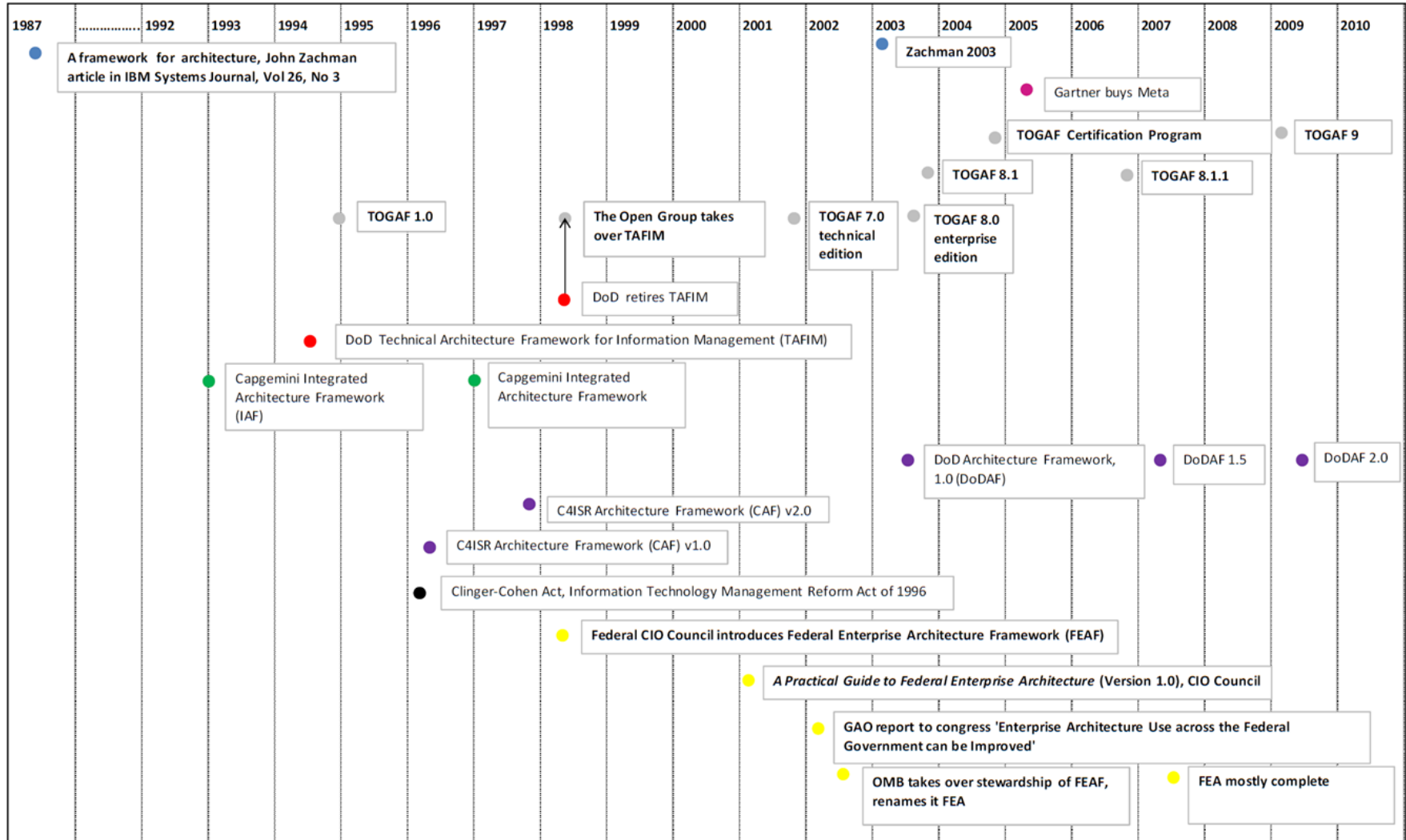
Architecture Framework

- An *Architecture Framework* establishes a common practice for creating, interpreting, analyzing and using architecture descriptions (Views and Viewpoints) within a particular domain of application or stakeholder community.
- It is a structure for the content of an Enterprise Architecture Description



<http://www.iso-architecture.org/ieee-1471/cm/>

Timeline of Enterprise Architecture Frameworks



(Bespoke Systems 2012)

Enterprise Architecture Frameworks

- There are a number of Enterprise Architecture Frameworks
- We can distinguish two main types of structures:

- ◆ **Matrix** of aspects and perspectives, e.g.

- **Zachmann Enterprise Architecture Framework**

- An enterprise ontology

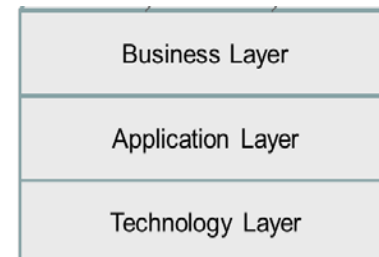
- ◆ **Three layer architecture** with business, applications and technology, e.g.

- **TOGAF - The Open Group Architecture Framework**

- A methodology for architecture development

- **ArchiMate – A modeling language for EA**

- **Best Practice Enterprise Architecture**



The Zachman Framework

Zachman Framework

- Regarded the origin of enterprise architecture frameworks (originally called "Framework for Information Systems Architecture")
- First version published in 1987 by John Zachman
- It is still further developed by Zachman International (<http://www.zachman.com>)
- Often referenced as a standard approach for expressing the basic elements of enterprise architecture
- The framework is a logical structure for classifying and organising the descriptive representations of an enterprise

Zachman, J.A., 1987. A framework for information systems architecture. *IBM Systems Journal*, 26(3).

Rationale of the Zachman Architecture

- There is not a single descriptive representation for a complex object ... there is a SET of descriptive representations.
- Descriptive representations (of anything) typically include:
 - ◆ Perspectives
 - ◆ Abstractions



(Zachman 2012)

Dimension 1 – Perspectives

Zachman originally used the analogy of classical architecture

For the different stakeholders different aspects of a building are relevant - models of the building from different perspectives

Bubble charts: conceptual representation delivered by the architect

Architect's drawing: transcription of the owner's perceptual requirements – *owner's perspective*

Architect's plans: translation of the owner's requirements into a product – *designer's perspective*

Contractor's plans: phases of operation, architect's plans constrained by nature and technology – *builder's perspective*

Shop plans: parts/sections/components of building details (out-of-context specification) – *subcontractor's perspective*

The building: physical building itself

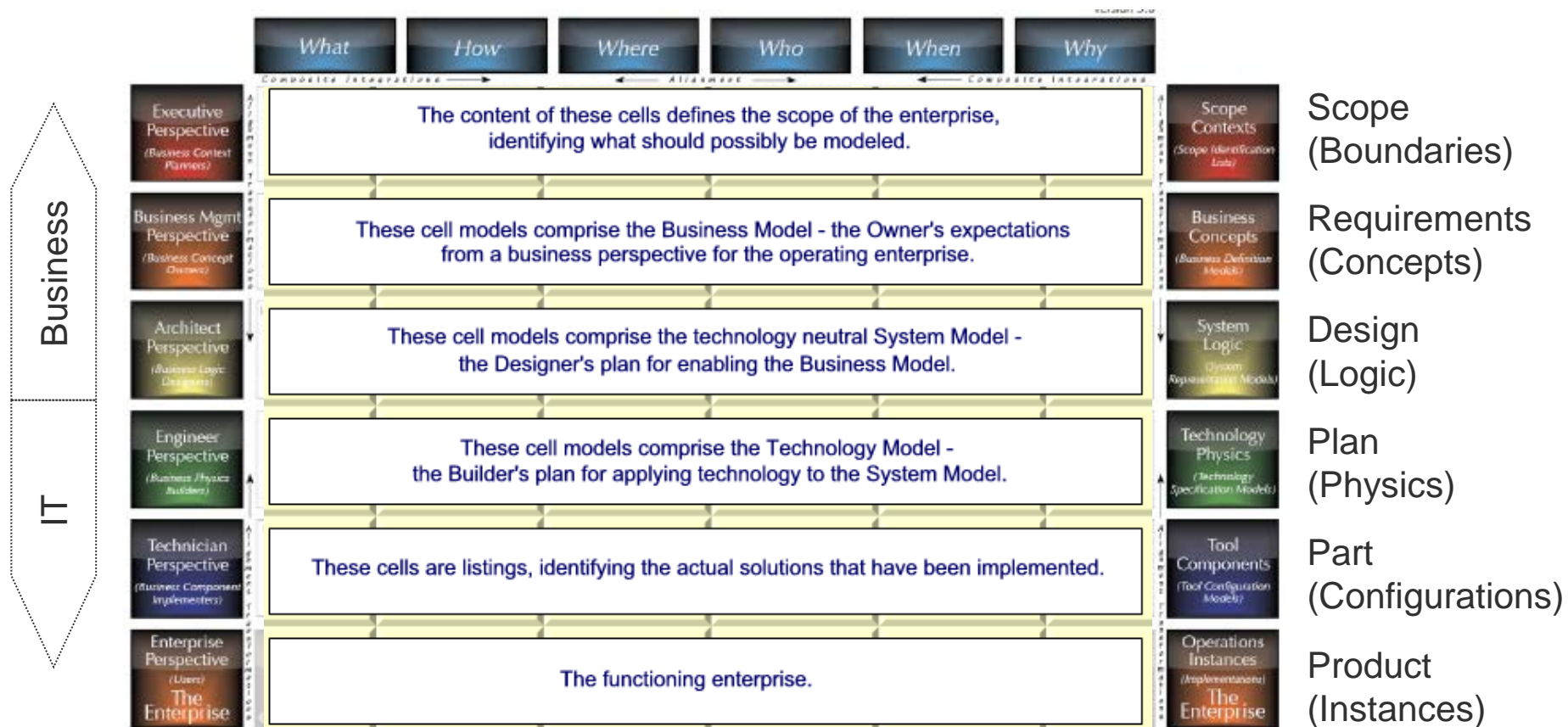
(Zachman 1987)

Dimension 1: Architectural Representations with analogies in Building and Information Systems

Generic	Buildings	Information Systems
Ballpark	Bubble charts	Scope/objectives
Owner's representation	Architect's drawings	Model of the business (or business description)
Designer's representation	Architect's plans	Model of the information system (or information system description)
Builder's representation	Contractor's plans	Technology model (or technology-constrained description)
Out-of-context representation	Shop plans	Detailed description
Machine language representation	—	Machine language description (or object code)
Product	Building	Information system

(Zachman 1987)

Perspectives



- Each row is different in nature, in content, in semantics from the others – representing different perspectives
- Representations do not correspond to different levels of details – level of detail is an independent variable, varying within one representation

Dimension 2: Aspects of an Architecture

- There exist different types of descriptions oriented to different aspects
- Zachman associates each aspect with a question word
 - WHAT inventory models
 - HOW functional/process models
 - WHERE location/distribution models
 - WHO organisation models
 - WHEN timing models
 - WHY motivation models

(Zachman 1987)

Abstractions for Manufacturing



(Zachman 2012)

The Zachman Framework for Enterprise Architecture

- Enterprise Ontology

Abstractions/Aspects

Each cell contains models

Perspectives

Business
Information Technology



© 1987-2011 John A. Zachman, all rights reserved. Zachman® and Zachman International® are registered trademarks of John A. Zachman

The Zachman Framework is not a Methodology

ONTOLOGY

The Zachman Framework™ schema technically is an ontology -
a theory of the existence of a structured set
of essential components of an object
(the object being an Enterprise, a department, a value chain,
a "sliver," a solution, a project,
an airplane, a building, a bathtub or whatever or whatever).

A Framework is a STRUCTURE.
(A Structure DEFINES something.)

METHODOLOGY

A Methodology is a PROCESS.
(A Process TRANSFORMS something.)

A Structure IS NOT A Process
A Process IS NOT a Structure.

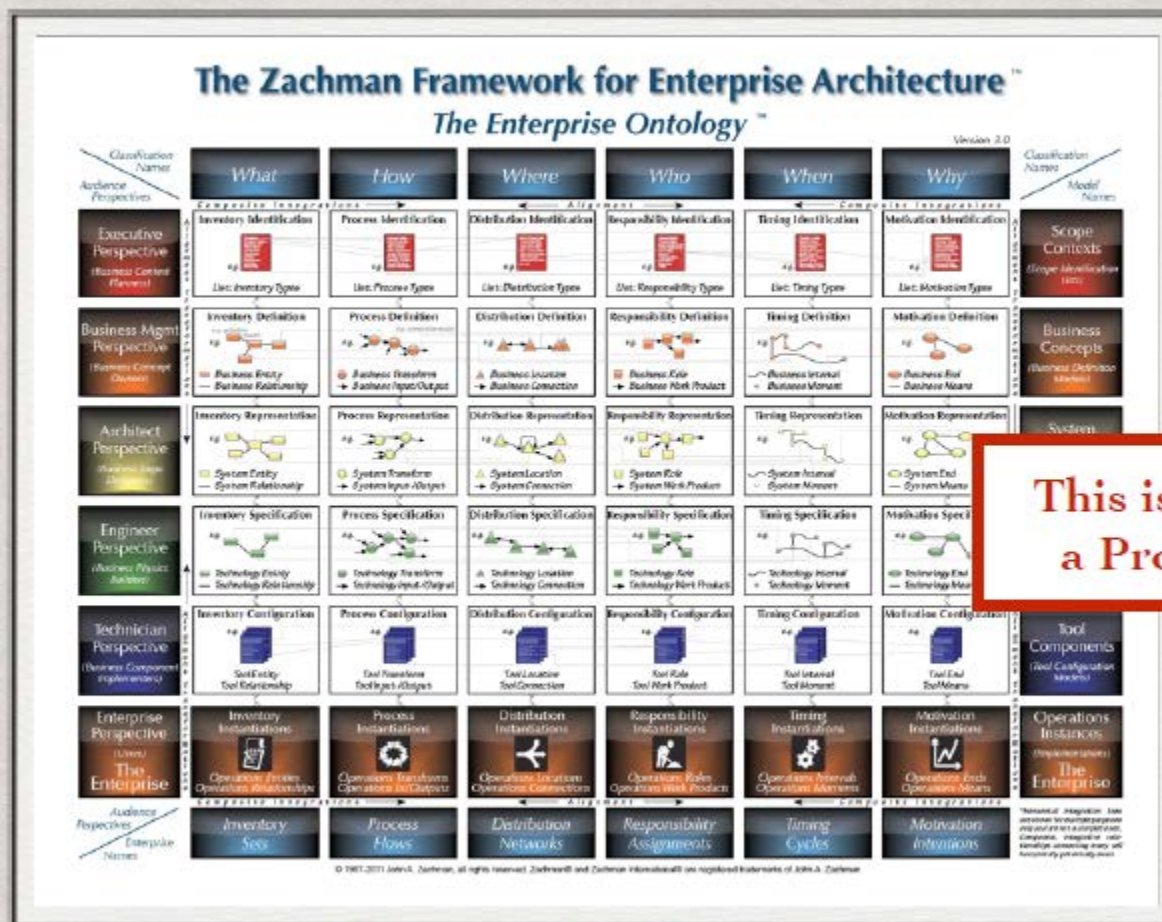
© 1990-2011 John A. Zachman, Zachman International®

ONTOLOGY VS METHODOLOGY

An Ontology is the classification of the total set of “**Primitive**” (elemental) components that exist and that are relevant to the existence of an object.

A Methodology produces “**Composite**” (compound) implementations of the Primitives.

ONTOLOGY

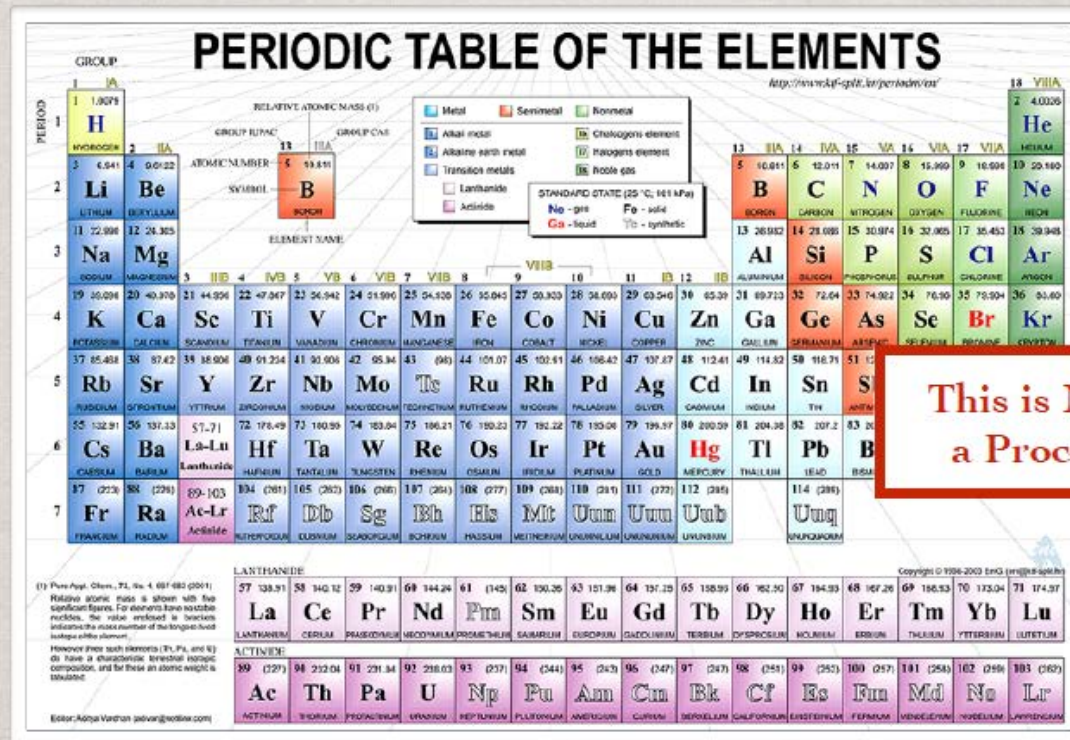


This is NOT a Process.

“Primitives” are Timeless.

Analogy: Chemistry

ONTOLOGY



Elements are Timeless

Until an ontology exists, nothing is repeatable, nothing is predictable.

There is no DISCIPLINE.

© 2012 John A. Zachman, Zachman International®



Analogy: Chemistry

PROCESS (METHODOLOGY)

Add Bleach to an Alkali and
it is transformed into Saltwater.



COMPOUNDS

Salt	NaCl
Aspirin	C ₉ H ₈ O ₄
Vicodin	C ₁₈ H ₂₁ NO ₃
Naproxen	C ₁₄ H ₁₄ O ₃
Ibuprophen	C ₁₃ H ₁₈ O ₂
Viagra	C ₂₂ H ₃₀ N ₆ O ₄ S
Sulphuric Acid	H ₂ SO ₄
Water	H ₂ O

etc., etc., etc.



This is NOT an
Ontology.

Model and the Zachman Framework

- Concepts for modelling are related to cells.
- Models are composites, they can roughly be assigned to cells, if they are composed of elements (concepts) of this cell.
- The elements of models can (roughly) be assigned to cells, but often cover



Viewpoints in Zachmann

- A viewpoint is characterized by the row, the column and the domain, e.g. the «How» and «Who» of the Sales Processing from the Architect Perspective.
- There can be different model kinds for a viewpoint
- There can be different modeling languages to represent a kind of model



Data model

- UML class
- ERM

IT model

- IT systems

Organisation model

- org chart
- Interaction

Motivation Model

- BMM
- ArchiMate Ext.

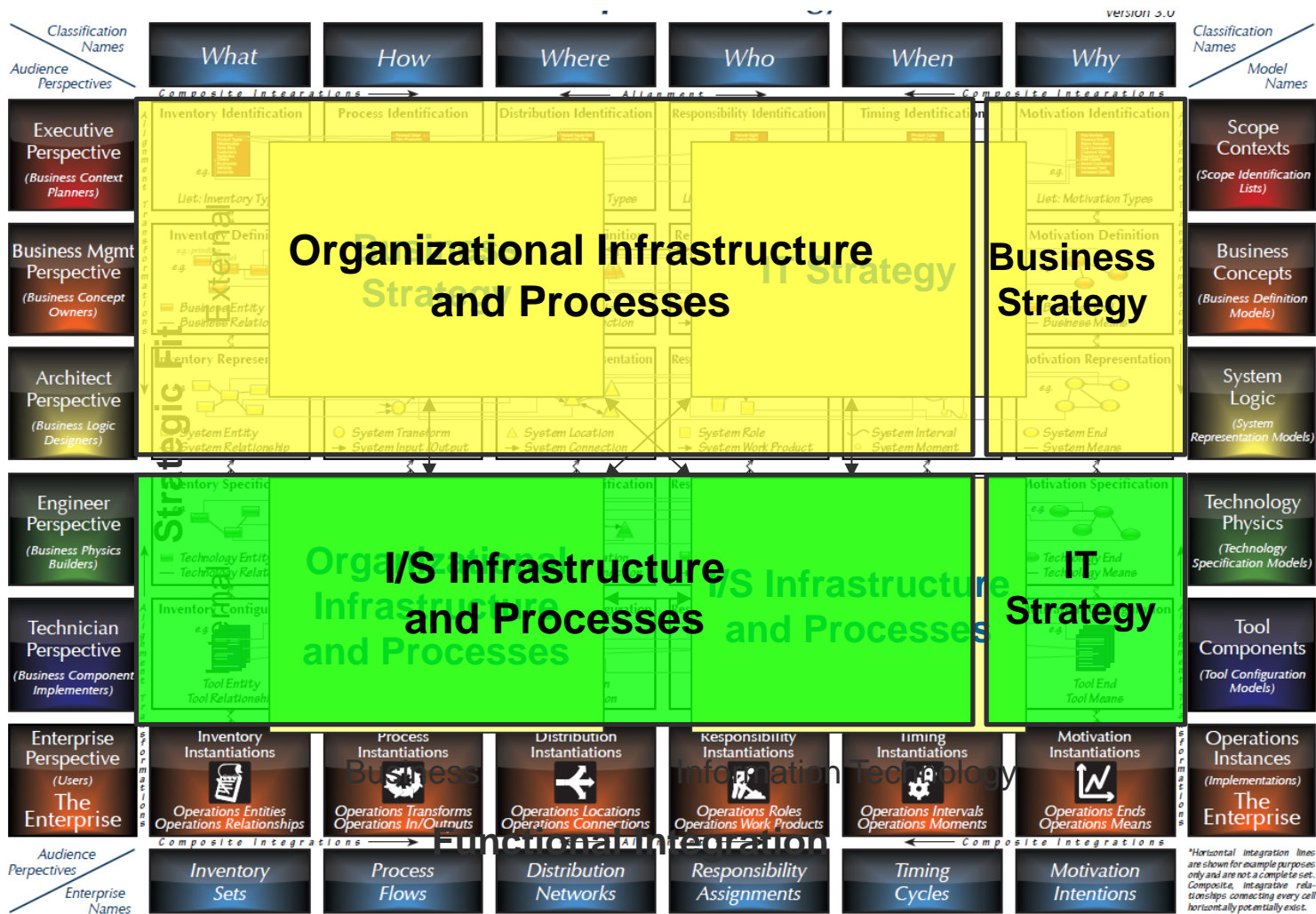
Process model

- Flow diagram
- BPMN
- Petri Net

Workflow model

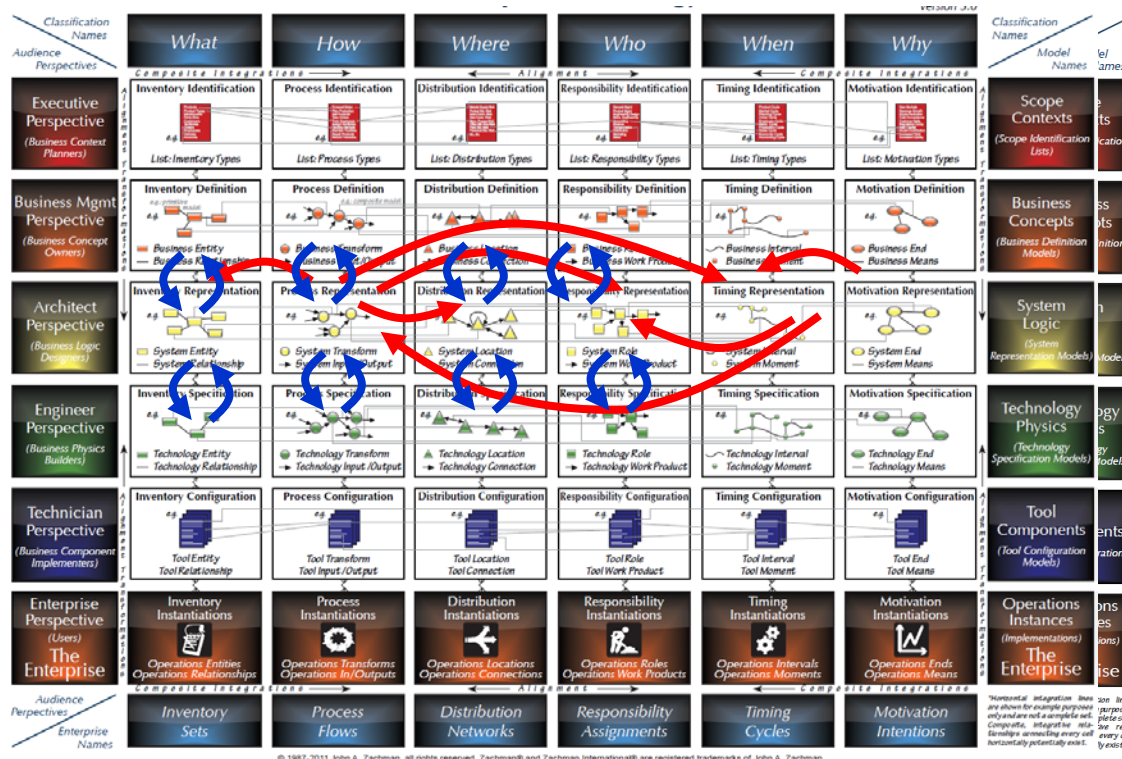
- BPEL

Strategic Alignment Model and Zachman Framework



Relations between Models and Model Elements (Composites)

- There are relations between (elements of) the models



Horizontal Relations: In same perspective, e.g.

- Data used in a process
- Application implementing a process activity

Vertical relations: Between different perspectives

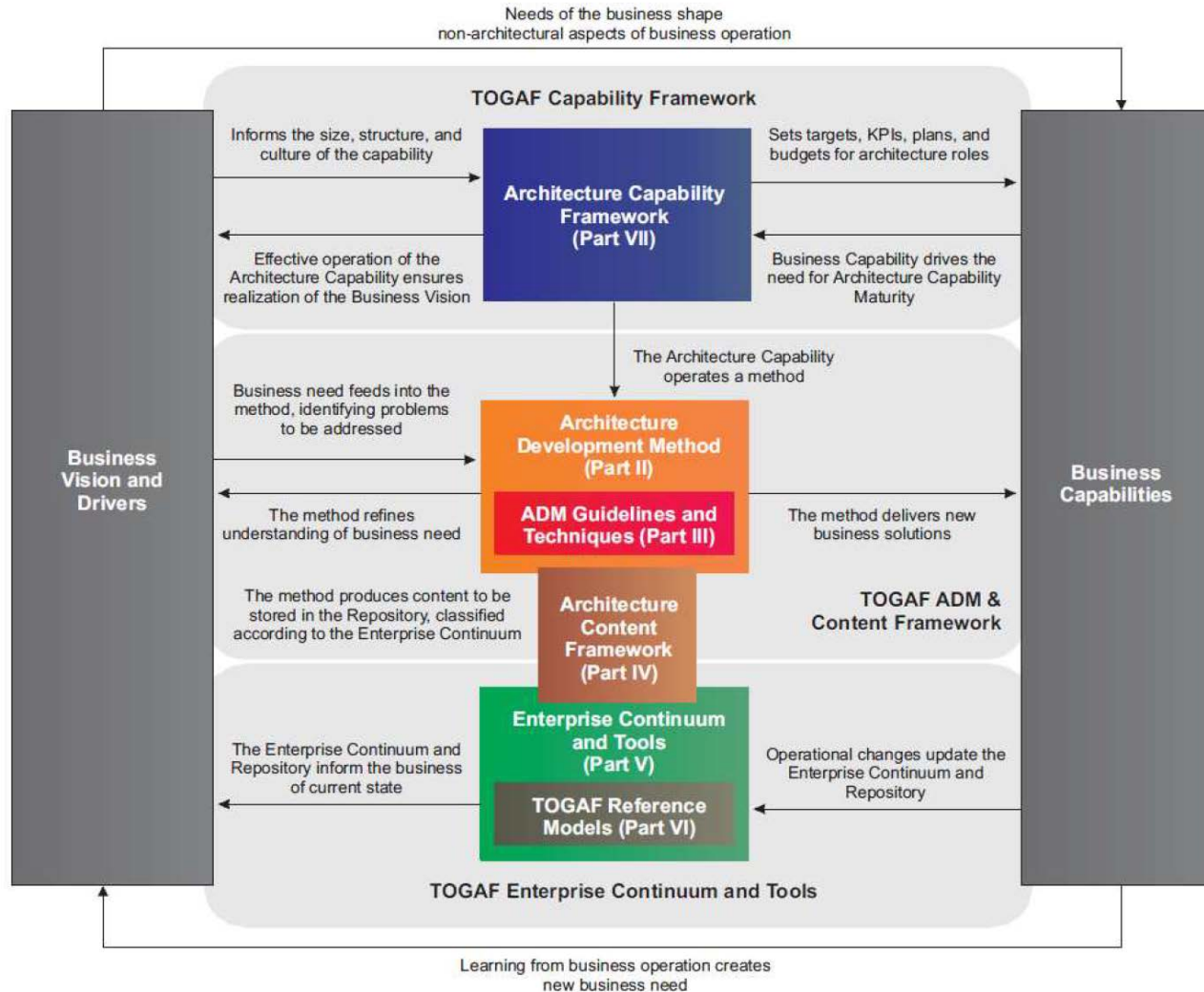
- Implementation of an application
- Database model for an entity relationship model

TOGAF – The Open Group Architecture Framework

TOGAF – The Open Group Architecture Framework

- Developed and continuously evolved since the mid-90's by The Open Group's Architecture Forum
- While Zachman is more an ontology, TOGAF is a methodology
- At the heart of the framework is the Architecture Development Method (ADM)
- <http://www.opengroup.org/togaf/>

Structure of the TOGAF Document



(The Open Group 2009, p. 4)

TOGAF Architecture Views

Business Architecture

Data
Architecture

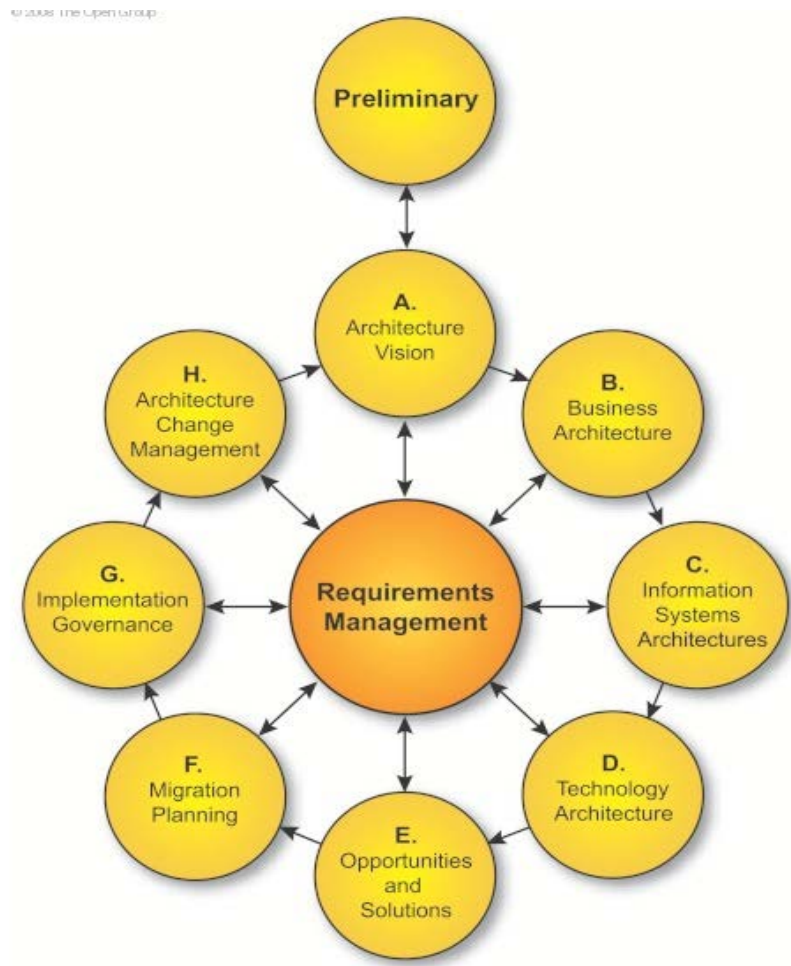
Application
Architecture

Technology Architecture

The TOGAF enterprise architecture model is organised in four partial sub-architectures:

- ◆ **Business Architecture**
 - Strategies, governance, organisation and business processes of the enterprise
- ◆ **Information Systems Architecture** – consists of
 - **Data Architecture**
 - data and their relations as well as principles for the organisation and the management of resources
 - **Application Architecture**
 - information systems and their relations to business processes
- ◆ **Technology Architecture**
 - current technical realisation and future enterprise-specific standards like operating system, middleware, infrastructure

TOGAF Architecture Development Method (ADM)

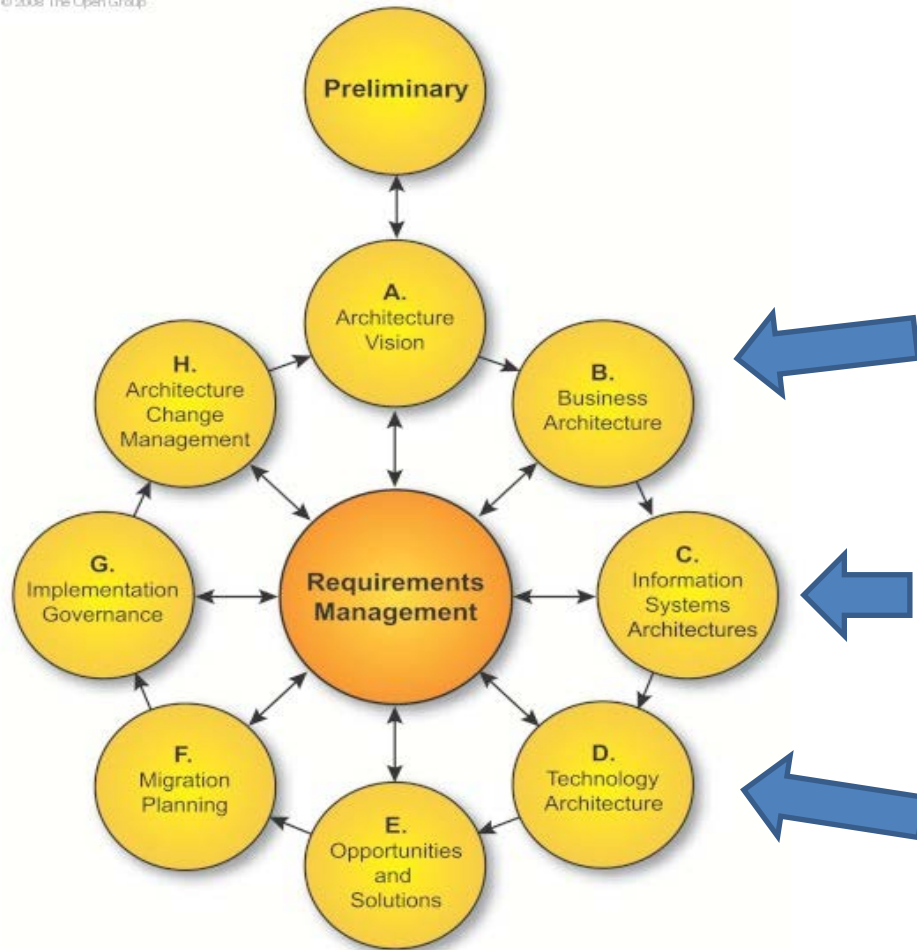


- TOGAF addresses the whole enterprise architecture lifecycle
- The TOGAF Architecture Development Method (ADM) is a generic method for developing an enterprise architecture
- The goals, approaches, required input, activities and deliverables are documented for each phase separately
- The ADM method is enriched by specific ADM guidelines and techniques.

(The Open Group 2009)

TOGAF Architecture Views

© 2008 The Open Group



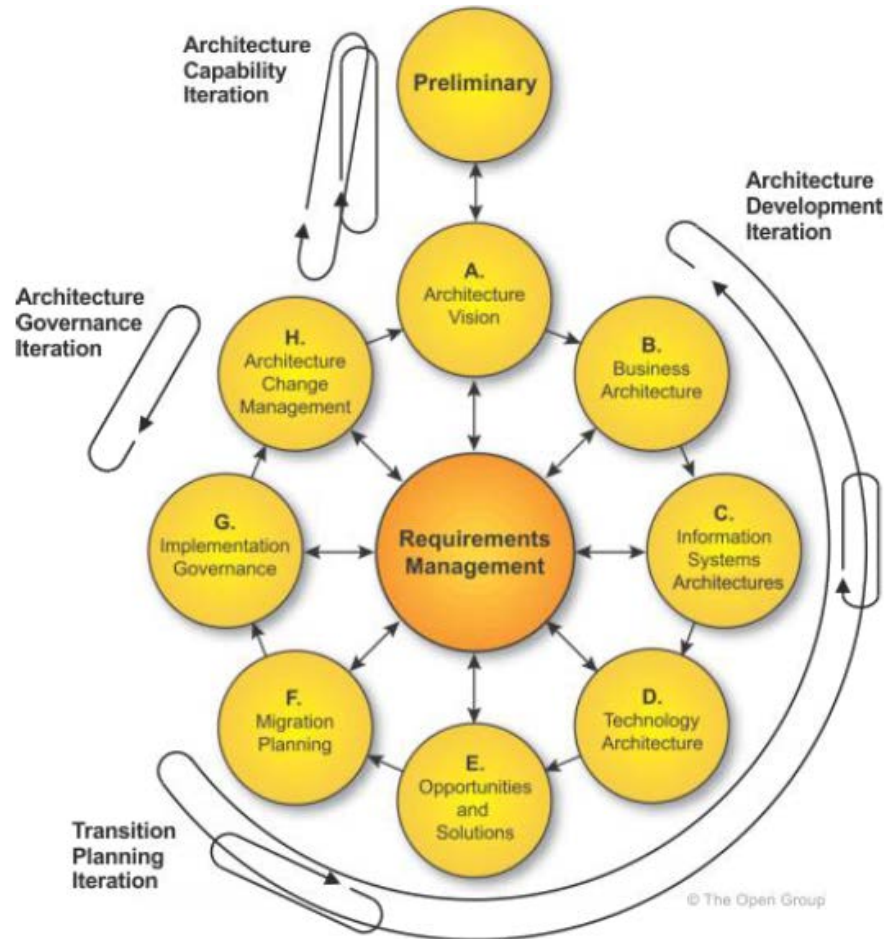
Business Architecture

Data Architecture

Application Architecture

Technology Architecture

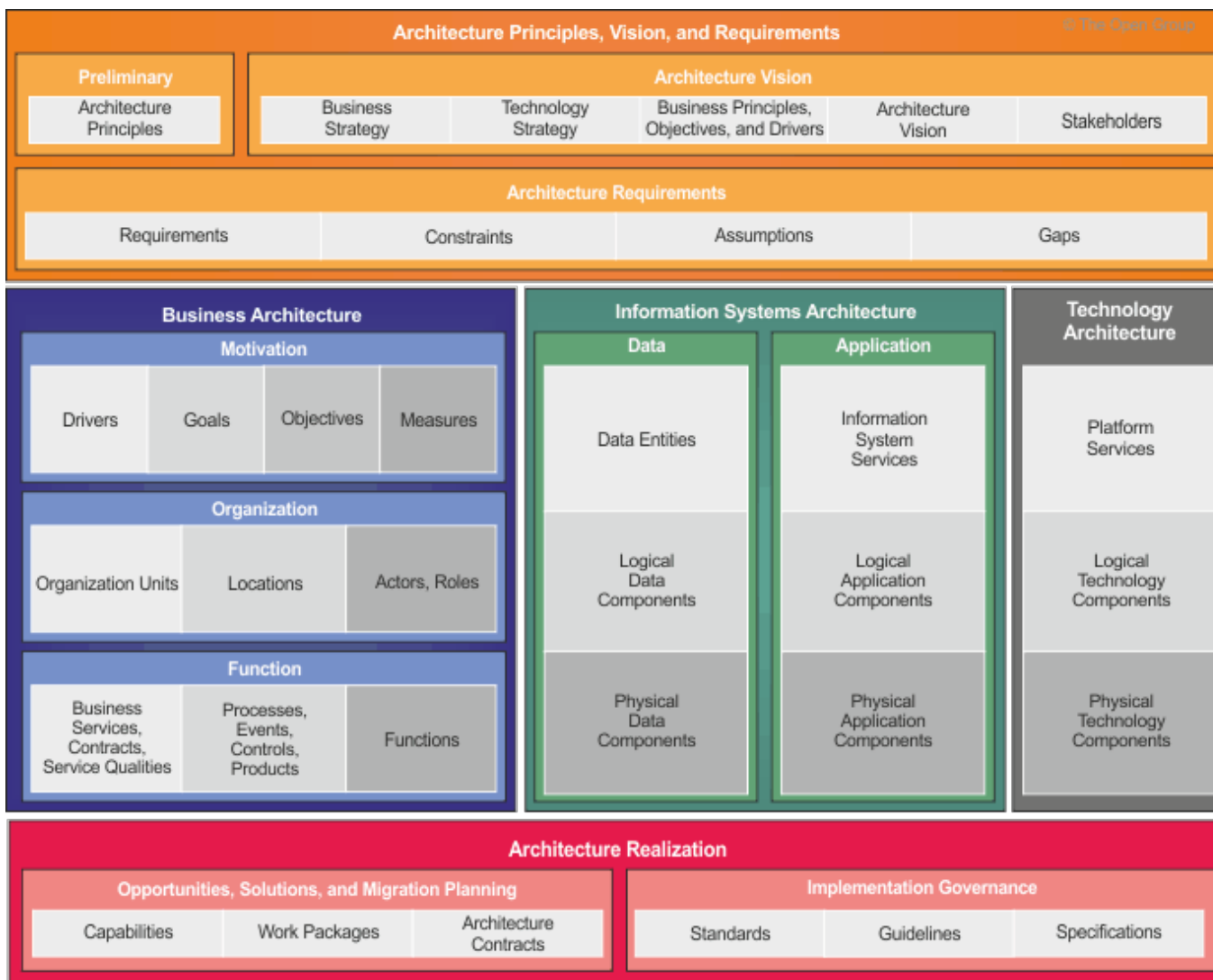
TOGAF Architecture Development Method (ADM)



- Although originally represented as a sequential method, chapter 19.2 of TOGAF describes also iteration cycles

(The Open Group 2011)

TOGAG Content Metamodel



<http://pubs.opengroup.org/architecture/togaf9-doc/arch/chap33.html>

(The Open Group 2011)



TOGAF Content Metamodel

- The content metamodel provides a definition of all the types of building blocks that may exist within an architecture.
- The content metamodel
 - ◆ identifies all of these building block (i.e., application, data entity, technology, actor, and business service),
 - ◆ shows the relationships that are possible between them, e.g.
 - actors consume business services
 - data entities are held within applications
 - technologies implement applications
 - applications support business users or actors
 - ◆ identifies artifacts that can be used to represent them.

TOGAF: Architecture Content Framework

- The content framework is intended to allow TOGAF to be used as a stand-alone framework for architecture.
- However, some enterprises may opt to use an external framework (such as the Zachman Framework or ArchiMate) in conjunction with TOGAF.
- In these cases, the content framework provides a useful reference and starting point for TOGAF content to be mapped to other frameworks

(The Open Group 2011, Part IV)

ArchiMate

ArchiMate

- ArchiMate is a modeling language that supports the TOGAF content metamodel and the TOGAF ADM

- Three architecture layers:

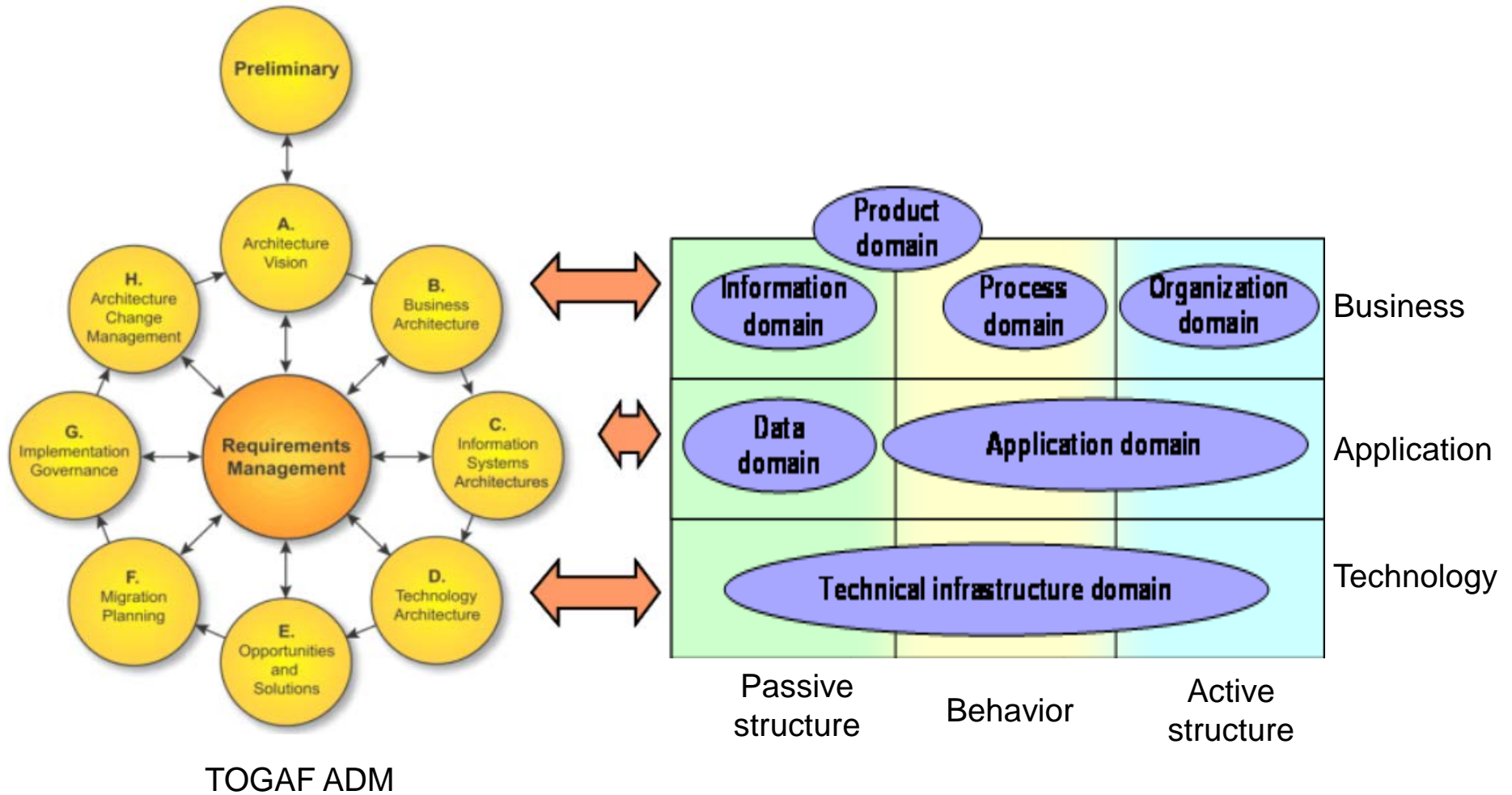
- ◆ *Business*
- ◆ *Application*
- ◆ *Technology*



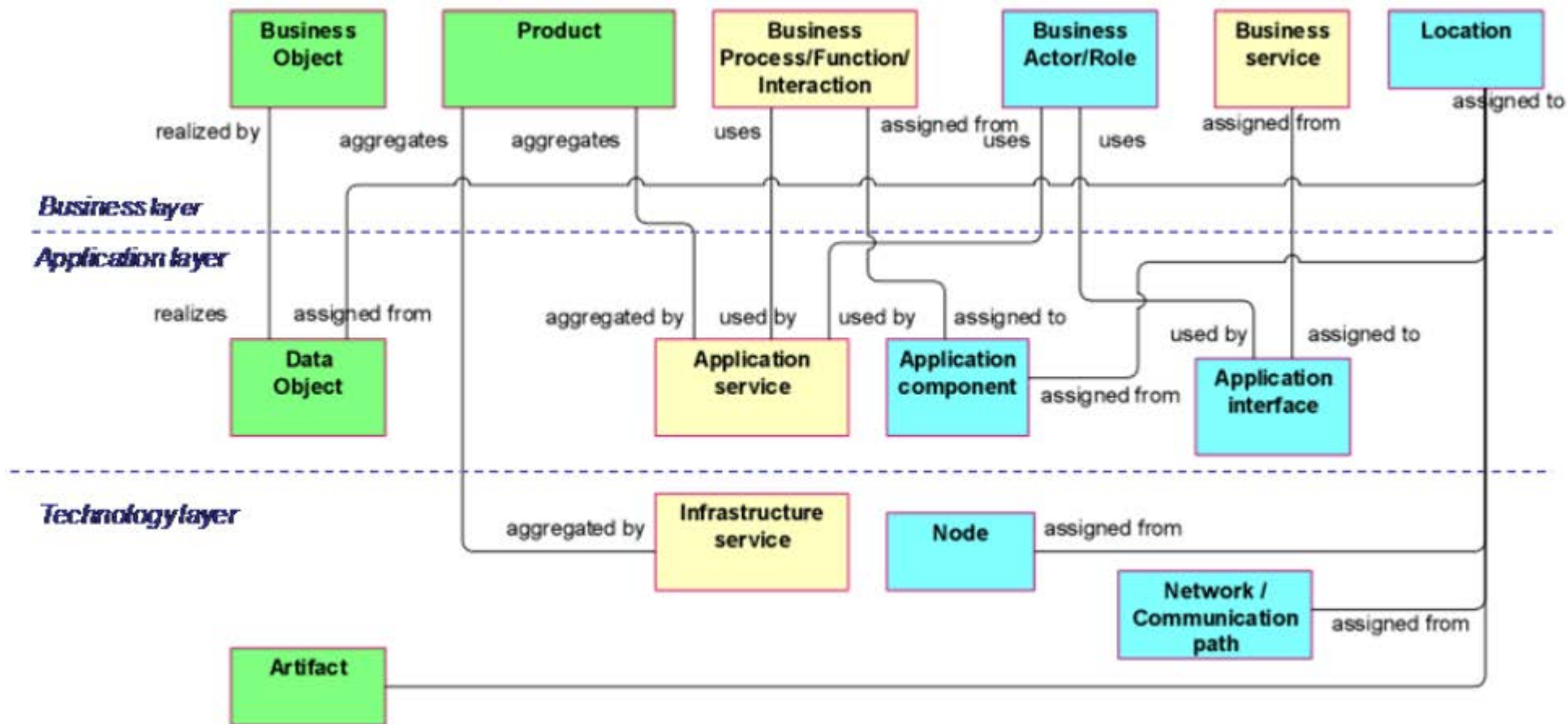
- Three main types of elements:

- ◆ *Active structure* element: an entity that is capable of performing behavior.
- ◆ *Behavior* element: a unit of activity performed by one or more active structure elements.
- ◆ *Passive structure* element: an object on which behavior is performed.

ArchiMate and TOGAF

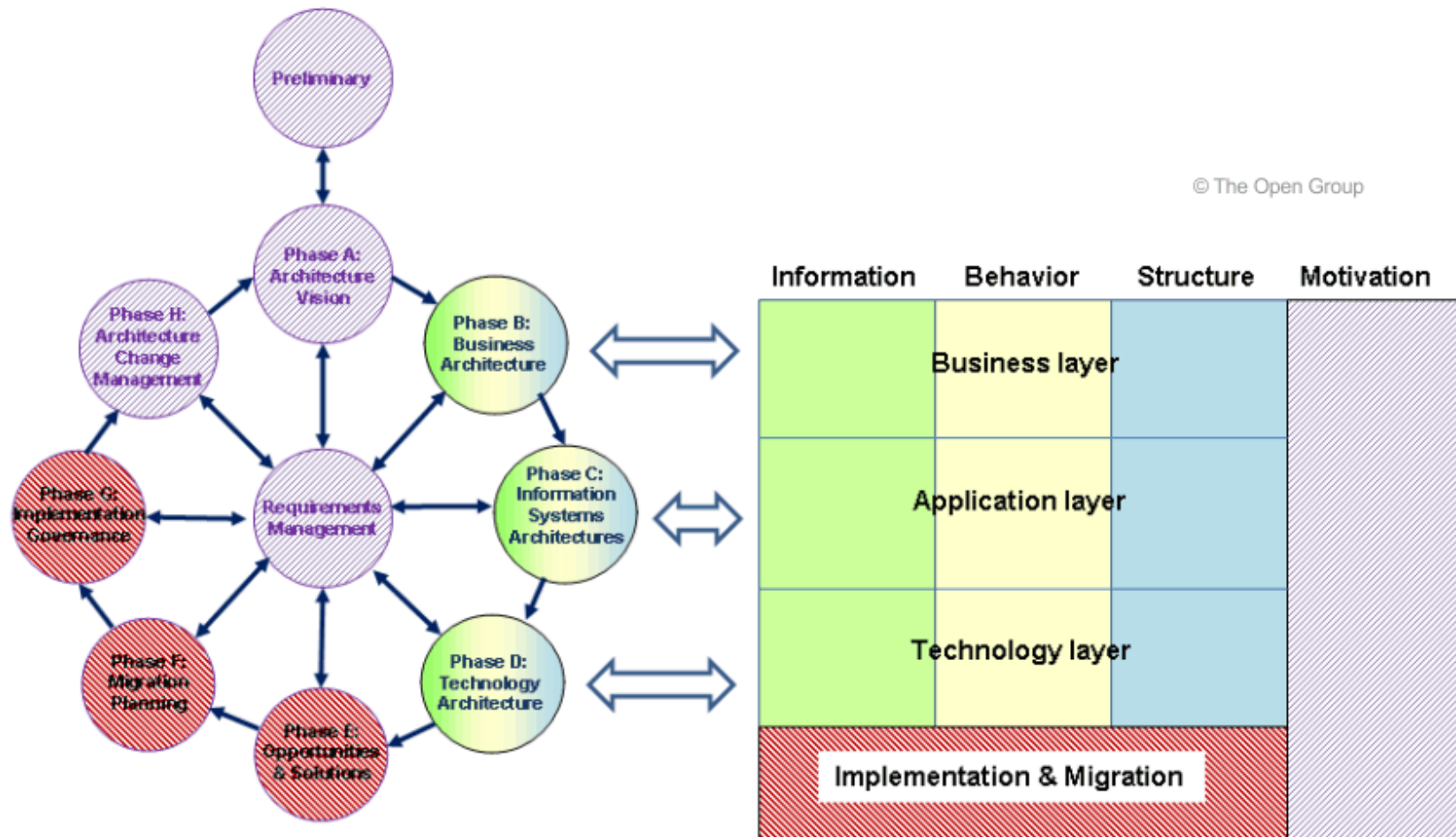


Cross-Layer Dependencies: Business-IT Alignment



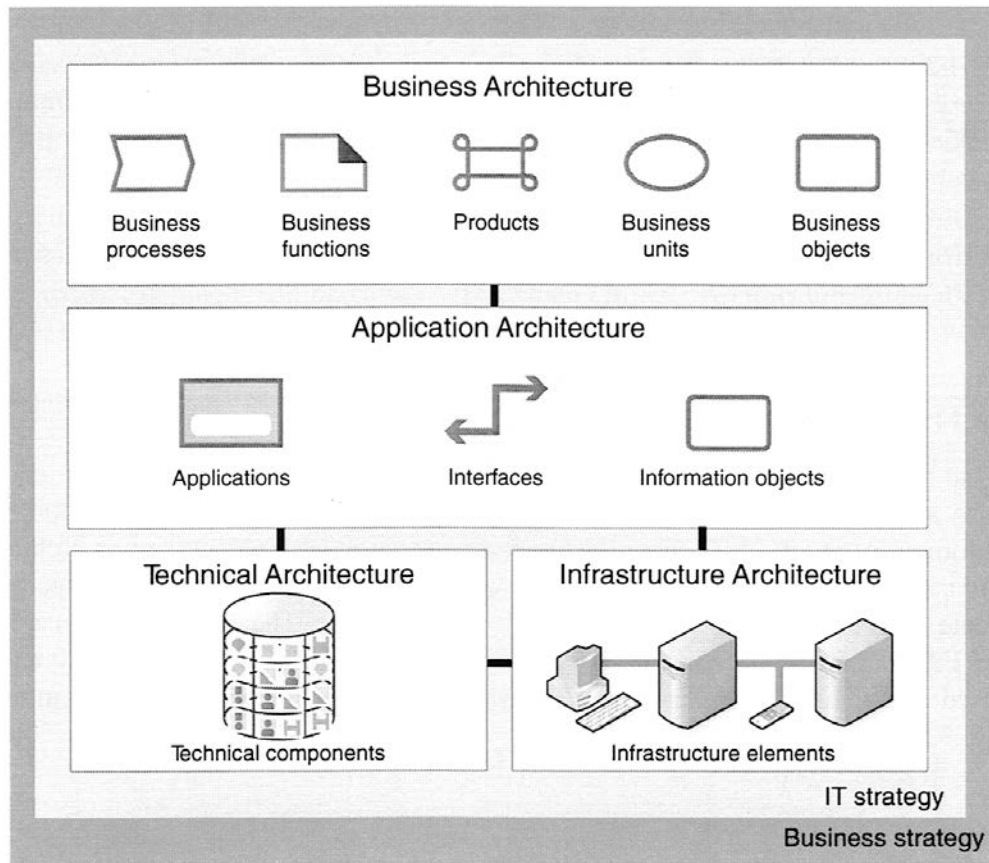
Source: ArchiMate 2.0 Specification, <http://pubs.opengroup.org/architecture/archimate2-doc/chap06.html>

Extensions of ArchiMate to cover the whole TOGAF ADM



Best Practice Enterprise Architecture

Best Practice Enterprise Architecture



- The Best Practice Architecture from Inge Hanschke (2010) is another example of a three-layer enterprise architecture framework.
- In contrast to TOGAF
 - ◆ it is quite simple
 - ◆ it differentiates between the technical architecture and the infrastructure architecture
 - ◆ it does not have a separate data or information architecture

from (Hanschke, 2010)

Partial Architectures of the Best Practice Architecture

■ Business Architecture

- ◆ Describing main entities that determine the business: business processes, functions, products, business units and business objects.

■ Application Architecture

- ◆ documentation of the information systems landscape, i.e. information systems, their data and interfaces and the information flow
- ◆ bridge between business architecture and the architectures of technology and infrastructure

■ Technology Architecture

- ◆ determination of enterprise-specific technical standards for information systems, interfaces and infrastructure

■ Infrastructure Architecture

- ◆ Entities of the infrastructure, on which the information systems are running

Enterprise Architecture Modeling

Overview: ArchiMate

Detailed Models

Overview: Zachman

