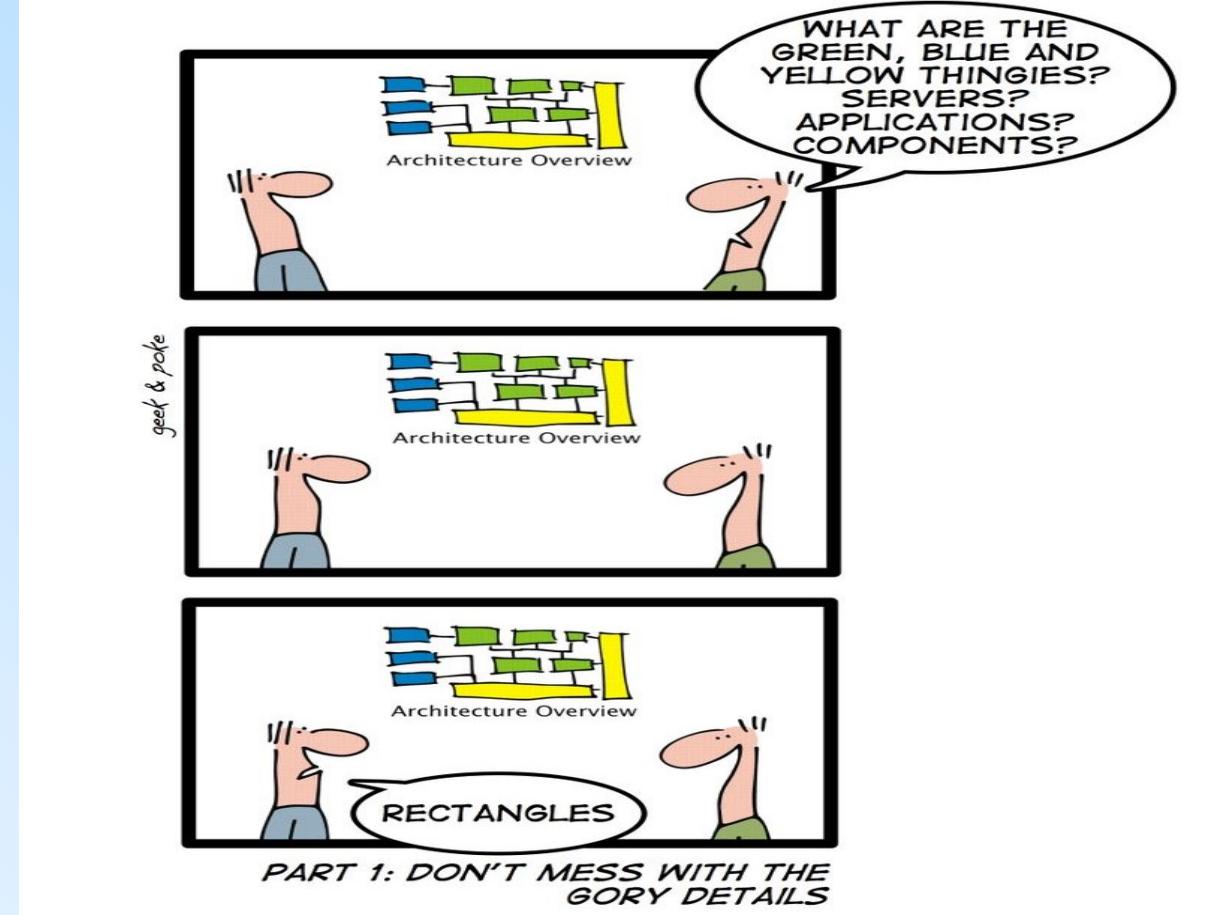


AMS

ENTERPRISE ARCHITECTURE MADE EASY



Enterprise Architecture

...and introductory overview...

Enterprise Architecture

- An **enterprise architecture** is a coherent body of **principles**, **methods**, **concepts and models** used in the **analysis**, **design**, **realization** and **implementation** of the artefacts of an organization
- The **organization** is the target system of the architecture.
- The artifacts of an organization are very varied and include :
 - People
 - Strategy
 - Business Processes
 - Information Systems
 - Support IT

What is “Enterprise Architecture”?

Enterprise architecture (EA) is a discipline for proactively and holistically leading enterprise responses to disruptive forces by identifying and analyzing the execution of change toward desired business vision and outcomes. EA delivers value by presenting business and IT leaders with signature-ready recommendations for adjusting policies and projects to achieve target business outcomes that capitalize on relevant business disruptions.

<http://www.gartner.com/it-glossary/enterprise-architecture-ea/>

TOGAF defines "enterprise" as any collection of organizations that has a common set of goals. For example, an enterprise could be a government agency, a whole corporation, a division of a corporation, a single department, or a chain of geographically distant organizations linked together by common ownership.

(...)

The purpose of enterprise architecture is to optimize across the enterprise the often fragmented legacy of processes (both manual and automated) into an integrated environment that is responsive to change and supportive of the delivery of the business strategy.

(...)

Furthermore, a good enterprise architecture enables you to achieve the right balance between IT efficiency and business innovation.

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

What is “Enterprise Architecture”?

Enterprise Architecture (EA) is a method and an organizing principle that aligns functional business objectives and strategies with an IT strategy and execution plan. The Enterprise Architecture provides a guide to direct the evolution and transformation of enterprises with technology. This in turn makes IT a more strategic asset for successfully implementing a modern business strategy.

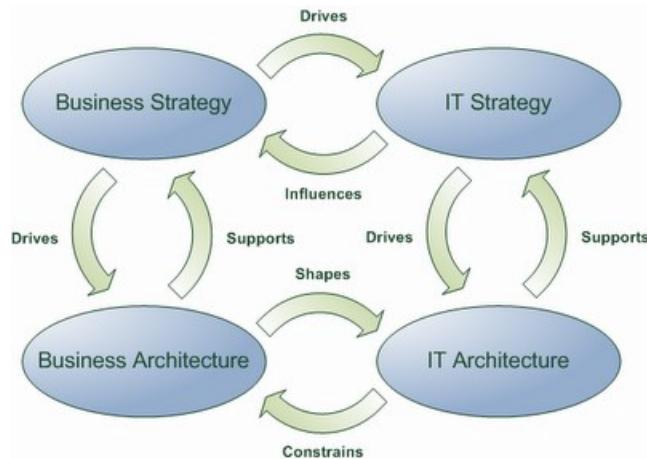
An Enterprise Architecture typically produces deliverables such as:

- **Current State** Enterprise Architecture model
- **Future State** Enterprise Architecture reference model that is needed to execute on the proposed business strategy
- **Gap analysis** that identifies the shortfalls of the current state in terms of its ability to support the objectives and strategies of the business
- **Architecture Roadmap** that defines the initiatives required to migrate from the current state into the future state.

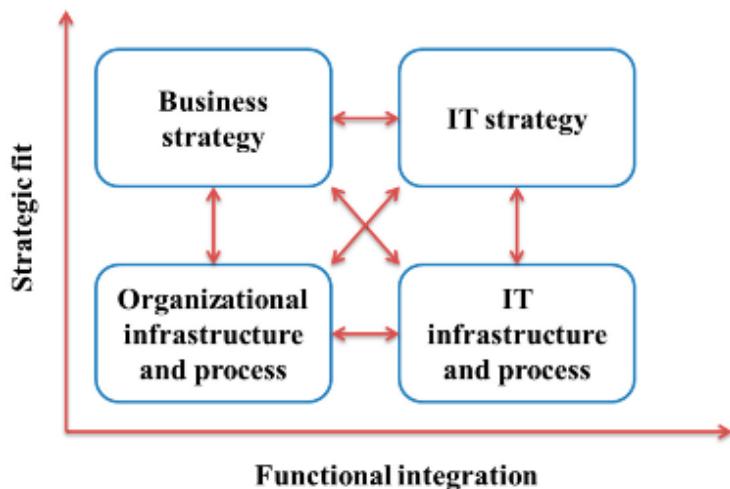
By taking an enterprise-wide perspective across business services, business processes, information, applications, and technology, an EA ensures the enterprise goals and objectives are addressed in a holistic way across all IT projects.

What is “Enterprise Architecture”?

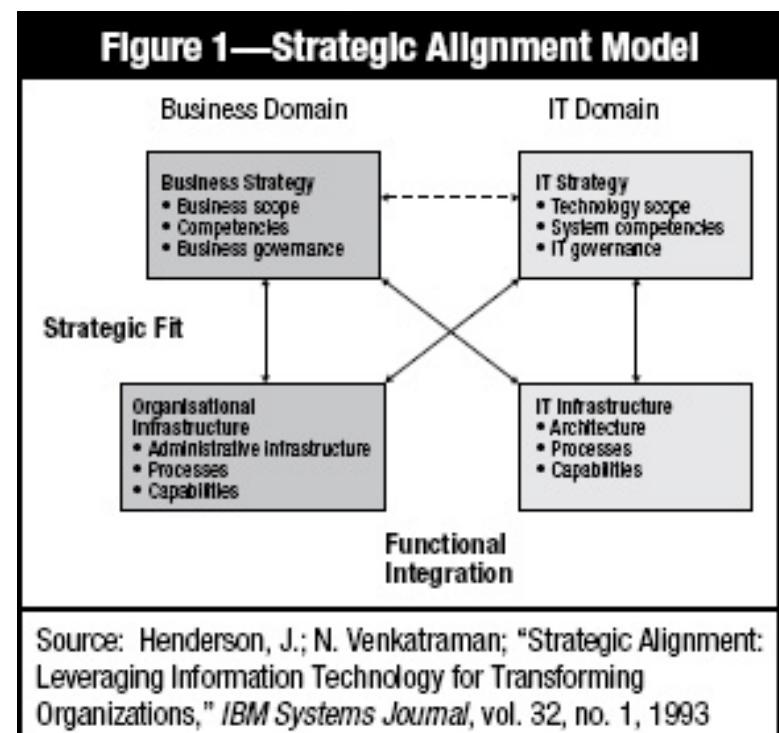
In simple terms,
EA is about
managing the
Business ↔ IT
Alignment



<http://www.btag.com/images/Business-IT%20Alignment.png>



http://www.mdpi.com/technologies/technologies-03-00198/article_deploy/html/images/technologies-03-00198-g002-1024.png



Business – IS – IT Dynamics

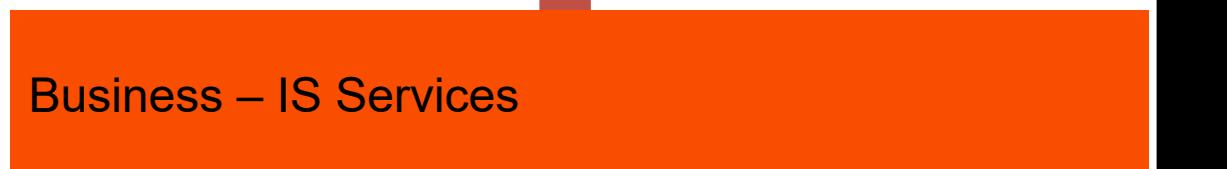
Strategy



Business



Decoupling Layer



Software



IT Infrastructure



Alignment with Management

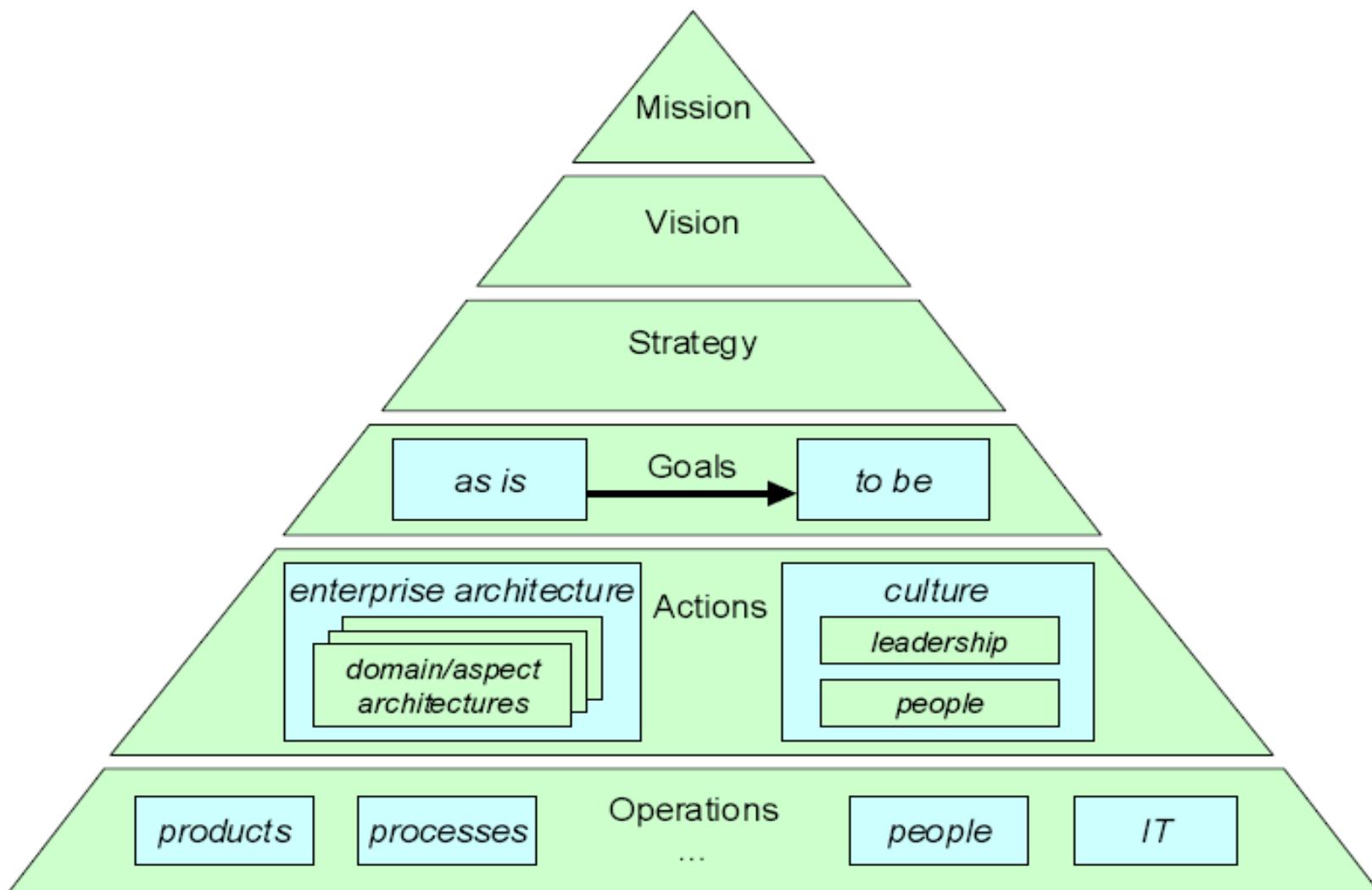
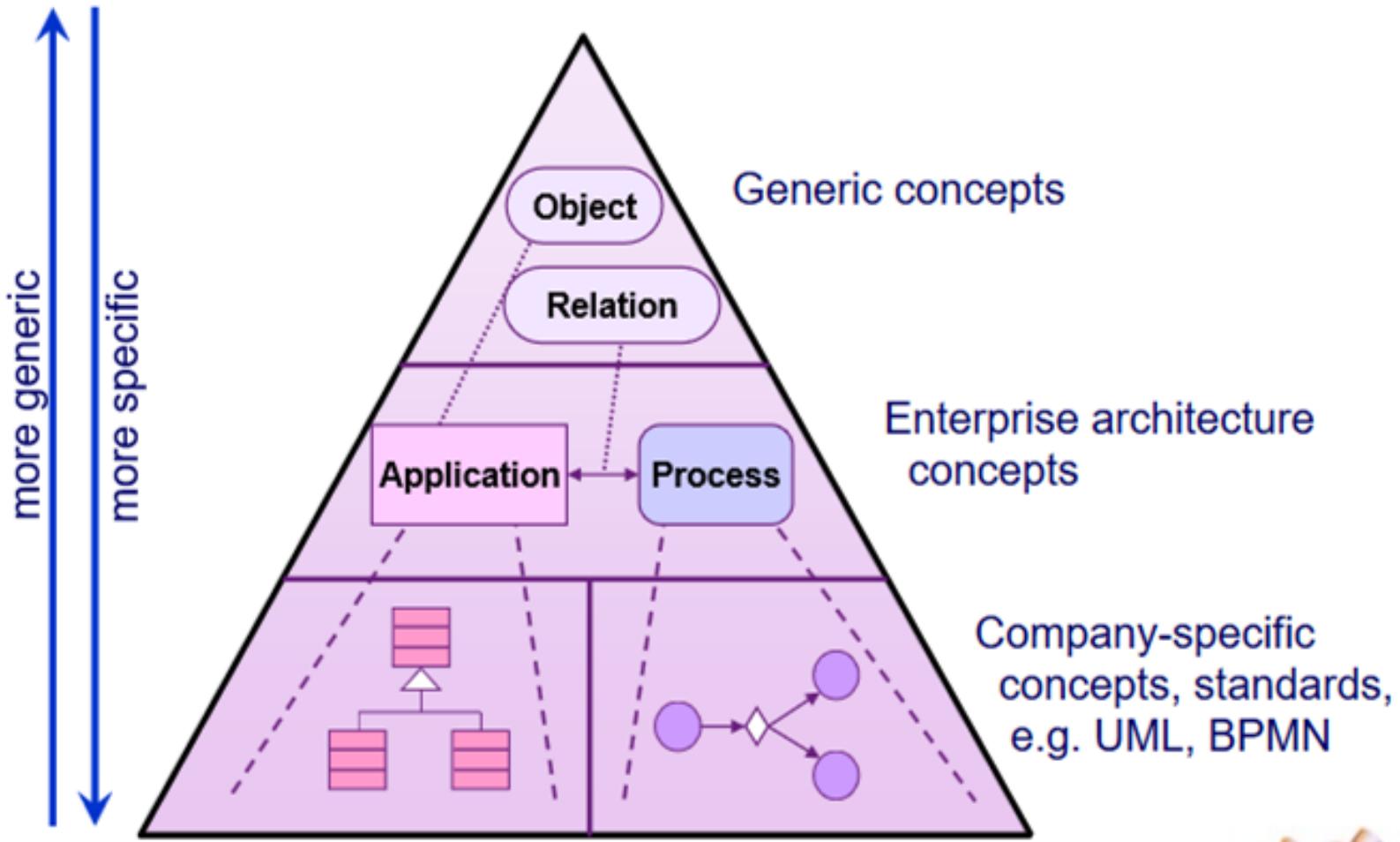
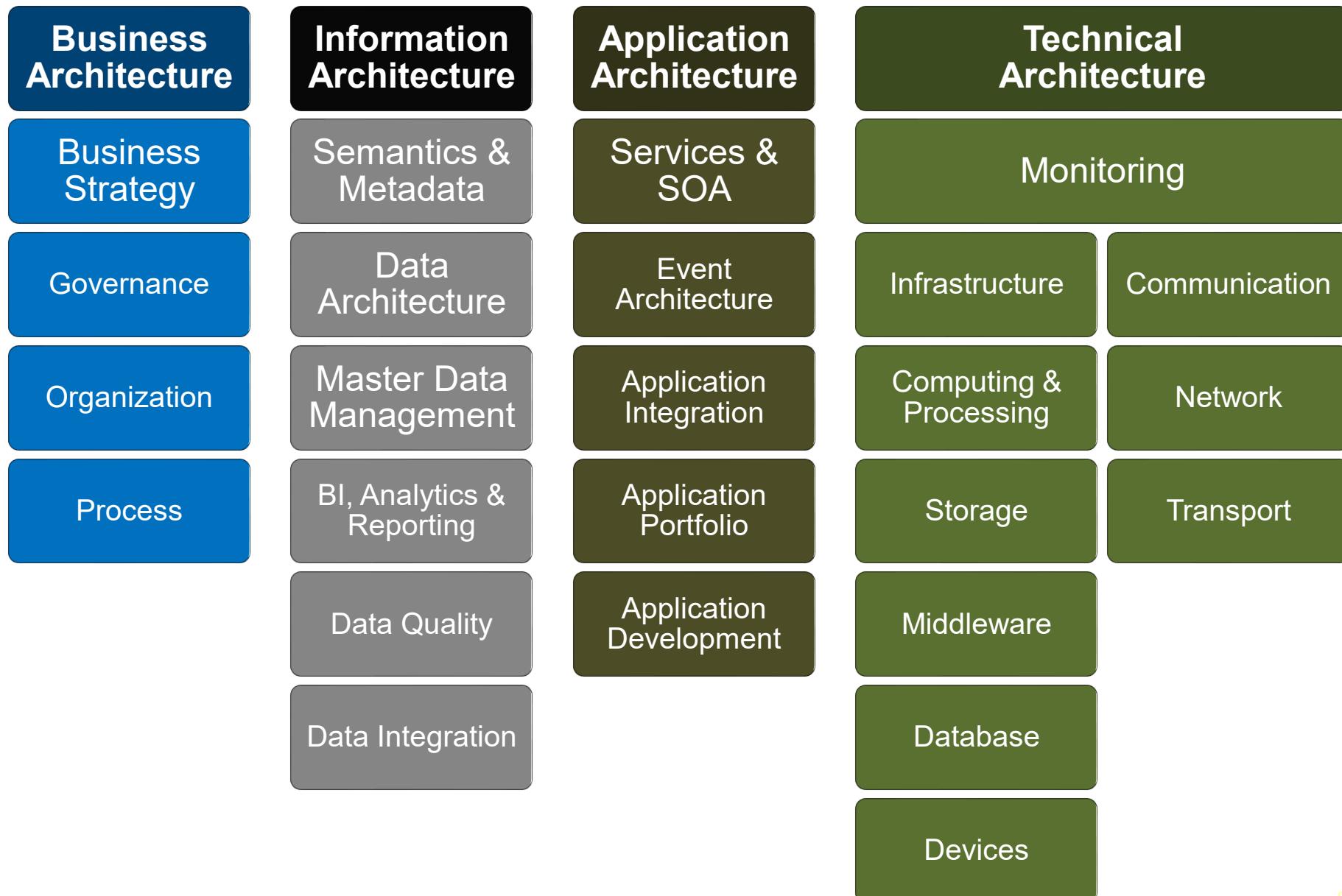


Fig. 1.4. Enterprise architecture as a management instrument.

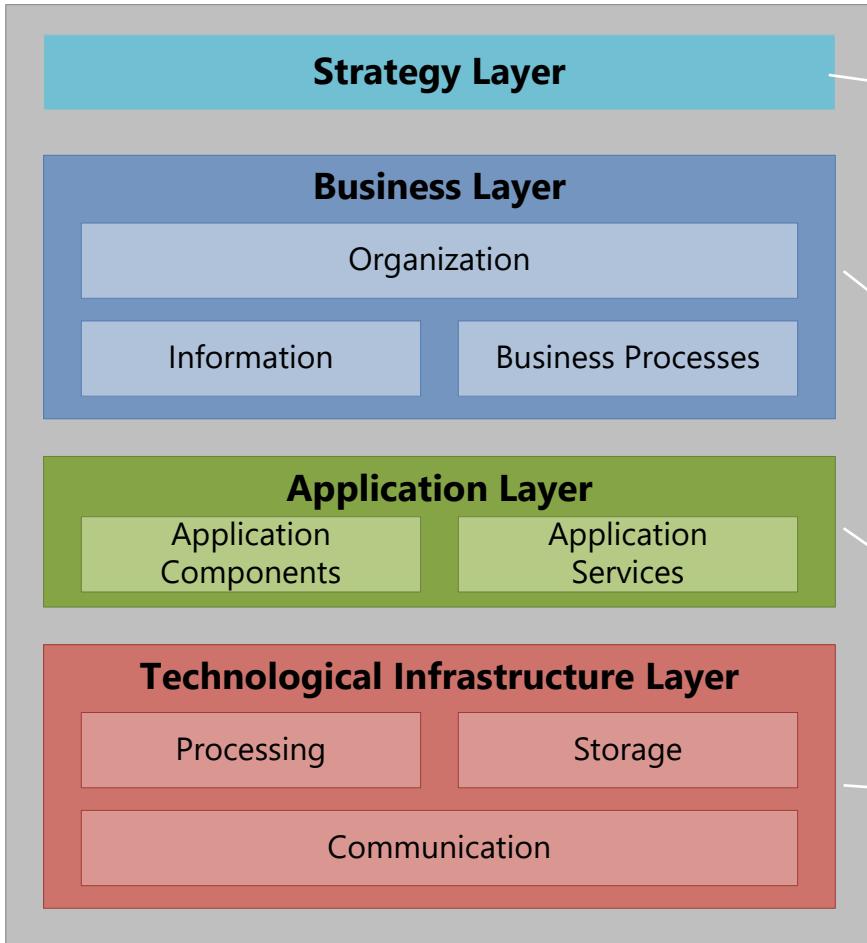
Abstraction Levels



EA Domains



EA Domains



Mission, Vision, Strategic Indicators, External Products/Services, Contracts, Rules and Regulations

Organizational Structure (Units, Functions), Actors, Information Entities, Business Processes and Activities, Goals, Operational Indicators (e.g. break points, control points)

(Technology neutral)
Applications, Services, Clusters of Applications, Clusters of Services

(Technology dependent)
Software applications and Services, Data Structures, Hardware

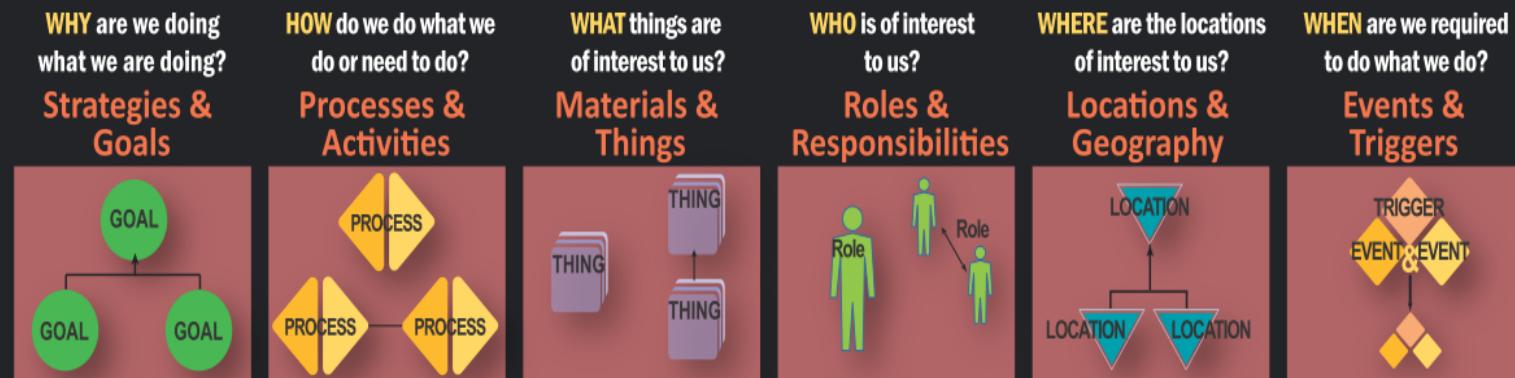
The Zachman Framework for Enterprise Architecture™

The Enterprise Ontology™



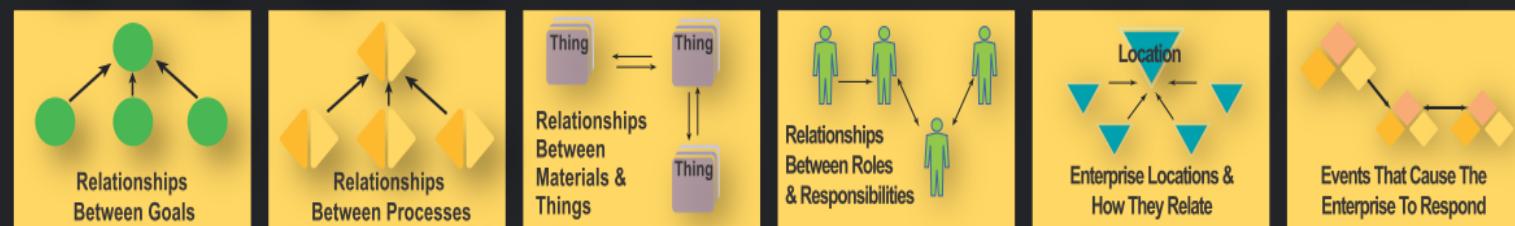
DESCRIBE

The Business



DEFINE

The Relationships



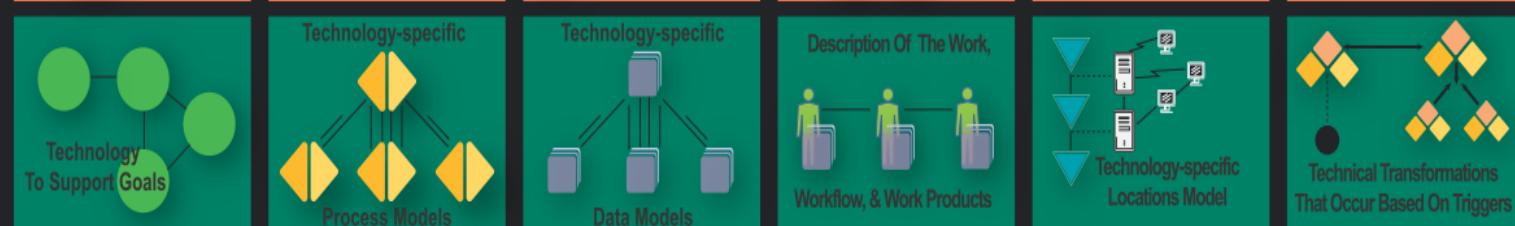
SPECIFY

Components & Services
(Technology-neutral)



IDENTIFY

Technology
(Technology-specific)



SELECT

Solutions



THE ENTERPRISE

GOALS

PROCESSES

MATERIALS

ROLES

LOCATIONS

EVENTS

What is “Enterprise Architecture”?

...in this course...

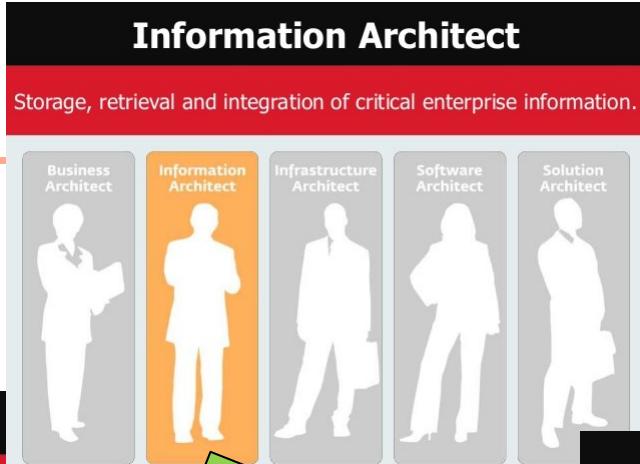
	DATA What	FUNCTION How	NETWORK Where	PEOPLE Who	TIME When	MOTIVATION Why
Objective/Scope (contextual) Role: Planner	List of things important in the business	List of Business Processes	List of Business Locations	List of important Organizations	List of Events	List of Business Goal & Strategies
Enterprise Model (conceptual) Role: Owner	Conceptual Data/Object Model	Business Process Model	Business Logistics System	Work Flow Model	Master Schedule	Business Plan
System Model (logical) Role: Designer	Logical Data Model	System Architecture Model	Distributed Systems Architecture	Human Interface Architecture	Processing Structure	Business Rule Model
Technology Model (physical) Role: Builder	Physical Data/Class Model	Technology Design Model	Technology Architecture	Presentation Architecture	Control Structure	Rule Design
Detailed Representation (out of context) Role: Programmer	Data Definition	Program	Network Architecture	Security Architecture	Timing Definition	Rule Speculation
Functioning Enterprise Role: User	Usable Data	Working Function	Usable Network	Functioning Organization	Implemented Schedule	Working Strategy

ArchiMate

BPMN

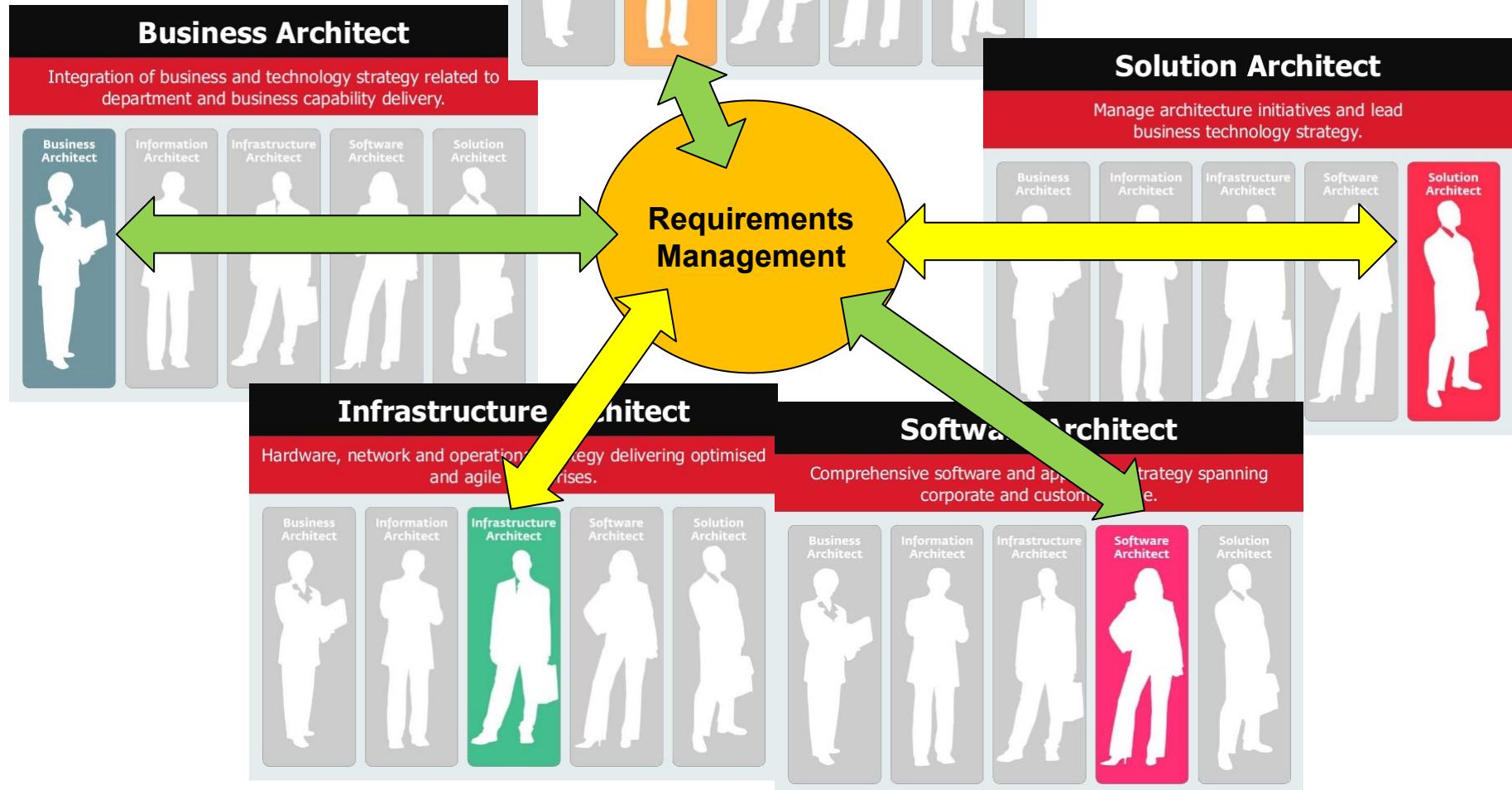
UML

Typical roles in Enterprise Architecture....



This course addresses relevant techniques and principles to use them for:

- ...primarily support those roles...
- ...and also inform those roles...





DEI
DEPARTAMENTO
DE ENGENHARIA INFORMÁTICA
TÉCNICO LISBOA

AMS

Enterprise Architecture

...and introductory overview...

**...ADM, the present reference method for Enterprise
Architecture...**

An Enterprise Architecture framework: TOGAF

The Open Group Library Sign In or Create an Account

THE OPEN GROUP TOGAF
The Open Group Library

Search Library here... 

Standards ▾ Certifications ▾ Guides ▾ Data Sheets Webinars ▾ White Papers ▾ Subject Areas More... ▾ The Open Group Press

Home > TOGAF® Library

TOGAF® Library

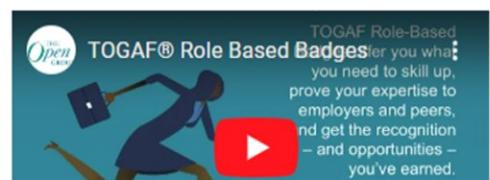
Quick Links: Browse by Document Type

- [The TOGAF Standard](#)
- [TOGAF Translation Glossaries](#)
- [TOGAF Series Guides](#)
- [TOGAF Study Guides](#)
- [TOGAF White Papers](#)
- [TOGAF Webinars](#)
- [TOGAF Reference Cards](#)



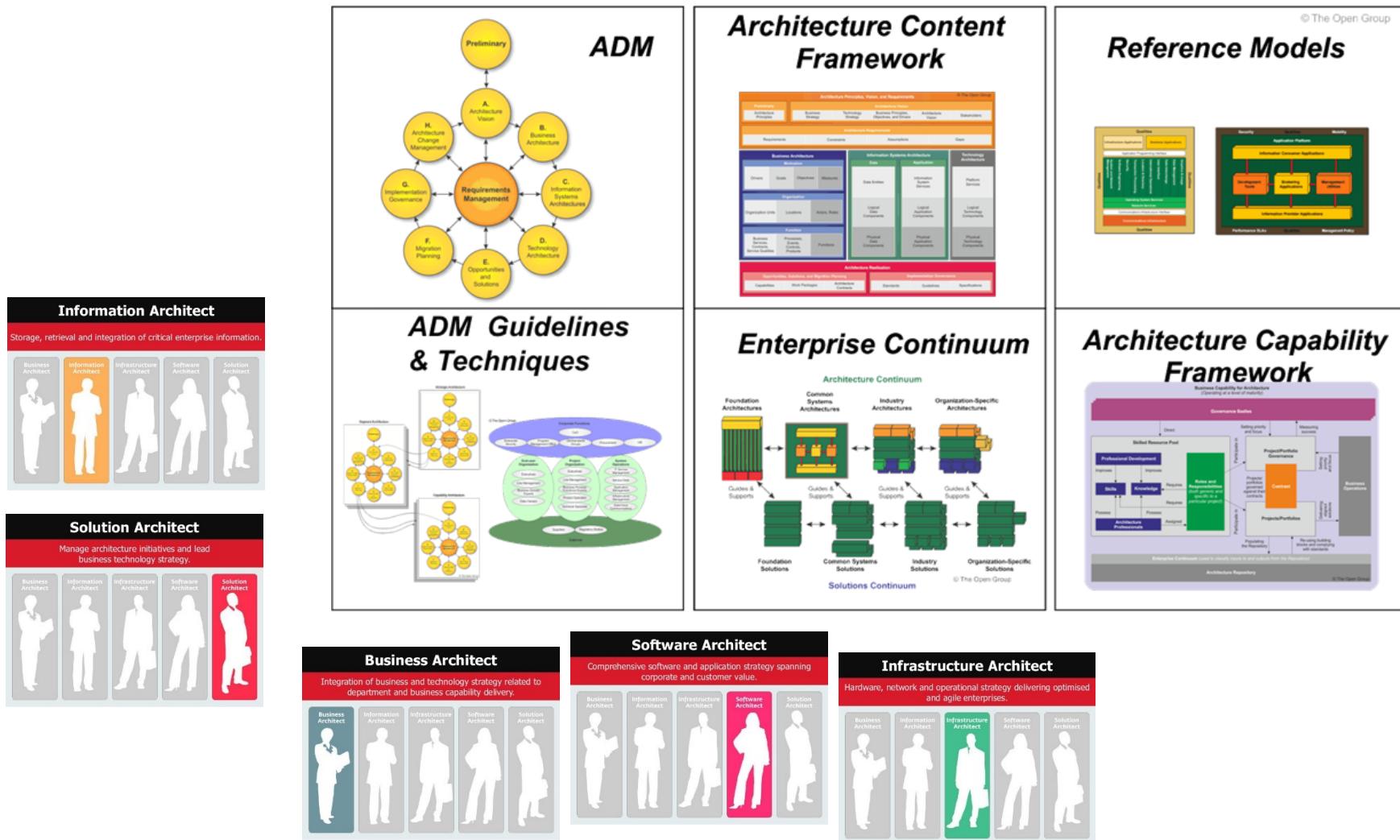
This is the online version of the TOGAF® Library — a Reference Library of potentially useful resources. The TOGAF Library follows a categorization model based on capabilities and features that can be delivered into the market through different sets of documents and resources.

You can also view a selection of the reference cards and information sheets online on the virtual bookshelf:



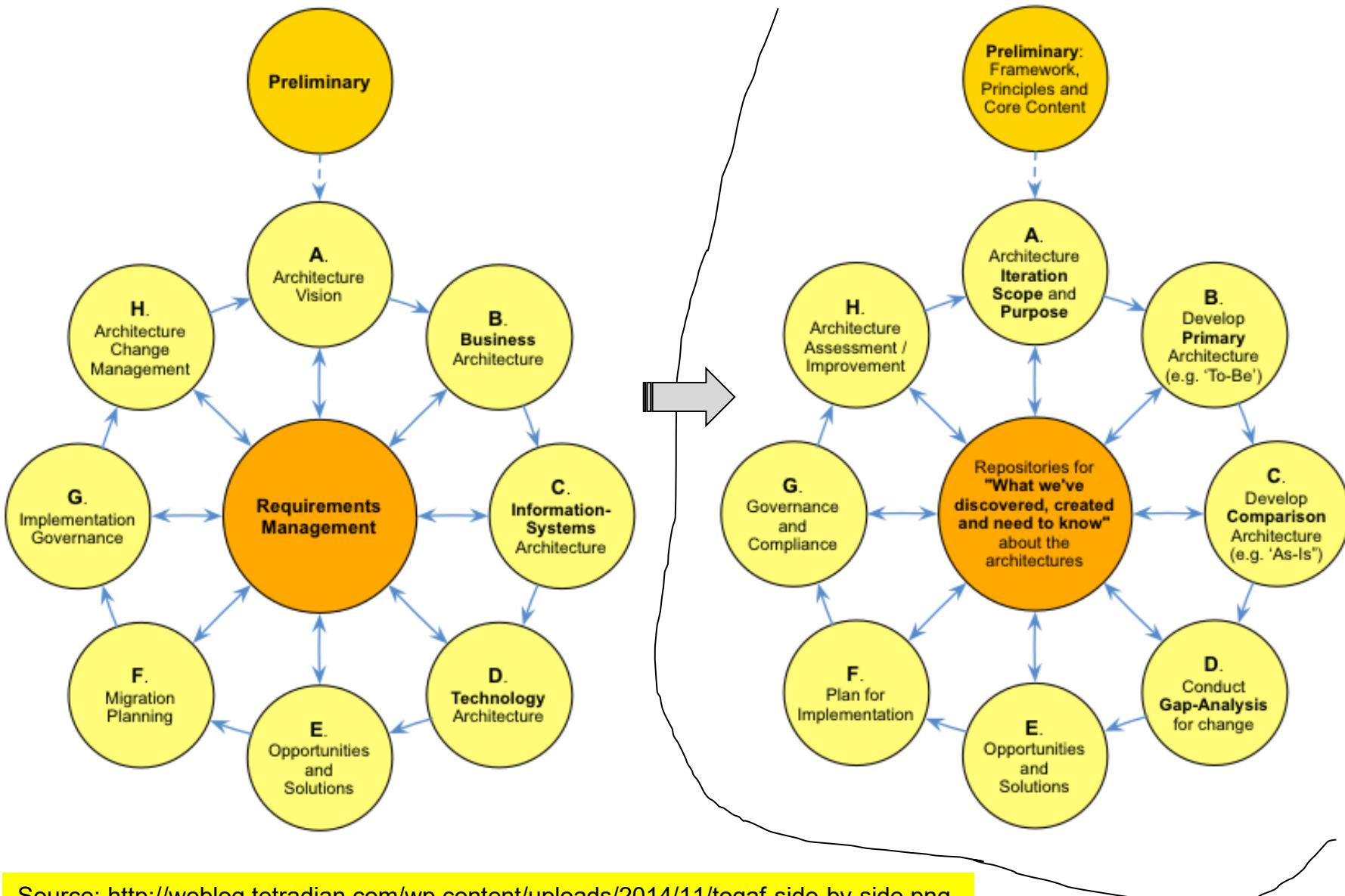
TOGAF

Methods are required for such the complex context that is Enterprise Architecture...
TOGAF is the present main framework reference for that...

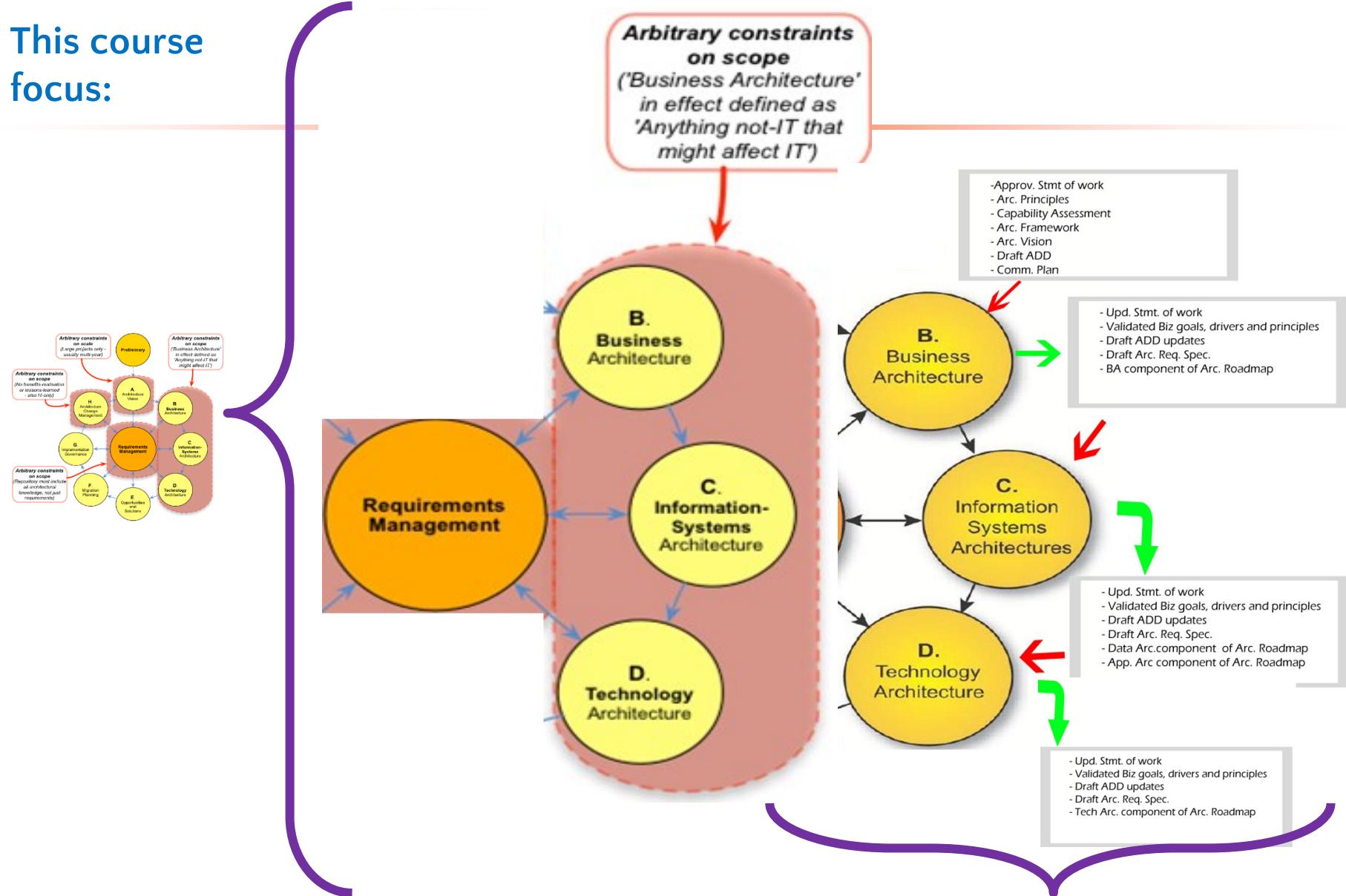


An Enterprise Architecture method: ADM (from the TOGAF framework...)

Architecture Development Method cycle



This course focus:



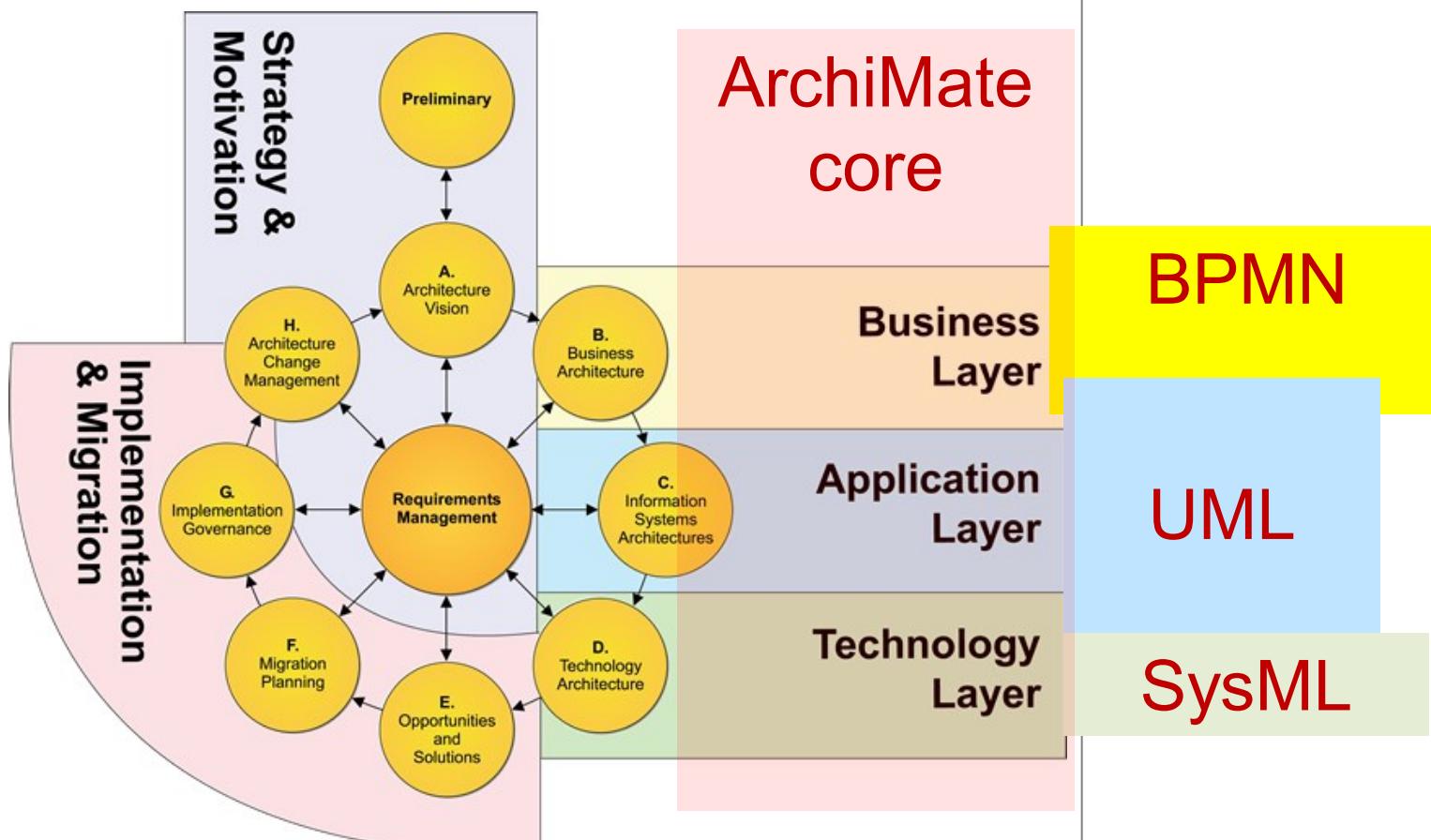
Sources:

<http://weblog.tetradian.com/wp-content/uploads/2014/11/togaf-side-by-side.png>

http://sarathc.com/wp-content/uploads/2015/01/TOGAF_IO_DIAGRAM.png

TOGAF ADM and ArchiMate core and o

ArchiMate





Standards ▾ Certifications ▾ Guides ▾ Data Sheets Webinars ▾ White Papers ▾ Subject Areas More... ▾ SOSA™ Library

The Open Group Press

Home > ArchiMate® Library

ArchiMate® Library

The ArchiMate® Specification, a standard of The Open Group, is an open and independent modeling language for Enterprise Architecture that is supported by different tool vendors and consulting firms. The ArchiMate visual modeling notation leverages your Enterprise Architecture practice and helps you describe and understand complex systems.

The Open Group makes a number of models available in the ArchiMate Model Exchange File format. Browse the ArchiMate model repository.

The following are available to read online or for immediate download (no registration required):

- [The online edition of the ArchiMate 3.2 Specification](#)
- [The ArchiMate 3.2 Specification Reference Cards](#)
- [The ArchiMate Exchange File Format Information Sheet](#)
- [The ArchiMate Certification Information Sheet](#)

You can also view a selection of the reference cards and information sheets online on the virtual bookshelf:



ArchiMate is a modeling language governed by The Open Group...
Please explore The Open Group web site and get familiar with it...





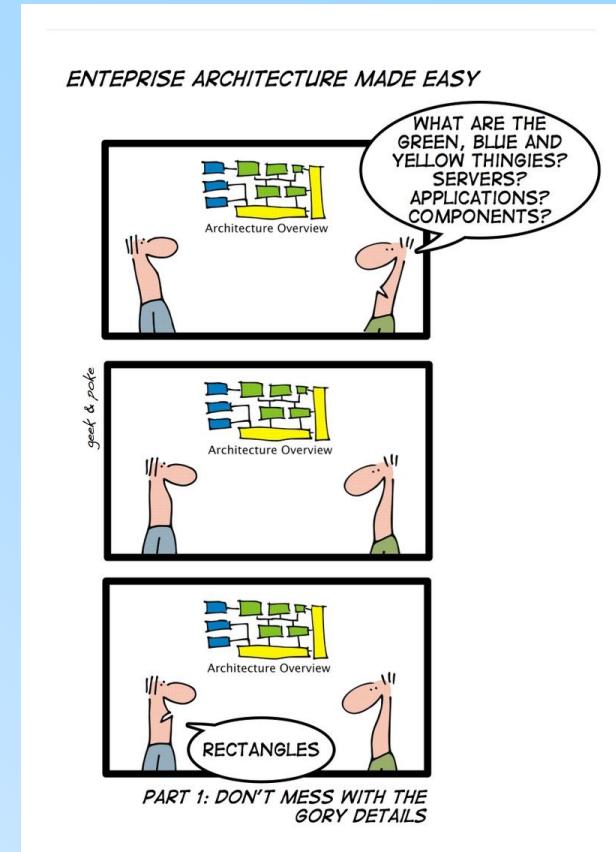
DEI
DEPARTAMENTO
DE ENGENHARIA INFORMÁTICA
TÉCNICO LISBOA

AMS

ArchiMate

...overview...

...and core concepts...



ArchiMate recommended sources:

Language reference: <http://pubs.opengroup.org/architecture/archimate3-doc/toc.html>

The ArchiMate® Enterprise Architecture Modeling Language

About the ArchiMate Modeling Language

The ArchiMate® Specification, a standard of The Open Group, is an open and independent modeling language for Enterprise Architecture that is supported by different tool vendors and consulting firms. The ArchiMate Specification provides instruments to enable Enterprise Architects to describe, analyze, and visualize the relationships among business domains in an unambiguous way. (Download a copy of the ArchiMate Specification)

Just as an architectural drawing in classical building architecture describes the various aspects of the construction and use of a building, the ArchiMate Specification defines a common language for describing the construction and operation of business processes, organizational structures, information flows, IT systems, and technical infrastructure. This insight helps stakeholders to design, assess, and communicate the consequences of decisions and changes within and between these business domains.

Resources

- ArchiMate Webinars
- Download a copy of the ArchiMate 3.1 Specification
- Read the ArchiMate 3.1 Specification online
- ArchiMate 3.1 Pocket Guide
- ArchiMate 3.1 Reference Cards
- The ArchiMate 3.1 Specification
- Blog: An Introduction to the ArchiMate® 3.1 Specification
- ArchiMate Certification Register
- Accredited ArchiMate Training
- Calendar of Accredited Training Courses
- Certified ArchiMate Tools
- ArchiMate Publications
- ArchiMate White Papers
- ArchiMate Certification
- ArchiMate Standard Courseware (training materials for trainers)
- ArchiMate Models

The ArchiMate Specification
Developed by The Open Group ArchiMate Forum, the latest version of the ArchiMate Specification is version 3.1.

Preface: ArchiMate® 3.2 Specific

ArchiMate® 3.2 Specification

Preface

- The Open Group
- This Document

Trademarks

Acknowledgements

Referenced Documents

- 1. Introduction
- 2. Definitions
- 3. Language Structure
- 4. Generic Metamodel
- 5. Relationships and Relationship Connectors
- 6. Motivation Elements
- 7. Strategy Layer
- 8. Business Layer
- 9. Application Layer
- 10. Technology Layer
- 11. Relationships Between Core Layers
- 12. Implementation and Migration Layer
- 13. Stakeholders, Architecture Views, and Viewpoints
- 14. Language Customization Mechanisms

Appendix A: Summary of Language Notation

Appendix B: Relationships (Normative)

Appendix C: Example Viewpoints

Appendix D: Relationship to Other Standards, Specifications, and

ArchiMate® 3.2 Specification

The Open Group

Preface

The Open Group

The Open Group is a global consortium that enables the achievement of business objectives through technology standards. With more than 900 member organizations, we have a diverse membership that spans all sectors of the technology community – customers, systems and solutions suppliers, tool vendors, integrators and consultants, as well as academics and researchers.

The mission of The Open Group is to drive the creation of Boundaryless Information Flow™ achieved by:

- Working with customers to capture, understand, and address current and emerging requirements, establish policies, and share best practices
- Working with suppliers, consortia, and standards bodies to develop consensus and facilitate interoperability, to evolve and integrate specifications and open source technologies
- Offering a comprehensive set of services to enhance the operational efficiency of consortia
- Developing and operating the industry's premier certification service and encouraging procurement of certified products

Further information on The Open Group is available at www.opengroup.org.

The Open Group publishes a wide range of technical documentation, most of which is focused on development of Open Group Standards and Guides, but which also includes white papers, technical studies, certification and testing documentation, and business titles. Full details and a catalog are available at www.opengroup.org/library.

This Document

This document is the ArchiMate® 3.2 Specification, a standard of The Open Group. It has been developed and approved by The Open Group.

This edition of the standard includes a number of corrections, clarifications, and improvements to the previous edition, as well as several additions.

Intended Audience

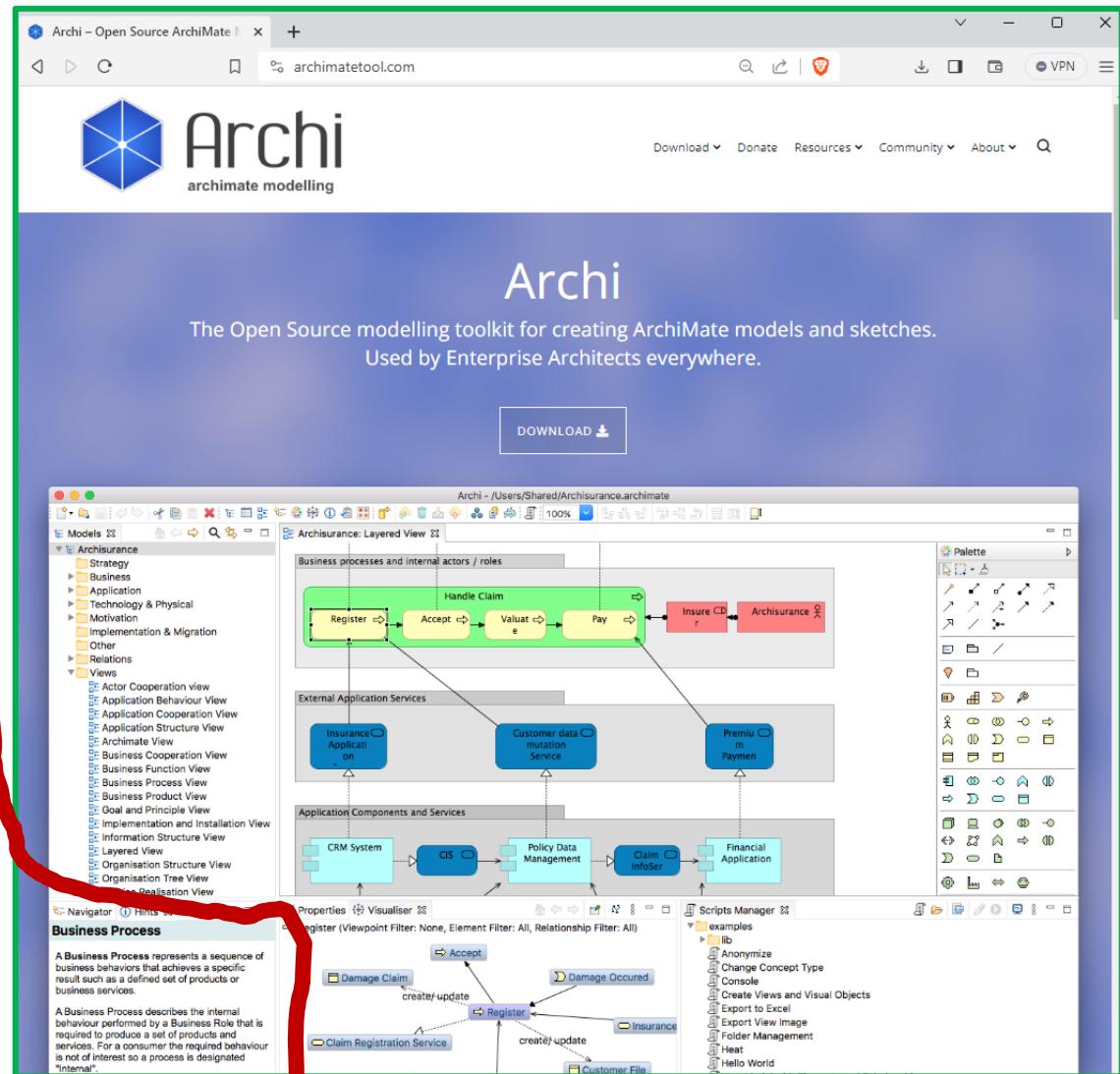
The intended audience of this standard is threefold:

- Those working to shape and implement complex organization change

ArchiMate recommended sources:

Great tool (free):

<https://www.archimatetool.com/>



Very good HELP:

This screenshot shows a section of the ArchiMate Help documentation for "Business Process". The title "Business Process" is at the top. Below it, a definition states: "A Business Process represents a sequence of business behaviors that achieves a specific result such as a defined set of products or business services." Another definition follows: "A Business Process describes the internal behaviour performed by a Business Role that is required to produce a set of products and services. For a consumer the required behaviour is not of interest so a process is designated "internal"." At the bottom, a note says: "A complex Business Process may be an → Register". The entire help page is framed by a green border, and a red wavy line highlights this border, indicating it's a valuable resource.

ArchiMate recommended sources:

From one other modelling toolmaker (a good reference for beginners):

<https://circle.visual-paradigm.com/category/archimate>

R&AIT Strategy & Architecture MASTERING ARCHIMATE – ED. 3.1 EA CHES

Free ArchiMate 3.2 Overview PDFs in multiple languages (Business Process / Function)

HOME > ARCHIMATE > FREE ARCHIMATE 3.2 OVERVIEW PDFS IN MULTIPLE LANGUAGES

In this diagram, a central box labeled '(Business Event)' has two outgoing arrows: one to the left pointing to '(Business Process / Function / Interaction)', and one to the right pointing to 'Modelling artifacts'. The '(Business Process / Function / Interaction)' box has two outgoing arrows: one up to 'Note: Element form shown is incomplete' and one down to 'Modelling artifacts'. A callout box next to the interaction box contains the text: 'Note: Element form shown is incomplete. It represents 3 concepts'.

For years, I've been distributing ArchiMate Overview PDFs, updated whenever I got around to adapting them after the ArchiMate standard was updated. Generally, the overview sheets are the first thing that is updated when I start to work with a new version of the standard. The current version is from ArchiMate 3.2. They are available in English, French, German, Japanese, Portuguese, and Russian.

Also explore:
<https://ea.rna.nl/archimate/free-archimate-overview-pdf/>

ArchiMate examples - Visual Paradigm Online

Home » Business » ArchiMate examples

ArchiMate examples

Posted on March 8, 2022 | Under Business | With 0 Comments

In this article, you will see a rich collection of ArchiMate examples views, organized into a layered framework following the ArchiMate standard. These ArchiMate views demonstrate how ArchiMate elements can be used. Some of the examples can be used as design patterns.

The examples are designed with Visual Paradigm Online, based on the examples in the ArchiMate Cookbook. If you have not yet checked out the cookbook, recommend you take a look. Link: <http://www.hosiaisluoma.fi/ArchiMate-Cookbook.pdf>

ArchiMate Example Views

Framework View

VisualParadigm Online

home > Business > ArchiMate examples

ArchiMate examples

Posted on March 8, 2022 | Under Business | With 0 Comments

In this article, you will see a rich collection of ArchiMate examples views, organized into a layered framework following the ArchiMate standard. These ArchiMate views demonstrate how ArchiMate elements can be used. Some of the examples can be used as design patterns.

The examples are designed with Visual Paradigm Online, based on the examples in the ArchiMate Cookbook. If you have not yet checked out the cookbook, recommend you take a look. Link: <http://www.hosiaisluoma.fi/ArchiMate-Cookbook.pdf>

Table of Contents [hide]

- 1 ArchiMate Example Views
- 2 Strategy Views
- 3 Business Views
- 4 Application Views
- 5 Technology Views
- 6 Implementation and Migration Layer / Transformation Architecture Layer Views
- 7 Extras
- 8 Other Resources

Motivation Views

This motivation views can be used to examine the motivations, or causes, that drive an organization's design or transformation, as well as its enterprise architecture, which serves as the foundation for all change operations and business transformations within a company. This view depicts the development effort's vision, whether the scale and scope encompass the entire organization, a subset of it (e.g., a line of business), or a particular program or project (solution level). Note that a value can be added to any ArchiMate element, such as the outcome (or any other ArchiMate element), to show what the real value add is.

The Business Motivation Model (BMM) [specification v1.3, 2015, OMG] is used to define the motivational elements.

Mission-Values-Vision View

The purpose, vision, and basic values of the organization can all be represented using the Mission-Values-Vision view. It helps you identify an organization's purpose, what is the organization actually doing or intends to do, and what is the primary reason for its existence. The vision is the desired state of the organization in the future. The vision, the culture, and the ideals of the organization are all supported by core values. Strategic goals must be met in order for the organization's vision to be realized.

References: Aldea, A., Iacob, M. E., Hillegensberg, J., Querol, D., Franken, H. (2015) Modelling strategy with ArchiMate.

Strategic Value Map View

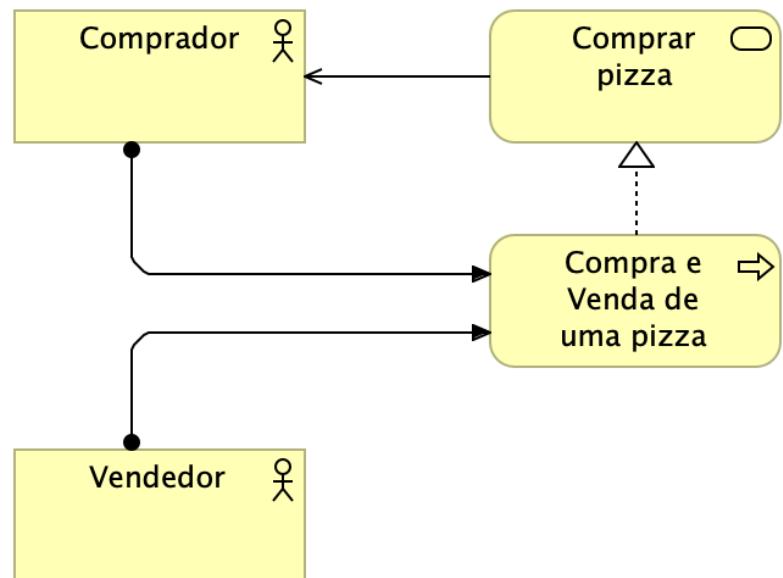
27

Hello World! ArchiMate

Ao **processo de negócio** de compra e venda de uma pizza **estão assignados** dois **atores de negócio**: o comprador e o vendedor.

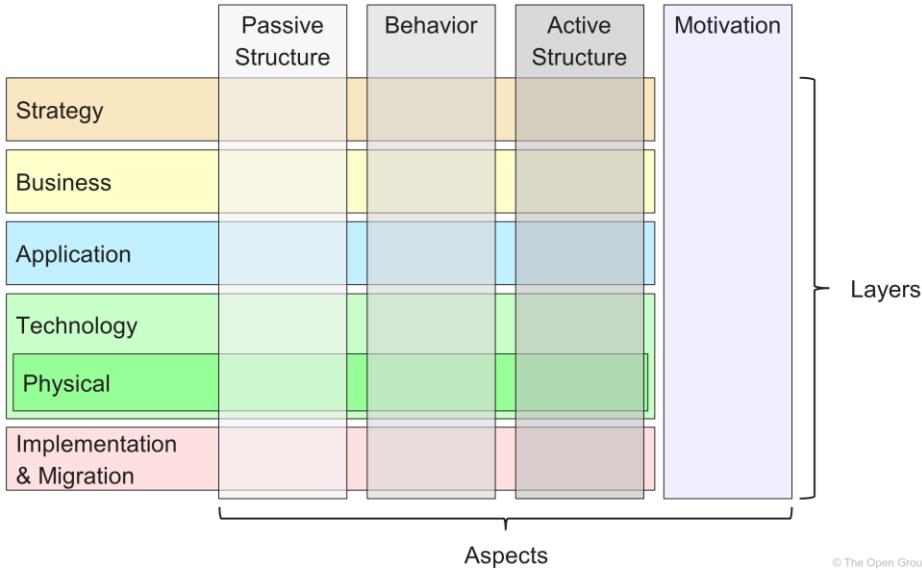
O **serviço de negócio** comprar pizza é **realizado** por este mesmo **processo de negócio** de compra e venda de uma pizza.

Por sua vez, o **serviço de negócio** comprar pizza **serves** o comprador.



ArchiMate Framework – Aspects x Layers

Archimate Framework: “A reference structure used to classify elements of the ArchiMate core language. It consists of layers and aspects.”



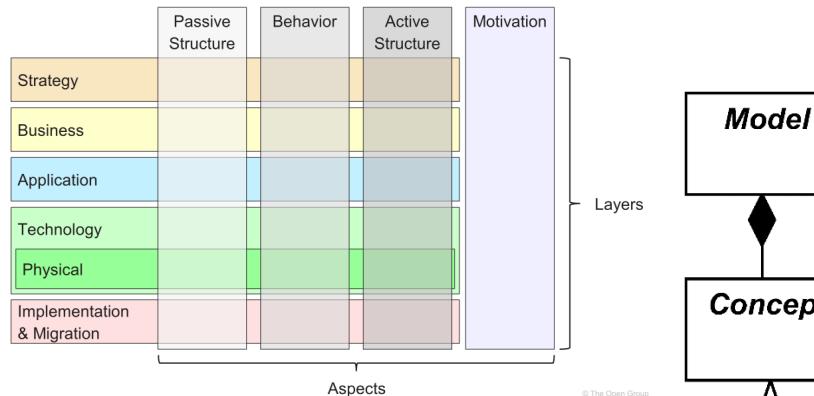
- **Layers:**

- “An abstraction of the ArchiMate framework at which an enterprise can be modeled.”
- In other words, the layers provide descriptions of how the elements are related in order to make structures...

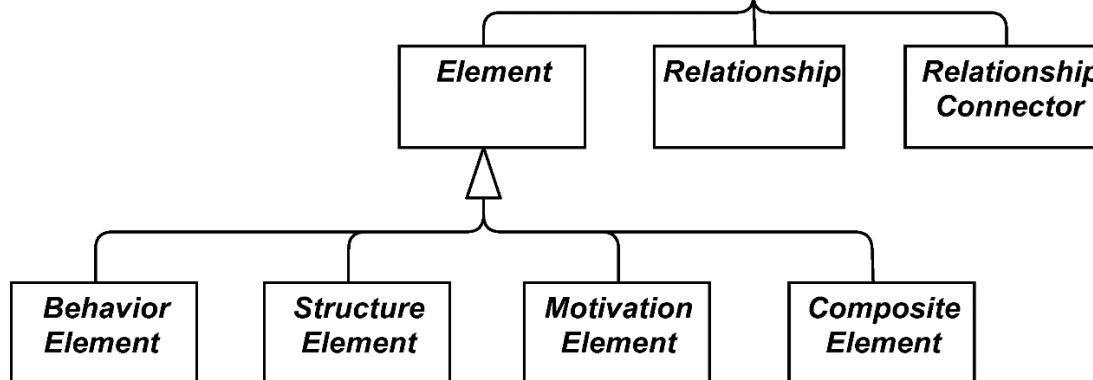
- **Aspects:**

- “Classification of elements based on layer-independent characteristics related to the **concerns of different stakeholders.**”
- In other words, aspects refer to the elements (things...) of the organization...

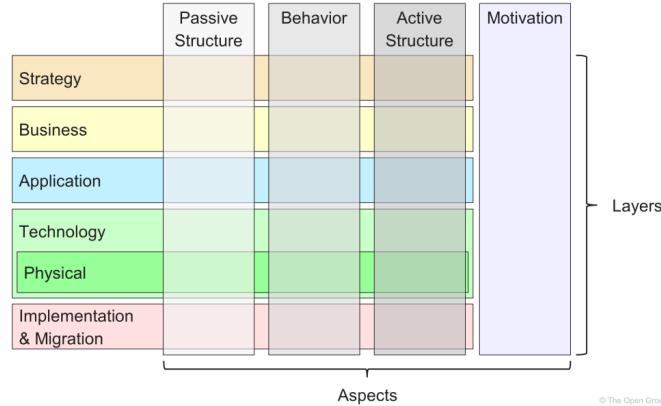
Top-Level Language Structure



- A model is a collection of *concepts*, which can be:
 - *A relationship*
 - *An element*, which can be:
 - A behavior element
 - A structure element
 - A motivation element
 - A composite element.



Note that this is a METAMODEL, meaning these are *abstract* concepts; they are not intended to be used directly in models. The concepts to use in models will be specific classes derived from these (metamodel) concepts...



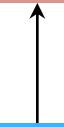
Layers

- Conceptually, one layer “serves” another layer above
- This “layered service orientation” concept leads to a **system based view** of “pairs of layers” made of:
 - Service interface layers.
 - Service implementation layers.
- **Services** are the foundation for the interconnection of layers.
- **Implementation Layers realize** the **Service Layers** of its layer.
- **Implementation** of one layers **serves** the **Services** of the layer below

Service Layer $N+1$



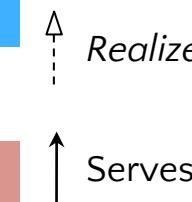
ImplementationLayer $N+1$



Service Layer N



ImplementationLayer N



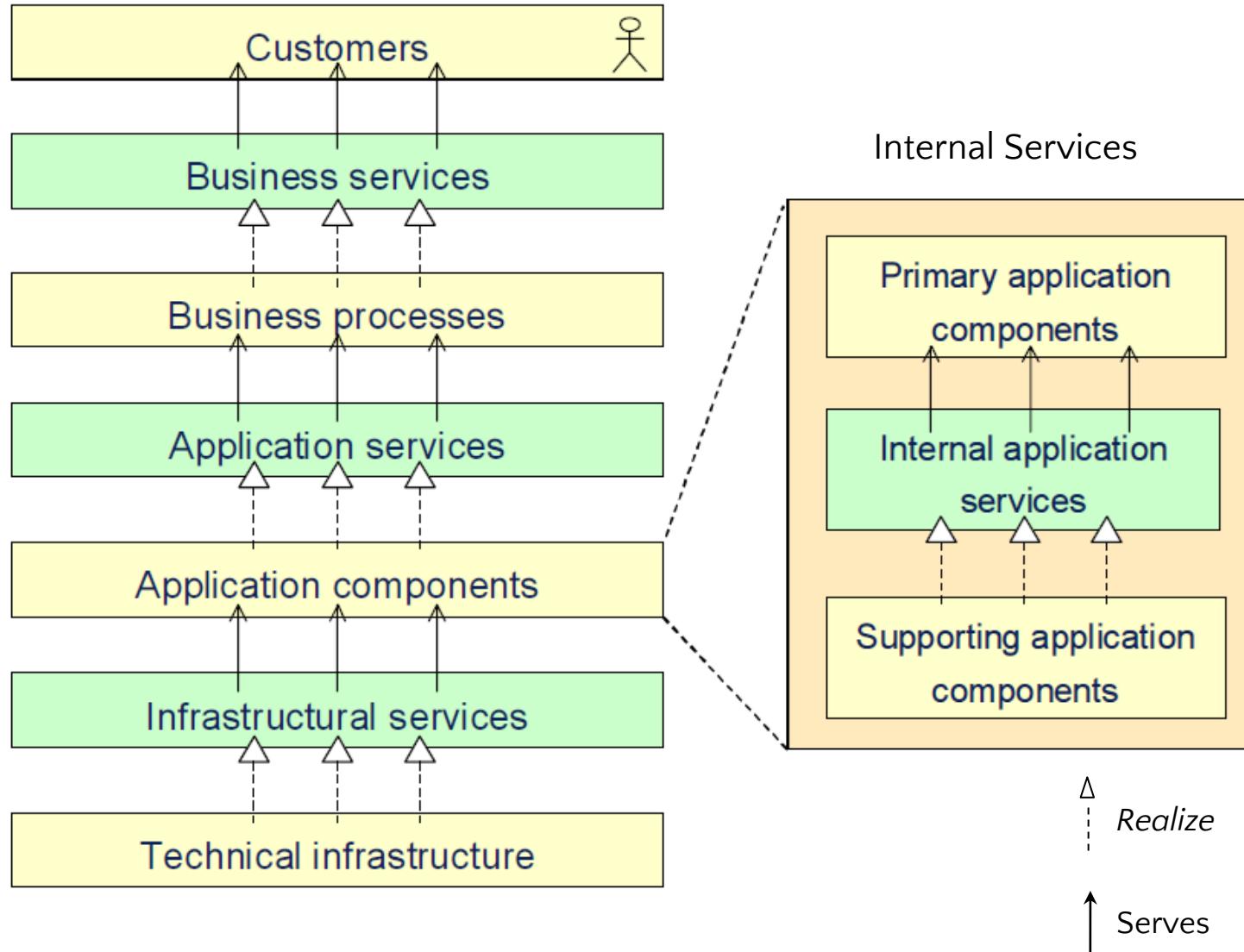
There may be:

- services between layers => horizontal integration (external services)
- services within the same layer => vertical integration (internal services)

Layers: Service Concept

- A **service** is:
 - A **functional unit** that an **entity** provides to its environment.
 - A service has **value** to at least one entity (the service user).
 - An **entity** is, for example, a person, an application, an organization, a department.
- Services may vary in nature and granularity:
 - Service provided by an organization to a customer.
 - Service provided by an application to a process.
 - Service provided by a person to a process.
 - Service provided by a technological component to an application
 - ...

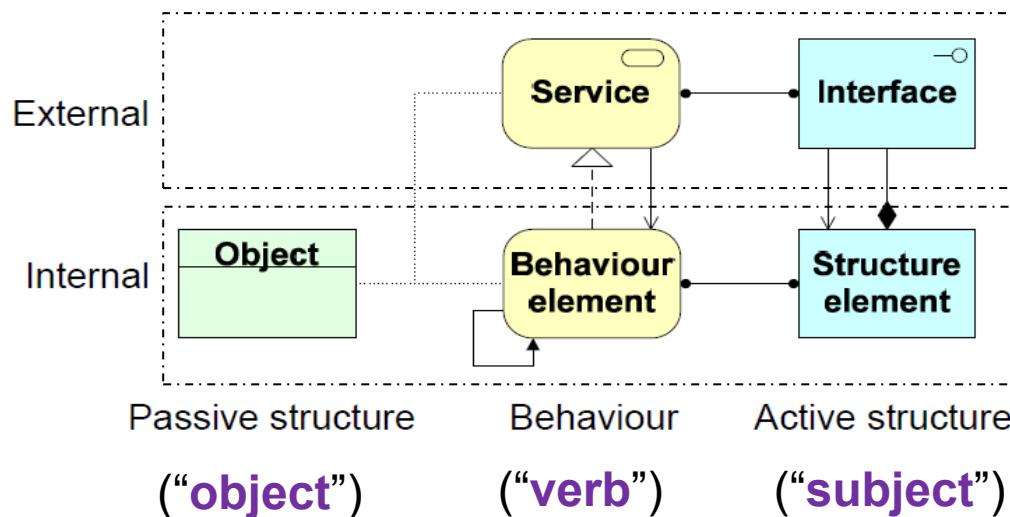
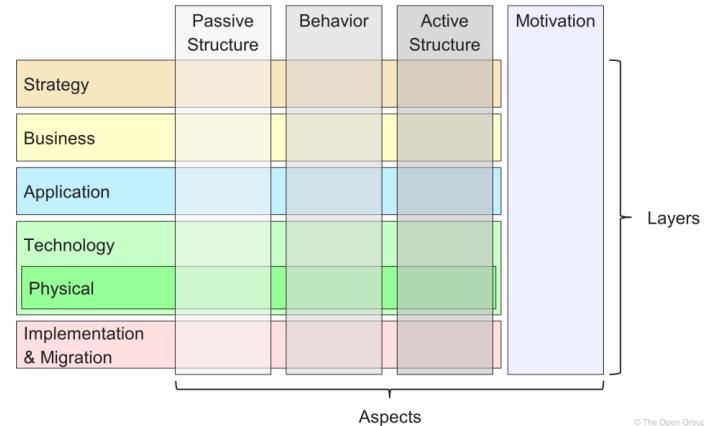
A generic view of layers...



ArchiMate Framework – Layers

All the layers share a similar overall structure, as the **Concepts** and **Relationships** of each layer are the same, but have different granularity and nature.

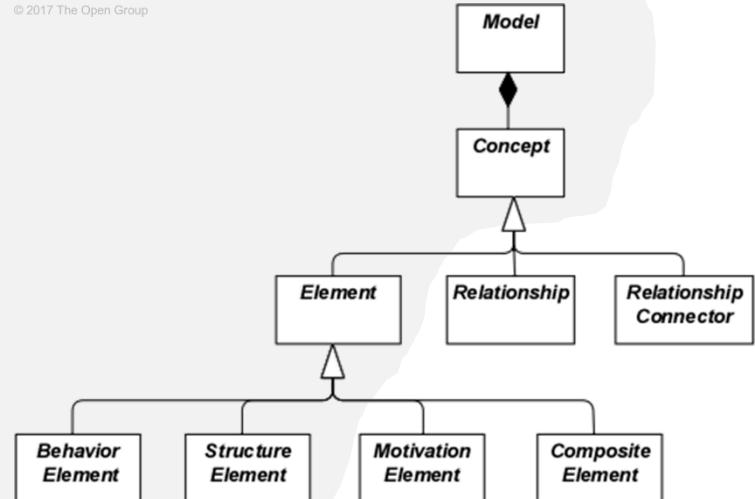
- An internal active structure element represents an entity that is capable of performing behavior.
- An external active structure element, called an interface, represents a point of access where one or more services are provided to the environment.
- An internal behavior element represents a unit of activity performed by one or more active structure elements.
- An external behavior element, called a service, represents an explicitly defined exposed behavior.



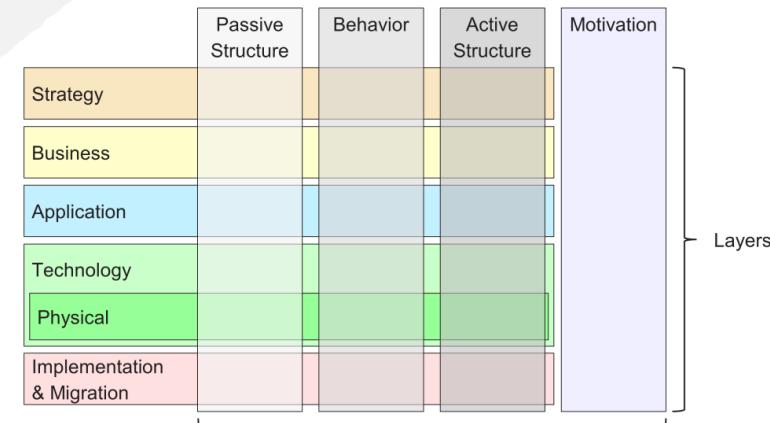
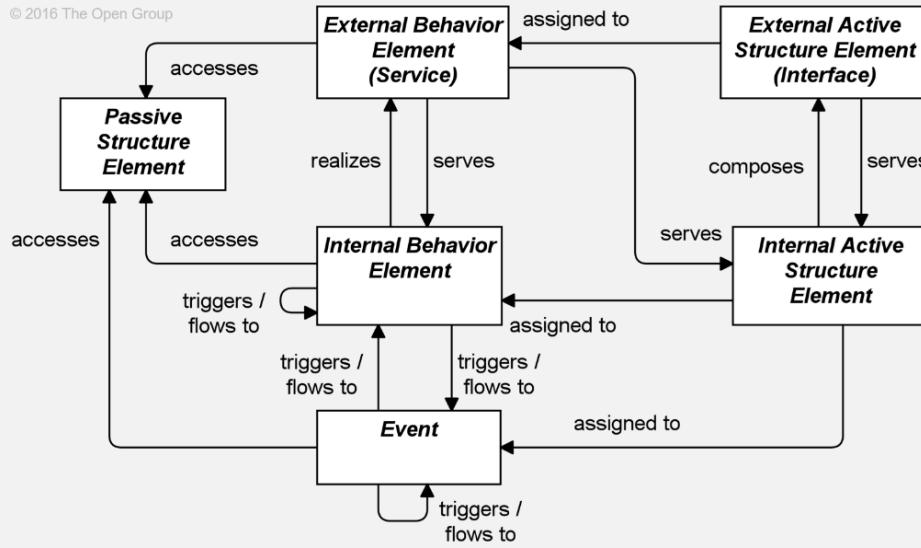
ArchiMate Framework - Aspects

- Structural elements can be subdivided into:
 - Passive structure elements
 - Active structure elements, further subdivided into:
 - External active structure elements (**interfaces**);
 - Internal active structure elements.
- Behavior elements can be subdivided into:
 - Internal behavior elements;
 - External behavior elements (**services**);
 - Events.

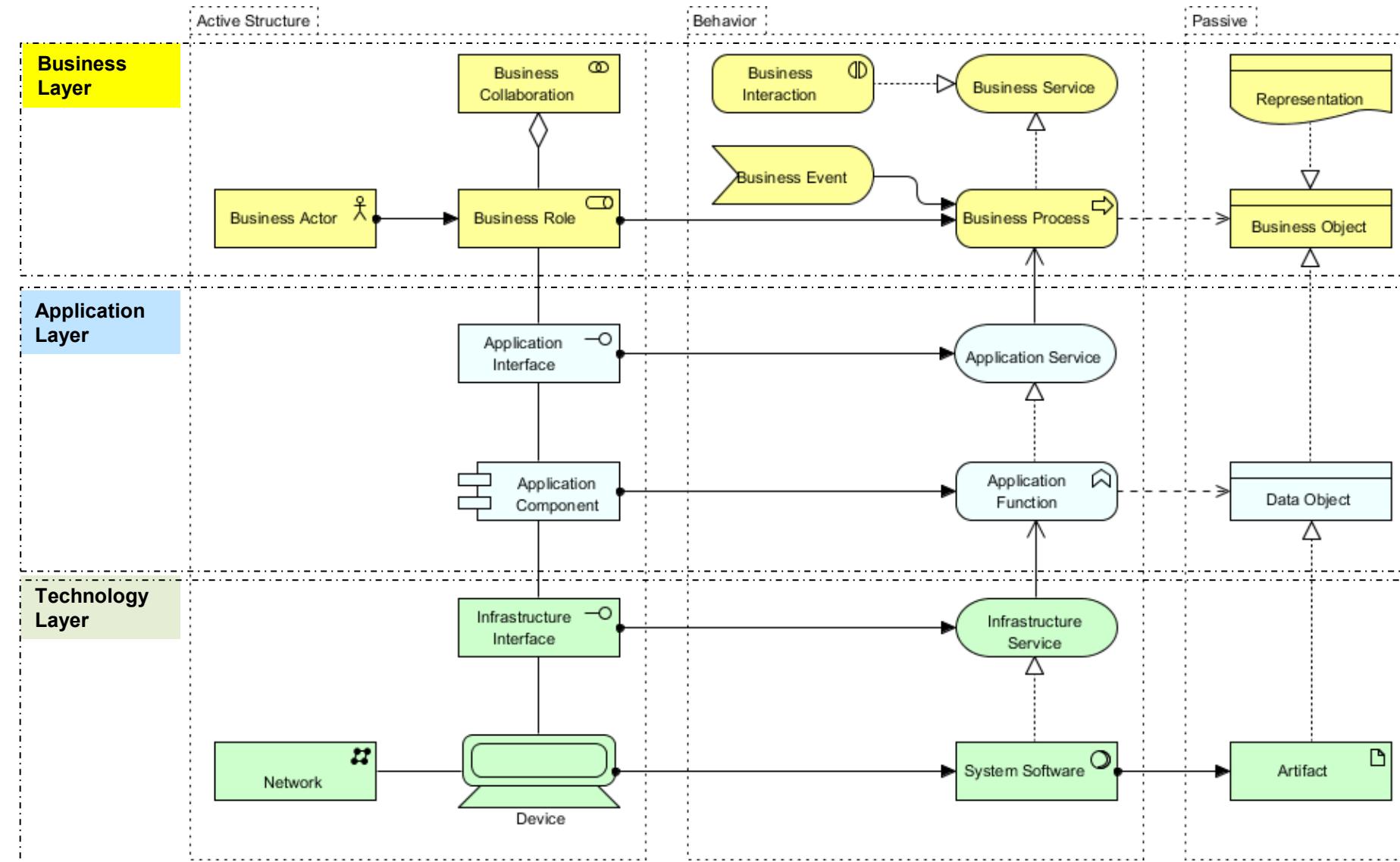
© 2017 The Open Group



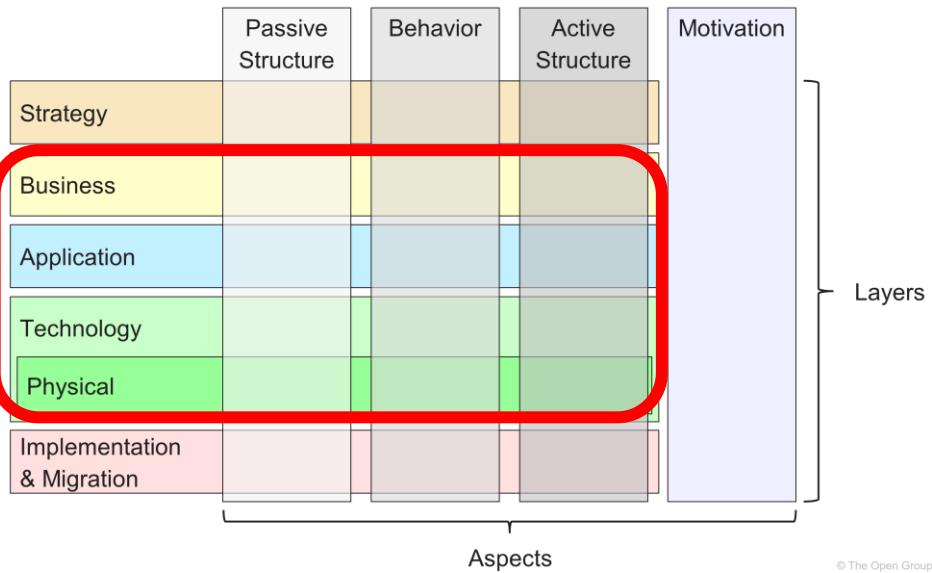
© 2016 The Open Group



ArchiMate Framework – Aspects x Layers

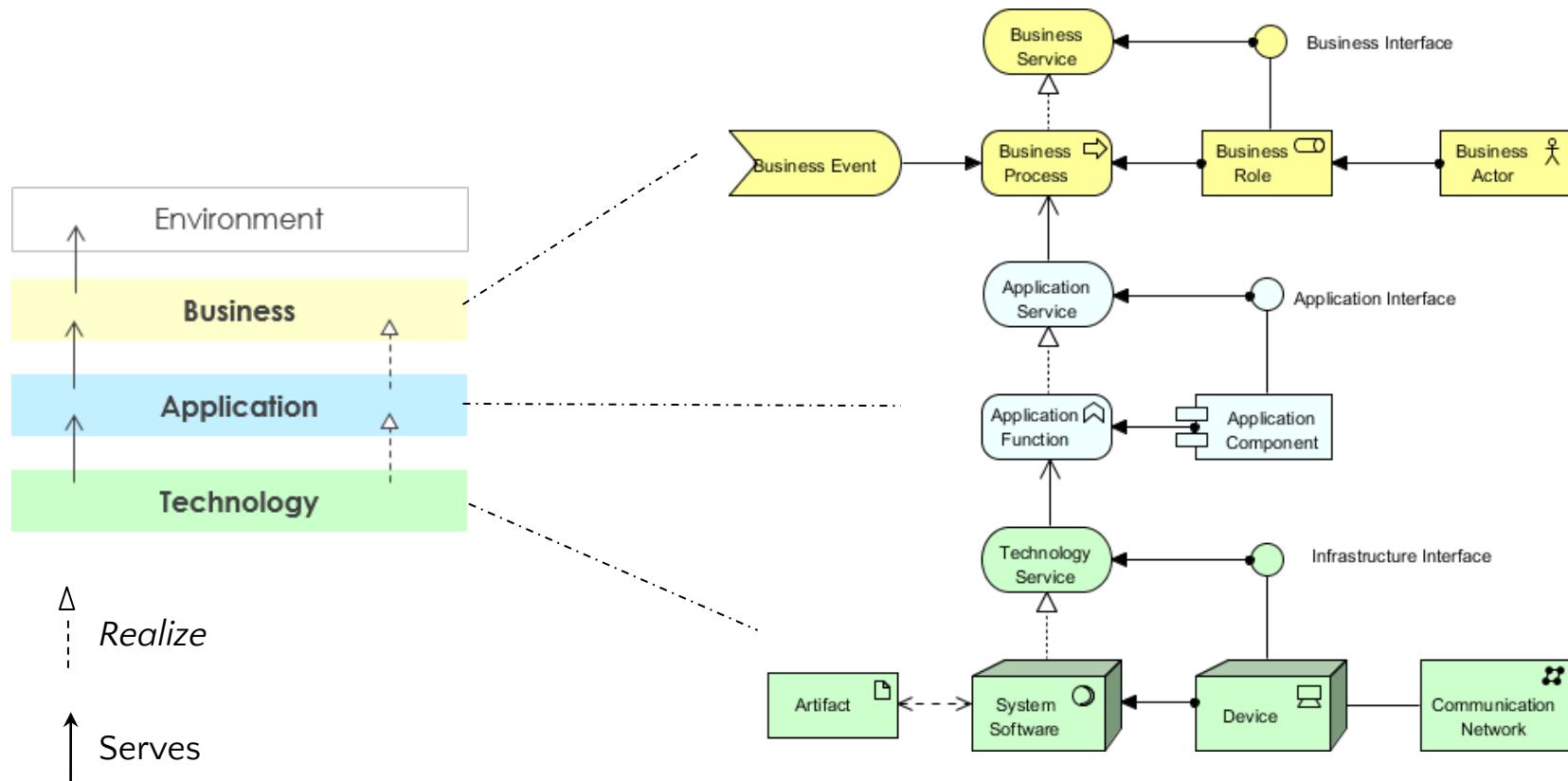


ArchiMate Framework – Core Layers



- **Business Layer:** **business services offered to customers**, which are **realized** in the organization by **business processes** performed by **business actors**.
- **Application Layer:** **application services that support the business**, and the **applications that realize them**.
- **Technology Layer:** **technology services** such as processing, storage, and communication services needed to run the applications, and the computer and communication hardware and system software that realize those services.

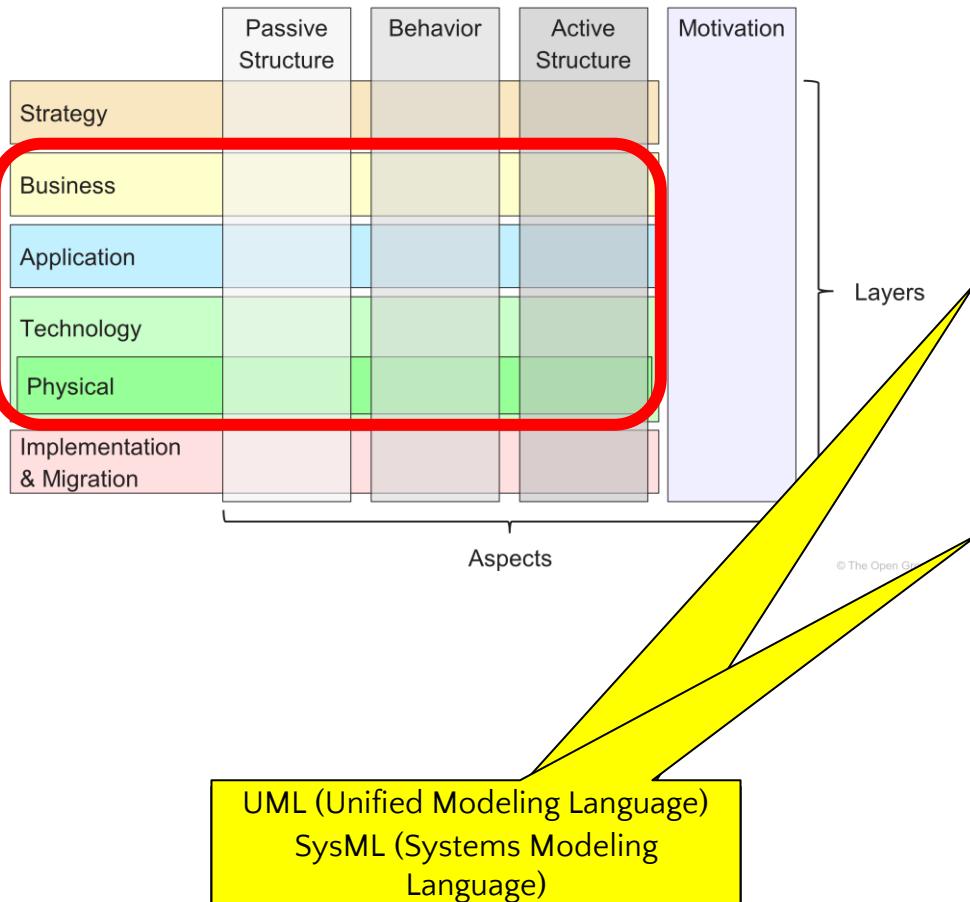
ArchiMate Framework – Core Layers



<https://www.visual-paradigm.com/guide/archimate/what-is-archimate>

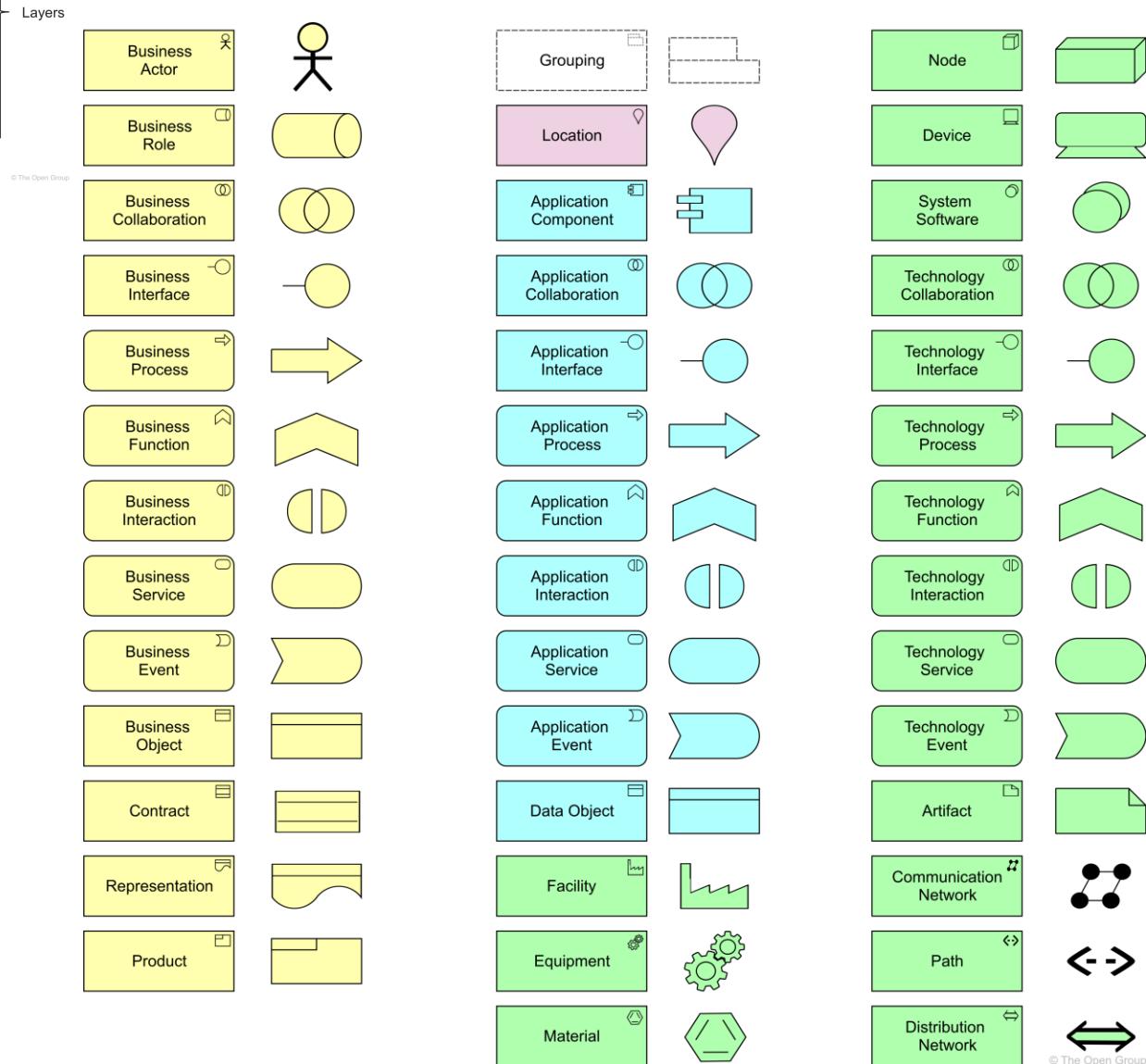
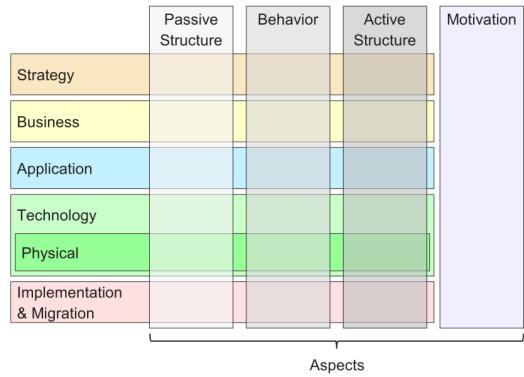
By The Way... concerning our future modules:

BPMN (Business Process Modeling Notation)...



- **Business Layer:** **business services offered to customers**, which are **realized** in the organization by **business processes** performed by **business actors**.
- **Application Layer:** **application services that support the business**, and the **applications that realize them**.
- **Technology Layer:** **technology services** such as processing, storage, and communication services needed to run the applications, and the computer and communication hardware and system software that realize those services.

ArchiMate Framework – Core Elements



© The Open Group

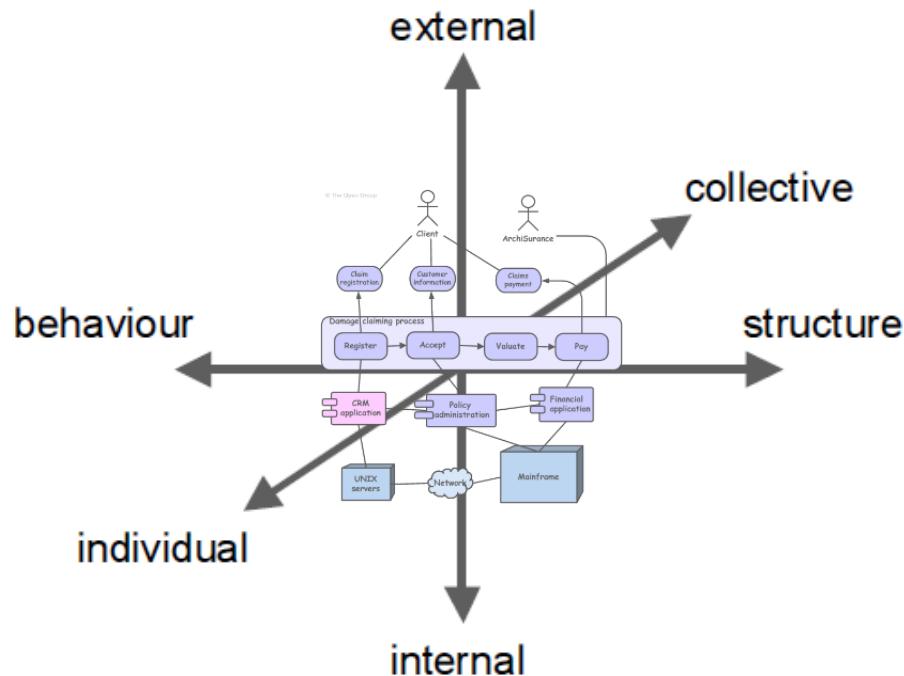


DEI
DEPARTAMENTO
DE ENGENHARIA INFORMÁTICA
TÉCNICO LISBOA

ArchiMate

...modelling dimensions...

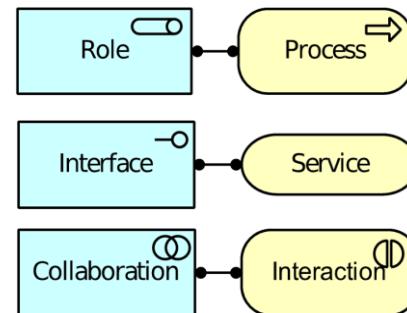
ArchiMate Modeling Dimensions



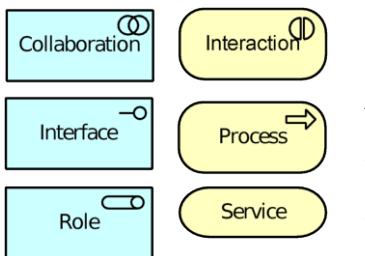
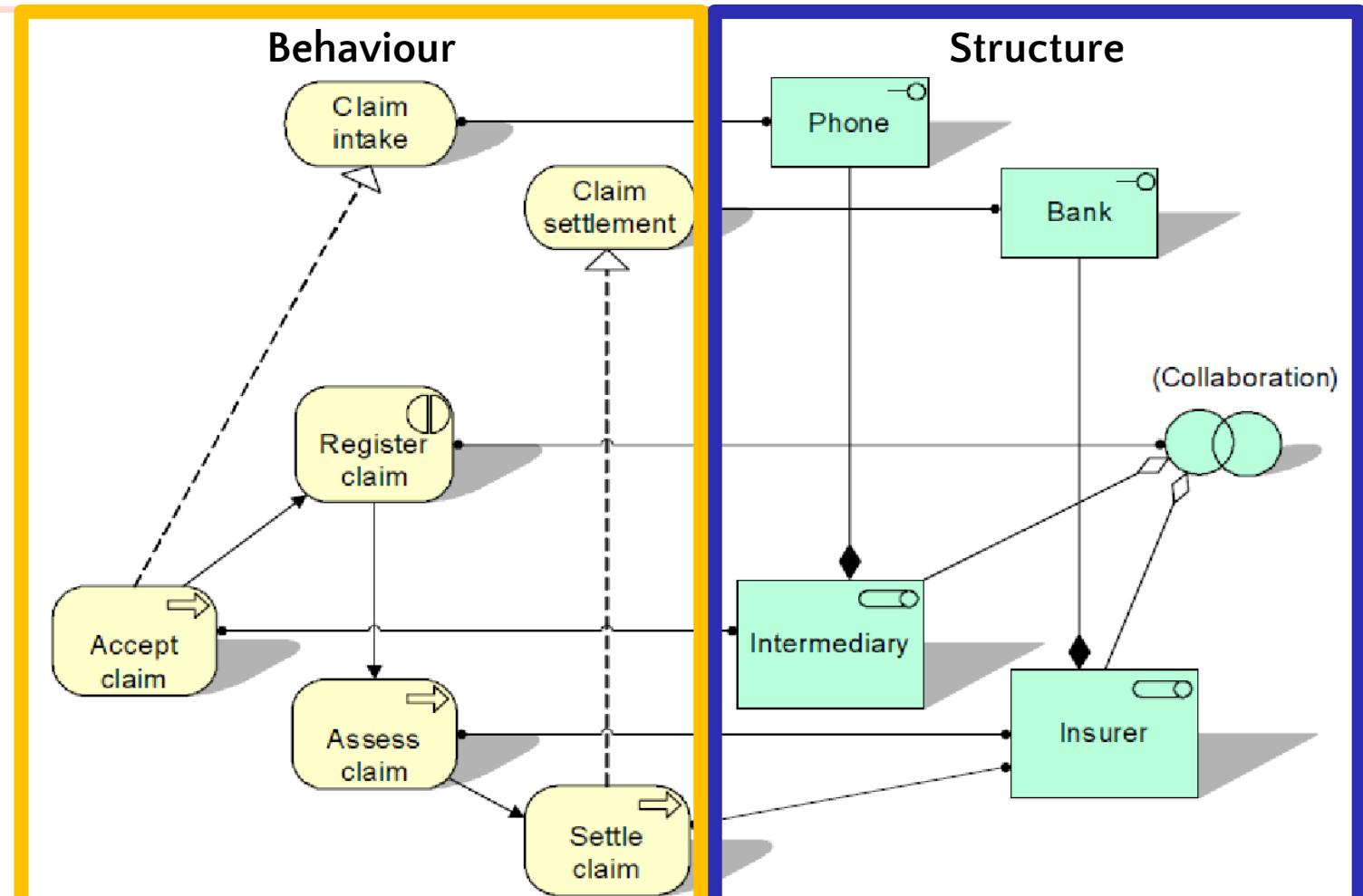
- x. Behaviour \Leftrightarrow Structure
- y. External \Leftrightarrow Internal
- z. Individual \Leftrightarrow Collective

ArchiMate Modeling Dimensions: Behavior – Structure

- Separates dynamic aspects from static.
- Assigns behavior to **structural** elements.
- A structural element can be active or passive.
 - Active: element with capacity to act, ie to demonstrate behavior.
 - Passive: object on which the behavior is applied.
- Assignments between behavior and structure:
 - Process – Role
 - Service – Interface
 - Interaction – Collaboration



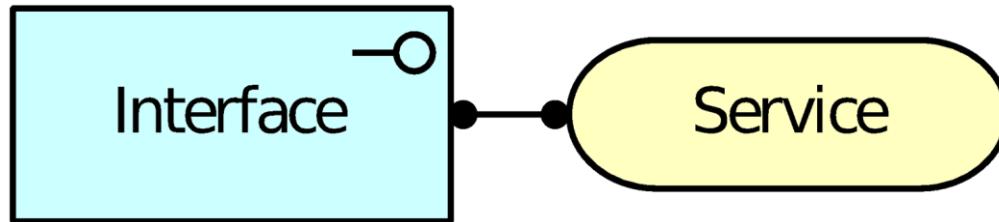
Example



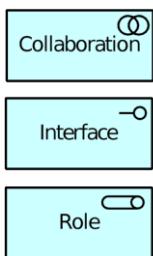
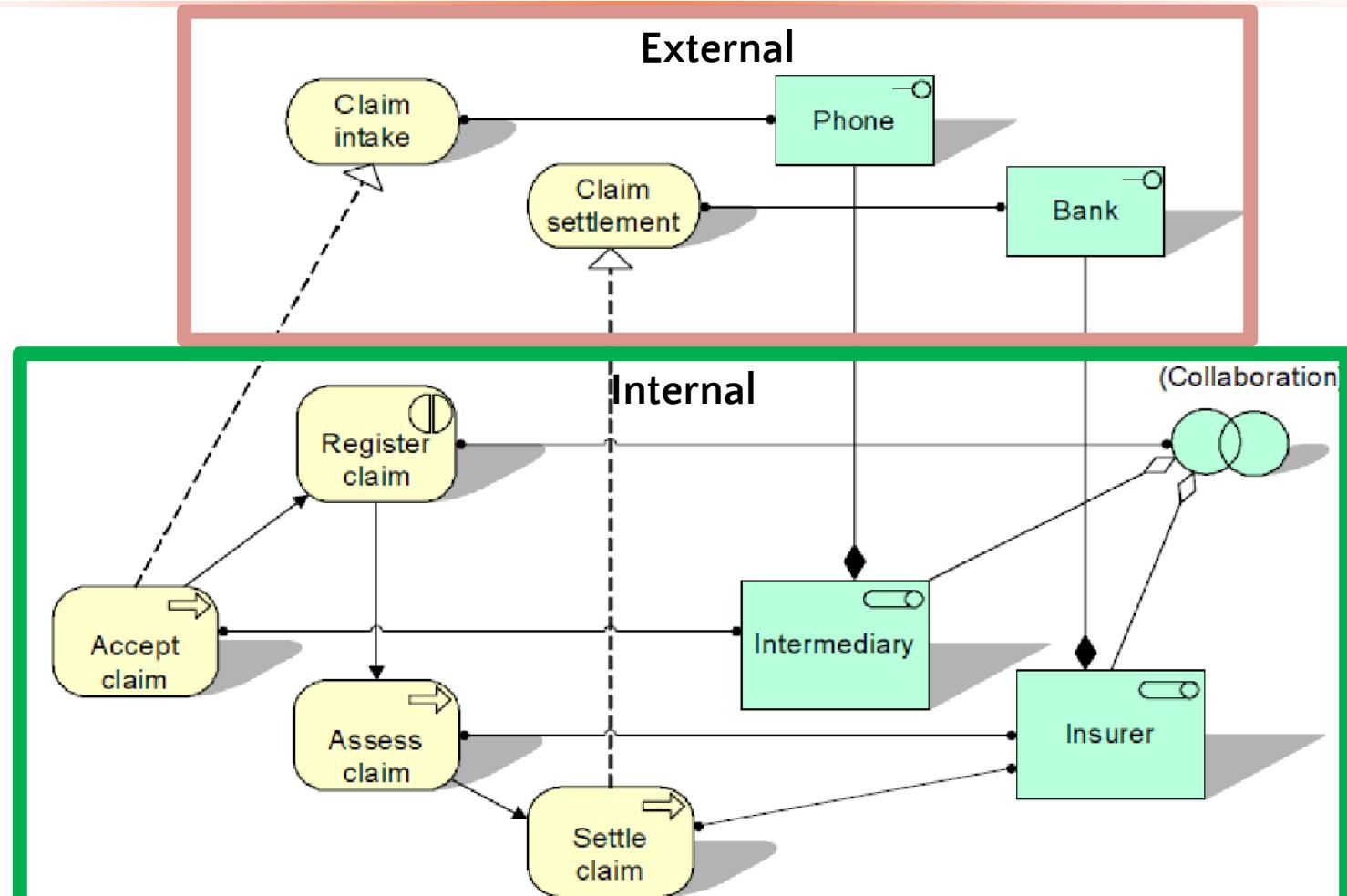
- → Assignment
- ◇ → Aggregation
- ◆ → Composition
- △ ← Realization

Archimate Modeling Dimensions: External – Internal

- Separate **external** and **internal** aspects.
- A service represents a structural functionality the systems that can be exposed to the exterior
- The behaviour of services are exposed by interfaces

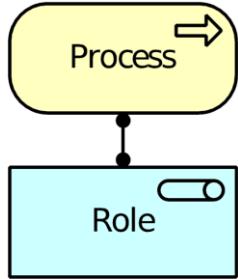


Example



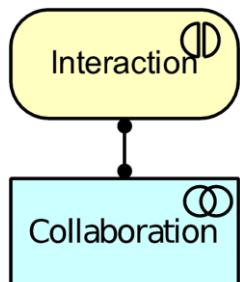
- Assignment
- ◆ Aggregation
- ◆ Composition
- ▷— Realization

ArchiMate Modeling Dimensions: Individual – Collective



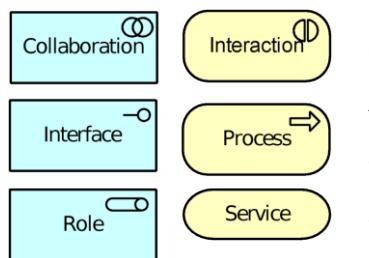
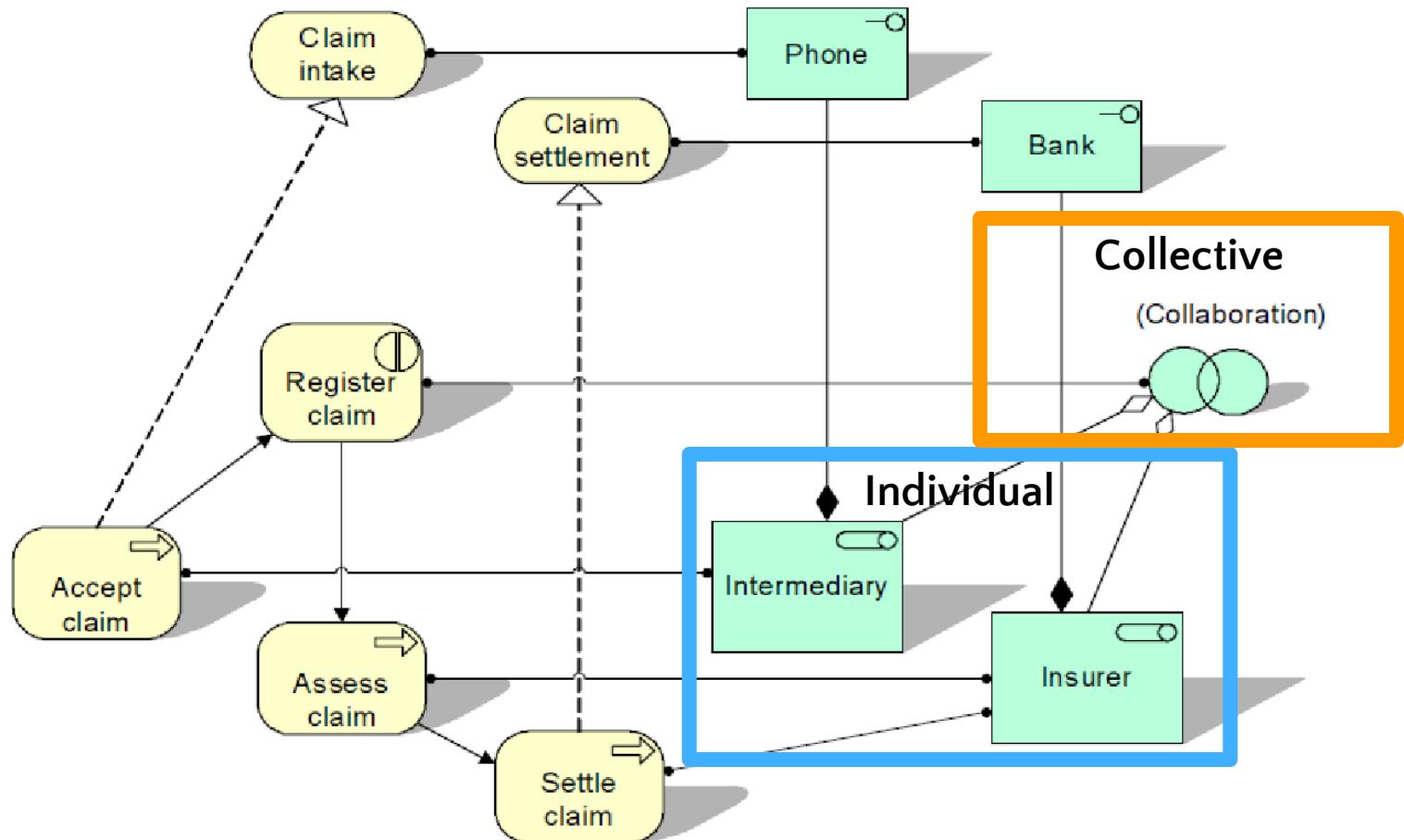
- Separation of the behaviours of the **individual** and **collective** structural elements
- A **collective** structural element represents an interaction with multiple elements.

- To the exterior, it must be irrelevant if the behaviour is **individual** or **collective**...



- Internally:
 - **Interaction** is a behaviour element
 - **Collaboration** is a structural element

Example

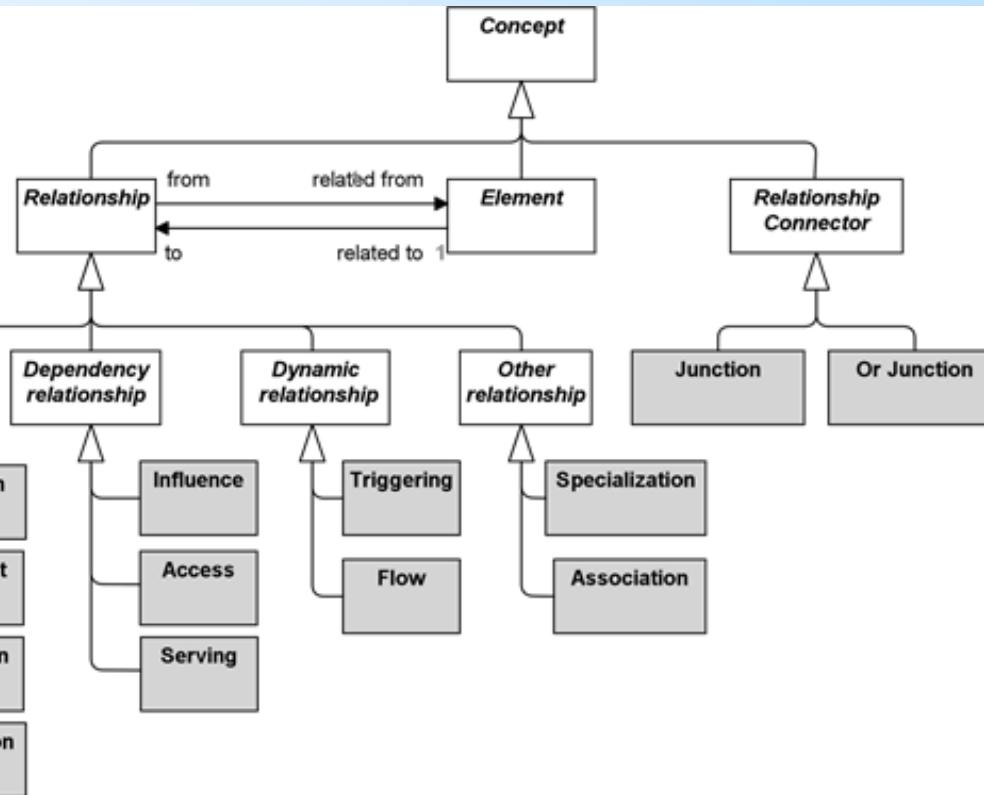


- Assignment
- ◇ Aggregation
- ◆ Composition
- △ Realization



ArchiMate

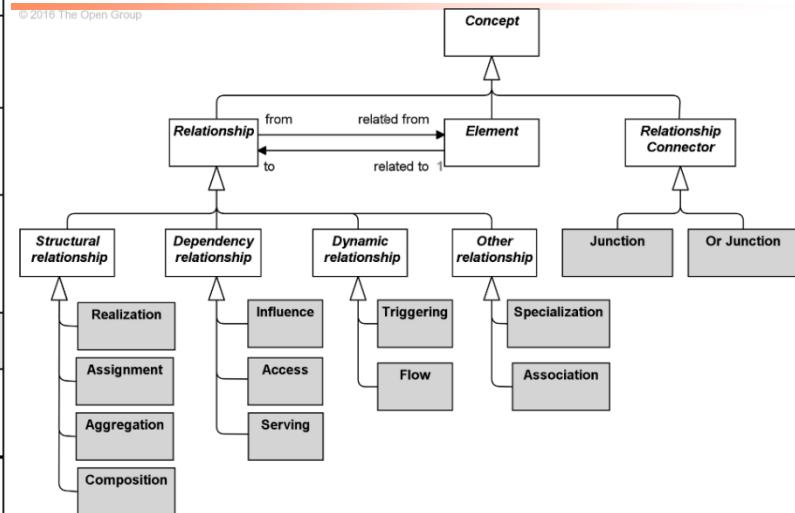
© 2016 The Open Group



...the
relationships...

ArchiMate Relationships

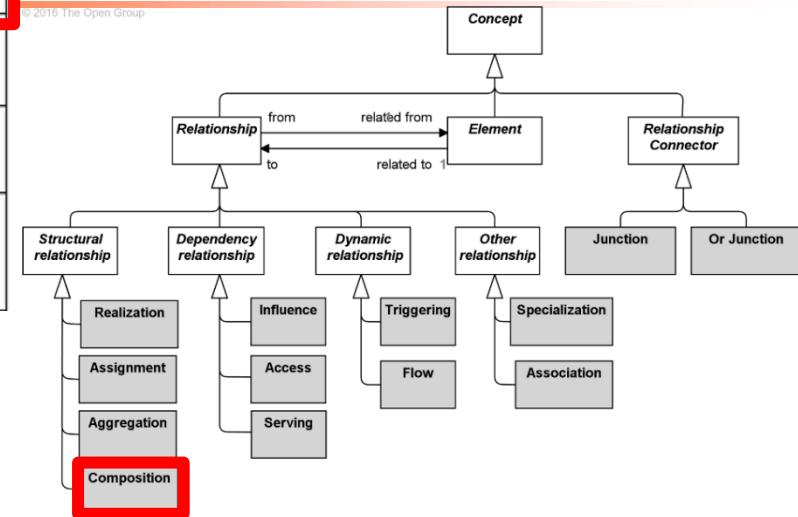
© 2016 The Open Group



Structural Relationships		Notation
Composition	Indicates that an element consists of one or more other elements.	◆—
Aggregation	Indicates that an element groups a number of other elements.	◇—
Assignment	Expresses the allocation of responsibility, performance of behavior, or execution.	●—►
Realization	Indicates that an entity plays a critical role in the creation, achievement, sustenance, or operation of a more abstract entity.►
Dependency Relationships		Notation
Serving	Models that an element provides its functionality to another element.	—►
Access	Models the ability of behavior and active structure elements to observe or act upon passive structure elements.► ↔
Influence	Models that an element affects the implementation or achievement of some motivation element.	—+/-►
Dynamic Relationships		Notation
Triggering	Describes a temporal or causal relationship between elements.	—►
Flow	Transfer from one element to another.	-----►
Other Relationships		Notation
Specialization	Indicates that an element is a particular kind of another element.	—►
Association	Models an unspecified relationship, or one that is not represented by another ArchiMate relationship.	—
Junction	Used to connect relationships of the same type.	● (And) Junction ○ Or Junction

Structural Relationships		Notation
Composition	Indicates that an element consists of one or more other elements.	
Aggregation	Indicates that an element groups a number of other elements.	
Assignment	Expresses the allocation of responsibility, performance of behavior, or execution.	
Realization	Indicates that an entity plays a critical role in the creation, achievement, sustenance, or operation of a more abstract entity.	

© 2016 The Open Group

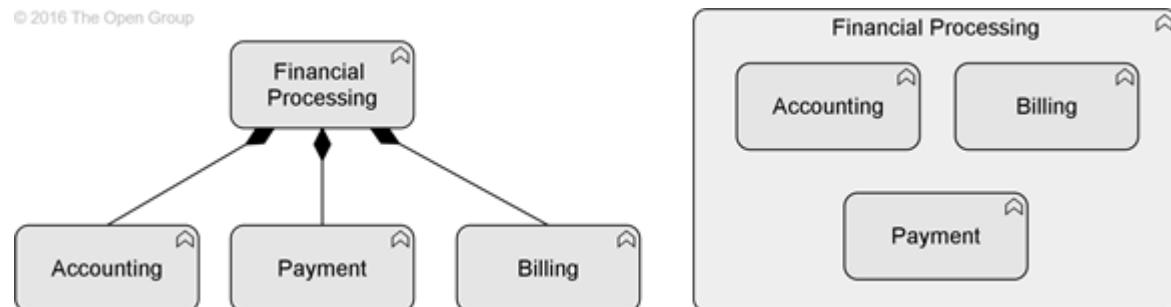


Structural relationships represent the ‘static’ coherence within an architecture:

- The ‘from’ side of the relationship is always an element;
- the ‘to’ side of the relationship may in some cases also be another relationship.

The composition relationship indicates that an element consists of one or more other elements:

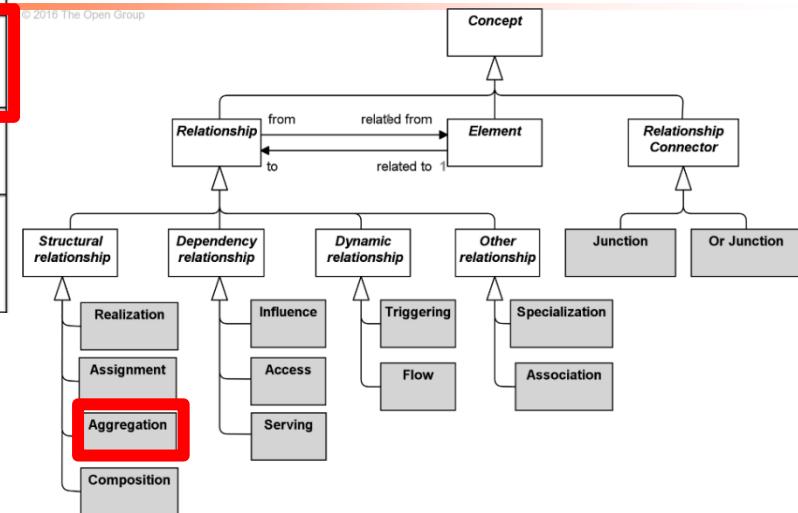
A composition relationship is always allowed between two instances of the same element type.



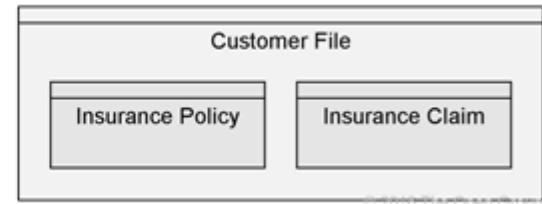
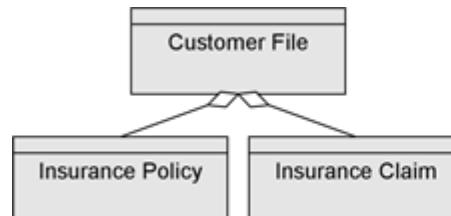
As an alternative to the graphical notations proposed in this section, structural relationships may also be expressed by means of nesting of the composed concept within the composing element. Note, however, that this can lead to ambiguous models, in case multiple structural relationships are allowed between these elements.

ArchiMate Relationships

Structural Relationships		Notation
Composition	Indicates that an element consists of one or more other elements.	
Aggregation	Indicates that an element groups a number of other elements.	
Assignment	Expresses the allocation of responsibility, performance of behavior, or execution.	
Realization	Indicates that an entity plays a critical role in the creation, achievement, sustenance, or operation of a more abstract entity.	



The aggregation relationship indicates that an element groups a number of other elements:

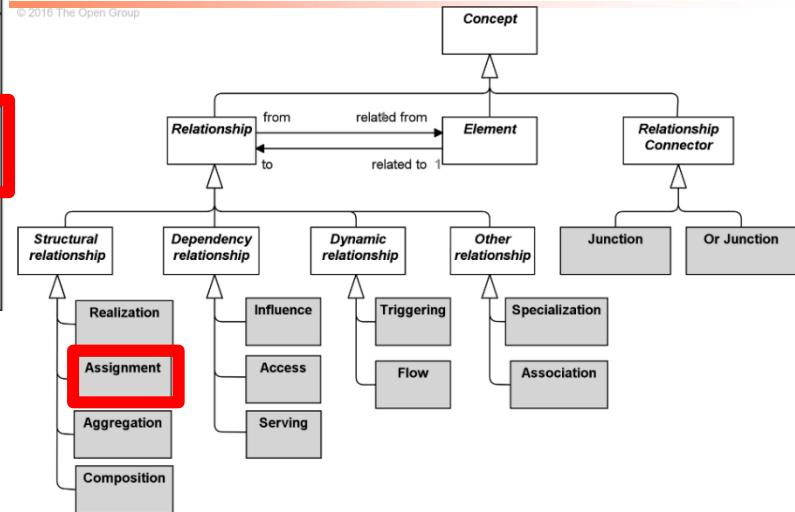


An aggregation relationship is always allowed between two instances of the same element type.

ArchiMate Relationships

Structural Relationships		Notation
Composition	Indicates that an element consists of one or more other elements.	
Aggregation	Indicates that an element groups a number of other elements.	
Assignment	Expresses the allocation of responsibility, performance of behavior, or execution.	
Realization	Indicates that an entity plays a critical role in the creation, achievement, sustenance, or operation of a more abstract entity.	

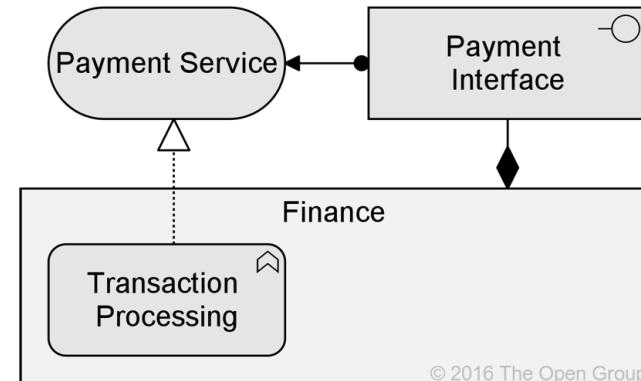
© 2016 The Open Group



The assignment relationship links:

- active structure elements with units of behavior that are performed by them,
- business actors with business roles that are fulfilled by them
- nodes with technology objects.

It can, for example, relate an internal active structure element with an internal behavior element, an interface with a service, or a node with a technology object:



© 2016 The Open Group

This model in the example includes the two ways to express the assignment relationship:

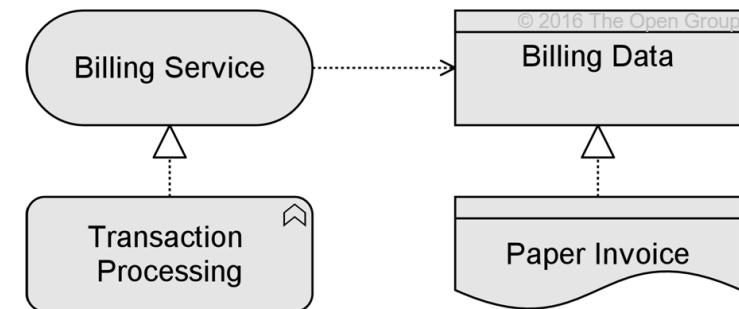
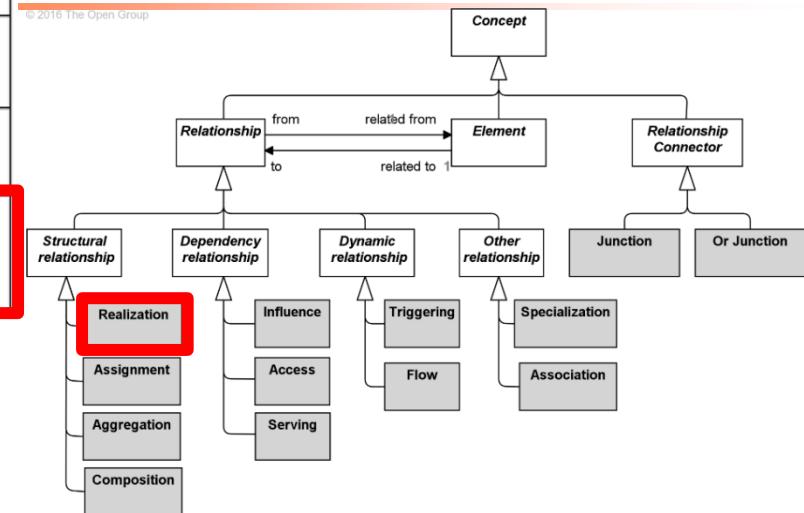
- (nesting) The **Finance** active structure element is assigned to the **Transaction Processing** function
- (relationship) The **Payment Interface** is assigned to the **Payment Service**.

ArchiMate Relationships

Structural Relationships		Notation
Composition	Indicates that an element consists of one or more other elements.	
Aggregation	Indicates that an element groups a number of other elements.	
Assignment	Expresses the allocation of responsibility, performance of behavior, or execution.	
Realization	Indicates that an entity plays a critical role in the creation, achievement, sustenance, or operation of a more abstract entity.	

The realization relationship indicates that more abstract entities (“what” or “logical”) are realized by means of more tangible entities (“how” or “physical”).

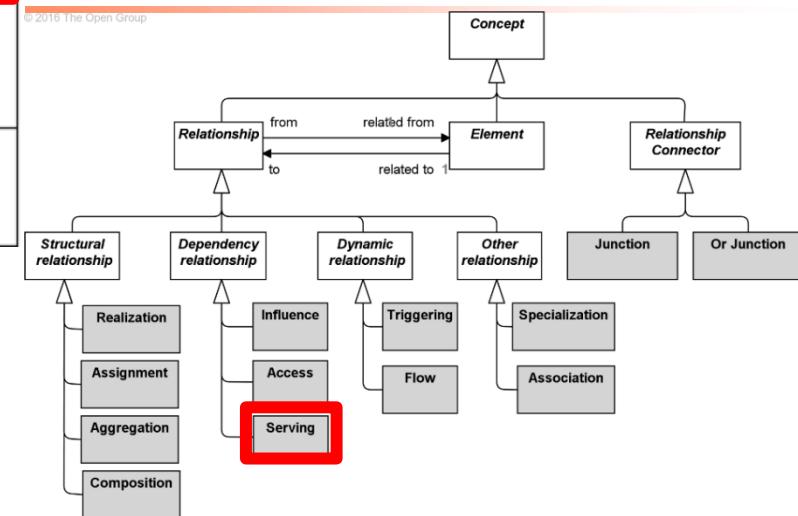
The realization relationship is used to model run-time realization; for example, that a business process realizes a business service, and that a data object realizes a business object, an artifact realizes an application component, or a core element realizes a motivation element:



This model in the example illustrates two ways to use the realization relationship:

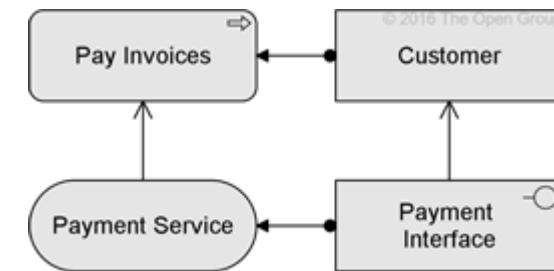
- A Transaction Processing function realizes a Billing Service;
- The Billing Data object is realized by the representation Paper Invoice.

Dependency Relationships		Notation
Serving	Models that an element provides its functionality to another element.	→
Access	Models the ability of behavior and active structure elements to observe or act upon passive structure elements.	↔ →
Influence	Models that an element affects the implementation or achievement of some motivation element.	↔ +/→



Dependency relationships describe how elements support or are used by other elements.

The serving relationship describes how the services or interfaces offered by a behavior or active structure element serve entities in their environment. This relationship is applied for both the behavior aspect and the active structure aspect:



This model in the example illustrates the serving relationship. The Payment Interface serves the Customer, while the Payment Service serves the Pay Invoices process of that customer.

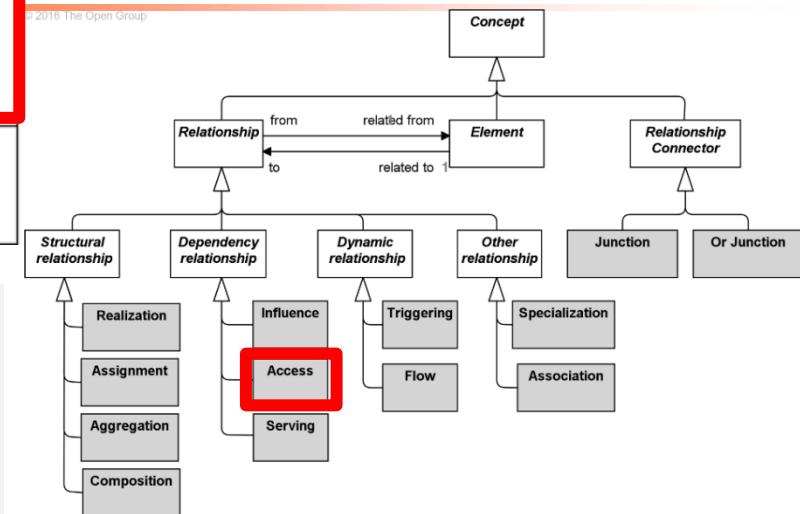
ArchiMate Relationships

Dependency Relationships		Notation
Serving	Models that an element provides its functionality to another element.	→
Access	Models the ability of behavior and active structure elements to observe or act upon passive structure elements.	↔, →, ←
Influence	Models that an element affects the implementation or achievement of some motivation element.	→/+/-→

The access relationship indicates that a process, function, interaction, service, or event “does something” with a passive structure element; e.g., create a new object, read data from the object, write or modify the object data, or delete the object.

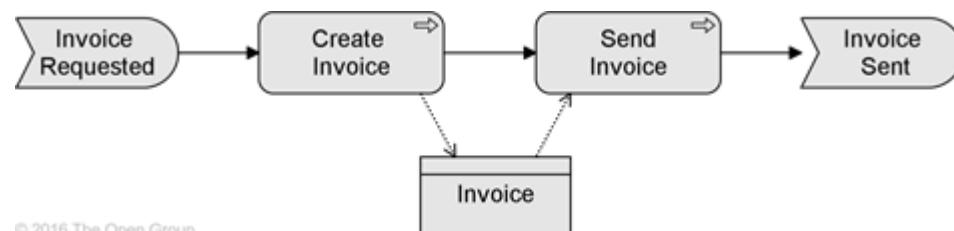
The relationship can also be used to indicate that the object is just associated with the behavior; e.g., it models the information that comes with an event, or the information that is made available as part of a service.

The arrow head, if present, indicates the direction of the flow of information:



This model in the example illustrates the access relationship:

- The Create Invoice sub-process writes/creates the Invoice object;
- The Send Invoice sub-process reads that object.



ArchiMate Relationships

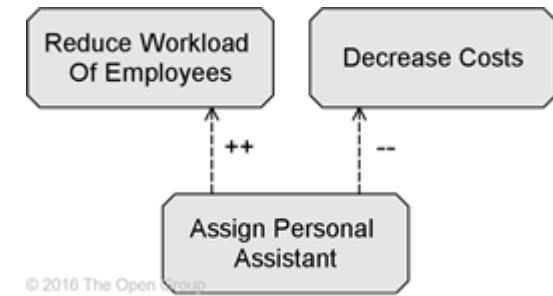
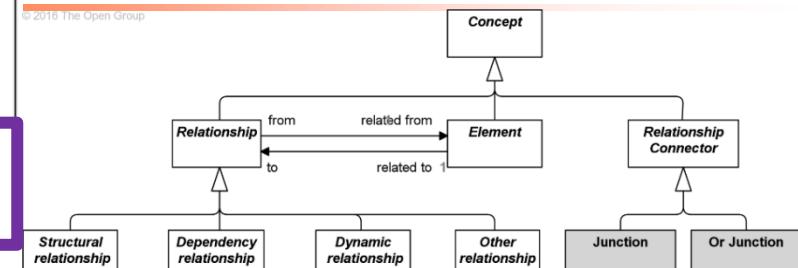
Dependency Relationships		Notation
Serving	Models that an element provides its functionality to another element.	→
Access	Models the ability of behavior and active structure elements to observe or act upon passive structure elements.	↔ → ↔
Influence	Models that an element affects the implementation or achievement of some motivation element.	↔ +/→

The influence relationship is used to describe that some architectural element influences achievement or implementation of a **motivation** element, such as a goal or a principle. In general, a motivation element is realized to a certain degree. For example, consistently satisfying the principle ‘serve customers wherever they are’ will help making the goal ‘increase market share’ come true. In other words, the principle contributes to the goal. In turn, to implement the principle ‘serve customers wherever they are’, it may be useful to impose a requirement of ‘24x7 web availability’ on some customer-facing application component. This can be modeled as a requirement that has an influence on that principle, and as an application component that in turn influences the requirement. Consistently modeling these dependencies with an influence relationship yields a traceable motivational path that explains why (in this example) a certain application component contributes to the corporate goal to ‘increase market share’.

Moreover, an influence relationship can be used to model either:

- The fact that an element positively contributes to the achievement or implementation of some motivation element, or
- The fact that an element negatively influences – i.e., prevents or counteracts – such achievement

Attributes can be used to indicate the sign and/or strength of the influence. The choice of possible attribute values is left to the modeler; e.g., {++, +, 0, -, -} or [0..10]. By default, the influence relationship models a contribution with unspecified sign and strength:



This model in the example illustrates the use of the influence relationship to model the different effects of the same motivation element, Assign Personal Assistant. This has a strongly positive influence on Reduce Workload Of Employees, but a strongly negative influence on Decrease Costs.

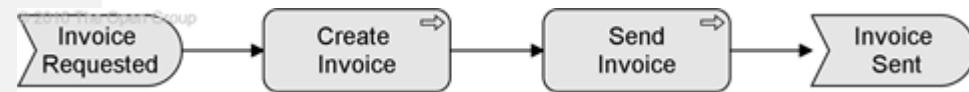
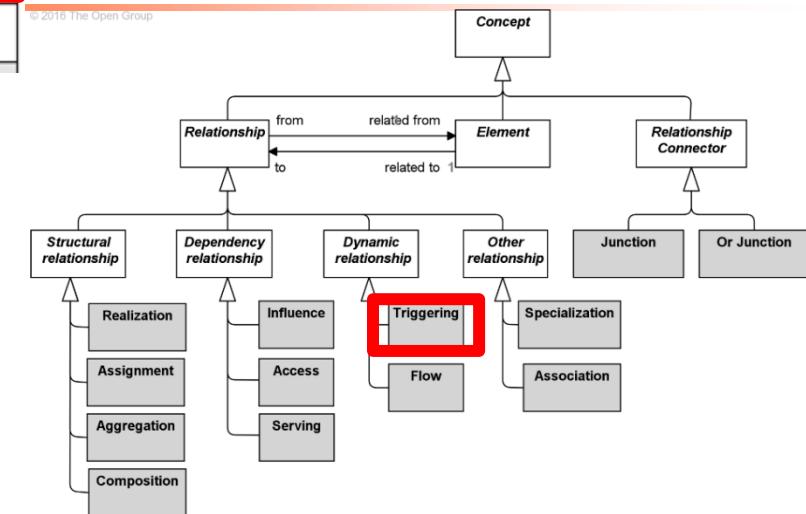
Dynamic Relationships		Notation
Triggering	Describes a temporal or causal relationship between elements.	→
Flow	Transfer from one element to another.	→→

The dynamic relationships describe temporal dependencies between elements within the architecture

The triggering relationship is used to model the temporal or causal precedence of behavior elements in a process.

The usual interpretation of a triggering relationship is that the source element should be completed before the target element can start, although weaker interpretations are also permitted.

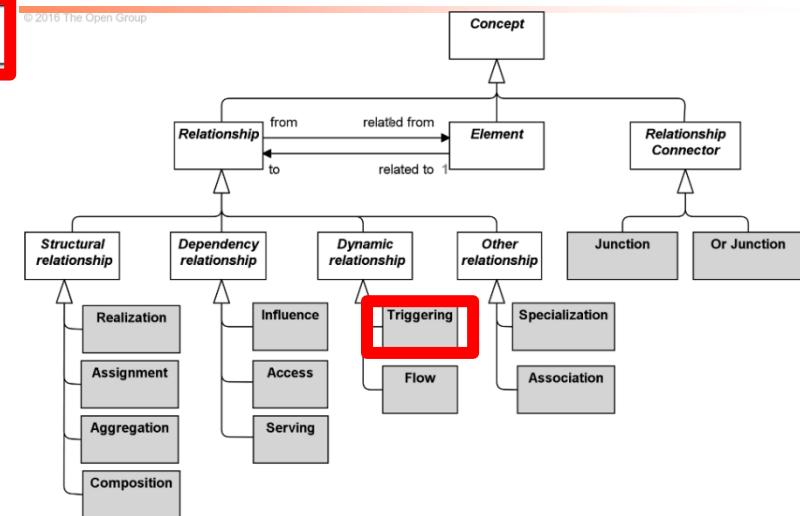
Note that this does not necessarily represent that one behavior element actively starts another; a traffic light turning green also triggers the cars to go through the intersection:



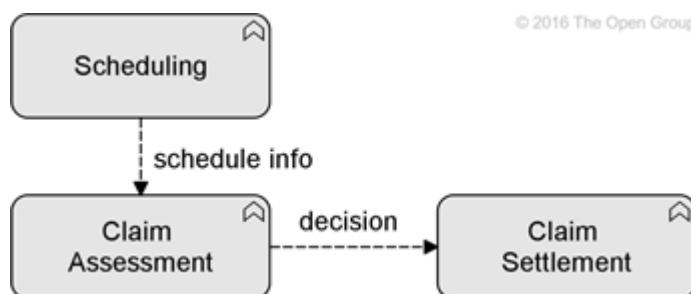
This model in the example illustrates that triggering relationships are mostly used to model causal dependencies between (sub-)processes and/or events.

ArchiMate Relationships

Dynamic Relationships		Notation
Triggering	Describes a temporal or causal relationship between elements.	→
Flow	Transfer from one element to another.	→



The flow relationship is used to model the flow of, for example, information, goods, or money between behavior elements. A flow relationship does not imply a causal relationship:



This model in the example shows a Claim Assessment function, which forwards decisions about the claims to the Claim Settlement function. In order to determine the order in which the claims should be assessed, Claim Assessment makes use of schedule information received from the Scheduling function.

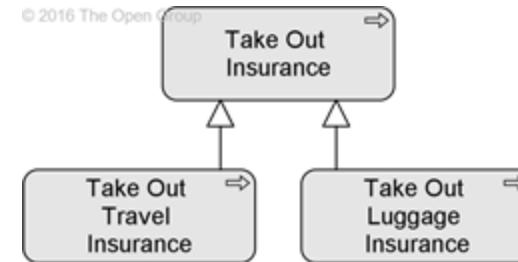
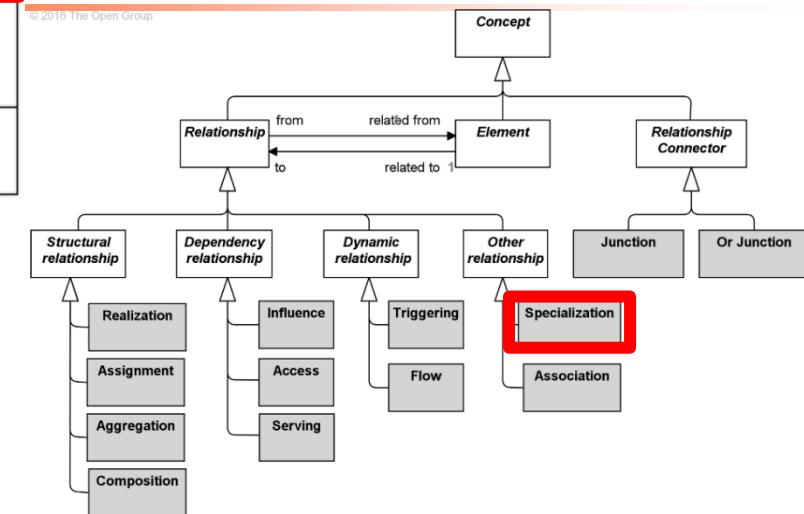
ArchiMate Relationships

Other Relationships		Notation
Specialization	Indicates that an element is a particular kind of another element.	→
Association	Models an unspecified relationship, or one that is not represented by another ArchiMate relationship.	—
Junction	Used to connect relationships of the same type.	(And) Junction Or Junction

The specialization relationship has been inspired by the generalization relationship in UML class diagrams, but is applicable to specialize a wider range of concepts.

The specialization relationship can relate any instance of a concept with another instance of the same concept.

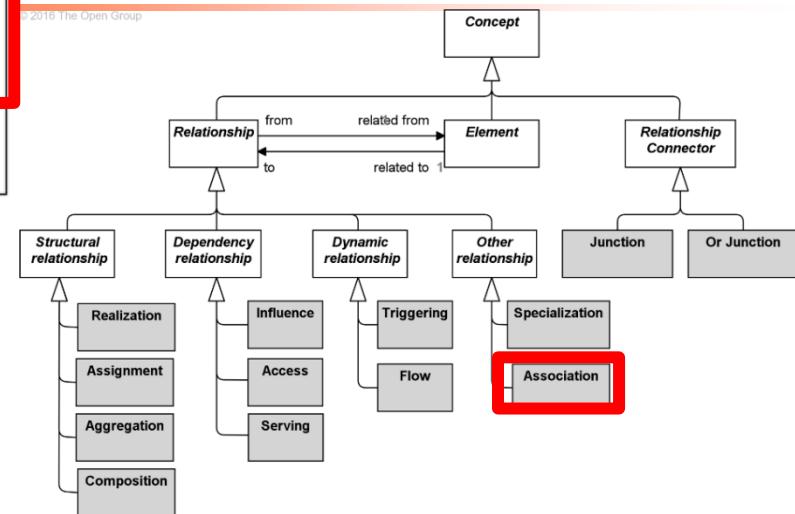
A specialization relationship is always allowed between two instances of the same element:



This model in the example illustrates the use of the specialization relationship for a process. In this case the Take Out Travel Insurance and Take Out Luggage Insurance processes are a specialization of a more generic Take Out Insurance process.

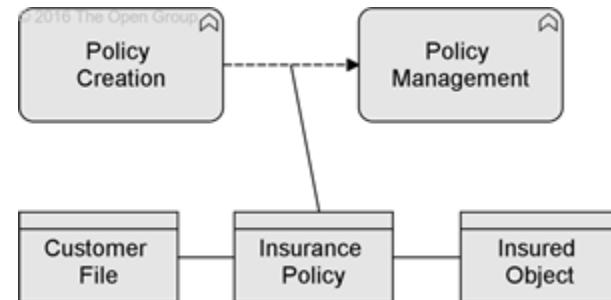
ArchiMate Relationships

Other Relationships		Notation
Specialization	Indicates that an element is a particular kind of another element.	→
Association	Models an unspecified relationship, or one that is not represented by another ArchiMate relationship.	—
Junction	Used to connect relationships of the same type.	(And) Junction Or Junction



An association relationship is always allowed between two elements, or between a relationship and an element.

The association relationship can be used when drawing a first high-level model where relationships are initially denoted in a generic way, and later refined to show more specific relationship types:



This model in the example illustrates a number of uses of the association relationship. It also shows an example of an association between a flow relationship and a passive structure element, to indicate the kind of information that is communicated between the two functions.

ArchiMate Relationships

Other Relationships		Notation
Specialization	Indicates that an element is a particular kind of another element.	→
Association	Models an unspecified relationship, or one that is not represented by another ArchiMate relationship.	—
Junction	Used to connect relationships of the same type.	(And) Junction Or Junction

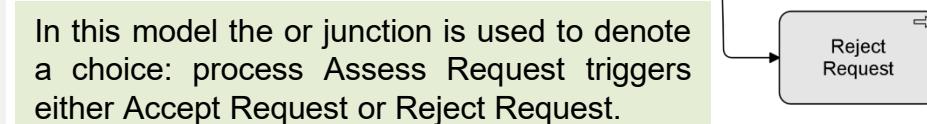
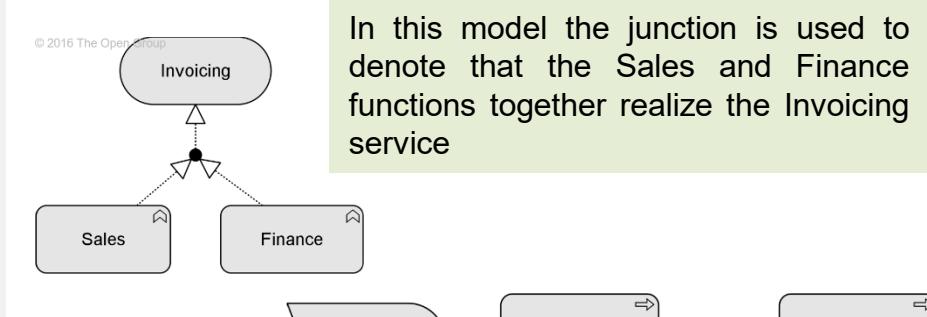
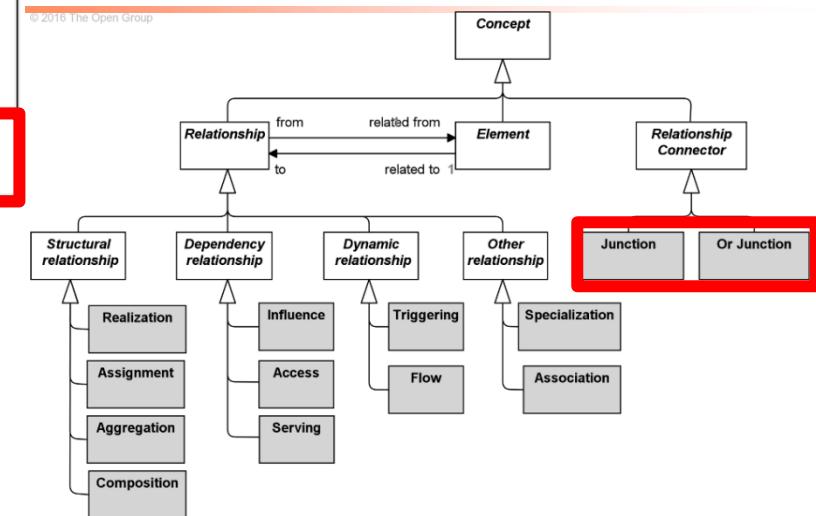
A junction is not an actual relationship in the same sense as the other relationships, but rather a relationship connector...

A junction may have multiple incoming relationships and one outgoing relationship, one incoming relationship and multiple outgoing relationships, or multiple incoming and outgoing relationships (the latter can be considered a shorthand of two subsequent junctions).

The relationships that can be used in combination with a junction are all the dynamic relationships, as well as assignment, realization, and association.

A junction is used to explicitly express that several elements *together* participate in the relationship (*and* junction) or that *one* of the elements participates in the relationship (*or* junction).

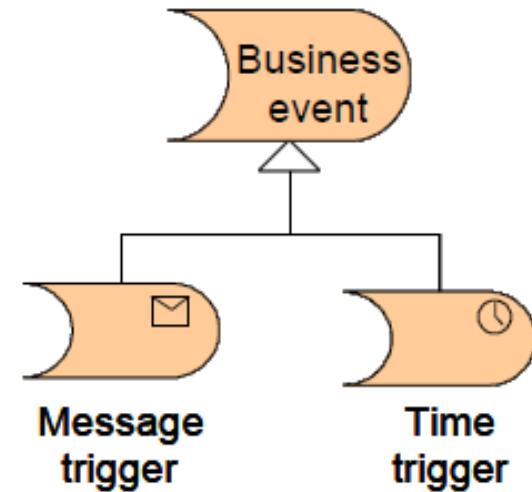
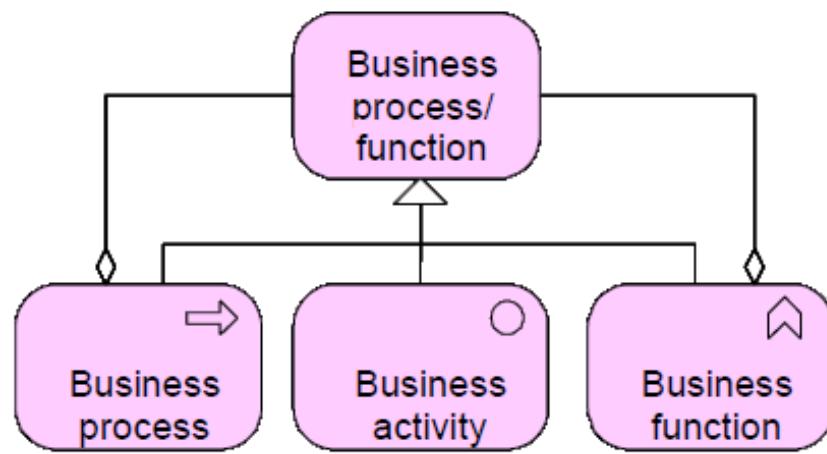
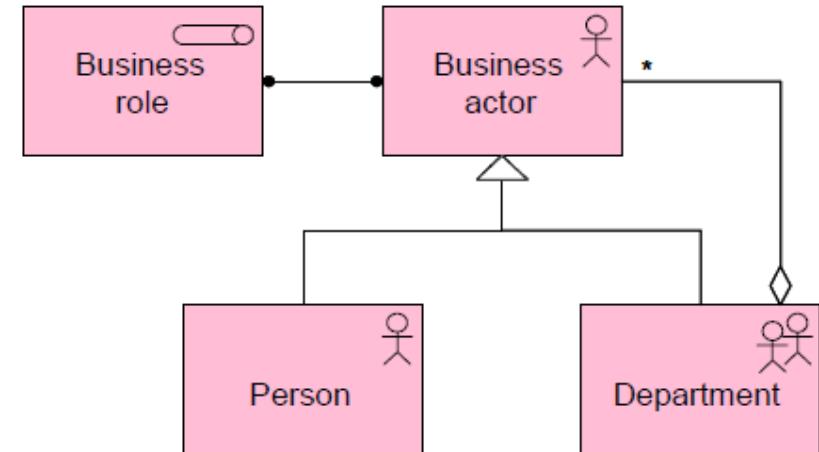
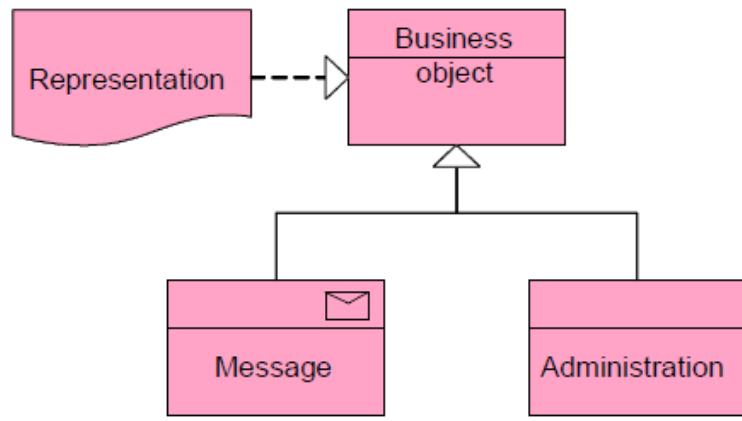
A junction should either have one incoming and more than one outgoing relationships, or more than one incoming and one outgoing. It is allowed to omit arrowheads of relationships leading into a junction:



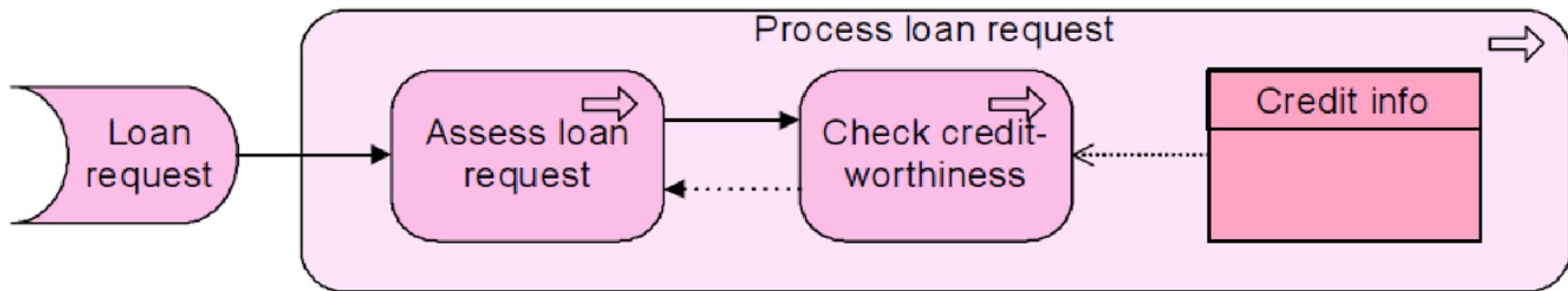
In this model the junction is used to denote that the Sales and Finance functions together realize the Invoicing service

In this model the or junction is used to denote a choice: process Assess Request triggers either Accept Request or Reject Request. The usual interpretation of two separate triggering relations, one from Assess Request to Accept Request and one from Assess Request to Reject Request, is that Assess Request triggers both of the other processes.

Examples of Specialization



Examples of Behavior Relationships



ArchiMate Core Framework: Derivation Rules

Derivation Rule for Structural and Dependency Relationships

Part of the language definition is an abstraction rule that states that two relationships that join at an intermediate element can be combined and replaced by the weaker of the two.

If two structural or dependency relationships $r:R$ and $s:S$ are permitted between elements a , b , and c such that $r(a,b)$ and $s(b,c)$, then a structural relationship $t:T$ is also permitted, with $t(a,c)$ and type T being the weakest of R and S .

The structural and dependency relationships can be ordered by ‘strength’. Structural relationships are ‘stronger’ than dependency relationships, and the relationships within these categories can also be ordered by strength:

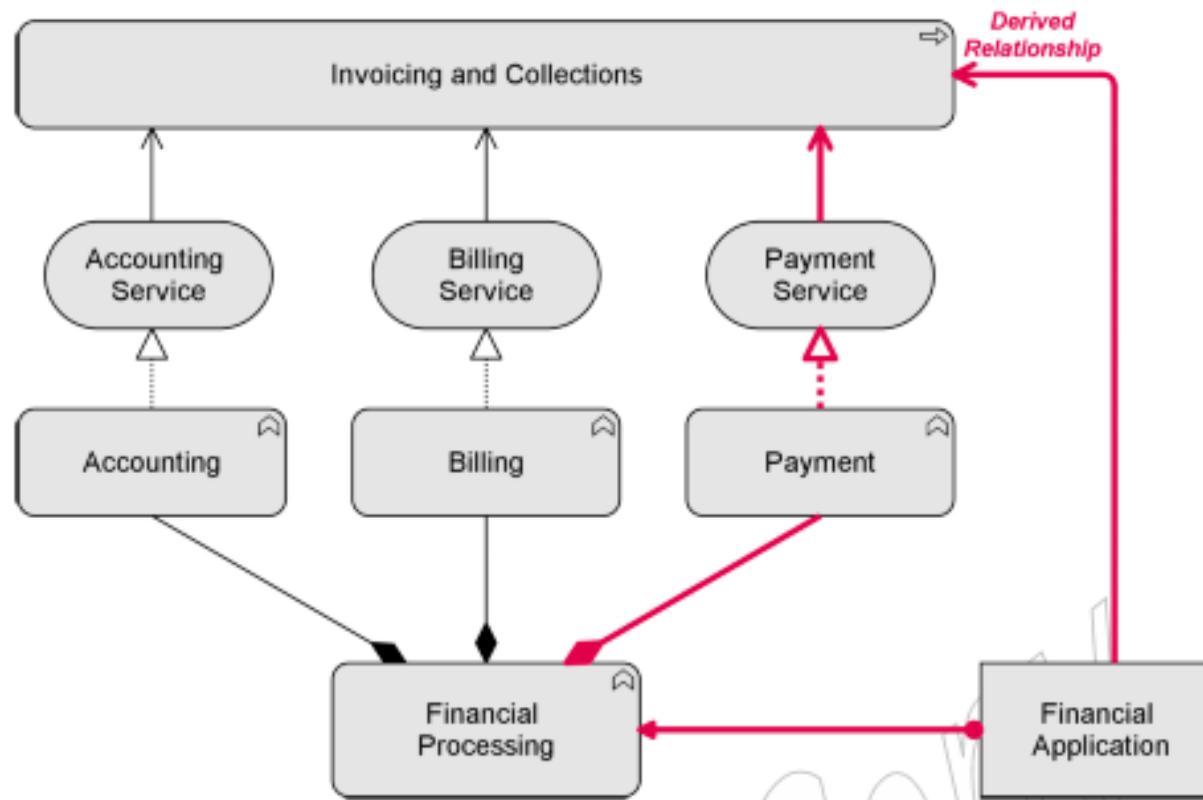
- Influence (weakest)
- Access
- Serving
- Realization
- Assignment
- Aggregation
- Composition (strongest)

ArchiMate Core Framework: Derivation Rules

Derivation Rule for Structural and Dependency Relationships

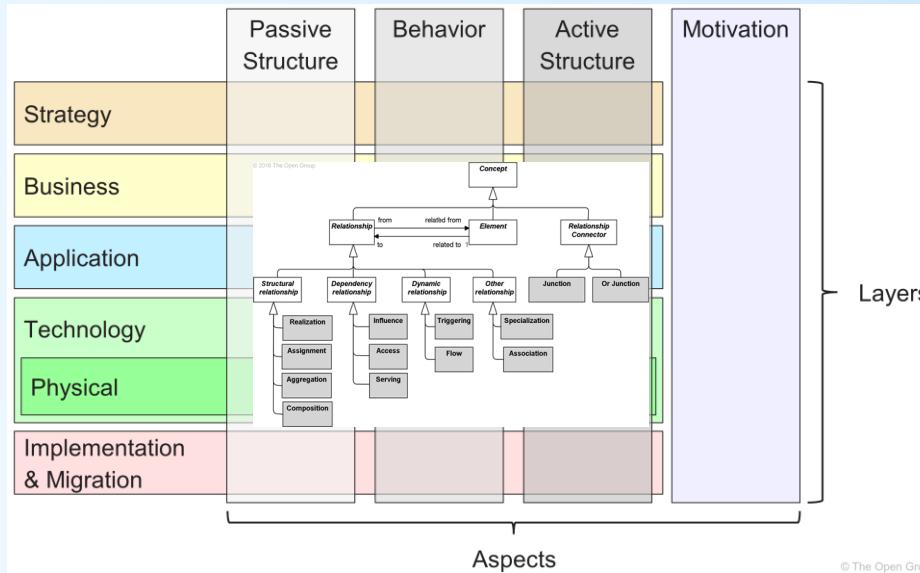
Part of the language definition is an abstraction rule that states that two relationships that join at an intermediate element can be combined and replaced by the weaker of the two.

If two structural or dependency relationships $r:R$ and $s:S$ are permitted between elements a , b , and c such that $r(a,b)$ and $s(b,c)$, then a structural relationship $t:T$ is also permitted, with $t(a,c)$ and type T being the weakest of R and S .



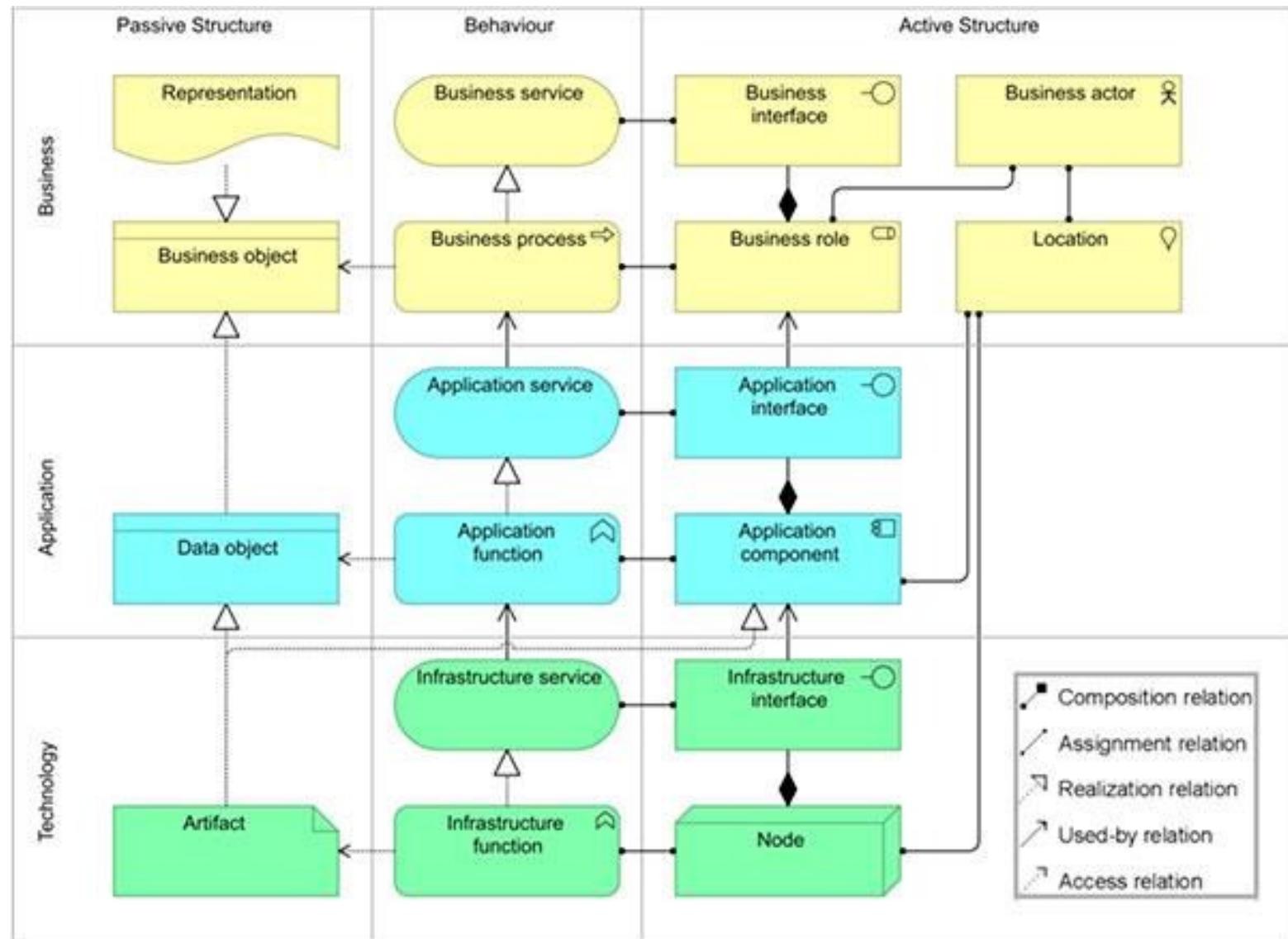


ArchiMate



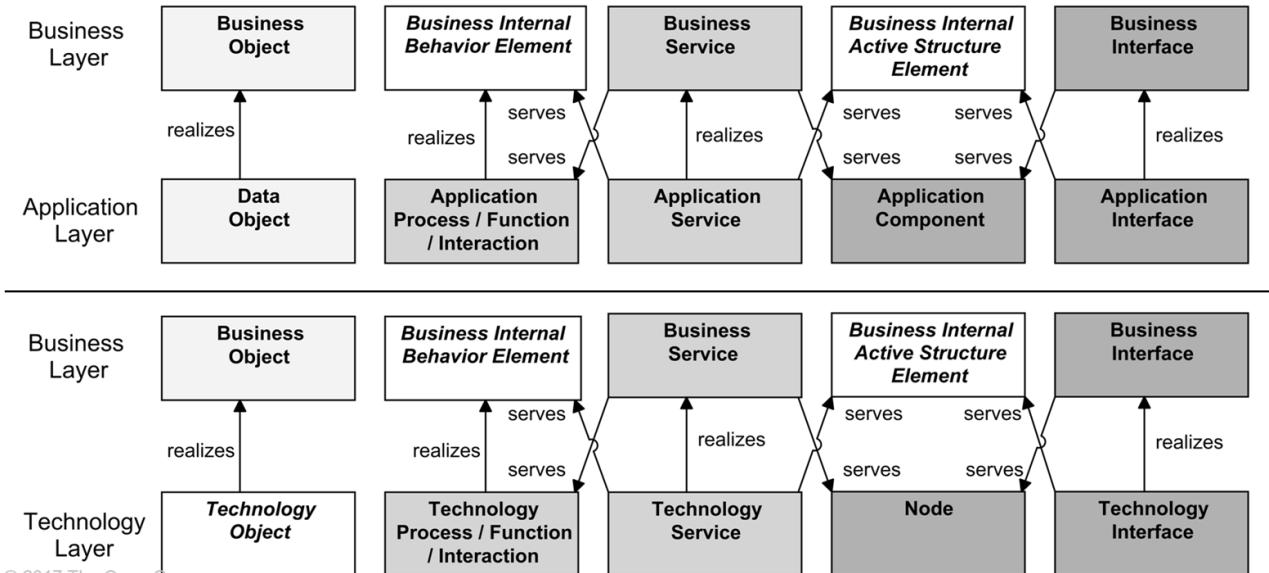
...the **core elements...**
...and **relationships...**

Simplified ArchiMate Metamodel Overview...

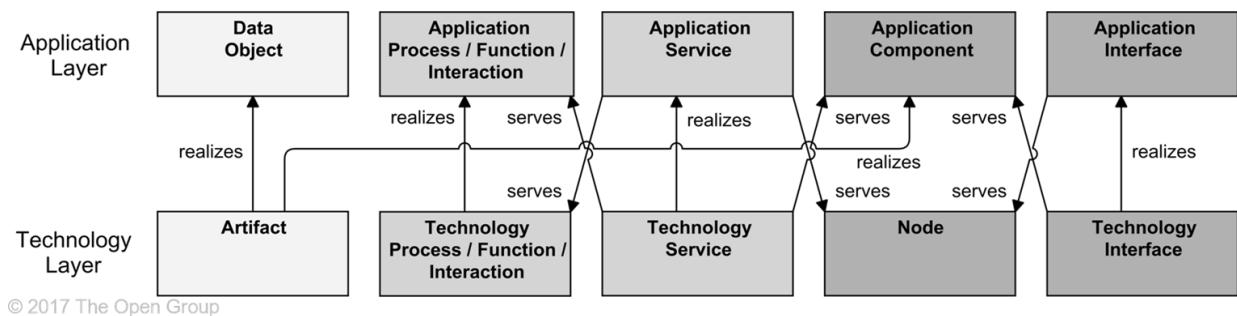


Alignment of the core layers

Figure 96: Relationships between Business Layer and Application and Technology Layer Elements



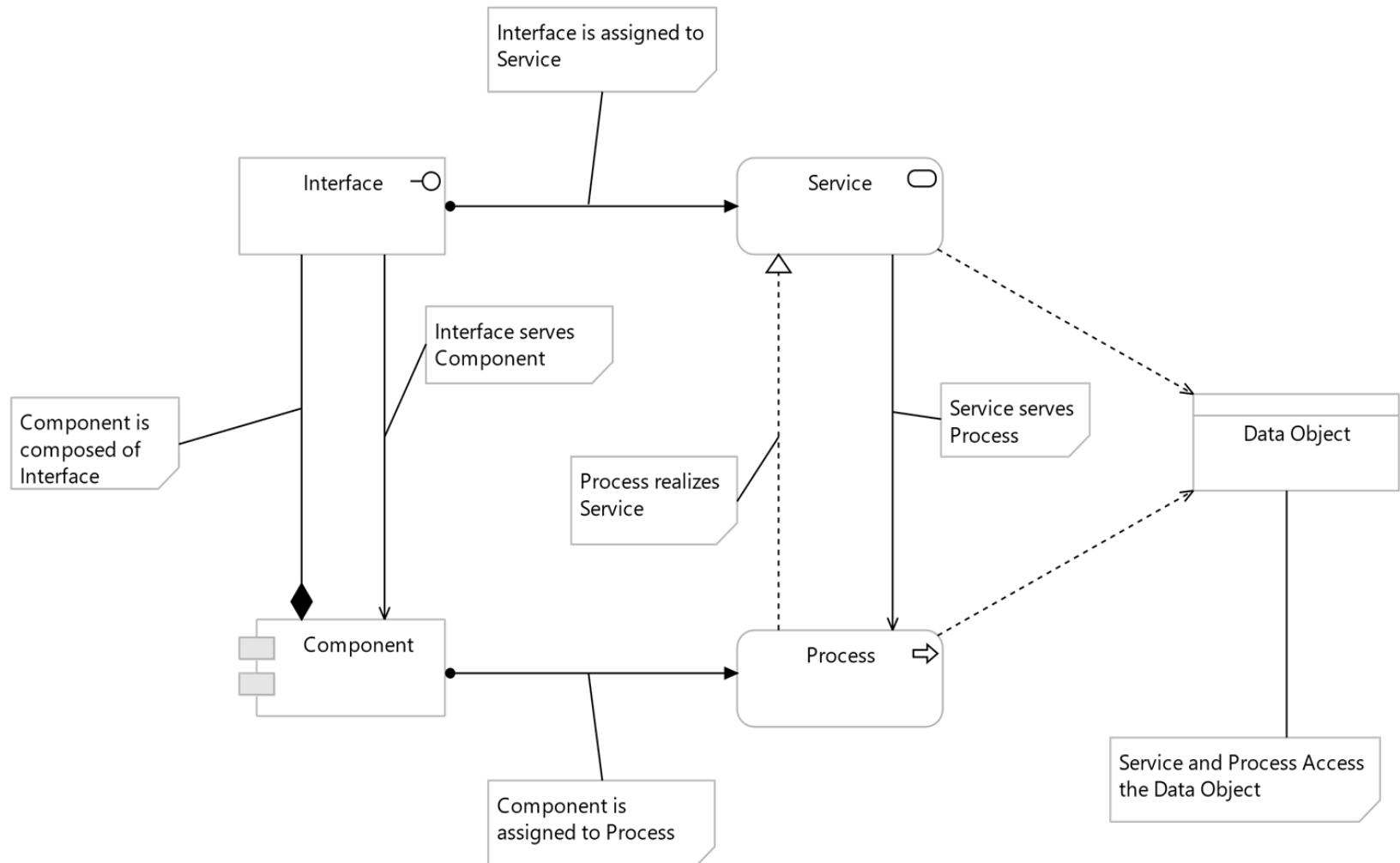
© 2017 The Open Group



© 2017 The Open Group

Figure 97: Relationships between Application Layer and Technology Layer Elements

Core elements – simple all layers metamodel





DEI
DEPARTAMENTO
DE ENGENHARIA INFORMÁTICA
TÉCNICO LISBOA

ArchiMate

**...clarification notes about the
elements notation and colors...**

About the ArchiMate concepts and their notation...

"The ArchiMate language separates the language concepts (i.e., the constituents of the metamodel) from their notation. (...)

Although the notation of the ArchiMate concepts can (and should) be stakeholder-specific, the standard provides one common graphical notation (...). This standard notation for most elements consists of a box with an icon in the upper-right corner. In several cases, this icon by itself may also be used as an alternative notation. This standard iconography should be preferred whenever possible so that anyone knowing the ArchiMate language can read the diagrams produced in the language.

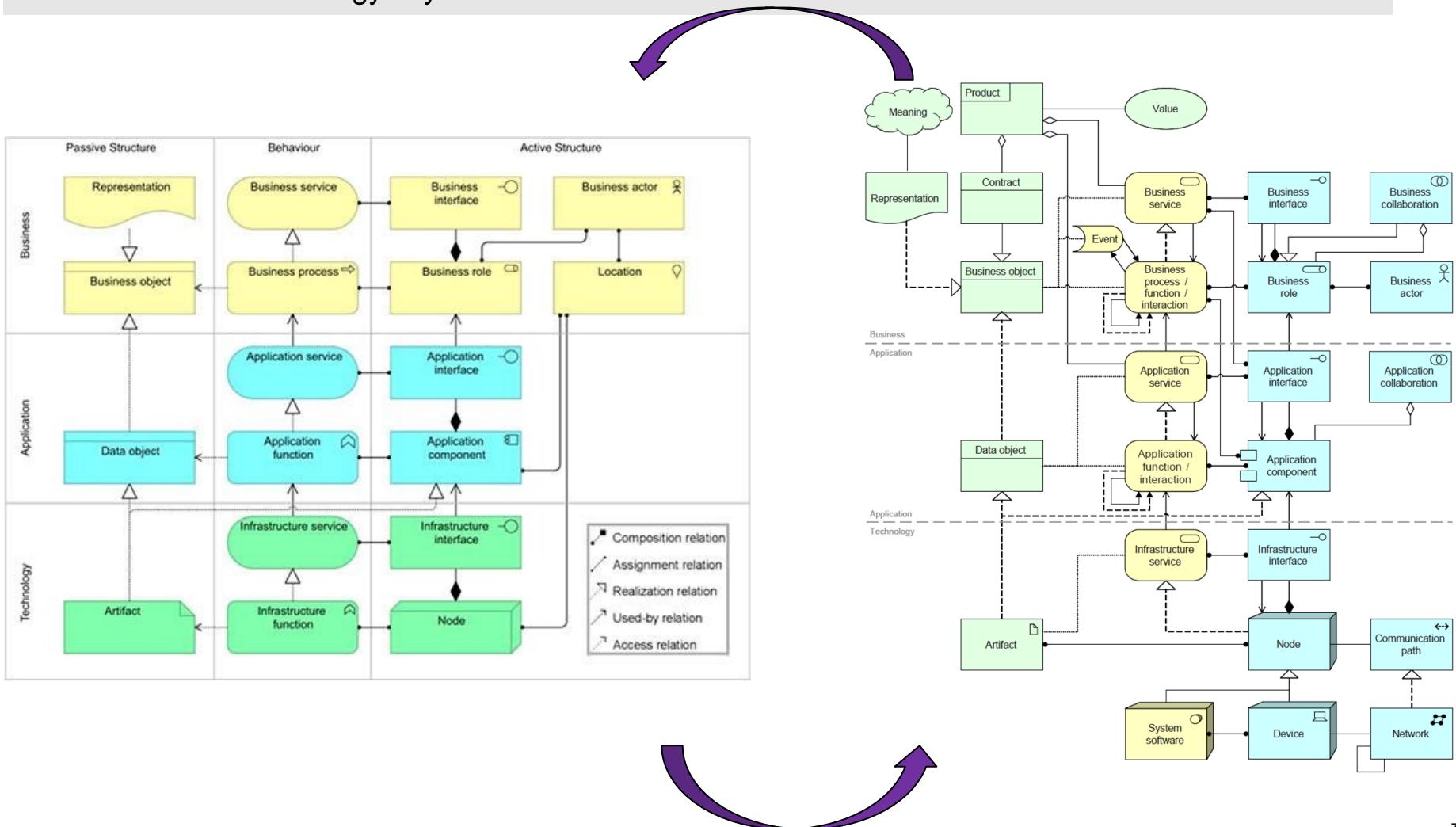
About colors and notational clues in ArchiMate...

There are no formal semantics assigned to colors!!!

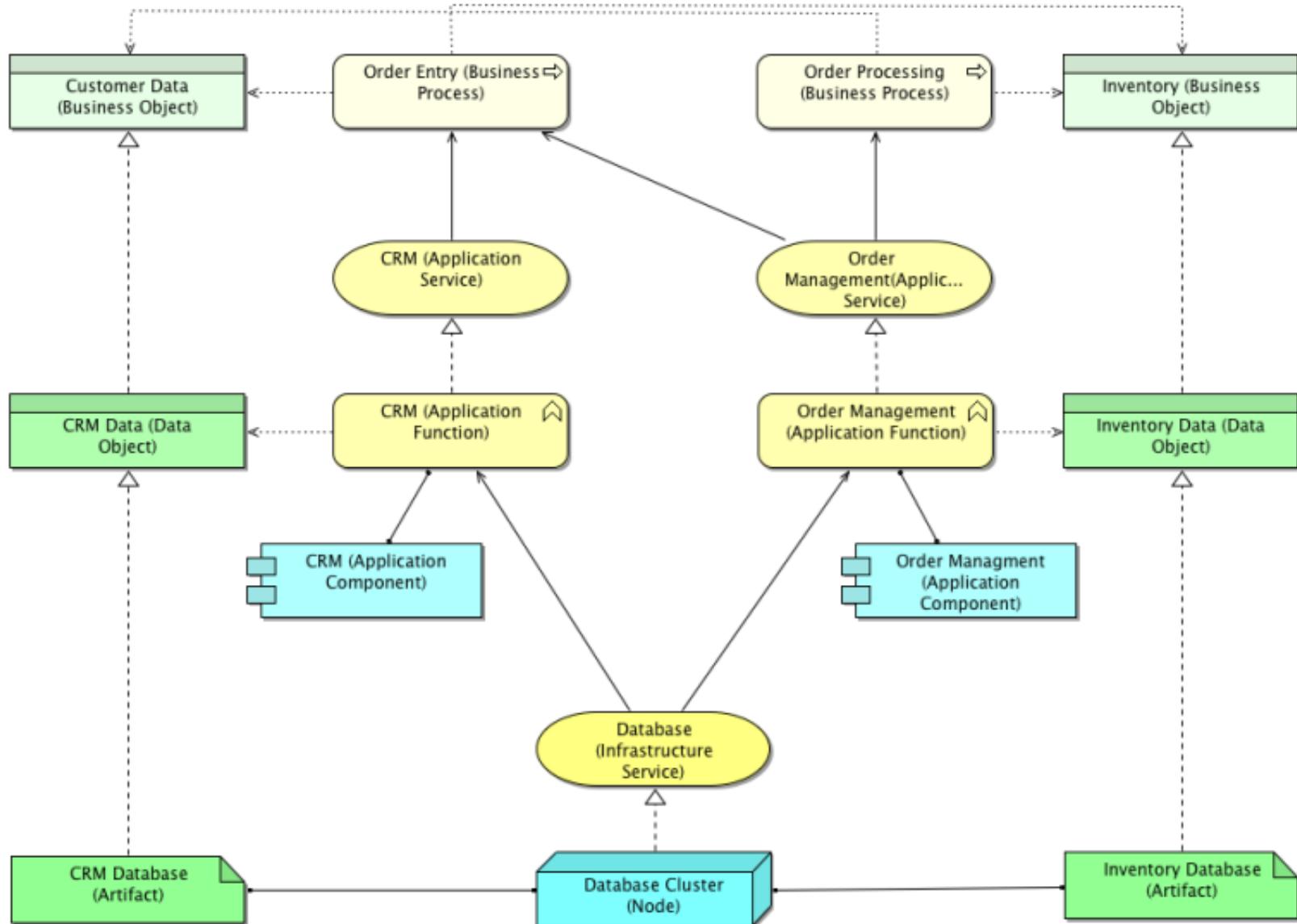
The use of color is left to the modeller, which can be used freely to stress certain aspects in models.

For instance, colors can be used to distinguish between the Core layers as for example:

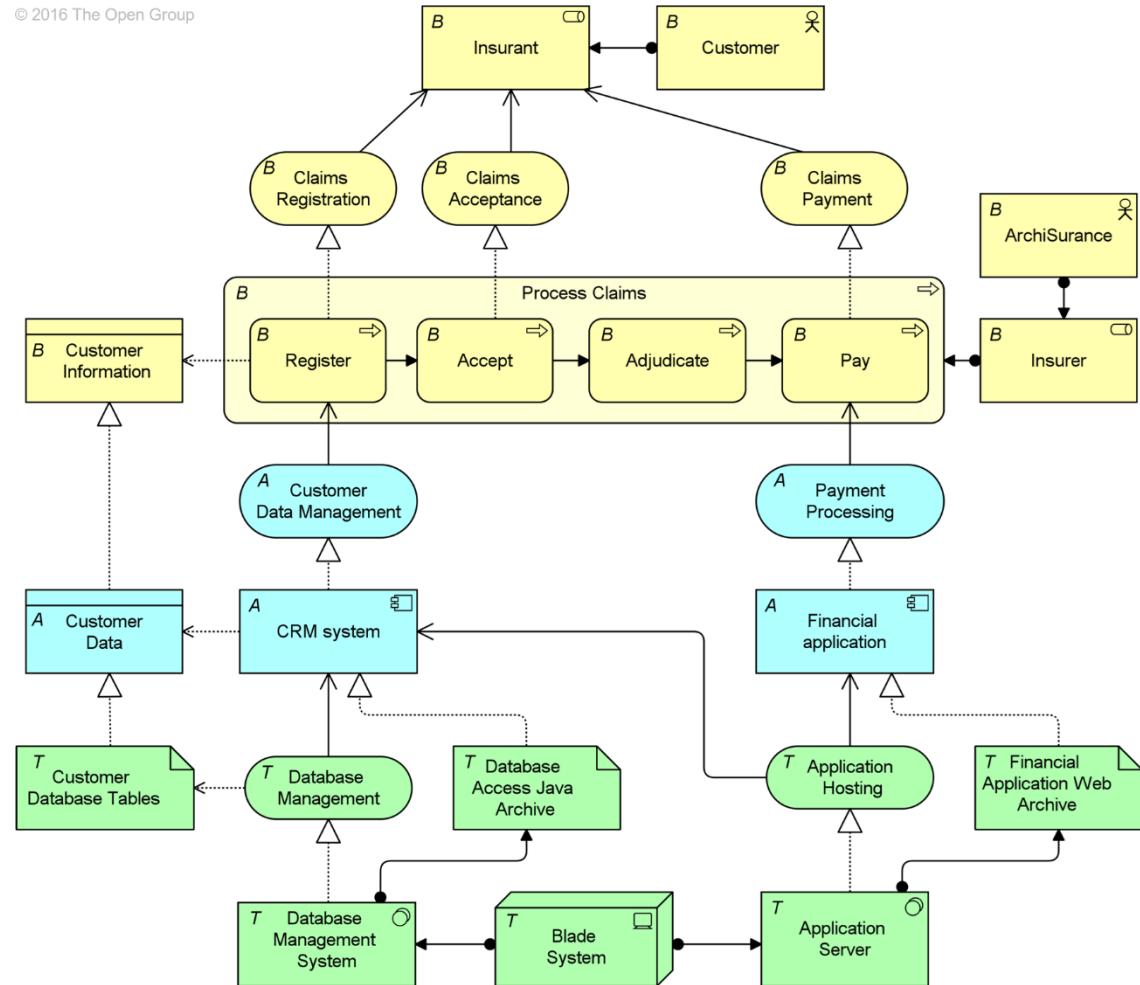
- Yellow for the Business Layer
- Blue for the Application Layer
- Green for the Technology Layer



The 9-Colour scheme of the Archi tool

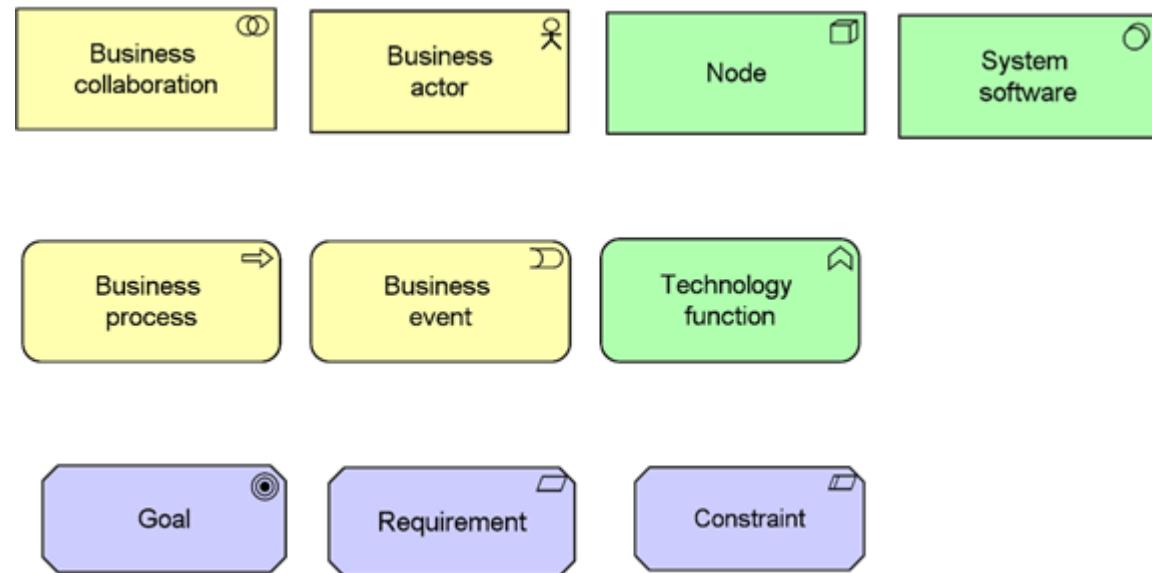


In addition to the colors, other notational cues can be used to distinguish between the layers of the framework. A letter 'M', 'S', 'B', 'A', 'T', 'P', or 'I' in the top-left corner of an element can be used to denote a Motivation, Strategy, Business, Application, Technology, Physical, or Implementation & Migration element, respectively. An example of this notation is depicted...



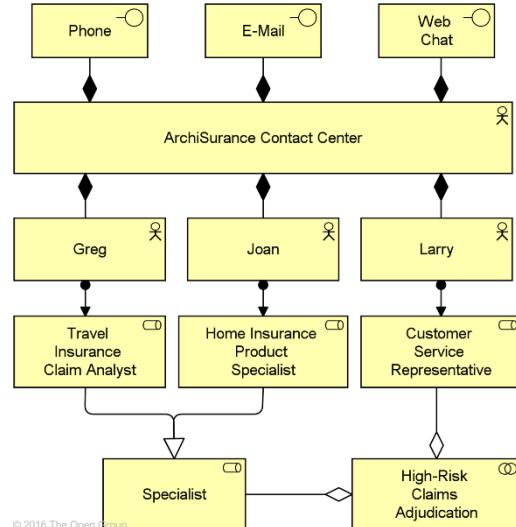
The standard notation also uses a convention with the shape of the corners of its symbols for different element types, as follows:

- Square corners are used to denote structure elements.
- Round corners are used to denote behavior elements.
- Diagonal corners are used to denote motivation elements.

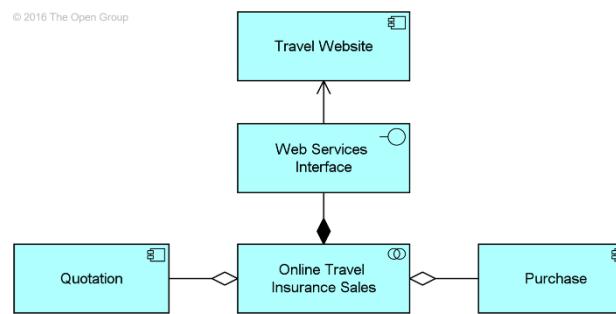


Examples...

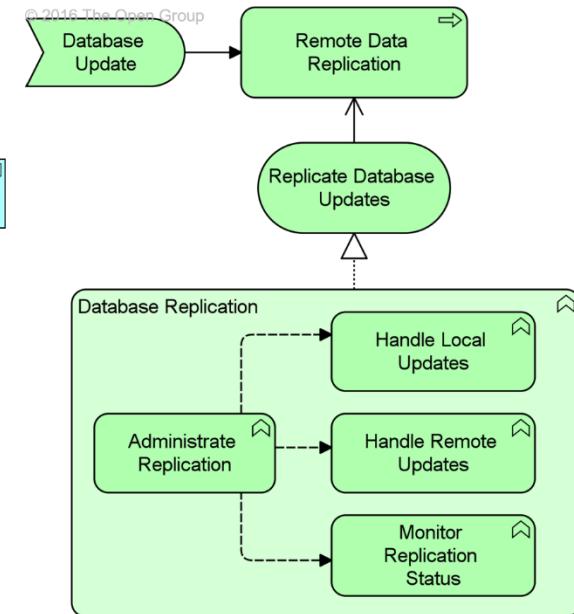
Example 22: Business Active Structure Elements



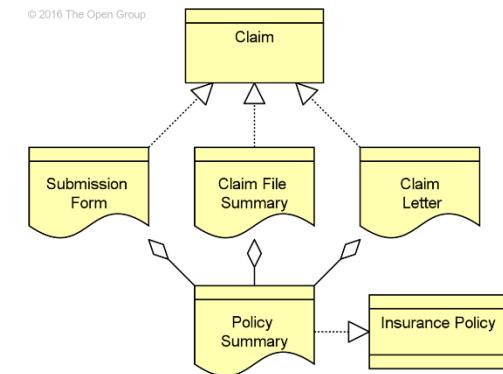
Example 26: Application Active Structure Elements



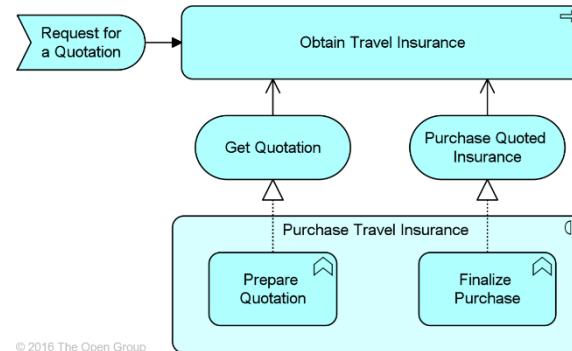
Example 30: Technology Behavior Elements



Example 24: Business Passive Structure Elements

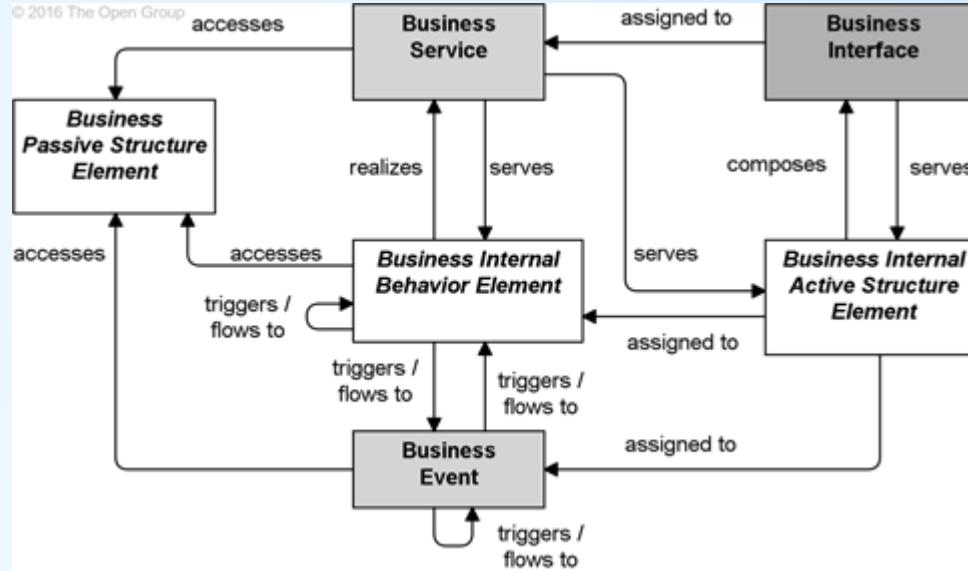


Example 26: Application Active Structure Elements



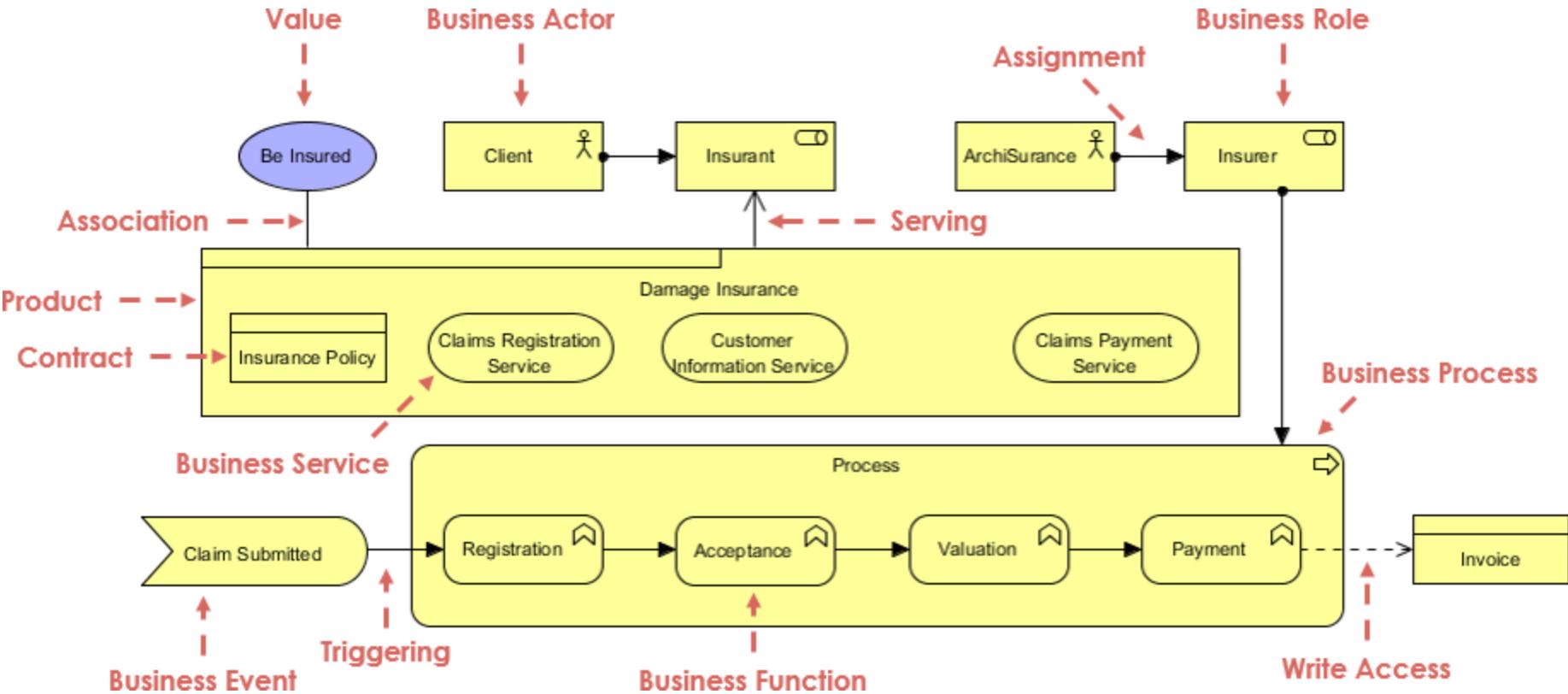


ArchiMate

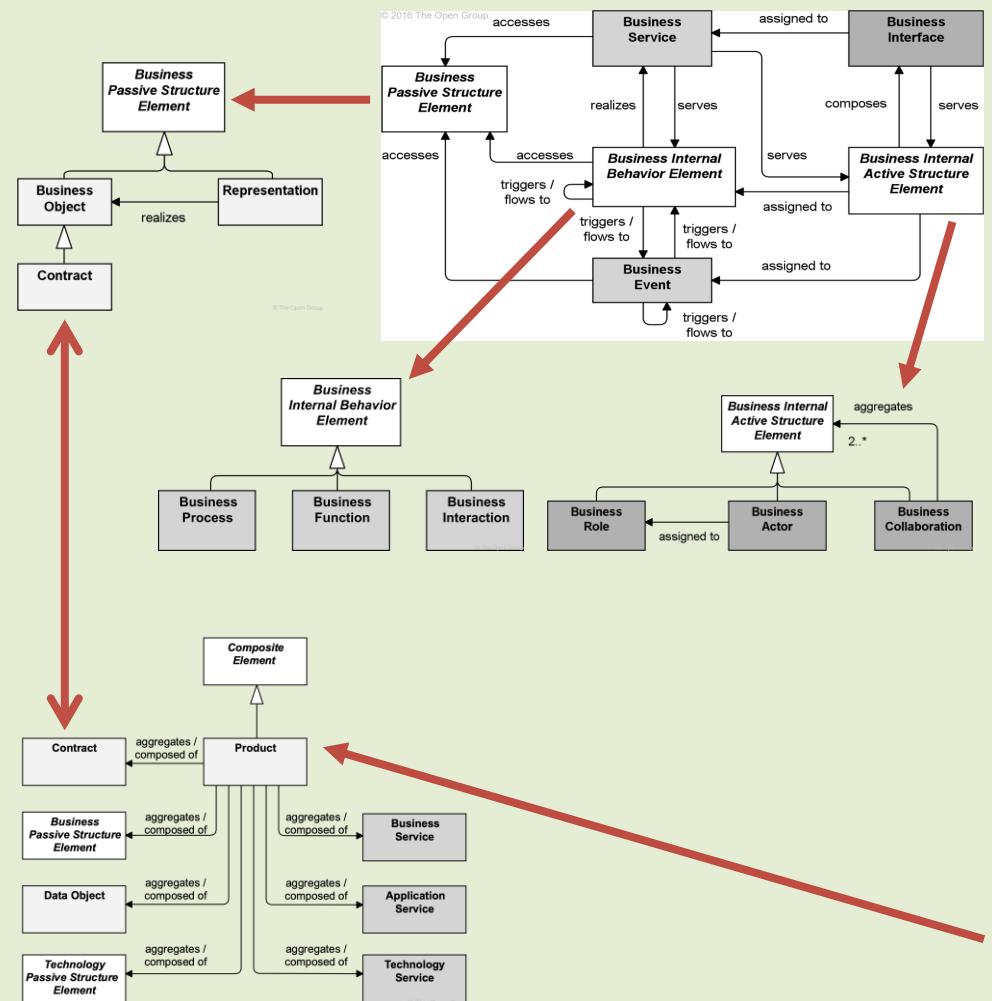


...Business Layer...

ArchiMate Business Layer Concepts

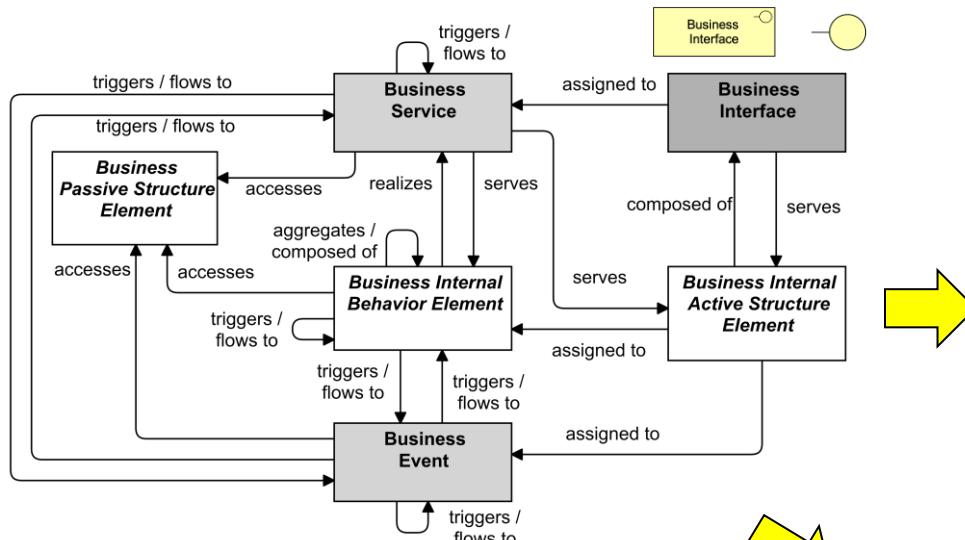


All Business Layer Elements



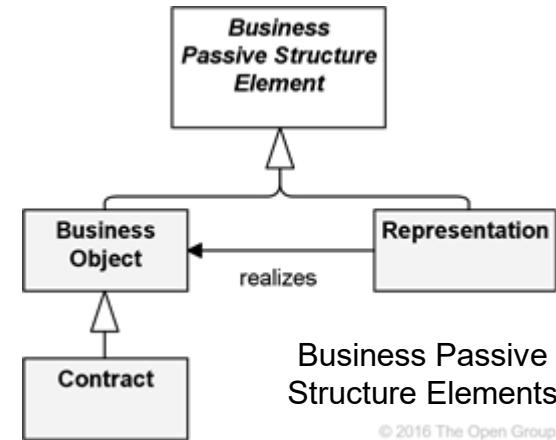
Element	Description	Notation
Business Actor	Represents a business entity that is capable of performing behavior.	
Business Role	Represents the responsibility for performing specific behavior, to which an actor can be assigned, or the part an actor plays in a particular action or event.	
Business Collaboration	Represents an aggregate of two or more business internal active structure elements that work together to perform collective behavior.	
Business Interface	Represents a point of access where business services are made available to the environment.	
Business Process	Represents a sequence of business behaviors that achieves a specific result such as a defined set of products or business services.	
Business Function	Represents a collection of business behavior based on a chosen set of criteria such as required business resources and/or competencies, and is managed or performed as a whole.	
Business Interaction	Represents a unit of collective business behavior performed by (a collaboration of) two or more business actors, business roles, or business collaborations.	
Business Event	Represents a business-related state change.	
Business Service	Represents explicitly defined behavior that a business role, business actor, or business collaboration exposes to its environment.	
Business Object	Represents a concept used within a particular business domain.	
Contract	Represents a formal or informal specification of an agreement between a provider and a consumer that specifies the rights and obligations associated with a product and establishes functional and non-functional parameters for interaction.	
Representation	Represents a perceptible form of the information carried by a business object.	
Product	Represents a coherent collection of services and/or passive structure elements, accompanied by a contract, which is offered as a whole to	

Business Layer Concepts



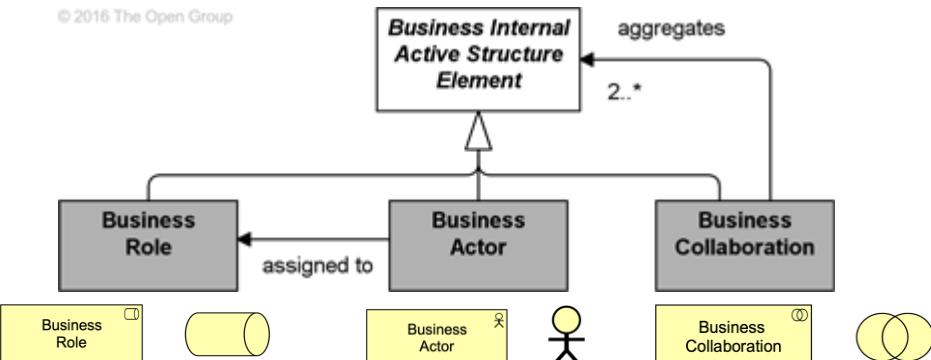
Business Layer Metamodel

© 2016 The Open Group



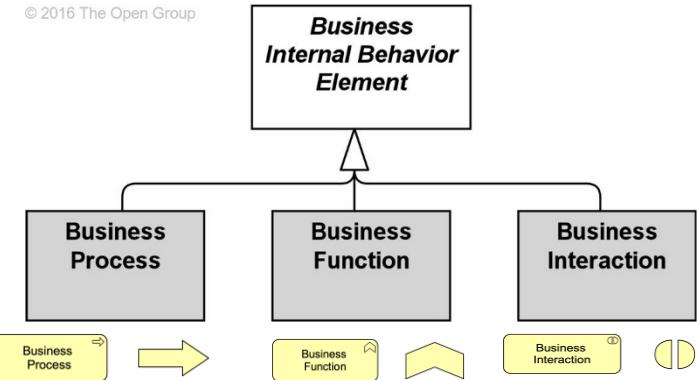
Business Passive Structure Elements

© 2016 The Open Group



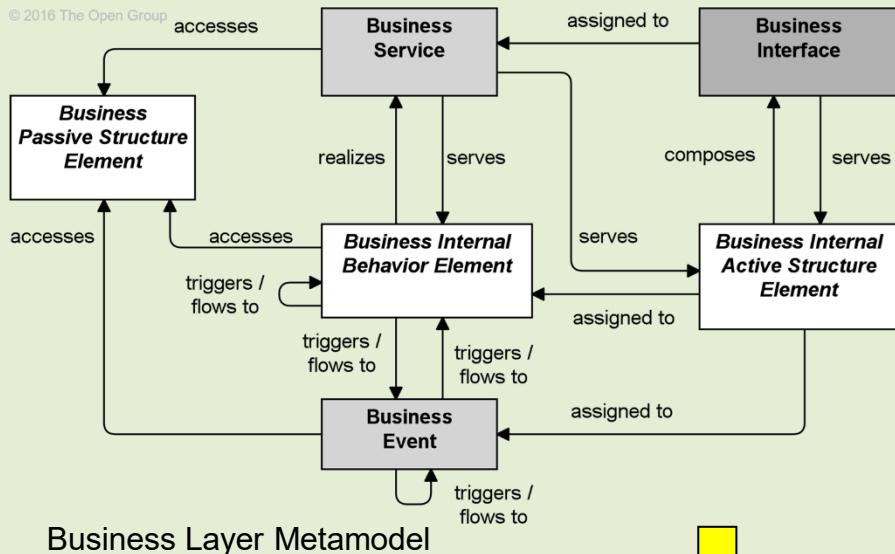
Business Internal Active Structure Elements

© 2016 The Open Group



Business Internal Behavior Elements

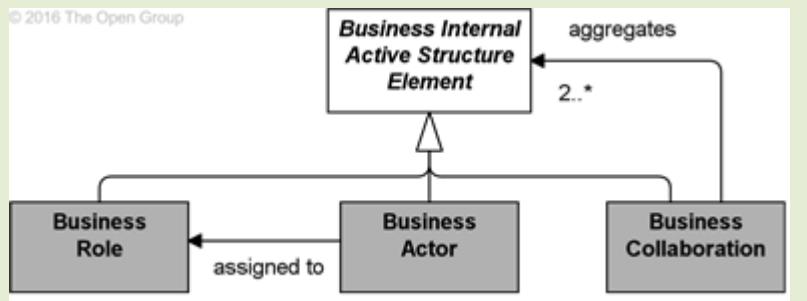
Business Internal Active Structure Elements



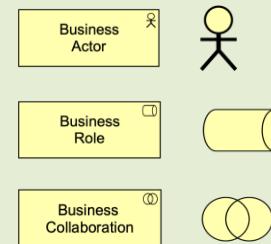
The active structure aspect of the Business Layer refers to the **static structure of an organization**, in terms of the entities that make up the organization and their relationships.

The *active entities* are the subjects (e.g., **business actors** or **business roles**) that perform behavior such as business processes or functions (capabilities).

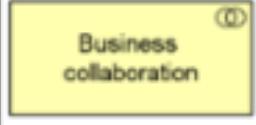
Business actors may be individual persons (e.g., customers or employees), but also groups of people (organization units) and resources that have a permanent (or at least long-term) status within the organizations. Typical examples of the latter are a department and a business unit.



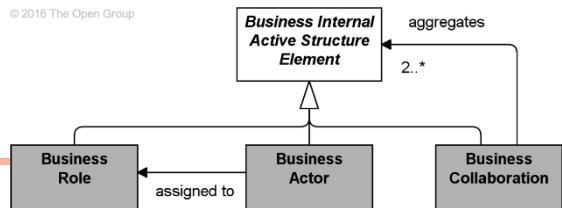
Business Internal Active Structure Elements



Business Internal active Structure Elements

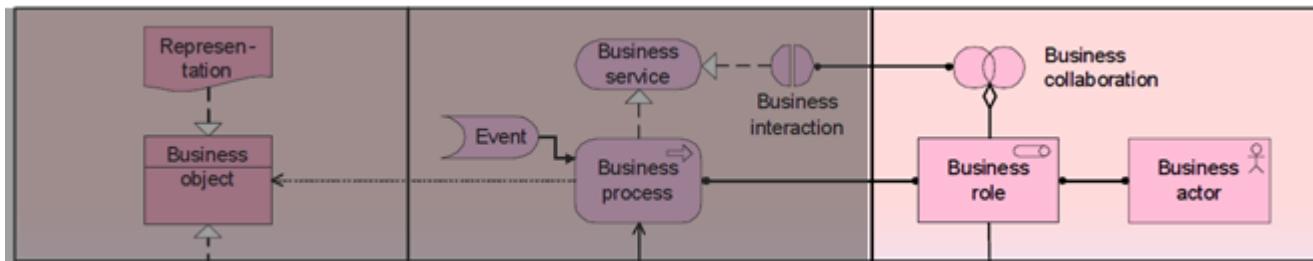
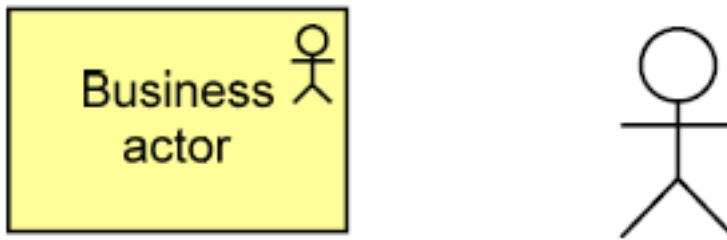
Element	Description	Notation
Business actor	A business entity that is capable of performing behavior.	 
Business role	The responsibility for performing specific behavior, to which an actor can be assigned, or the part an actor plays in a particular action or event.	 
Business collaboration	An aggregate of two or more business internal active structure elements that work together to perform collective behavior.	 
Business interface	A point of access where a business service is made available to the environment.	 

Structural Internal Active Concepts

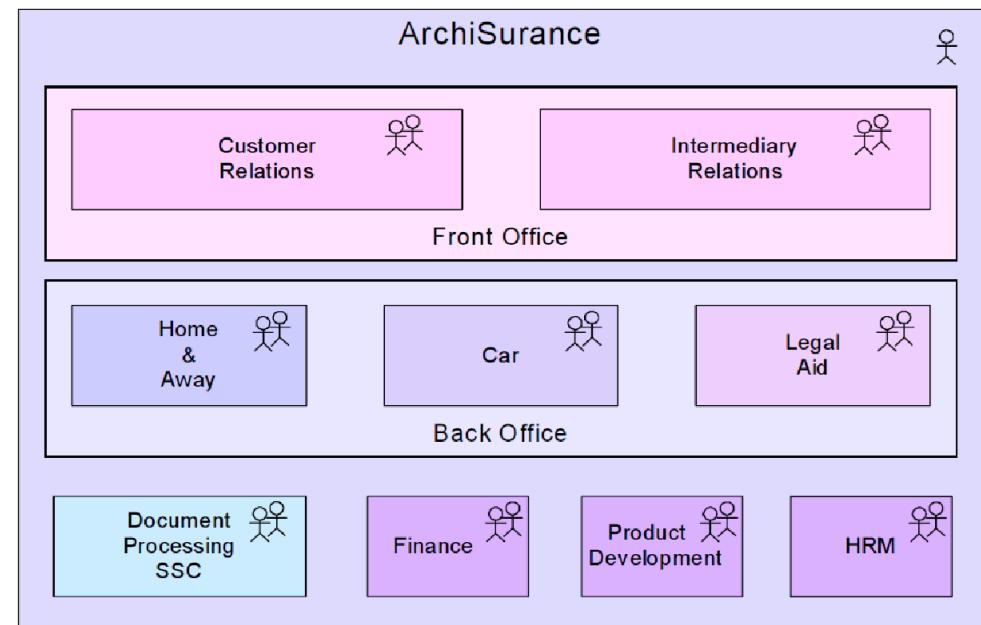
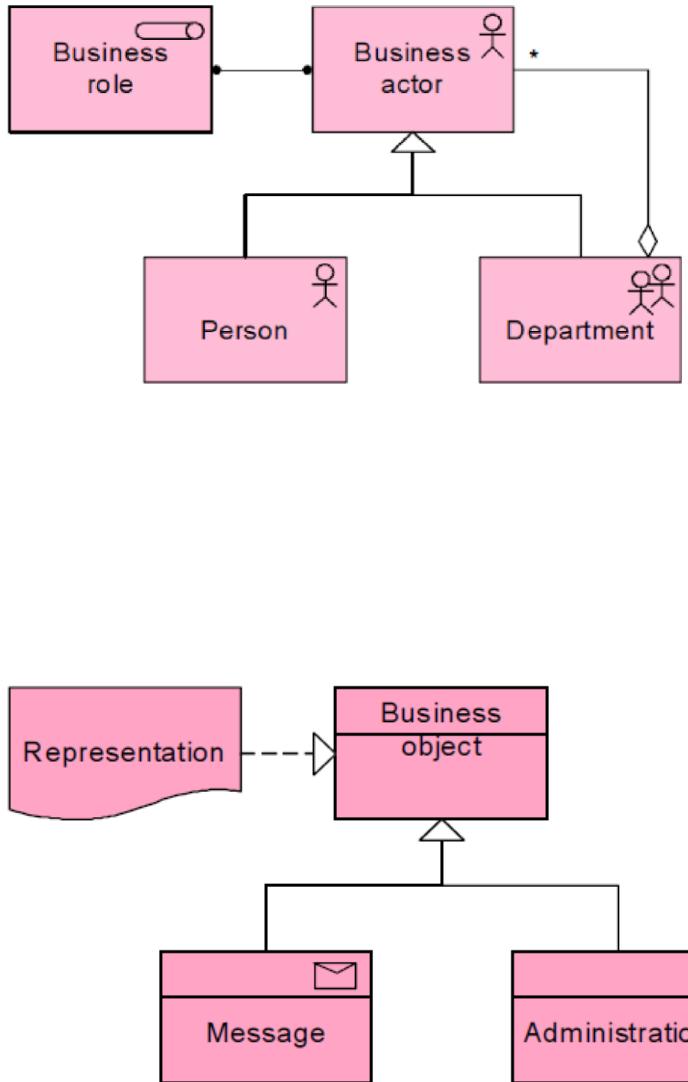


Business Actor: an **active entity** that performs behaviour (i.e., the ‘subject’ of behaviour).

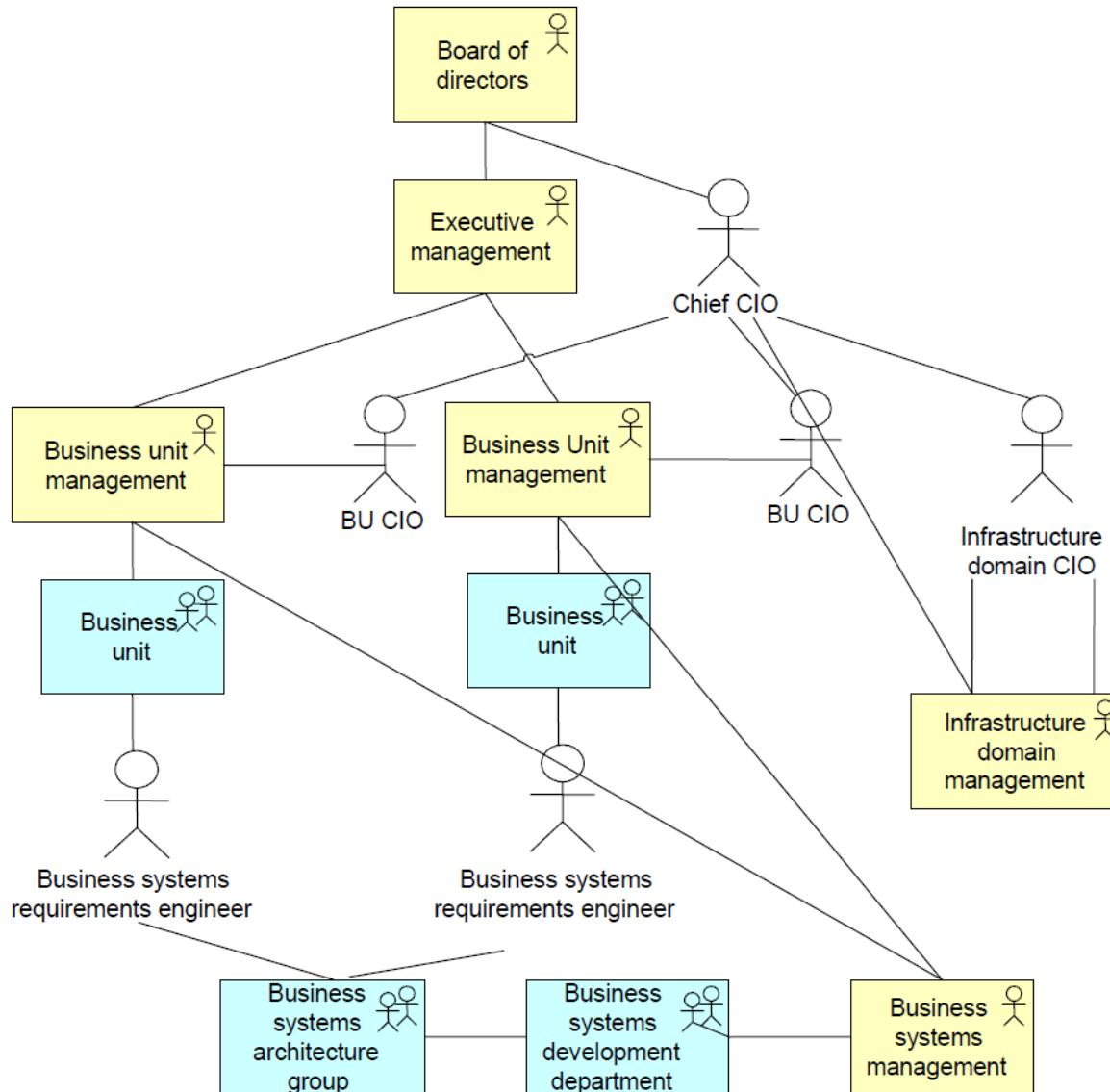
- Examples: Person (e.g. employee, customer), Department, Business Unit
- Symbol:



Example (Business Actors)



Example (Business Actors)



Structural Internal Active Concepts

Business Role: responsibility for performing specific behavior, to which an actor can be assigned:

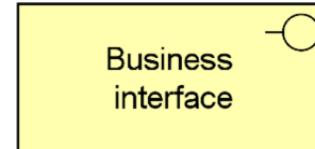
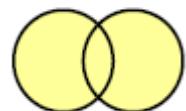
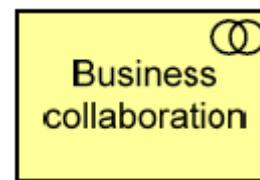
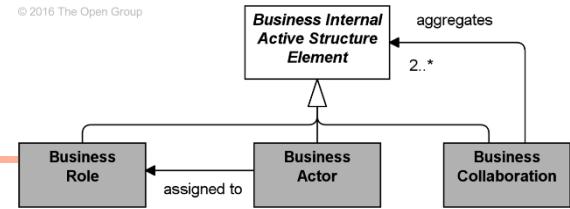
- The same actor can play multiple roles.
- The same role can be played by multiple actors.
- More stable paper structure of the actors.

Business Collaboration: aggregate of two or more business roles that work together to perform collective behavior ([interactions](#)):

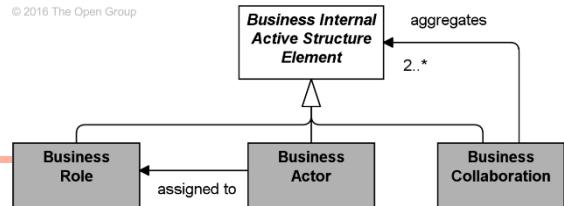
- Do not have a permanent / official status in the organization
- Is directed to a set of interactions between "roles"
- It is a role of expertise - "virtual role"

Business Interface: point of access where a business service is made available to the environment.

- Exposes the functionality of a business service (provided interface) or requires business services
- Used to set the "channel" in which a service is provided (telephone, web, face-to-face...)

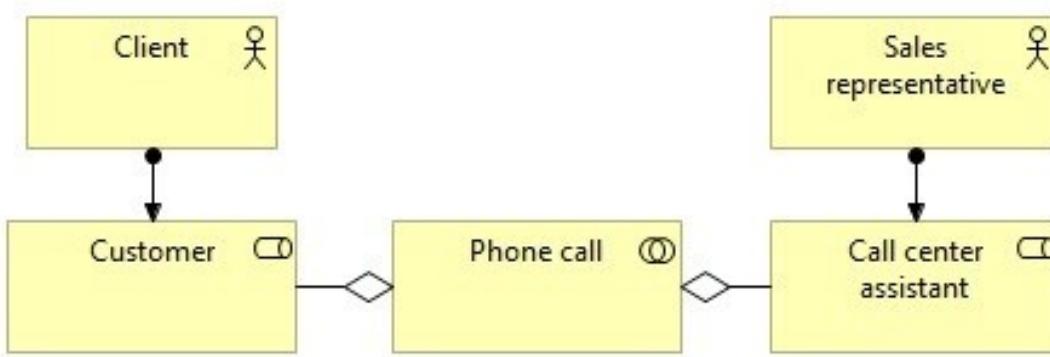


Structural Internal Active Concepts

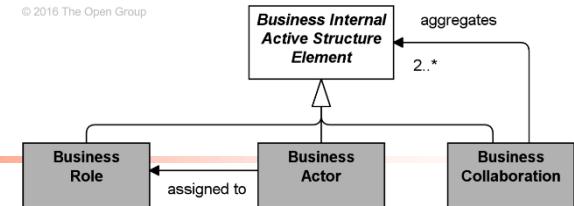


Example:

- An actor “Cliente” has assigned the role “Customer”
- An actor “Sales representative” has assigned the role “Call center assistant”
- The collaboration “Phone call” aggregates the roles “Customer” and “Sales representative”



Structural Internal Active Concepts



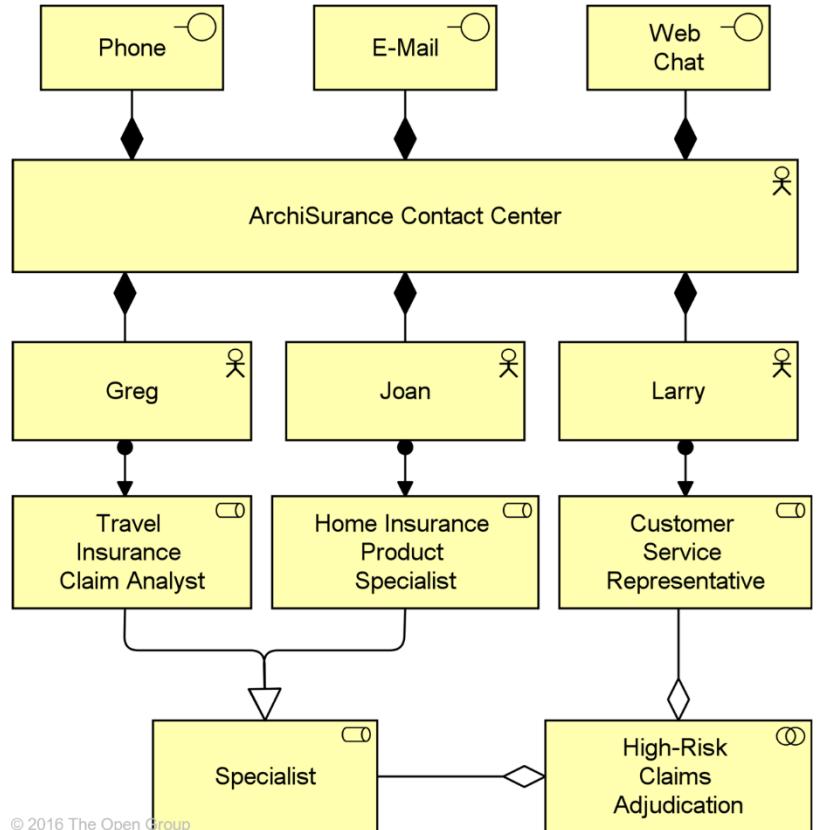
Exercise:

The ArchiInsurance Contact Center, modeled as a business actor, is composed of three employees, also modeled as business actors: Greg, Joan, and Larry.

The Contact Center has three business interfaces to serve customers: Phone, E-mail, and Web Chat. Greg fulfills the business role of Travel Insurance Claim Analyst, Joan fulfills the business role of Home Insurance Product Specialist, and Larry fulfills the business role of Customer Service Representative.

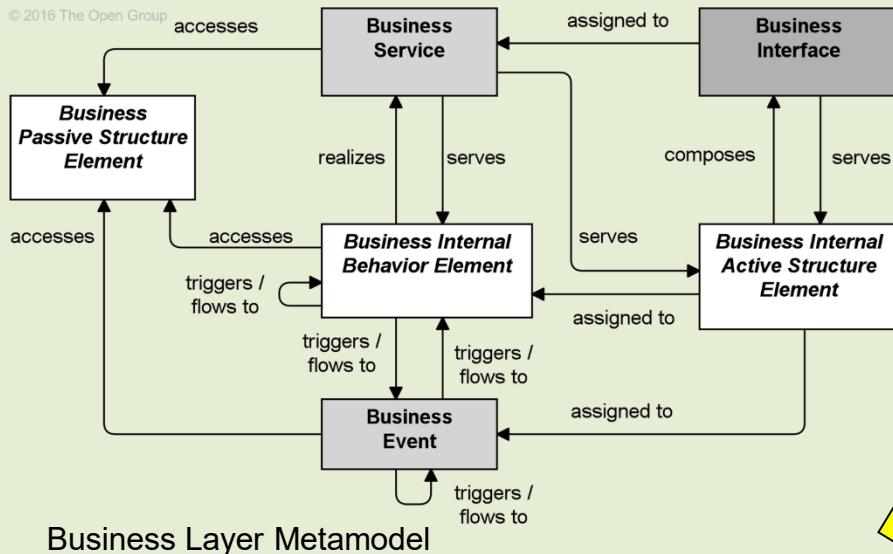
The former two business roles are specializations of a business role Specialist.

High-Risk Claims Adjudication is a business collaboration of two business roles: Specialist and Customer Service Representative.



Business Internal Behavior Elements

© 2016 The Open Group



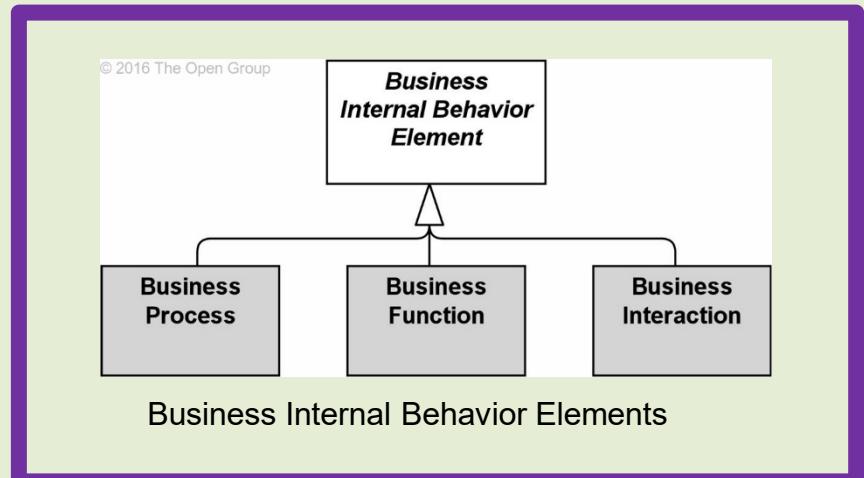
Business Layer Metamodel

A distinction can be made between “external” business services, offered to external customers, and “internal” business services, offering supporting functionality to processes or functions within the organization.

Based on service-orientation, a crucial design decision for the behavioral part is the distinction between “external” and “internal” behavior of an organization.

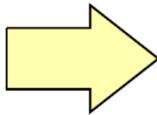
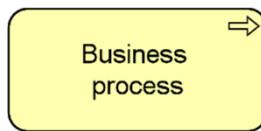
The externally visible behavior is modeled by the element **business service.**

A business service represents a coherent piece of functionality that offers added value to the environment, independent of the way this functionality is realized internally.

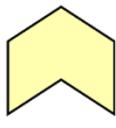


Business Internal Behavior Elements

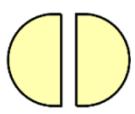
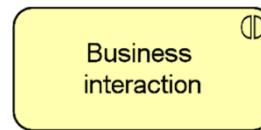
Business Internal Behavior Elements



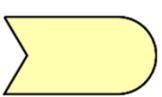
A business process represents a sequence of business behaviors that achieves a specific outcome such as a defined set of products or business services.



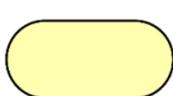
A business function is a collection of business behavior based on a chosen set of criteria (typically required business resources and/or competences), closely aligned to an organization, but not necessarily explicitly governed by the organization.



A business interaction is a unit of collective business behavior performed by (a collaboration of) two or more business roles.



A business event is a business behavior element that denotes an organizational state change. It may originate from and be resolved inside or outside the organization.



A business service represents an explicitly defined exposed business behavior.

Business Internal Behavior Elements

Claims Administration is a business function that is composed of a number of business processes and a business interaction.

This business function realizes a Claims Processing business service

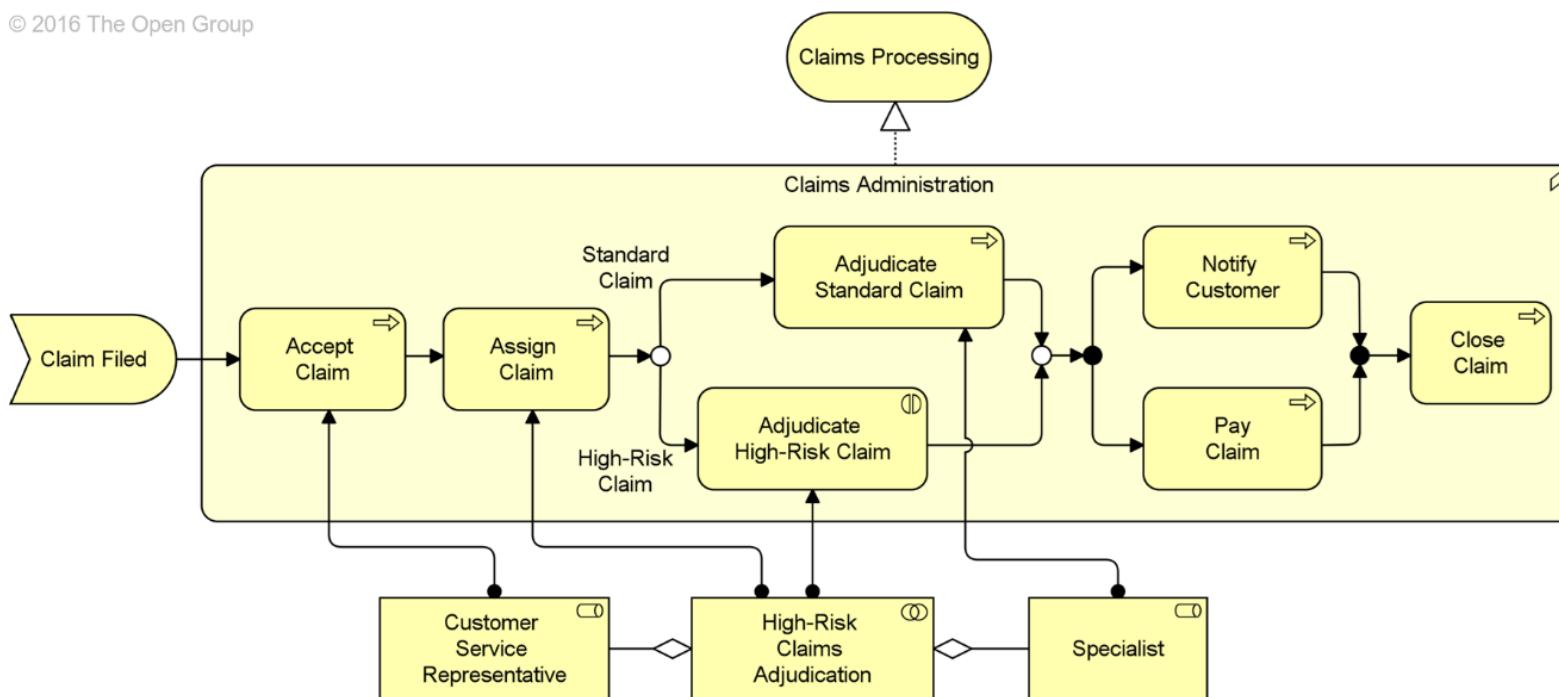
A business event Claim Filed triggers the first business process, Accept Claim, which in turn triggers a business process Assign Claim.

Depending on the type of claim, either the business process Adjudicate Standard Claim or the business interaction Adjudicate High-Risk Claim is performed.

Adjudication of high-risk claims is a business interaction because, according to the company policy, two people should always be involved in this activity to minimize the risk of fraud.

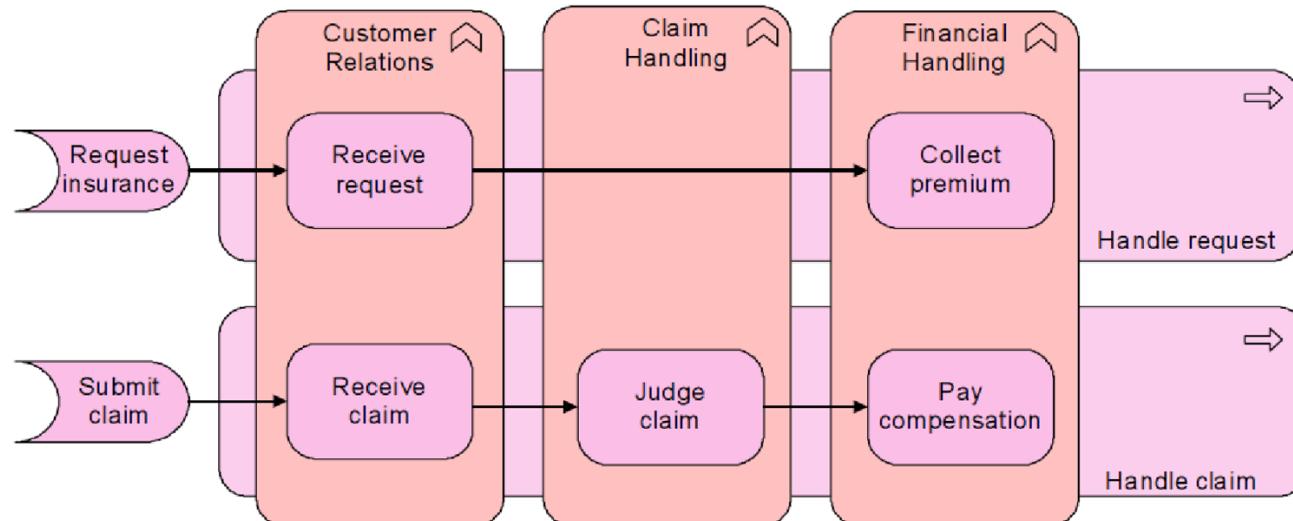
After adjudication, the business processes Notify Customer and Pay Claim are performed in parallel, and when both have finished, business process Close Claim is triggered.

© 2016 The Open Group

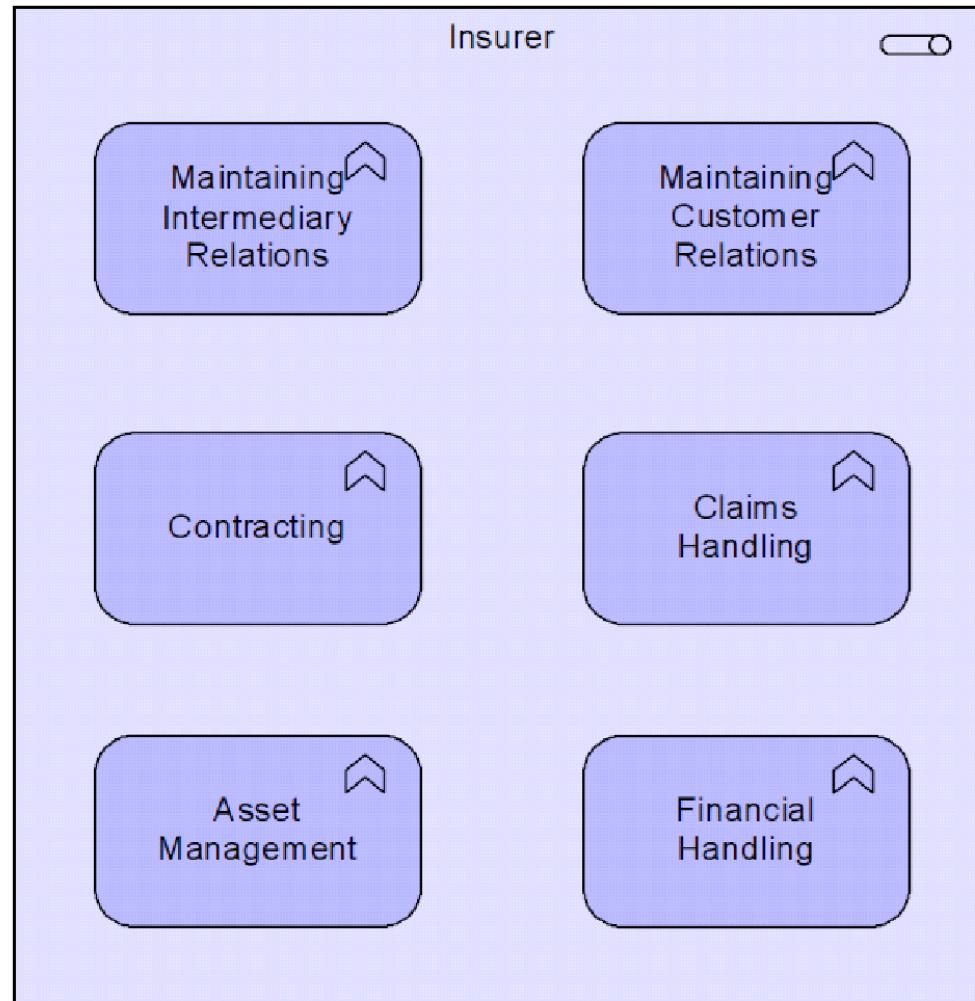


Business Internal Behavior Elements

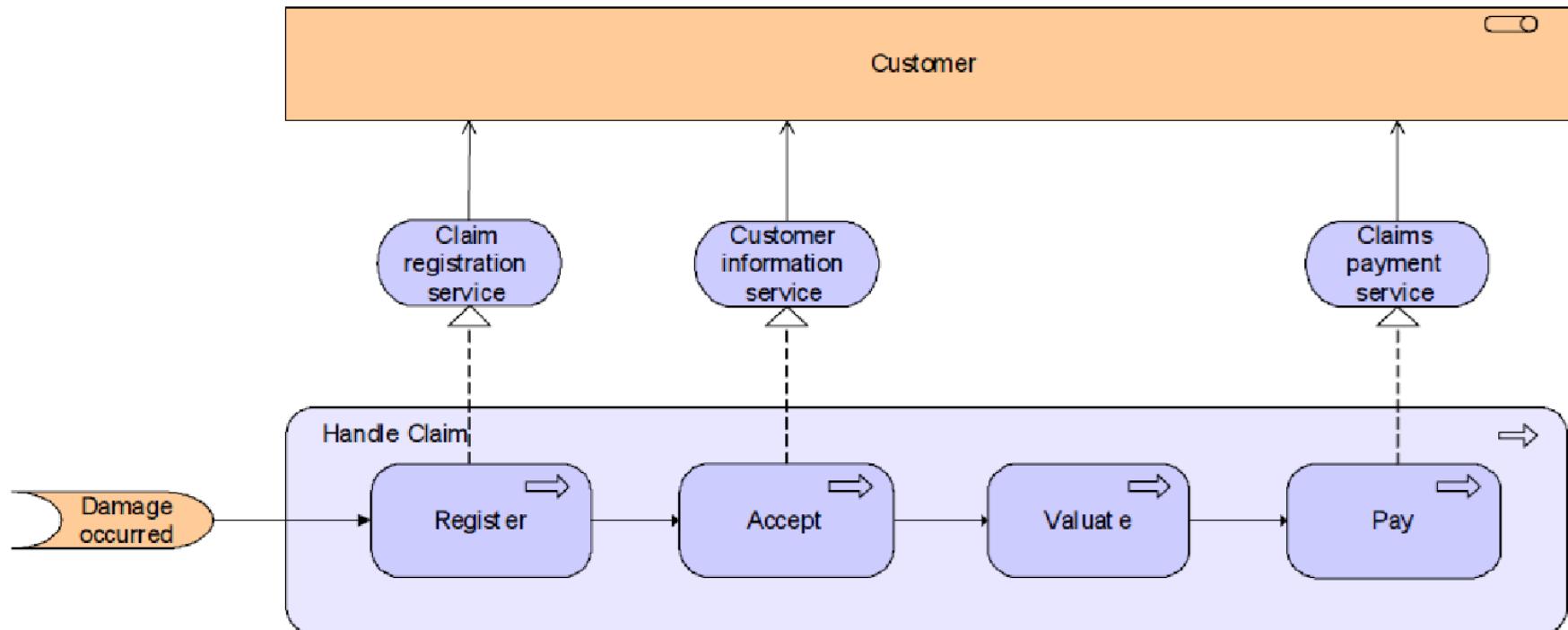
- Business functions and processes group behavior.
 - Process groups according to products or services;
 - Function groups based on other criteria, eg competence, resources.
 - There is a potential link n: n Relationshipship between processes and functions.



Example: Business Functions

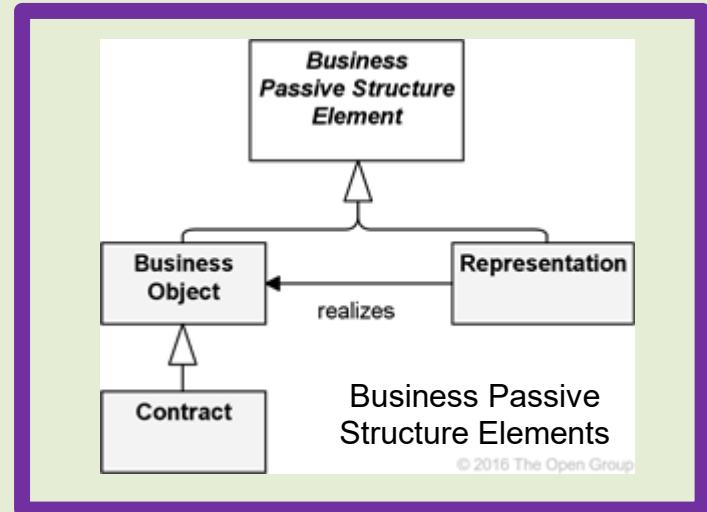
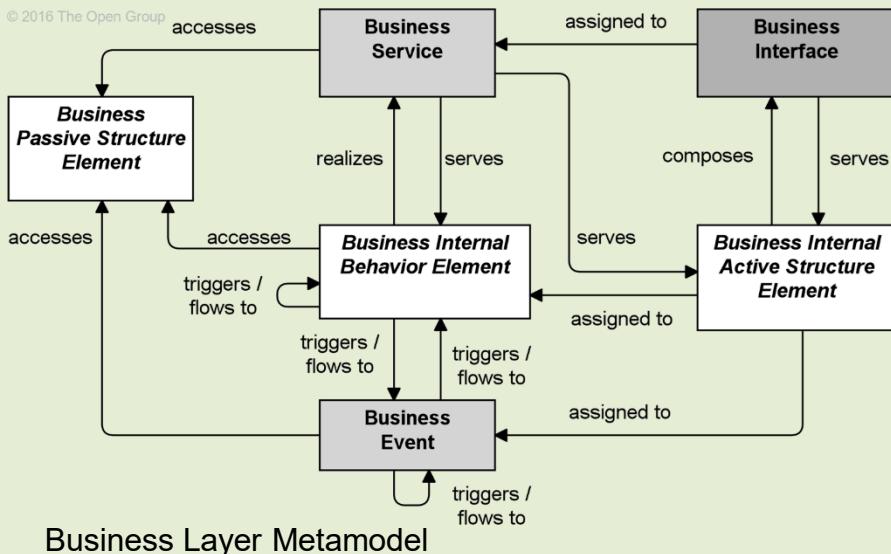


Example: services associated with a process



Business Passive Structure Elements

© 2016 The Open Group

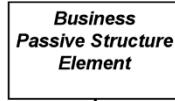


The passive structure aspect of the Business Layer contains the passive structure elements (business objects) that are manipulated by behavior, such as **business processes** or **functions**.

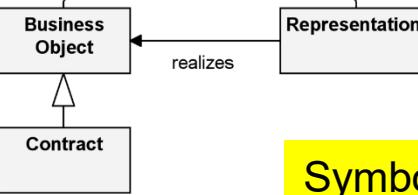
The passive entities represent the important concepts in which the business thinks about a domain.

In the Business Layer, there are two main types of passive structure elements: **business object** and **representation**.

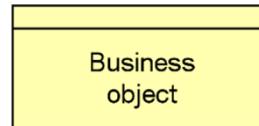
Furthermore, a **contract**, used in the context of a product, is a specialization of a business object.



Structural Passive Concepts

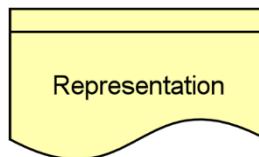


Symbols:

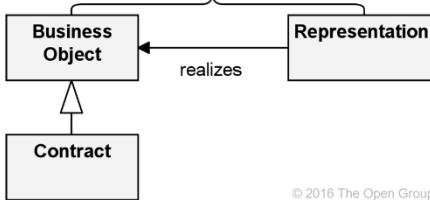
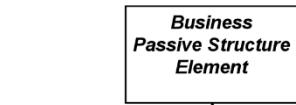


Elements:

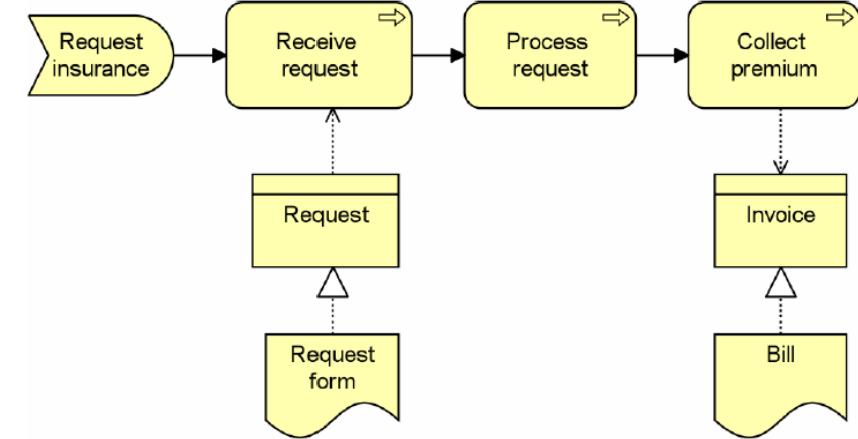
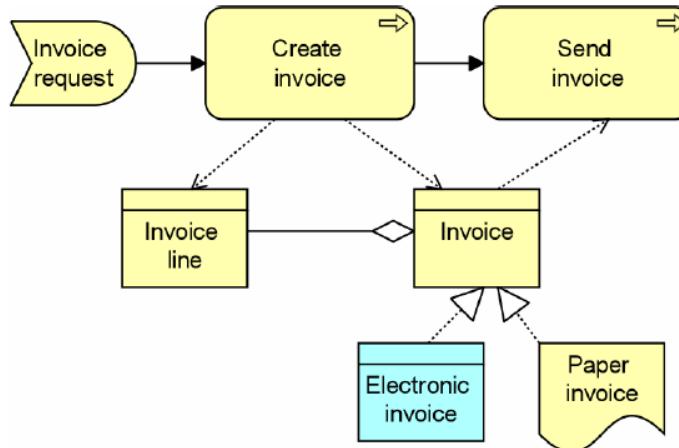
1. **Business Object:** passive element that has relevance from a business perspective (manipulated by behaviour):
 - Represent "conceptual" or "informational" business elements.
 - Typically used to model an object type (UML class)
2. **Representation:** the perceptible form of the information carried by a **business object**, such as a document.
 - Can be classified in different ways (medium (paper, electronic, audio), format (HTML, ASCII, ...))
 - A business object can have multiple representations
3. **Contract:** a formal or informal specification of agreement that specifies the rights and obligations associated with a product.
 - Can be used to model a formal contract, as an informal agreement on a product
 - May contain SLAs
 - It is a business object specialization



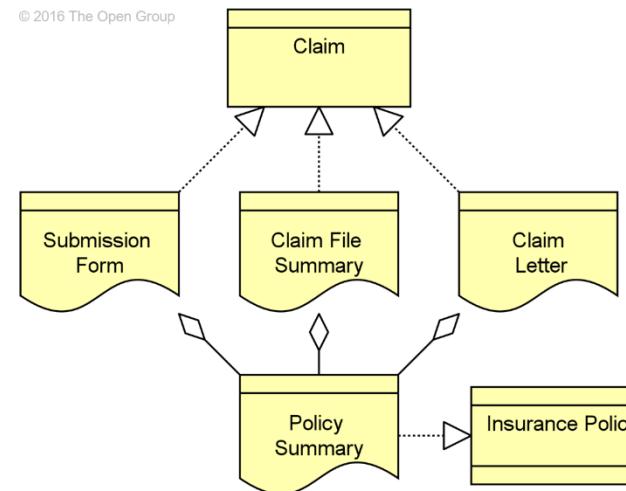
Structural Passive Concepts



Examples:



© 2016 The Open Group

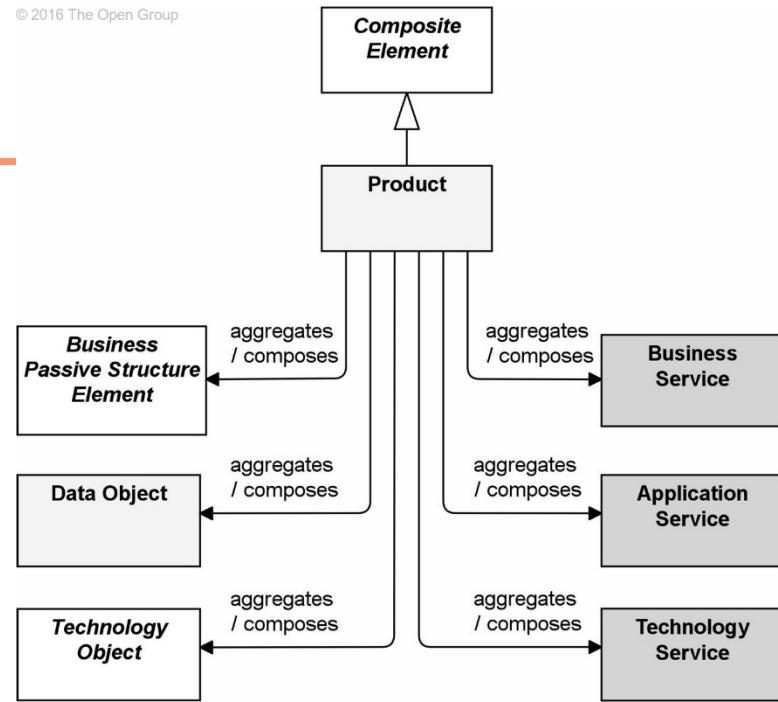
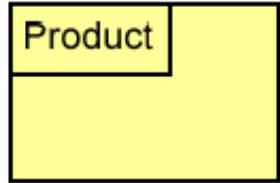


Structural Passive Concepts

4. Product: coherent collection of services, accompanied by a contract/set of agreements, which is offered as a whole to (internal or external) customers.

- Can be products based on services or information
- A product consists of a set of services
- The product name is the name used to communicate with the outside (eg, customers)

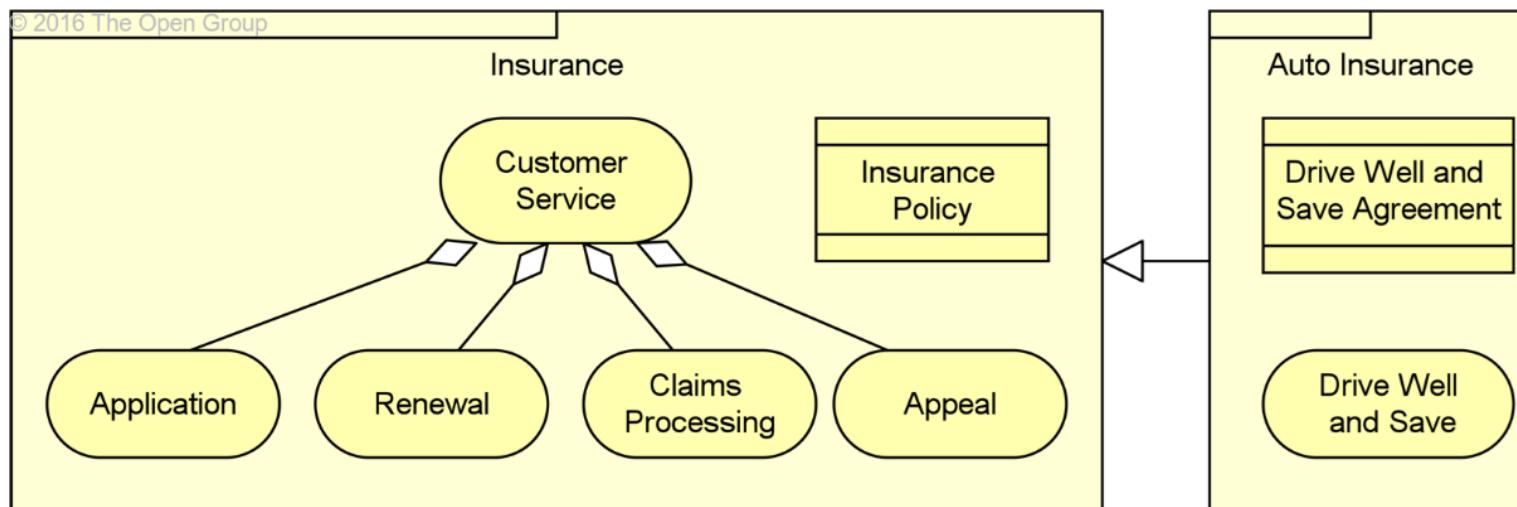
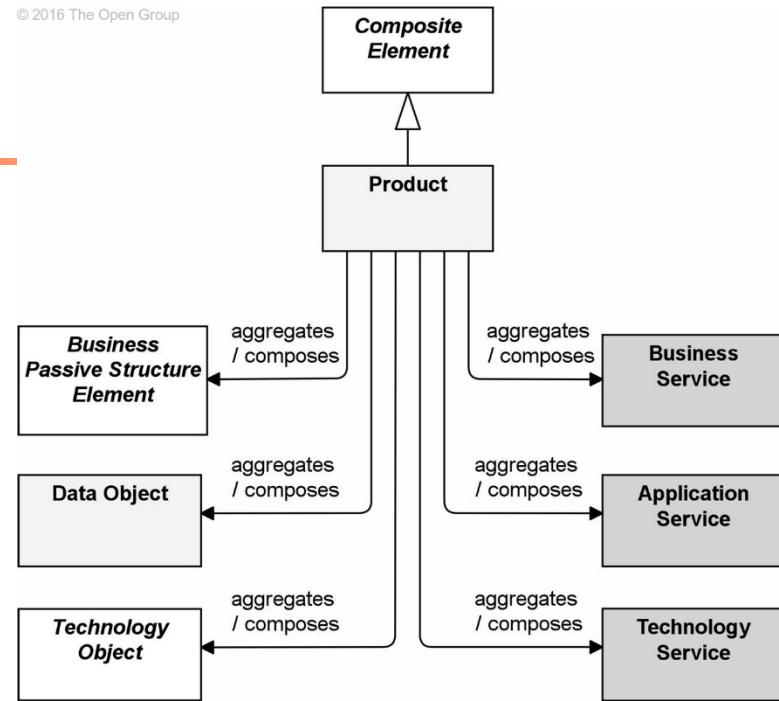
Symbol:



Structural Passive Concepts

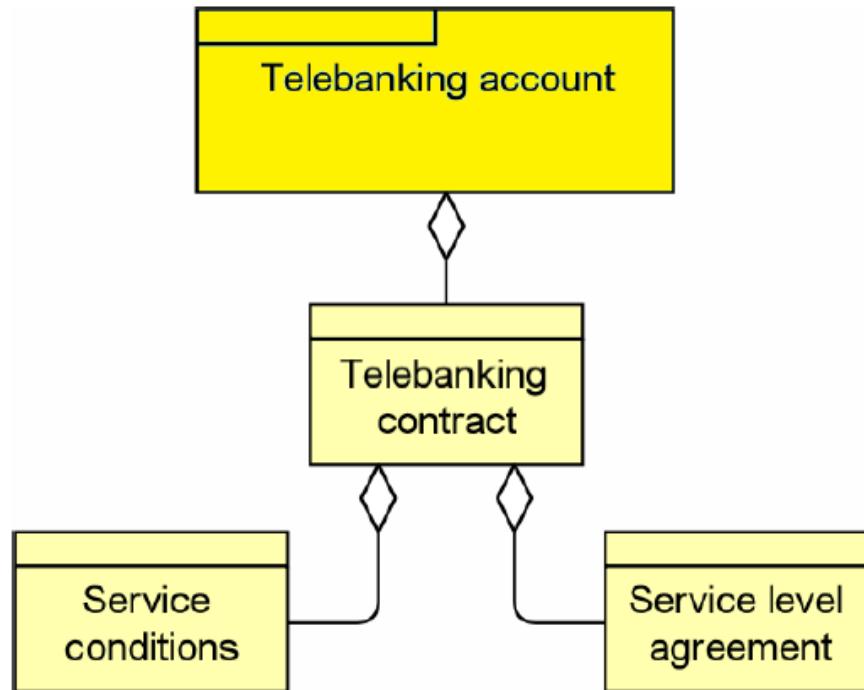
Example:

A product Insurance consists of a contract Insurance Policy and a business service Customer Service, which aggregates four other business services: Application, Renewal, Claims Processing, and Appeal. An Auto Insurance product is a specialization of the generic Insurance product, with an additional business service Drive Well and Save, and accompanying contract Drive Well and Save Agreement.

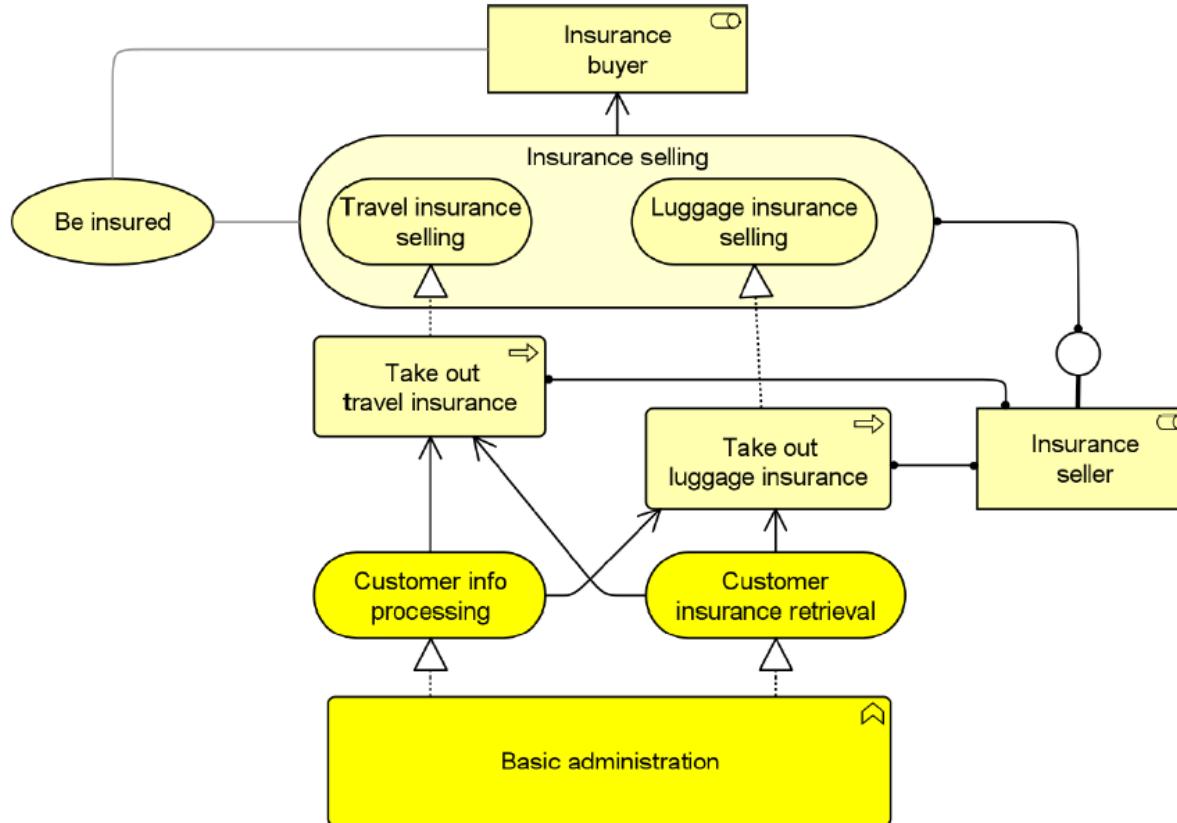


Structural Passive Concepts

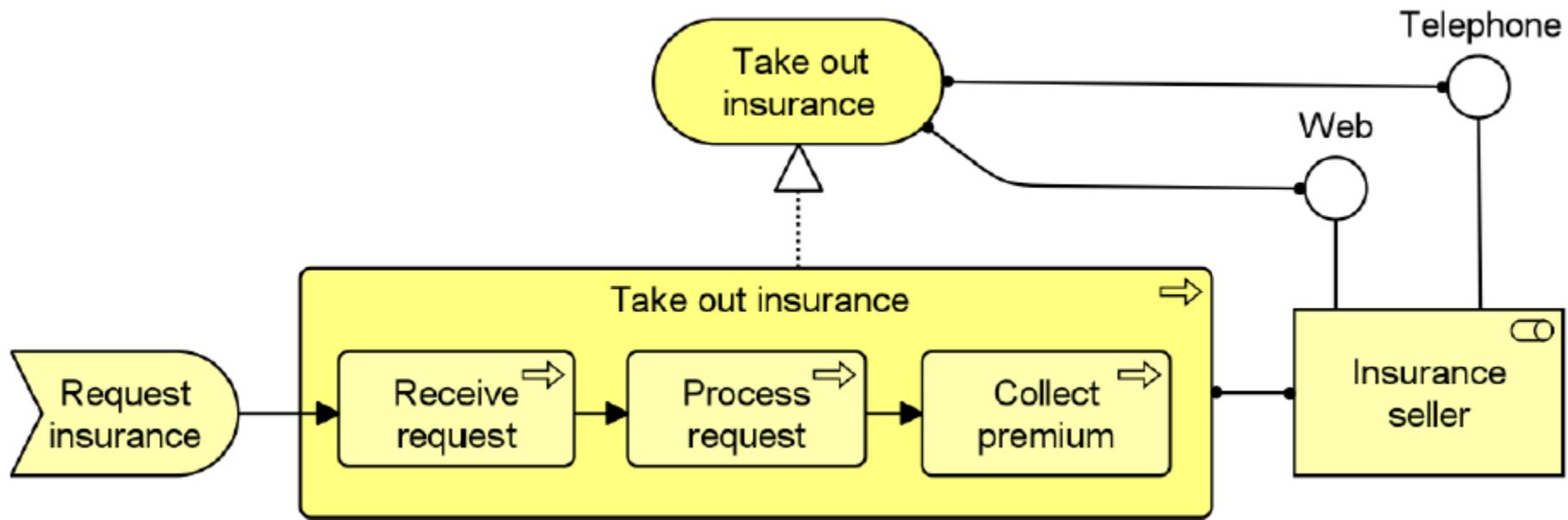
Example of a Product (“Telebanking account”) that aggregates a Contract (Telebanking contract”), which aggregates two Business Objects (“Services Conditions” and “Service level agreement”):



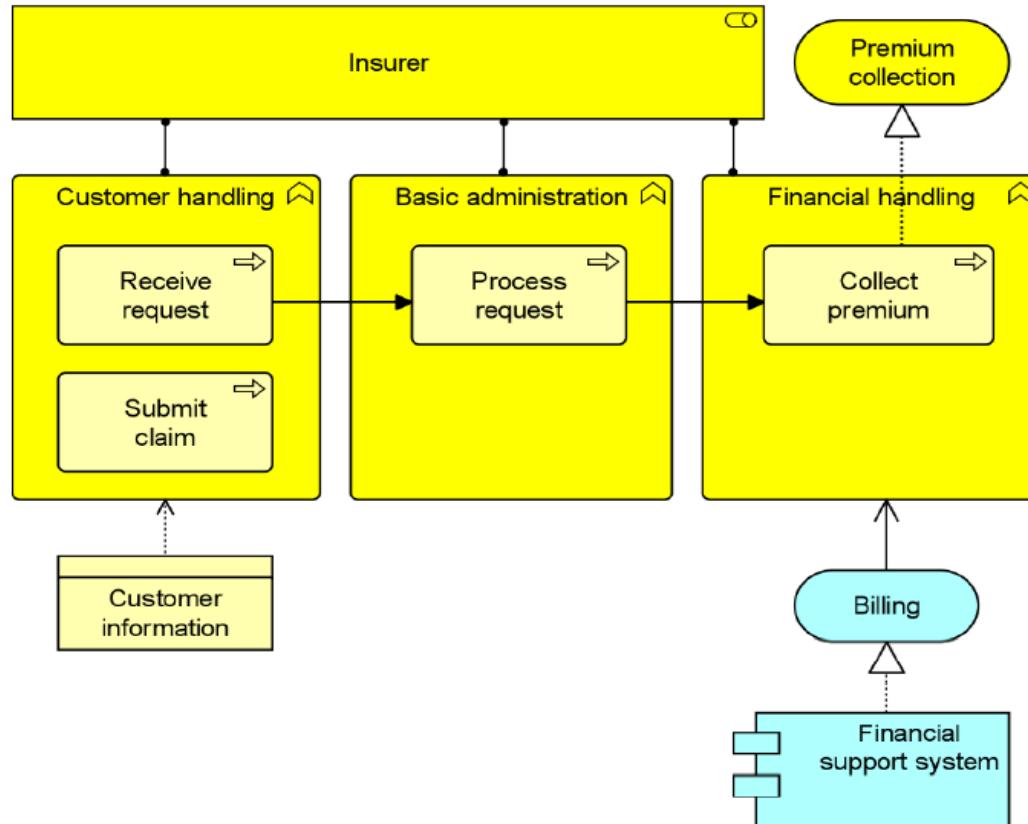
ArchiMate Business Layer Concepts



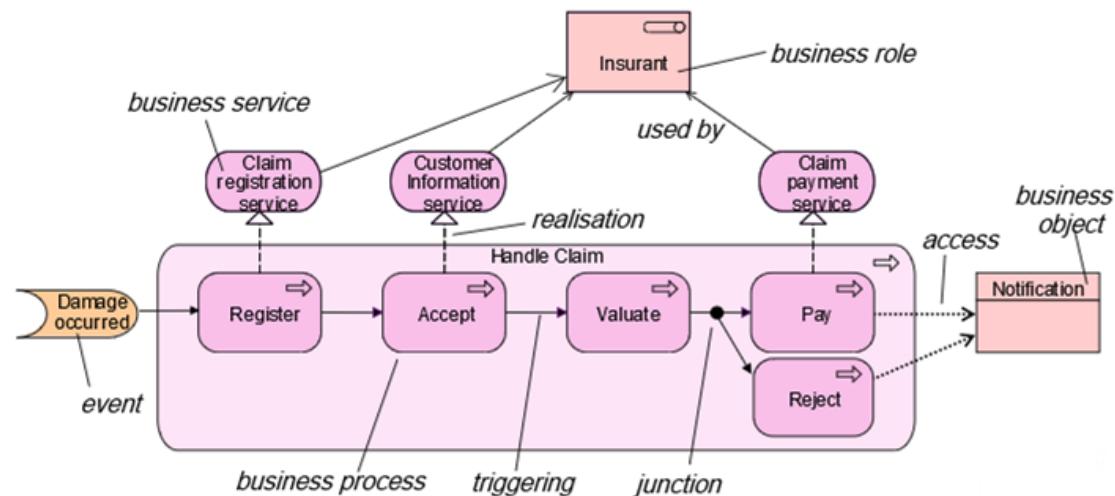
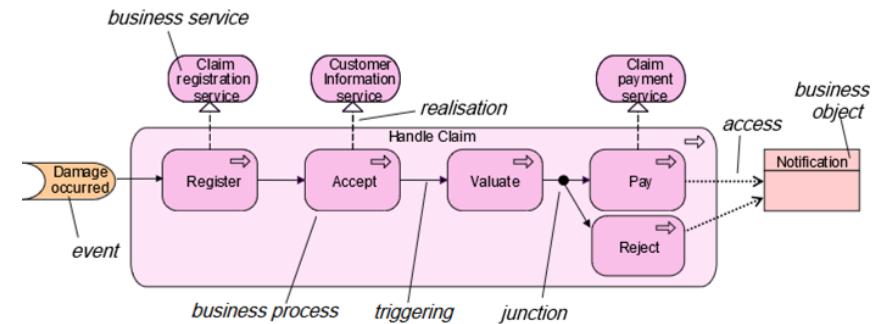
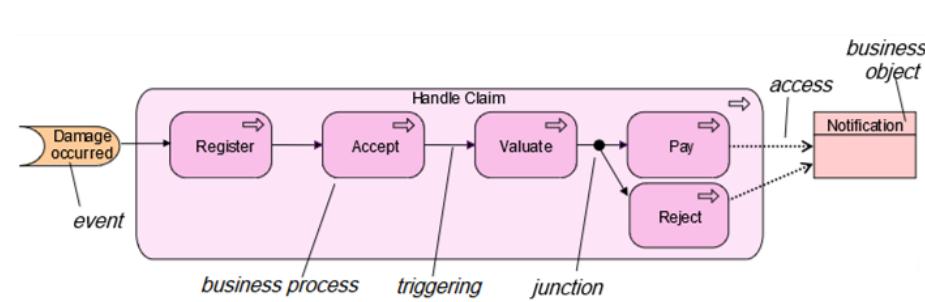
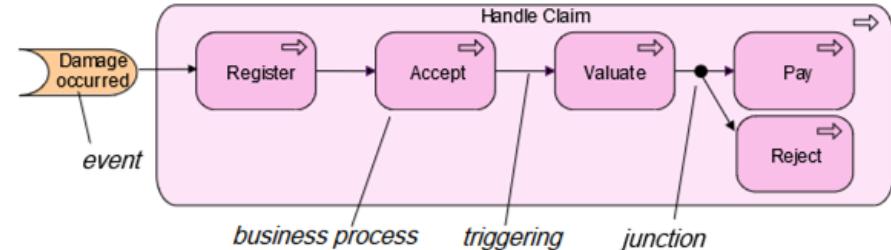
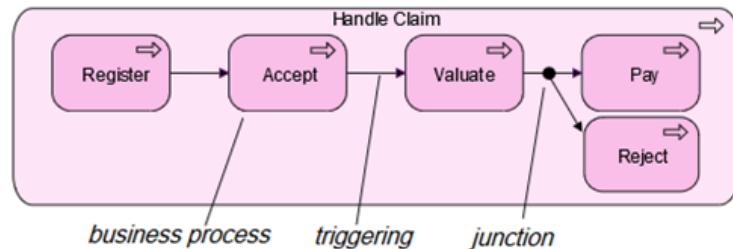
ArchiMate Business Layer Concepts



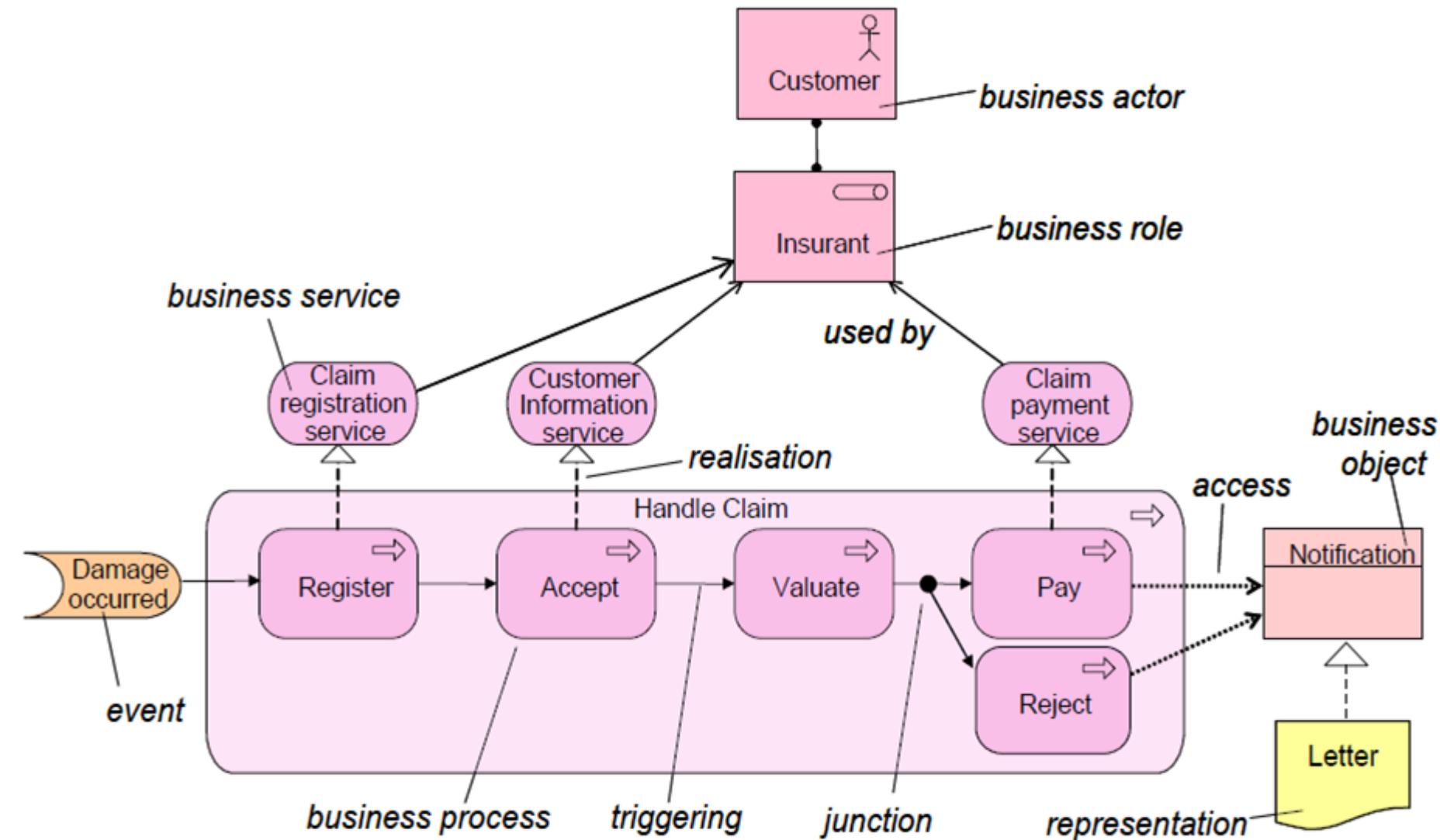
ArchiMate Business Layer Concepts



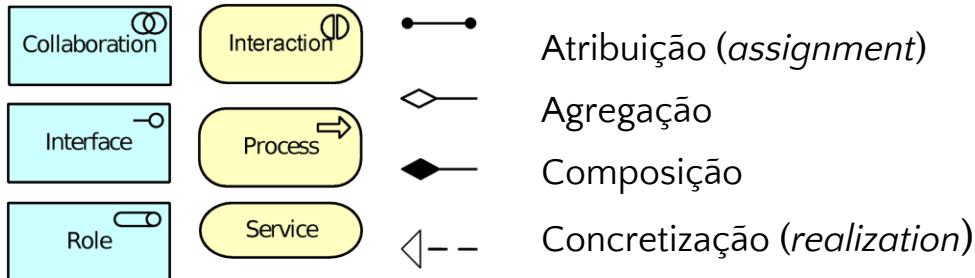
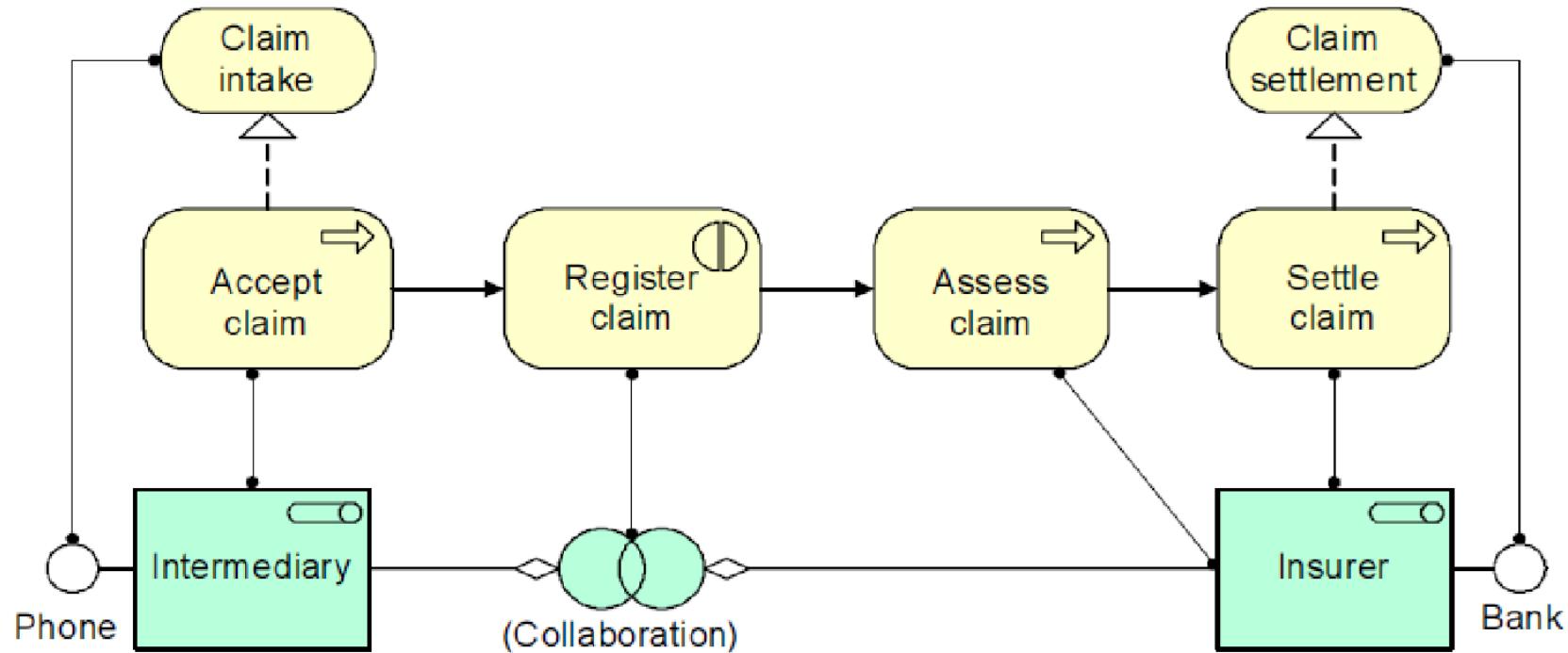
Business Processes Examples...



Business Processes Examples...

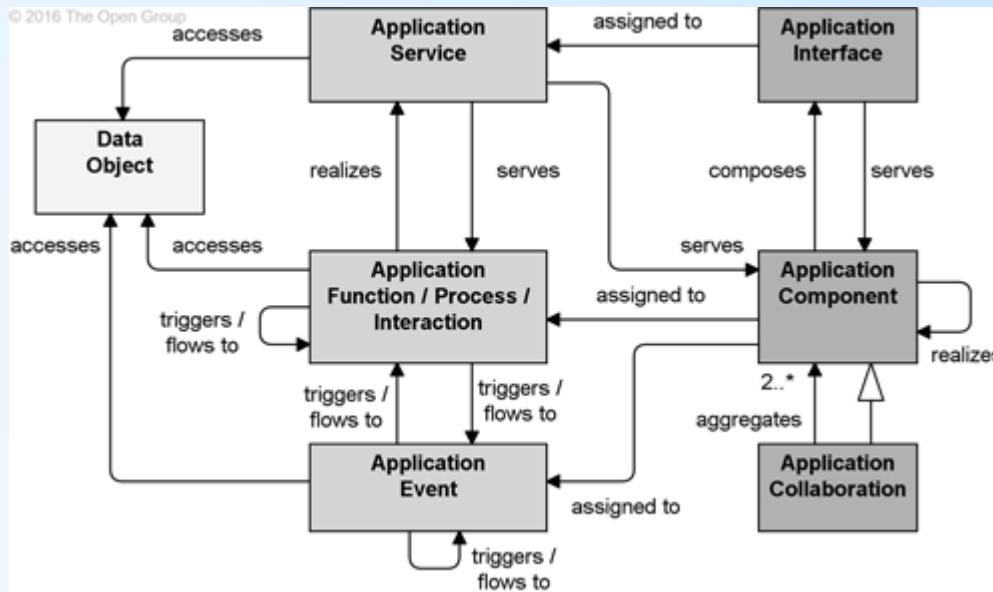


ArchiMate Business Layer Concepts



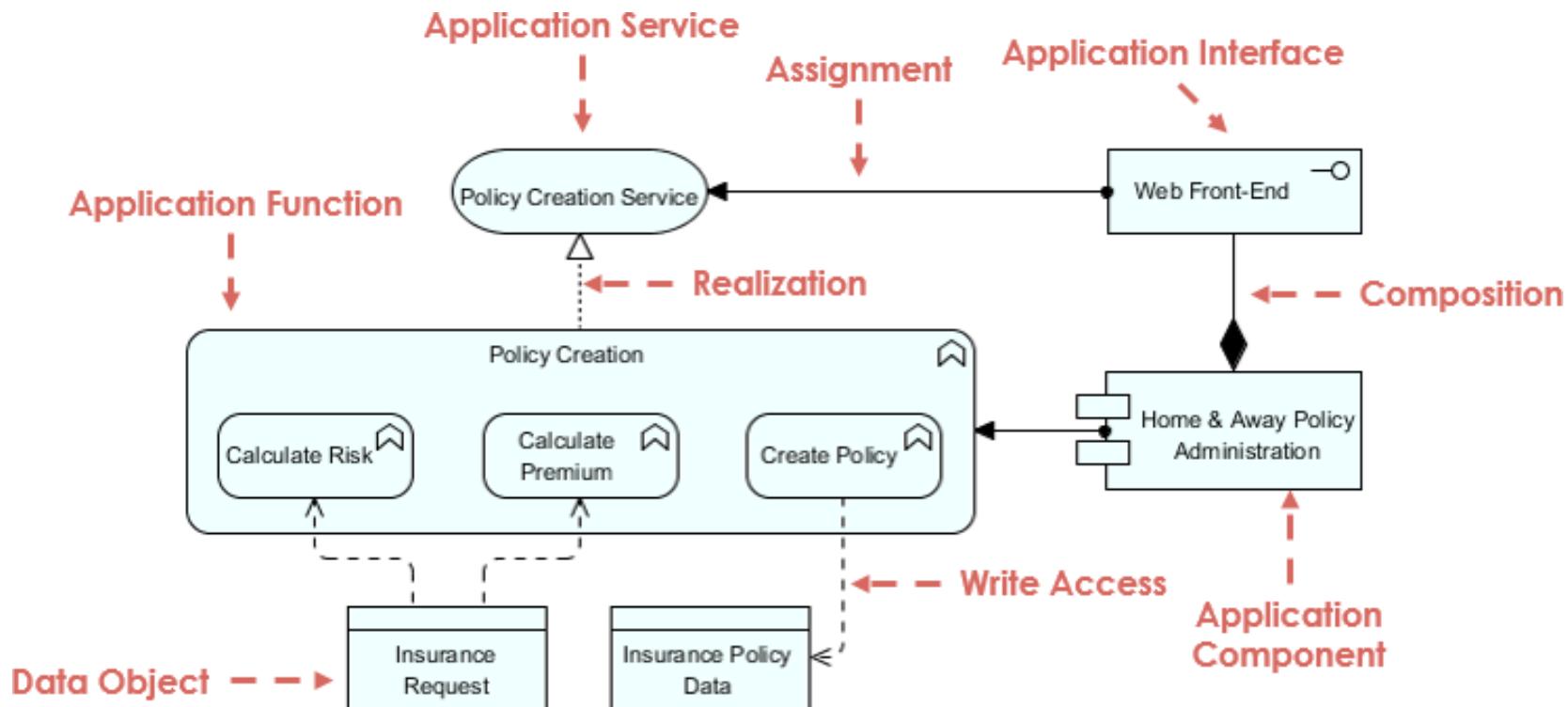


ArchiMate



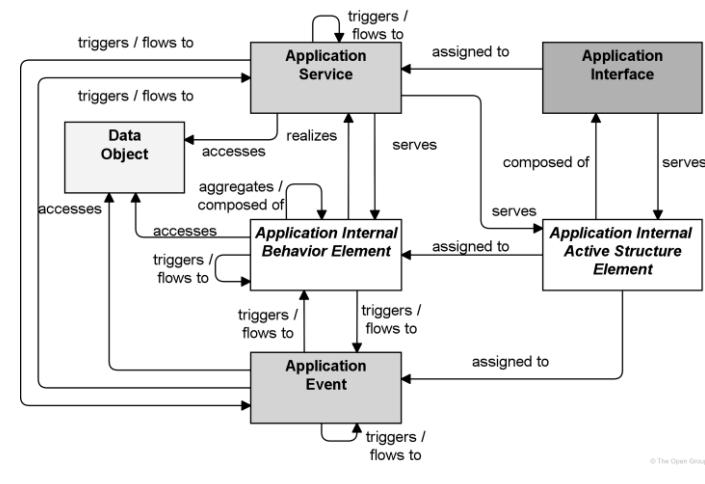
...Application Layer...

ArchiMate Application Layer Concepts



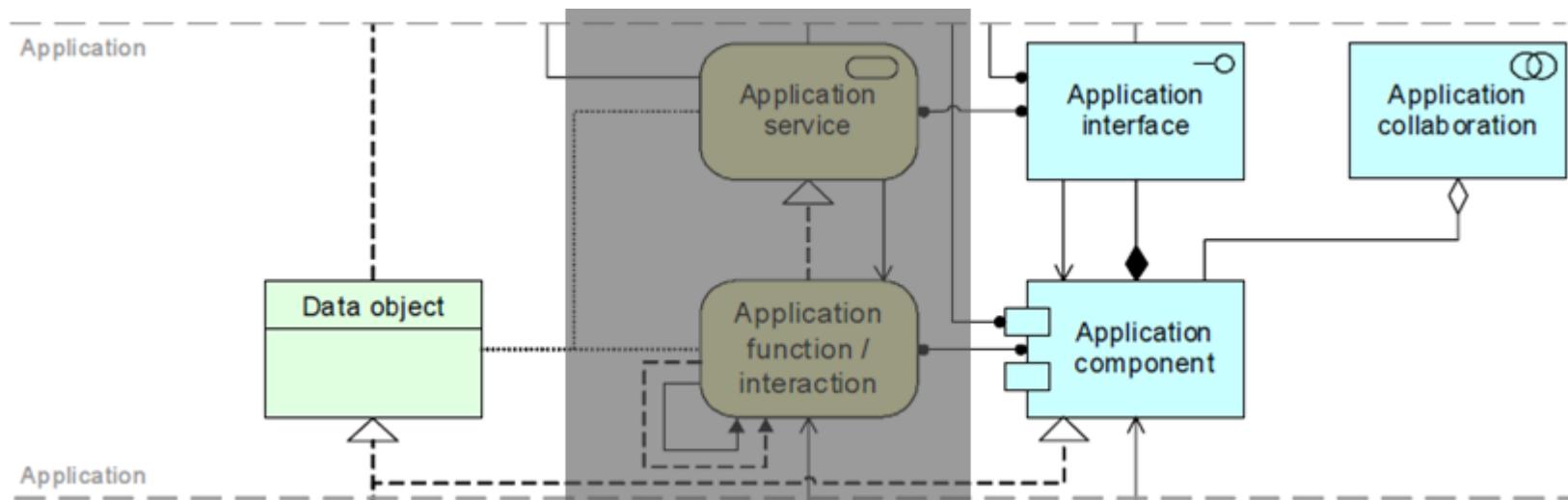
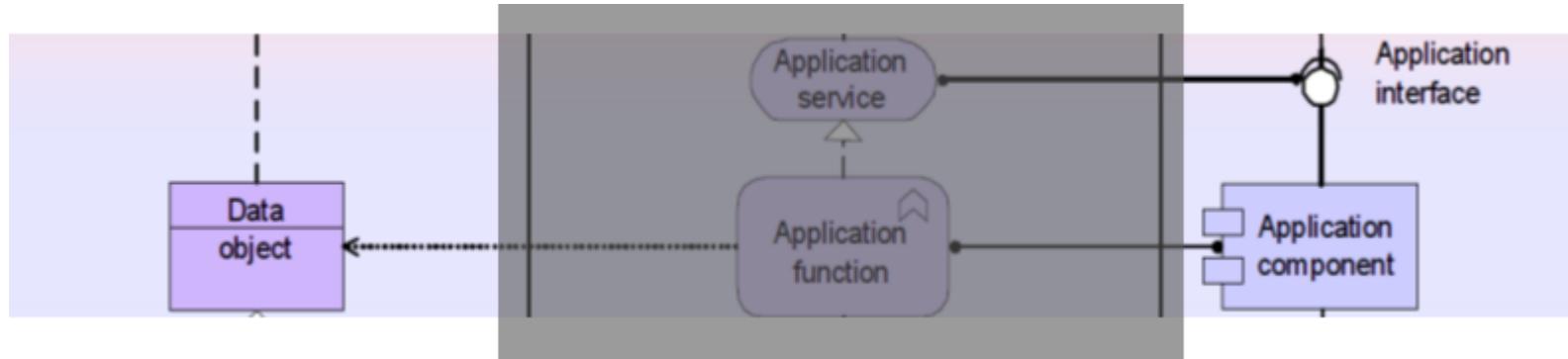
ArchiMate Application Layer Concepts

Element	Definition	Notation
Application Component	Represents an encapsulation of application functionality aligned to implementation structure, which is modular and replaceable.	
Application Collaboration	Represents an aggregate of two or more application internal active structure elements that work together to perform collective application behavior.	
Application Interface	Represents a point of access where application services are made available to a user, another application component, or a node.	
Application Function	Represents automated behavior that can be performed by an application component.	
Application Interaction	Represents a unit of collective application behavior performed by (a collaboration of) two or more application components.	
Application Process	Represents a sequence of application behaviors that achieves a specific result.	
Application Event	Represents an application state change.	
Application Service	Represents an explicitly defined exposed application behavior.	
Data Object	Represents data structured for automated processing.	



The Application Layer is typically used to model the information systems architectures of the enterprise, including the application architecture that describes the structure and interaction of the applications

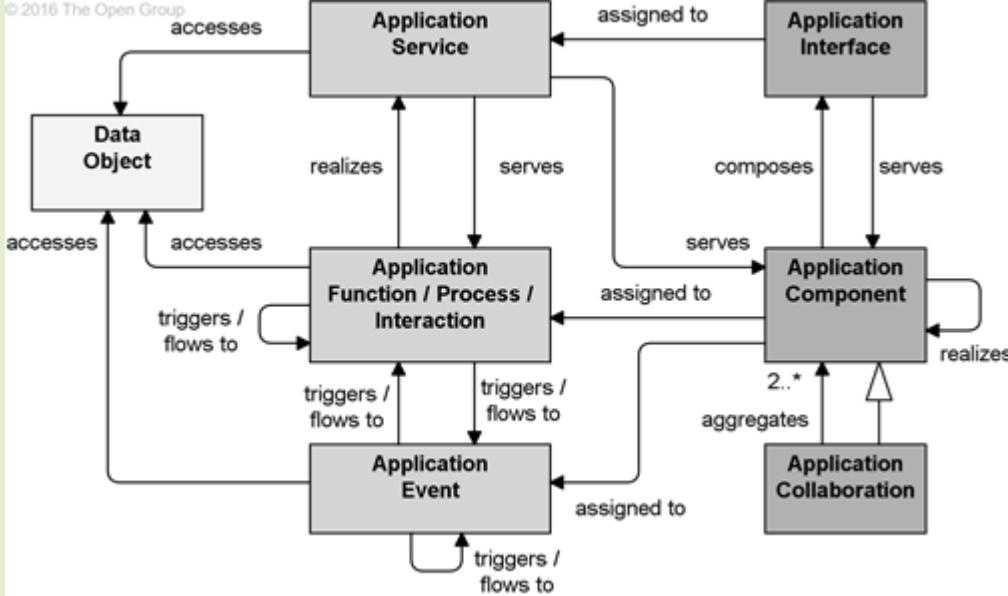
Structure Concepts



Structure Concepts

1. **Application component:** self-contained part of a system that encapsulates its contents and exposes its functionality through a set of **interfaces**.
2. **Application collaboration:** a collective of application **components**, which perform application interactions.
3. **Application interface:** defines the set of operations and events that are *provided* by the **component**, or those that are *required* from the environment.
4. **Data object:** a coherent, self-contained piece of **information** suitable for automated processing.

ArchiMate Application Layer



The main active structure element for the Application Layer is the **application component**. This element is used to model any structural entity in the Application Layer: not just (re-usable) software components that can be part of one or more applications, but also complete software applications, sub-applications, or information systems.

The application component element strictly models the structural aspect of an application; its behavior is modeled by an explicit relationship to the behavior element.

Also in the application architecture, the inter-relationships of components are an essential ingredient. Therefore, we also introduce the element of **application collaboration** here, defined as a collective of application components which perform application interactions.

In the purely structural sense, **an application interface** is the (logical) channel through which the services of a component can be accessed.

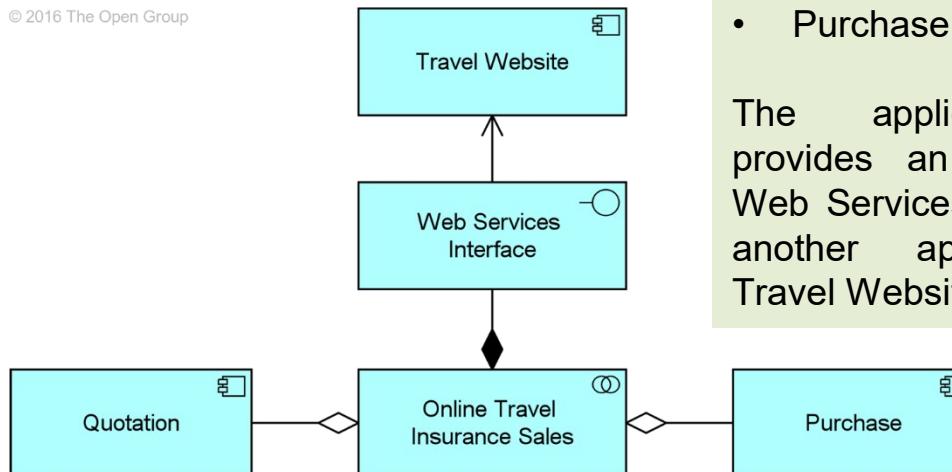
In a broader sense, an application interface defines some elementary behavioral characteristics: it defines the set of operations and events that are provided by the component, or those that are required from the environment. Thus, it is used to describe the functionality of a component.

The application interface element can be used to model both *application-to-application* interfaces, which offer internal application services, and *application-to business* interfaces (and/or *user interfaces*), which offer external application services.

ArchiMate Application Layer Active Structure

Element	Definition	Notation
Application component	An encapsulation of application functionality aligned to implementation structure, which is modular and replaceable. It encapsulates its behavior and data, exposes services, and makes them available through interfaces.	
Application collaboration	An aggregate of two or more application components that work together to perform collective application behavior.	
Application interface	A point of access where application services are made available to a user, another application component, or a node.	

© 2016 The Open Group



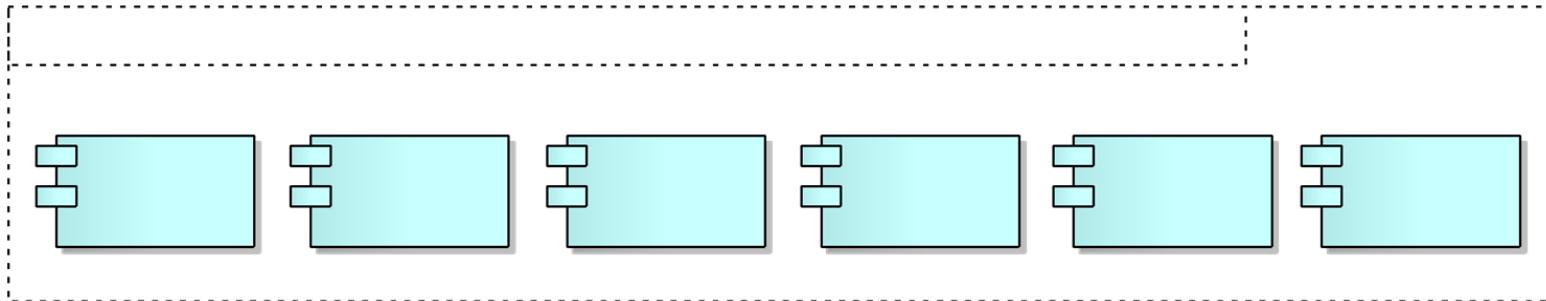
The Online Travel Insurance Sales application collaboration aggregates two application components:

- Quotation;
- Purchase.

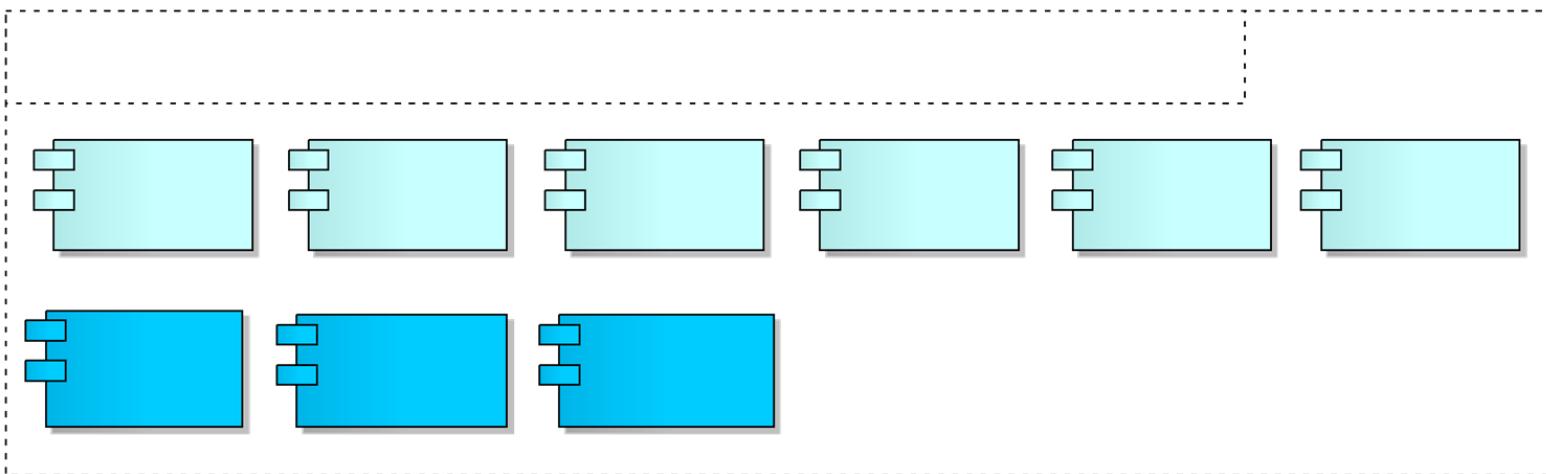
The application collaboration provides an application interface Web Services Interface that serves another application component Travel Website.

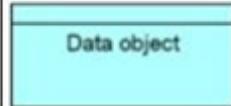
Application Landscape Viewpoint

Baseline Application Model



Target Application Model



Data object	Data structured for automated processing.	
-------------	---	---

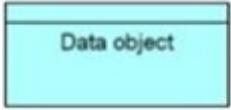
The passive counterpart of the application component in the Application Layer is called a *data object*.

This element is used in the same way as data objects (or object types) in well-known data modeling approaches, most notably the “class” concept in UML class diagrams.

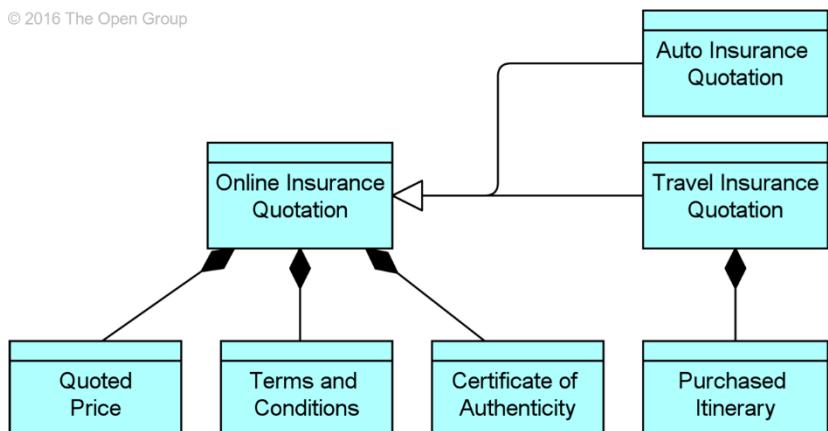
A data object can be seen as a representation of a business object, as a counterpart of the representation element in the Business Layer.

The ArchiMate language does not define a specific layer for information; however, elements such as business objects and data objects are used to represent the information entities and also the logical data components that realize the business objects.

ArchiMate Application Layer Passive Structure

Data object	Data structured for automated processing.	
-------------	---	---

© 2016 The Open Group



An Online Insurance Quotation data object is composed of three other data objects:

- Quoted Price
- Terms and Conditions
- Certificate of Authenticity

Auto Insurance Quotation and Travel Insurance Quotation are two specializations of the Online Insurance Quotation data object.

Travel Insurance Quotation contains an additional data object Purchased Itinerary.

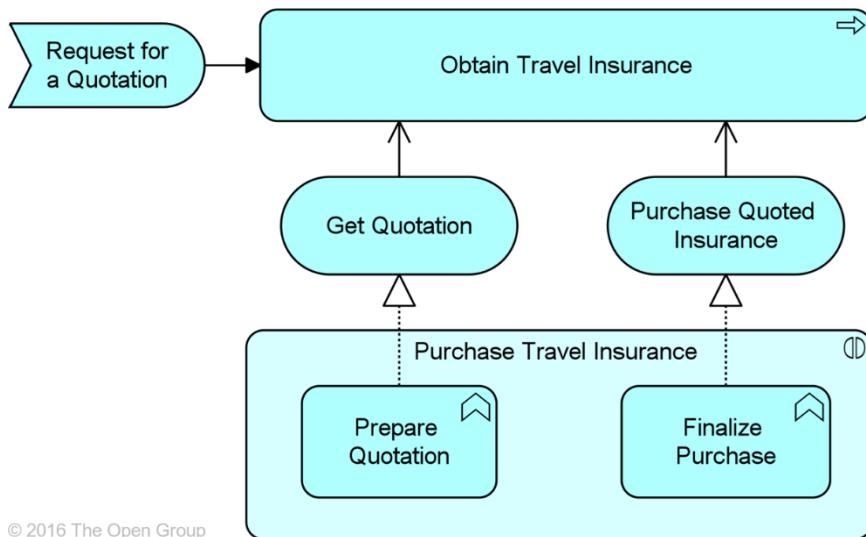
Behavior Concepts

1. **Application service:** an externally visible unit of functionality, provided by one or more **components**, exposed through well-defined **interfaces**, and meaningful to the environment.
2. **Application function:** the internal behaviour of a **component** needed to realise one or more application **services**.
3. **Application interaction:** the behaviour of a collaboration of two or more application **components**.

ArchiMate

Application Layer Behavior Structure

Application function	Automated behavior that can be performed by an application component.		
Application interaction	A unit of collective application behavior performed by (a collaboration of) two or more application components.		
Application process	A sequence of application behaviors that achieves a specific outcome.		
Application event	An application behavior element that denotes a state change.		
Application service	An explicitly defined exposed application behavior.		

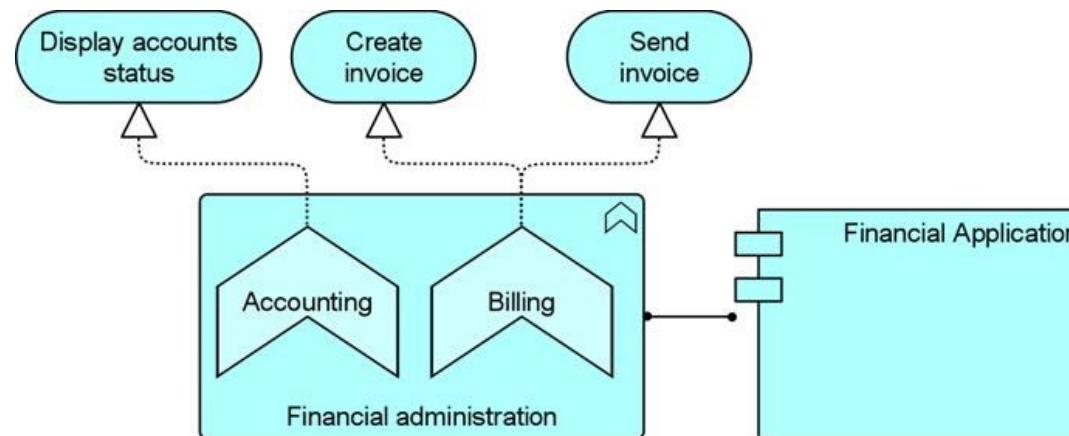
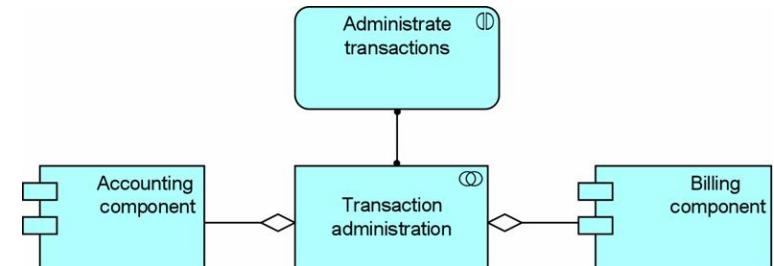
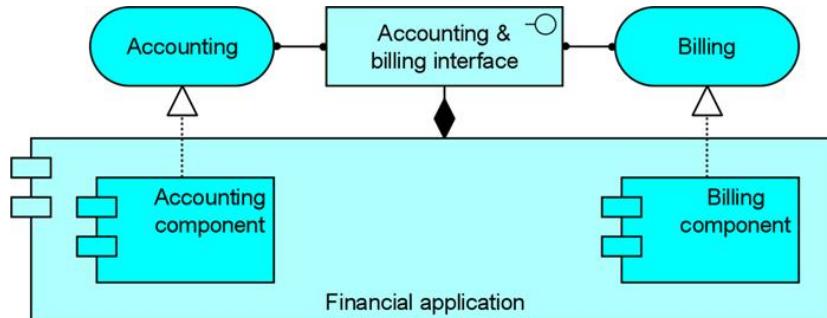


The Purchase Travel Insurance application interaction is composed of two application functions: Prepare Quotation, realizing an application service Get Quotation, and Finalize Purchase, realizing an application service Purchase Quoted Insurance.

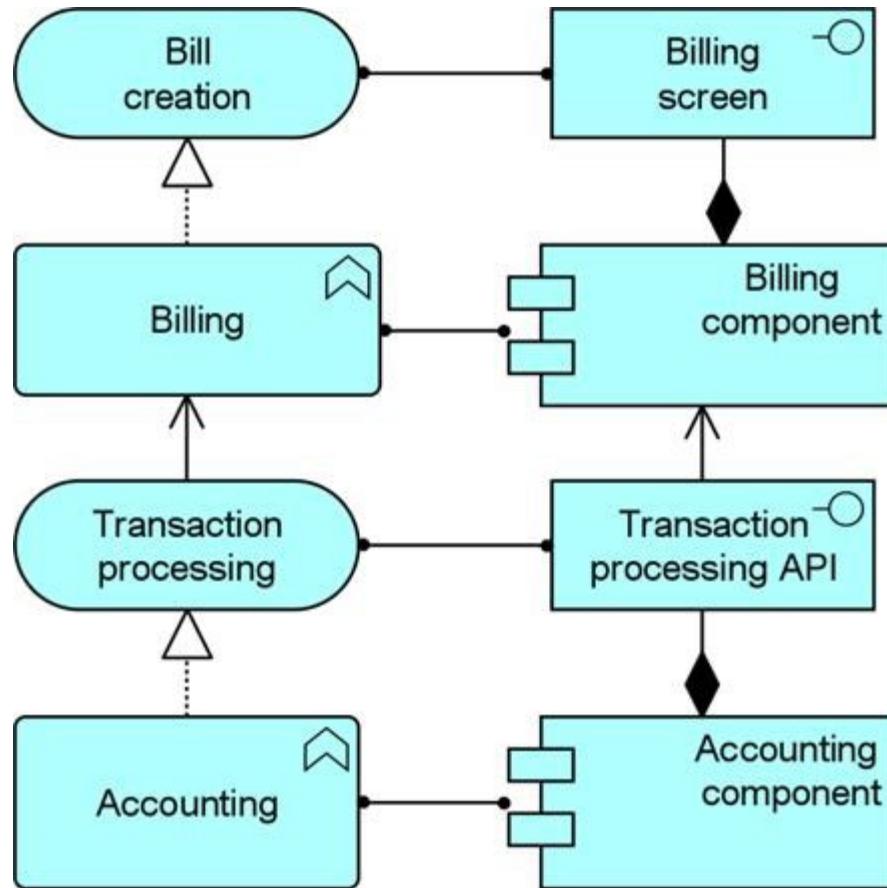
This application interaction models the cooperative behavior of the Quotation and Purchase application components, modeled as the application collaboration Online Travel Insurance.

An application event Request for a Quotation triggers an application process Obtain Travel Insurance, which is served by the two aforementioned application services.

Examples

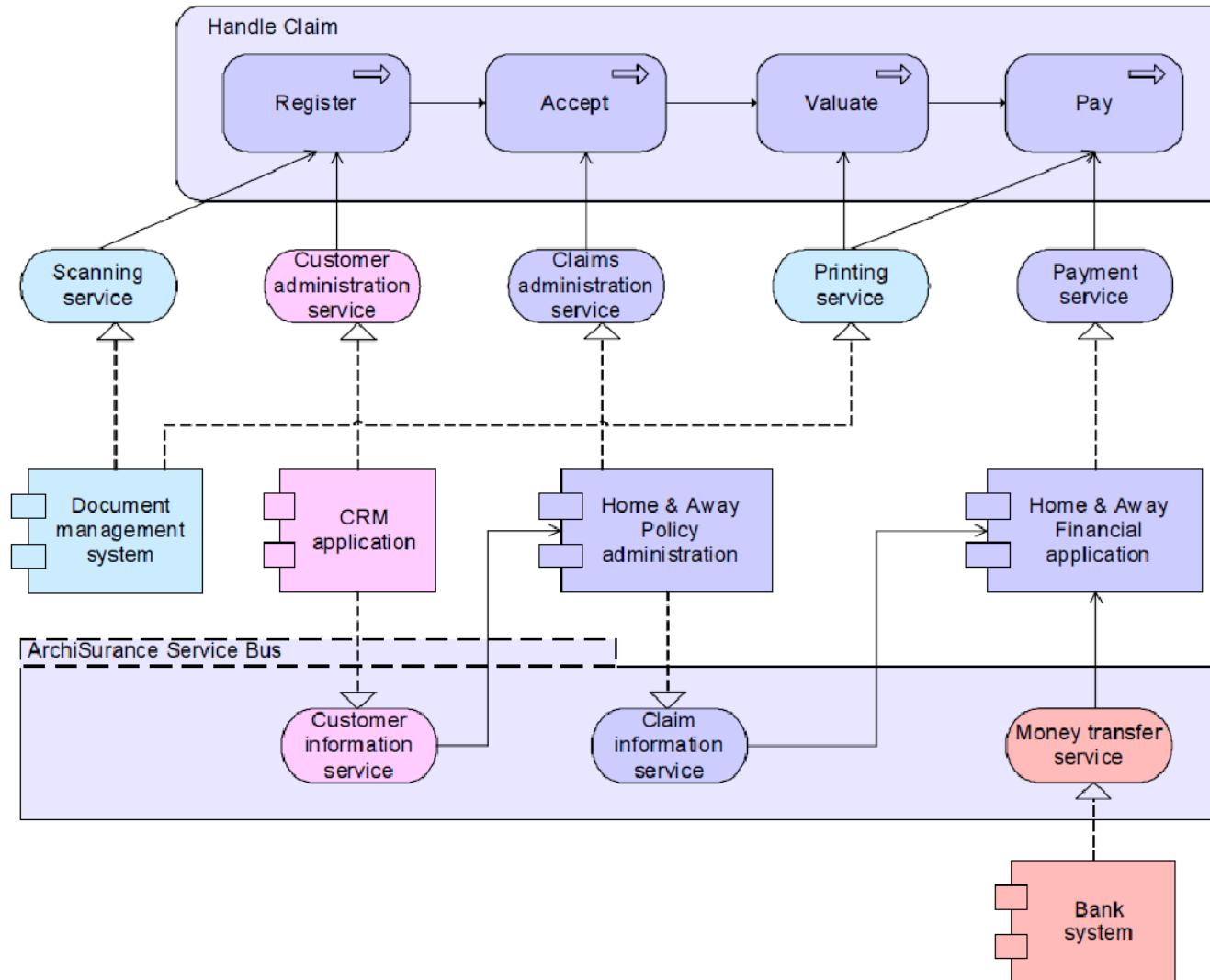


Example

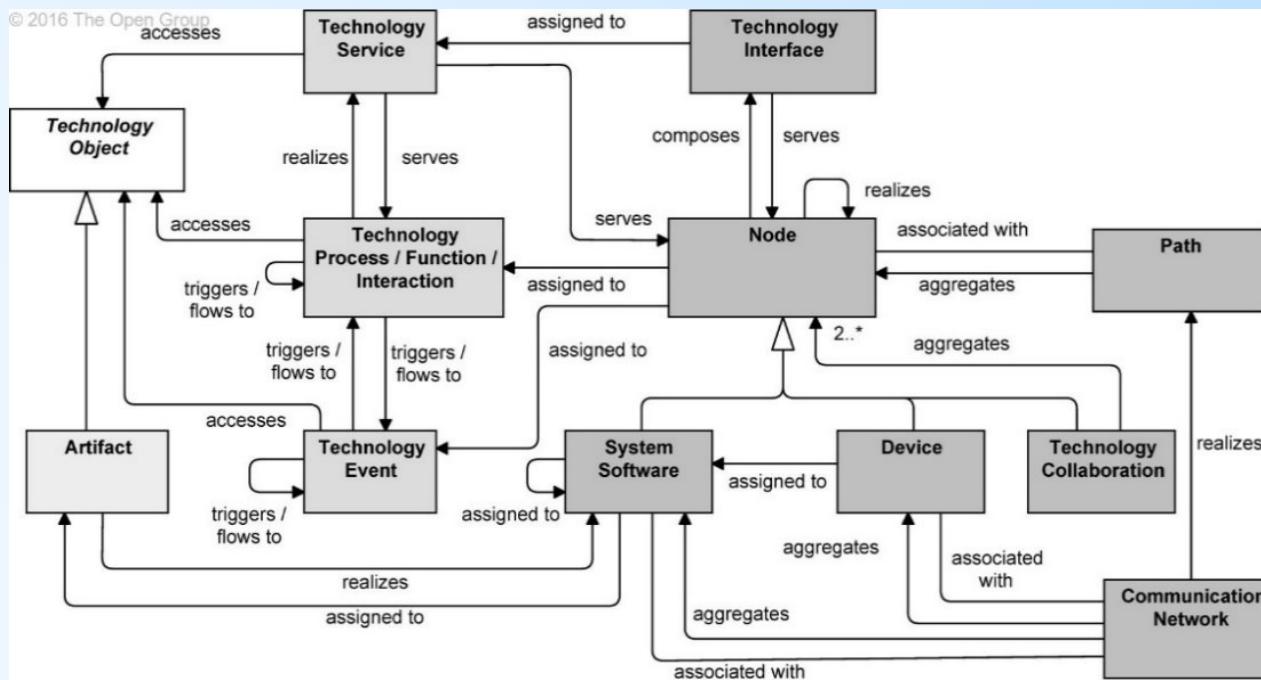


Example: processes, services and applications

- Business and Application Layer

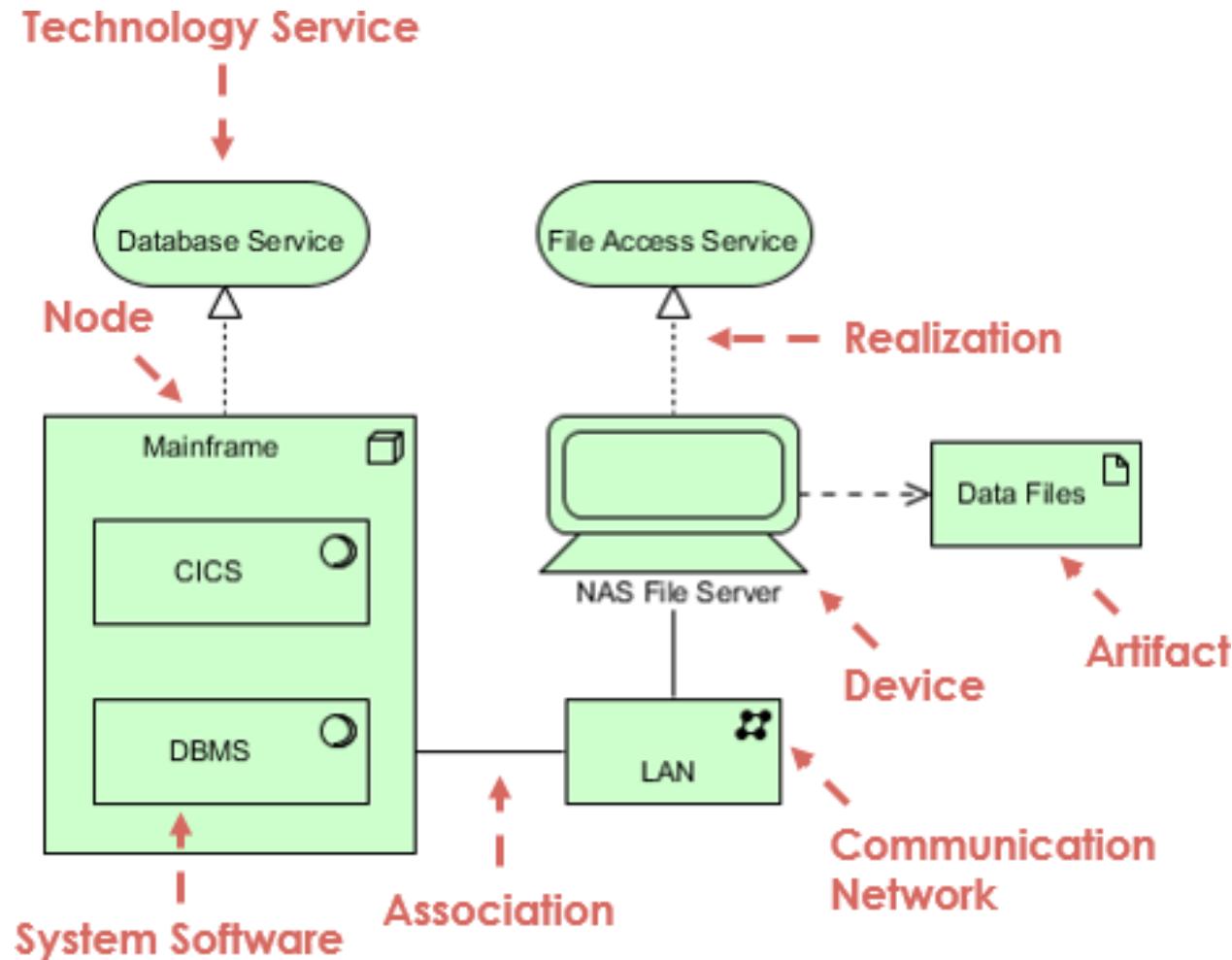


ArchiMate



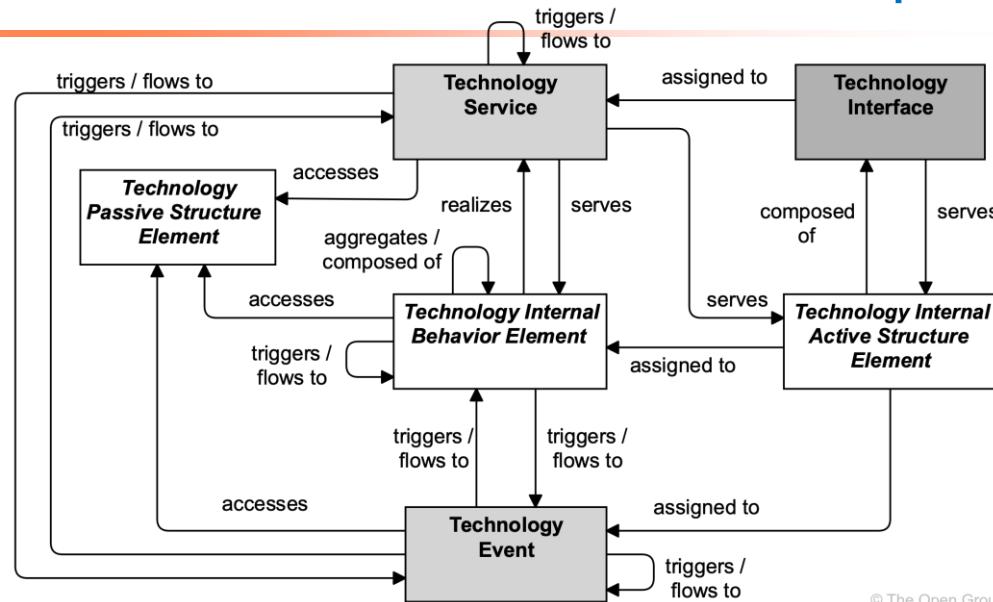
...Technology
Layer...

ArchiMate Technology Layer Concepts



ArchiMate Technology Layer Concepts

Element	Definition	Notation
Node	Represents a computational or physical resource that hosts, manipulates, or interacts with other computational or physical resources.	
Device	Represents a physical IT resource upon which system software and artifacts may be stored or deployed for execution.	
System Software	Represents software that provides or contributes to an environment for storing, executing, and using software or data deployed within it.	
Technology Collaboration	Represents an aggregate of two or more technology internal active structure elements that work together to perform collective technology behavior.	
Technology Interface	Represents a point of access where technology services offered by a technology internal active structure can be accessed.	
Path	Represents a link between two or more technology internal active structure elements, through which these elements can exchange data, energy, or material.	
Communication Network	Represents a set of structures that connects devices or system software for transmission, routing, and reception of data.	
Technology Function	Represents a collection of technology behavior that can be performed by a technology internal active structure element.	
Technology Process	Represents a sequence of technology behaviors that achieves a specific result.	
Technology Interaction	Represents a unit of collective technology behavior performed by (a collaboration of) two or more technology internal active structure elements.	
Technology Event	Represents a technology state change.	
Technology Service	Represents an explicitly defined exposed technology behavior.	
Artifact	Represents a piece of data that is used or produced in a software development process, or by deployment and operation of an IT system.	
Equipment	Represents one or more physical machines, tools, or instruments that can create, use, store, move, or transform materials.	
Facility	Represents a physical structure or environment.	
Distribution Network	Represents a physical network used to transport materials or energy.	
Material	Represents tangible physical matter or energy.	



© The Open Group

The Technology Layer is typically used to model the technology architecture of the enterprise, defined by the TOGAF framework as: “*the structure and interaction of the platform services, and logical and physical technology components*”.

Structural Concepts

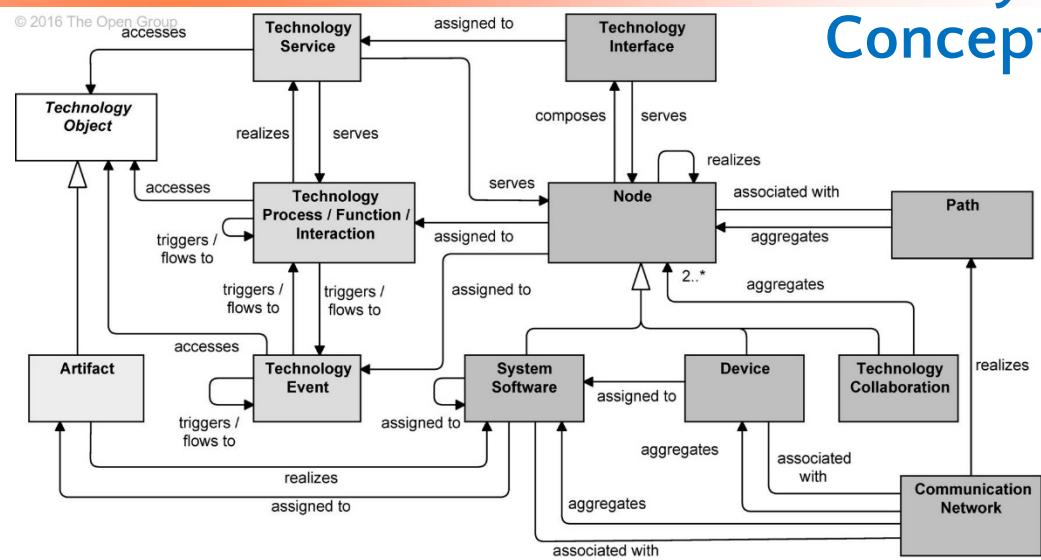
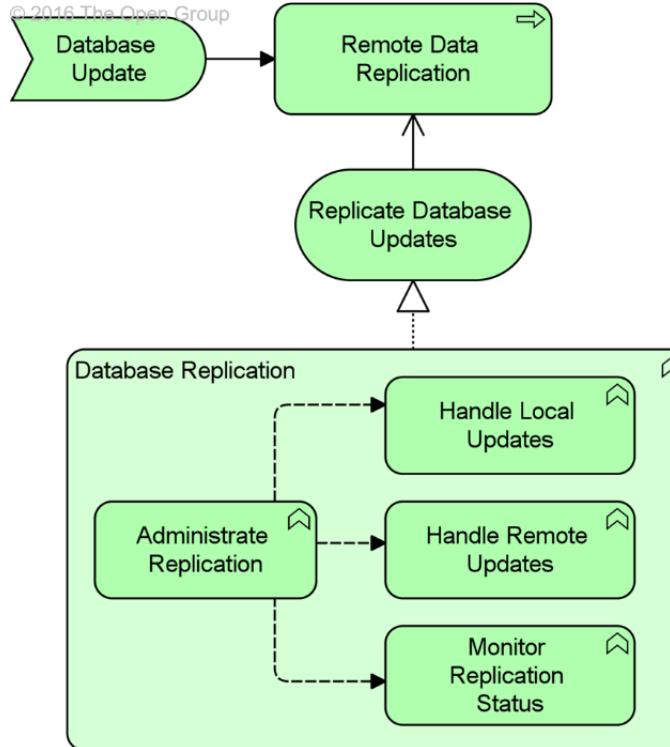
1. **Node:** a (logical) computational resource upon which artifacts may be deployed for execution.
2. **System Software:** : a software environment for specific types of application components and data objects that are deployed on it in the form of artifacts
3. **Device:** a physical computational resource, upon which **artifacts** may be deployed for execution.
4. **Artifact:** a physical piece of information that is used or produced in a software development process, or by deployment and operation of a system (similar to UML 2).
5. **Infrastructure interface:** the point of access location where the infrastructural services offered by a **node** can be accessed by other nodes or by **application components**
6. **Communication path:** a logical link between two or more **nodes**, through which these nodes can exchange information.
7. **Network:** a physical communication medium between two or more **devices**.

Behavioral Concepts

1. **Infrastructure function:** groups infrastructural behavior that can be performed by a node.
2. **Infrastructure service:** externally visible unit of functionality, provided by one or more **nodes**, exposed through well-defined **interfaces**, and meaningful to the environment.

ArchiMate Technology Layer Concepts

Technology function	A collection of technology behavior that can be performed by a node.	
Technology process	A sequence of technology behaviors that achieves a specific outcome.	
Technology interaction	A unit of collective technology behavior performed by (a collaboration of) two or more nodes.	
Technology event	A technology behavior element that denotes a state change.	
Technology service	An explicitly defined exposed technology behavior.	



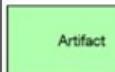
A technology event Database Update triggers a technology process Remote Data Replication, which is served by a technology service Replicate Database Updates.

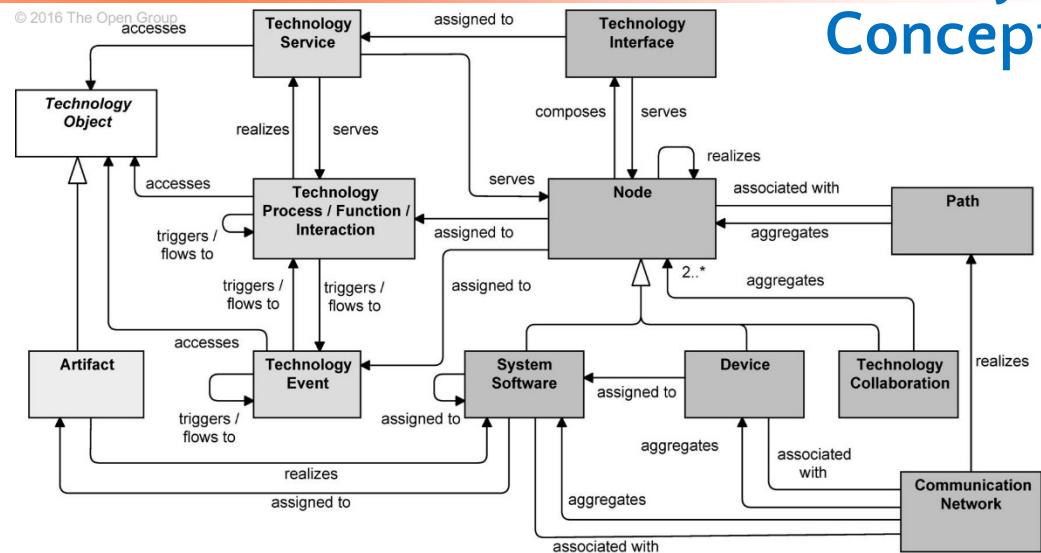
This technology service is realized by a technology function Database Replication, which is composed of four other technology functions:

- Administrate Replication,
- Handle Local Updates,
- Handle Remote Updates,
- Monitor Replication Status.

There are information flows from the Administrate Replication technology function to the other three technology functions.

ArchiMate Technology Layer Concepts

Technology object	A passive element that is used or produced by technology behavior.	Abstract element
Artifact	A piece of data that is used or produced in a software development process, or by deployment and operation of a system.	 

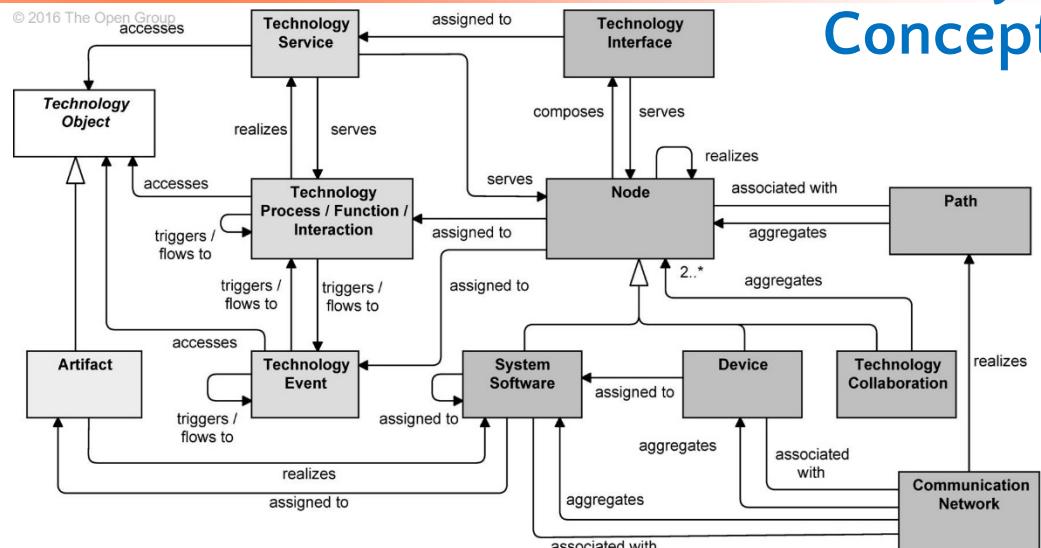


A *technology object* models the passive structure elements that are used and processed by the infrastructure.

An *artifact* is a physical piece of information that is used or produced in a software development process, or by deployment and operation of a system. It is the representation, in the form of, for example, a file, of a data object, or an application component, and can be deployed on a node.

ArchiMate Technology Layer Concepts

Technology object	A passive element that is used or produced by technology behavior.	Abstract element
Artifact	A piece of data that is used or produced in a software development process, or by deployment and operation of a system.	 



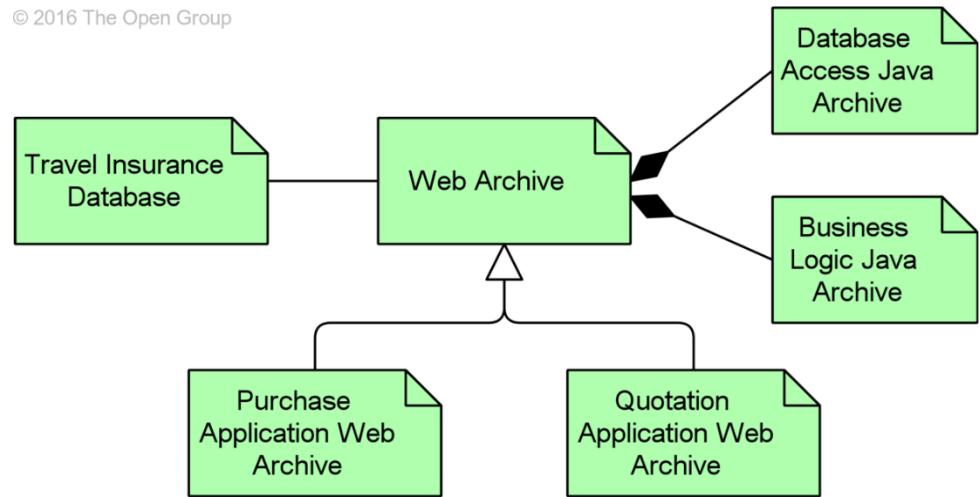
A Web Archive artifact (which may realize an application component) is composed of two other artifacts:

- Database Access Java Archive
 - Business Logic Java Archive.

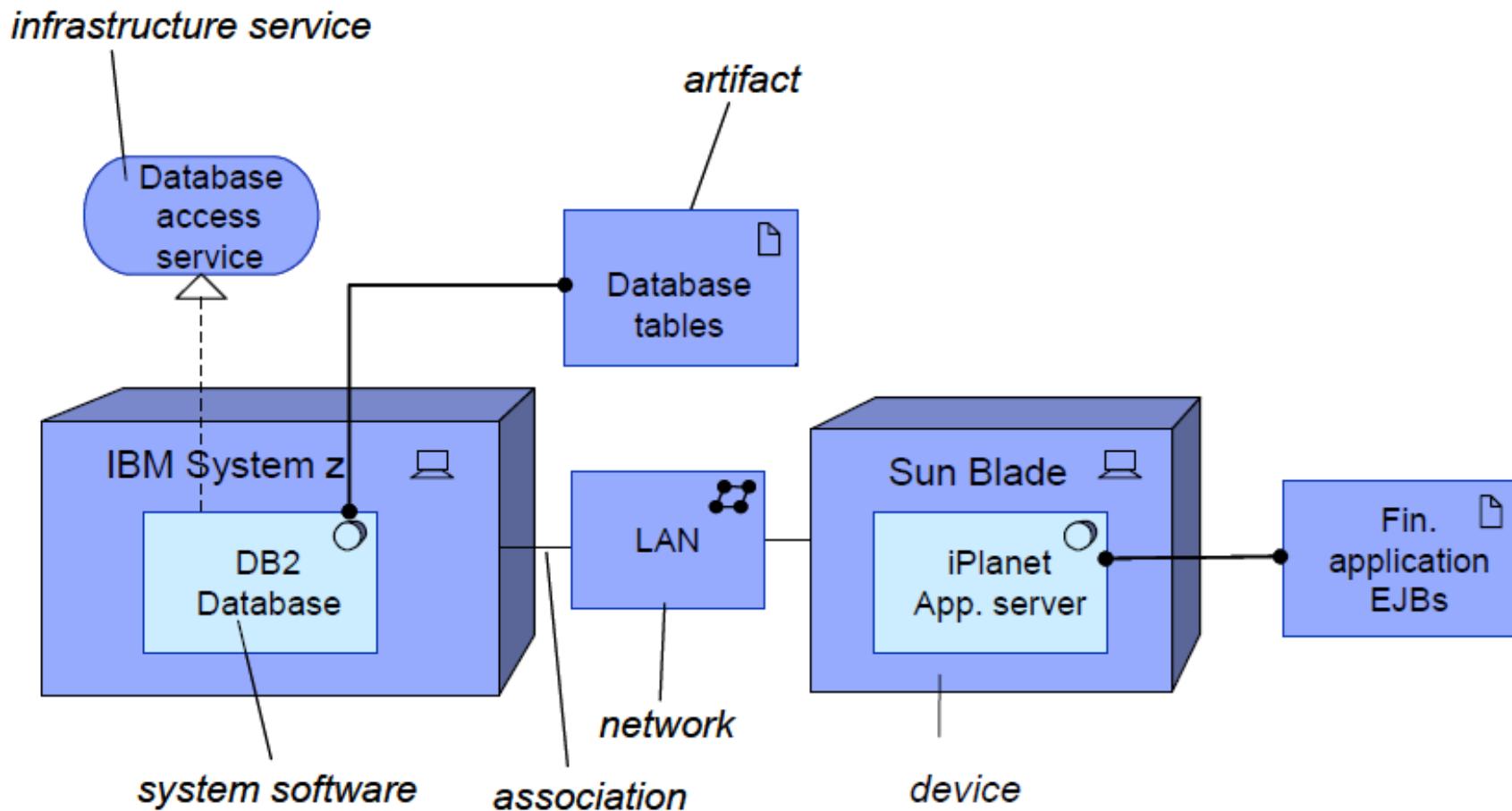
Two specializations of the Web Archive artifact are:

- A Purchase Application Web Archive
 - a Quotation Application Web Archive.

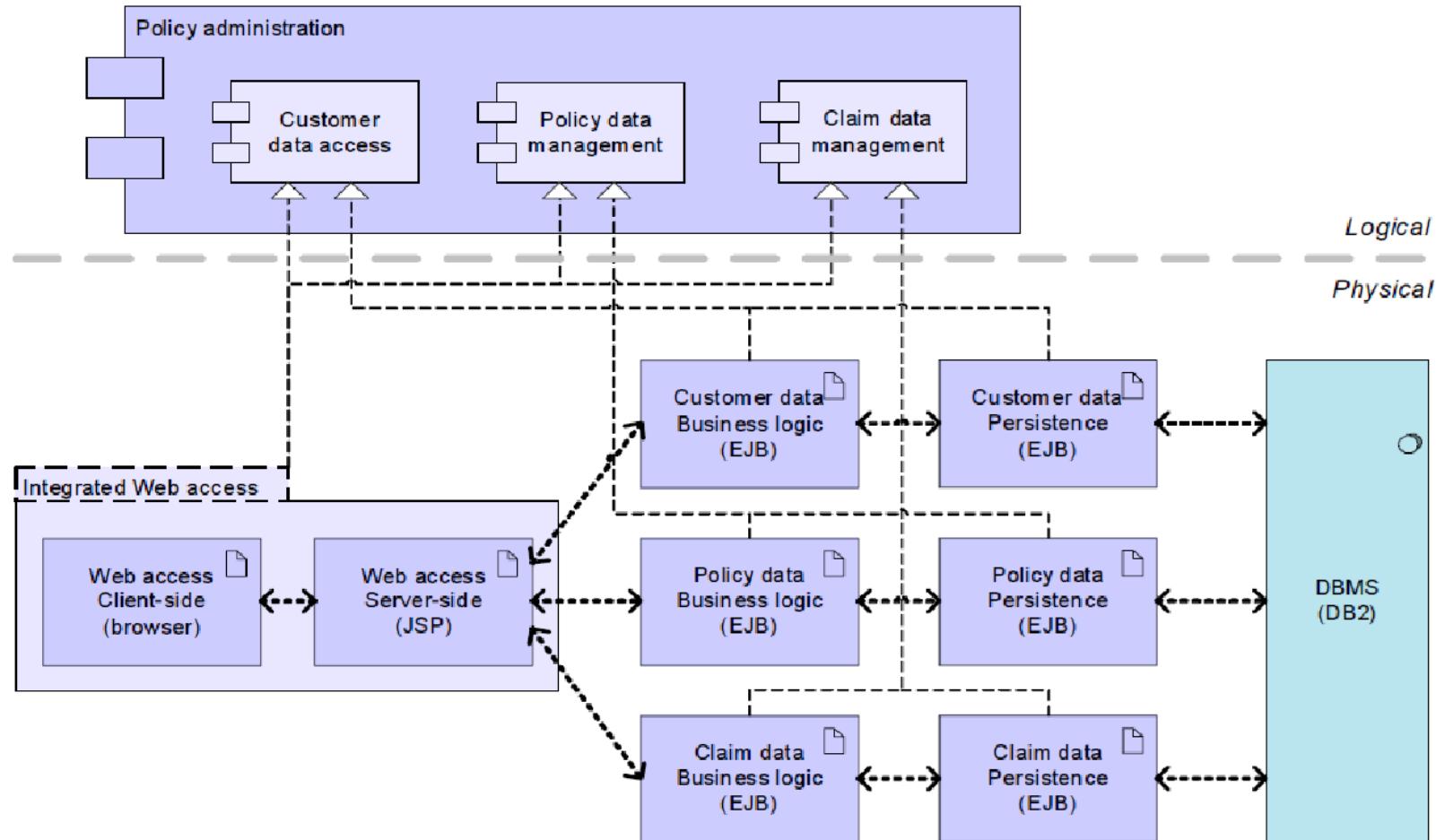
A Travel Insurance Database artifact (which may realize a data object) is associated with the Web Archive artifact.



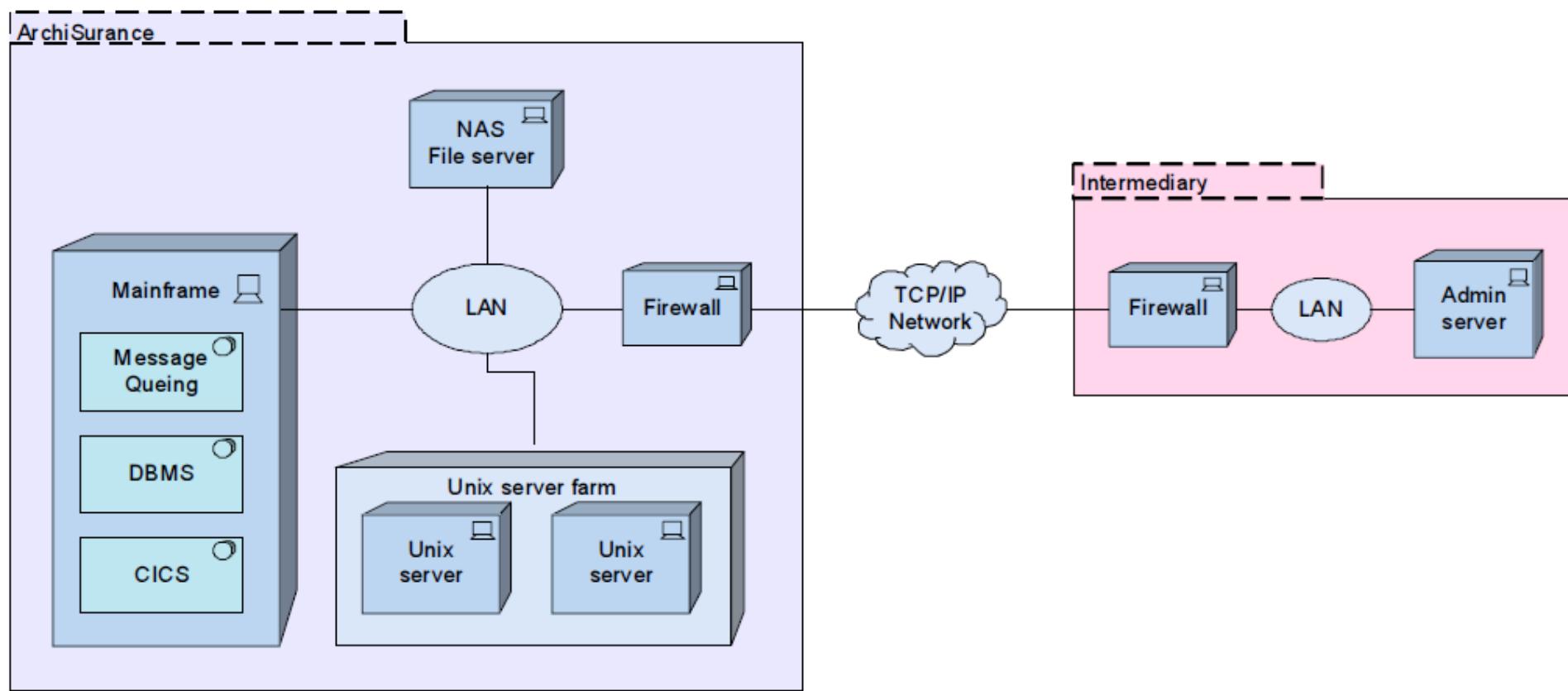
Example



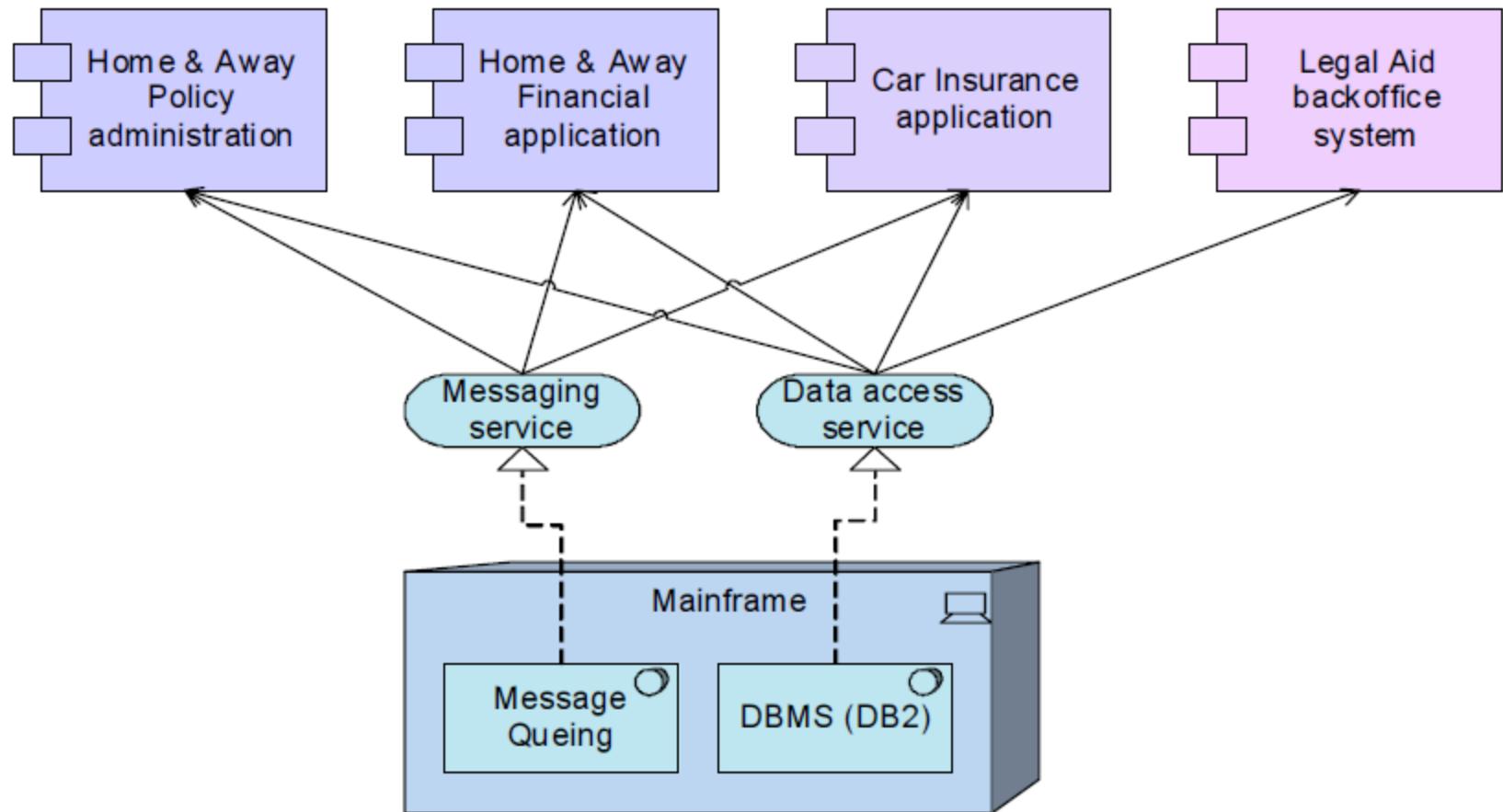
Example - Implementation & Deployment Viewpoint



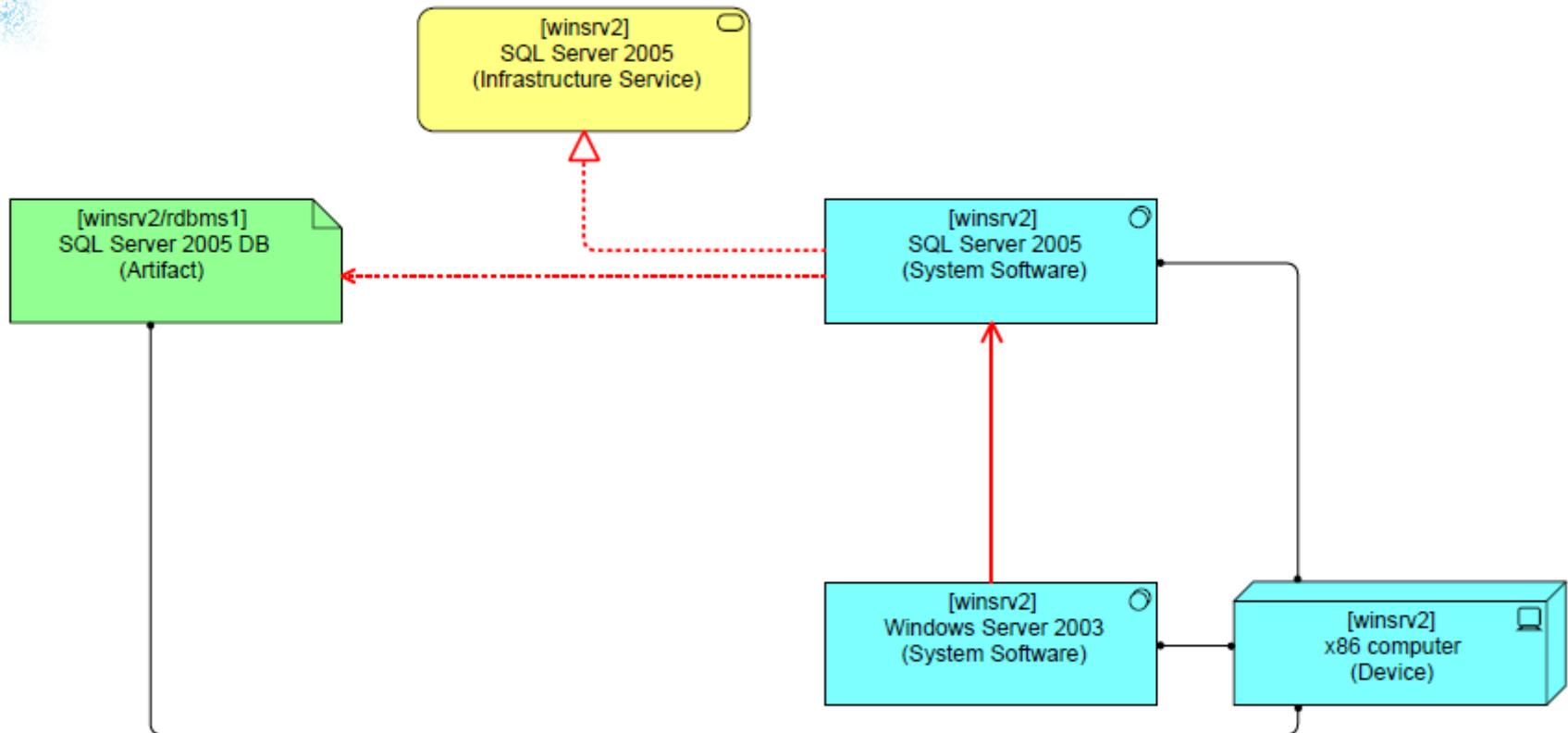
Example - Infrastructure Viewpoint



Example - Infrastructure Usage Viewpoint

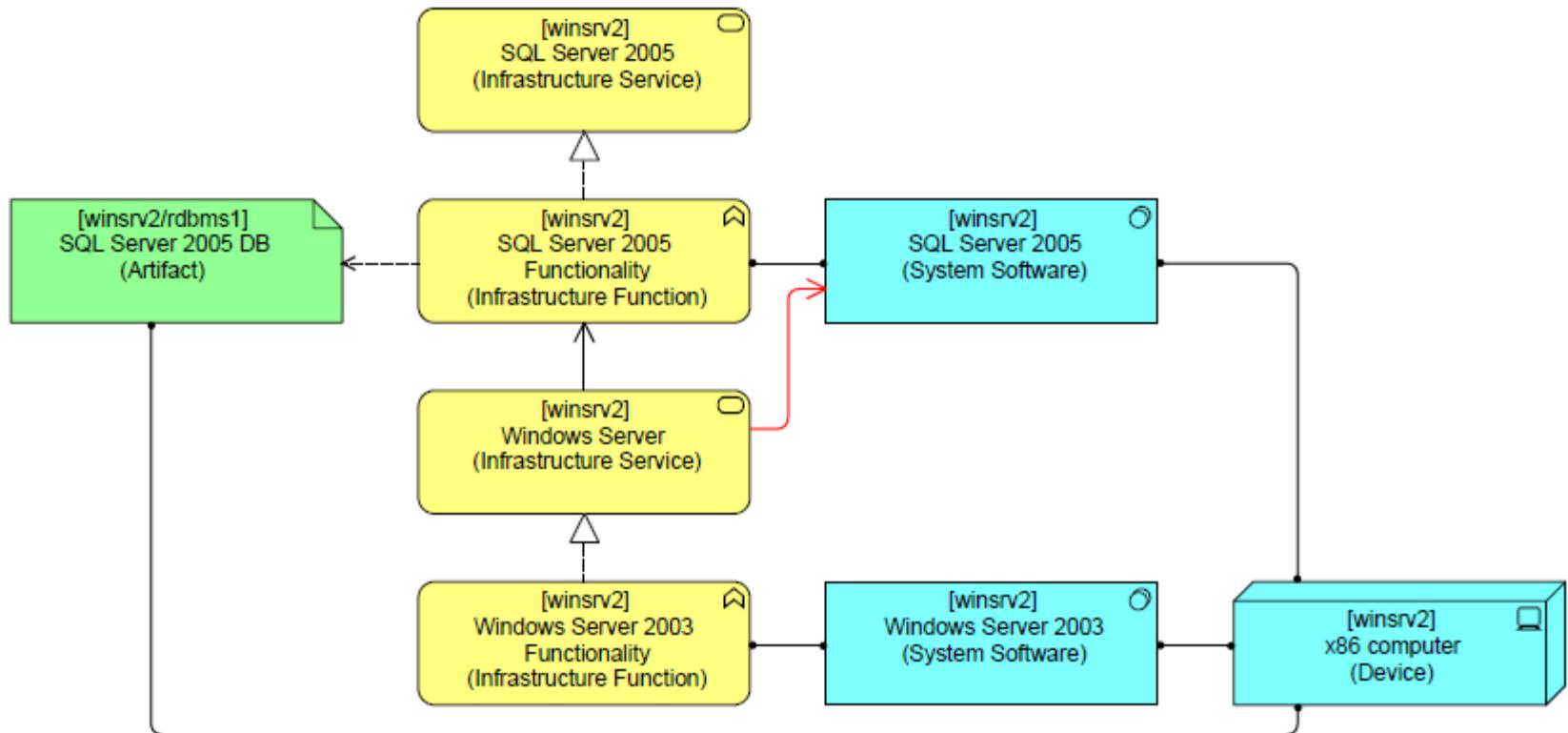


Example – Deploying a database...



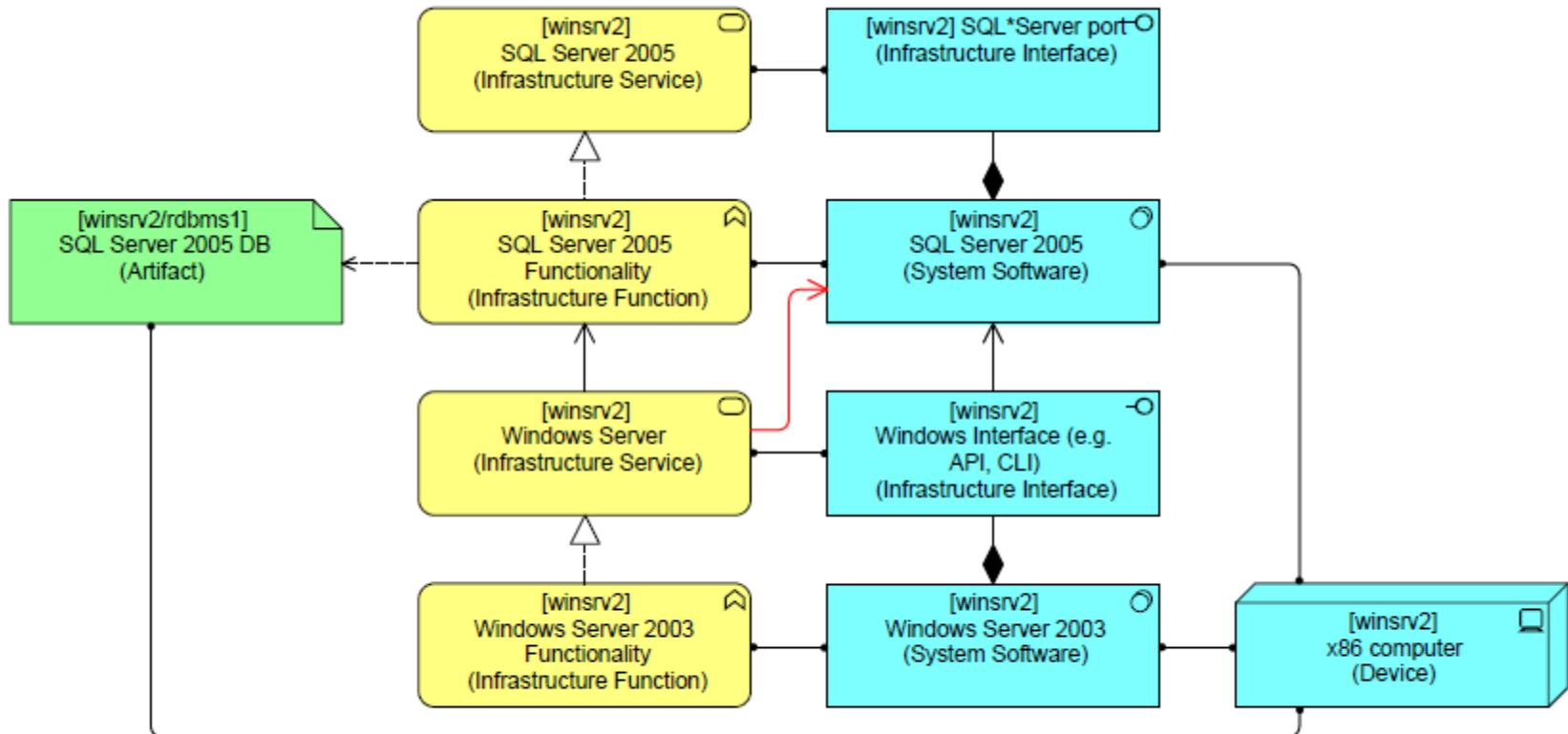
[View 63. Deploying a database, without hardware details, interfaces and internal behavior](#)

Example – Deploying a database...



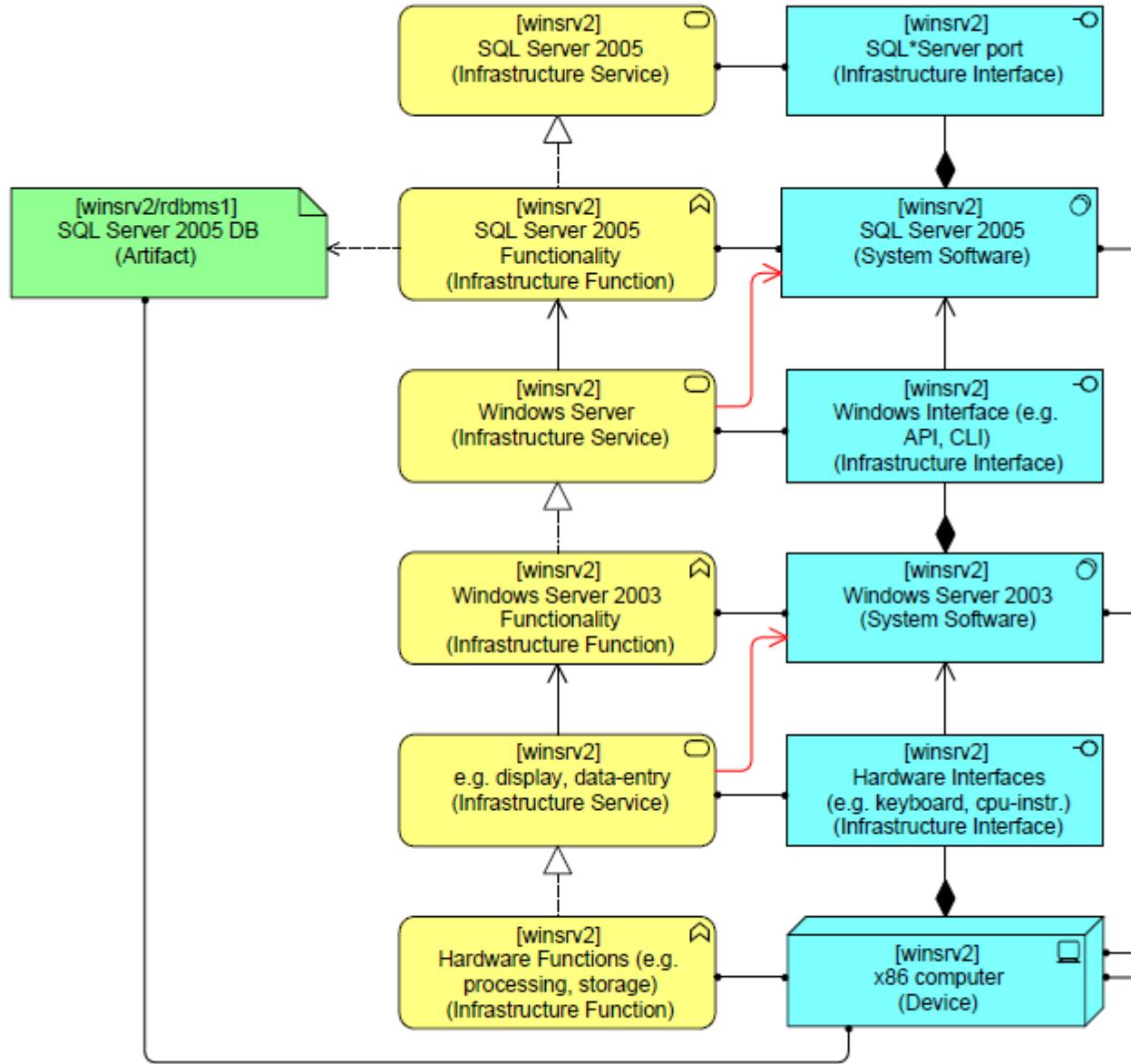
View 62. Deploying a database, without hardware details and interfaces

Example – Deploying a database...



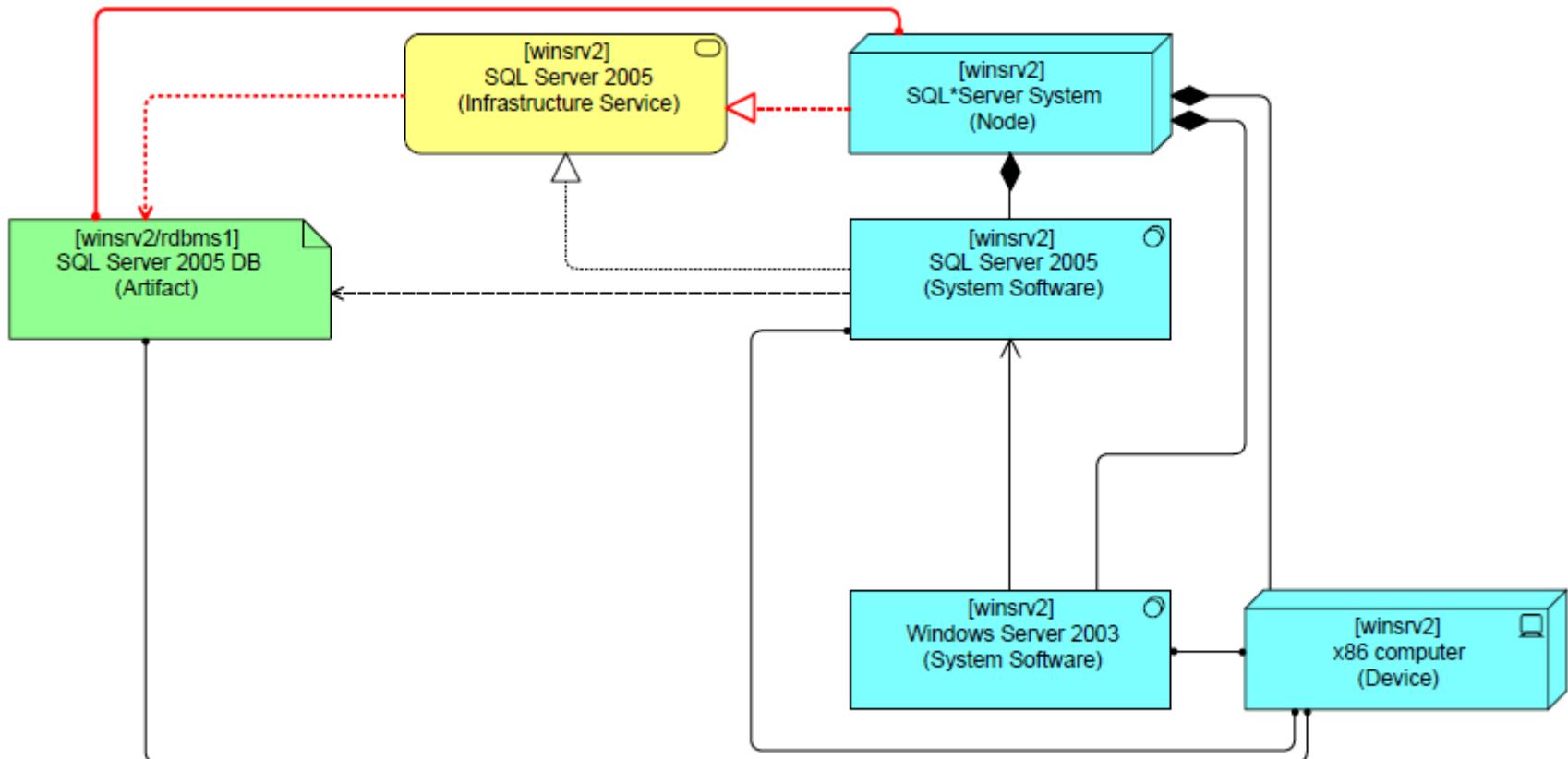
View 61. Deploying a database, without the hardware details

Example – Deploying a database...



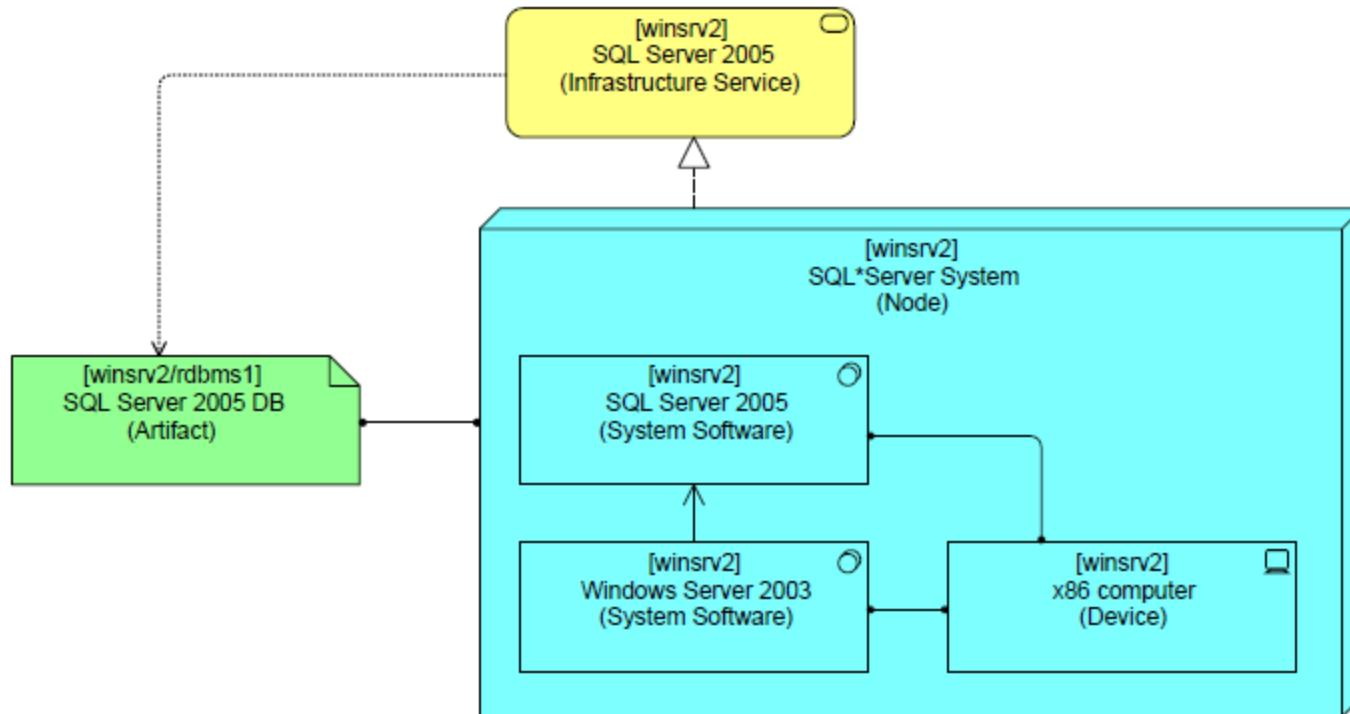
[View 60. Full details of deploying a database](#)

Example – Deploying a database...



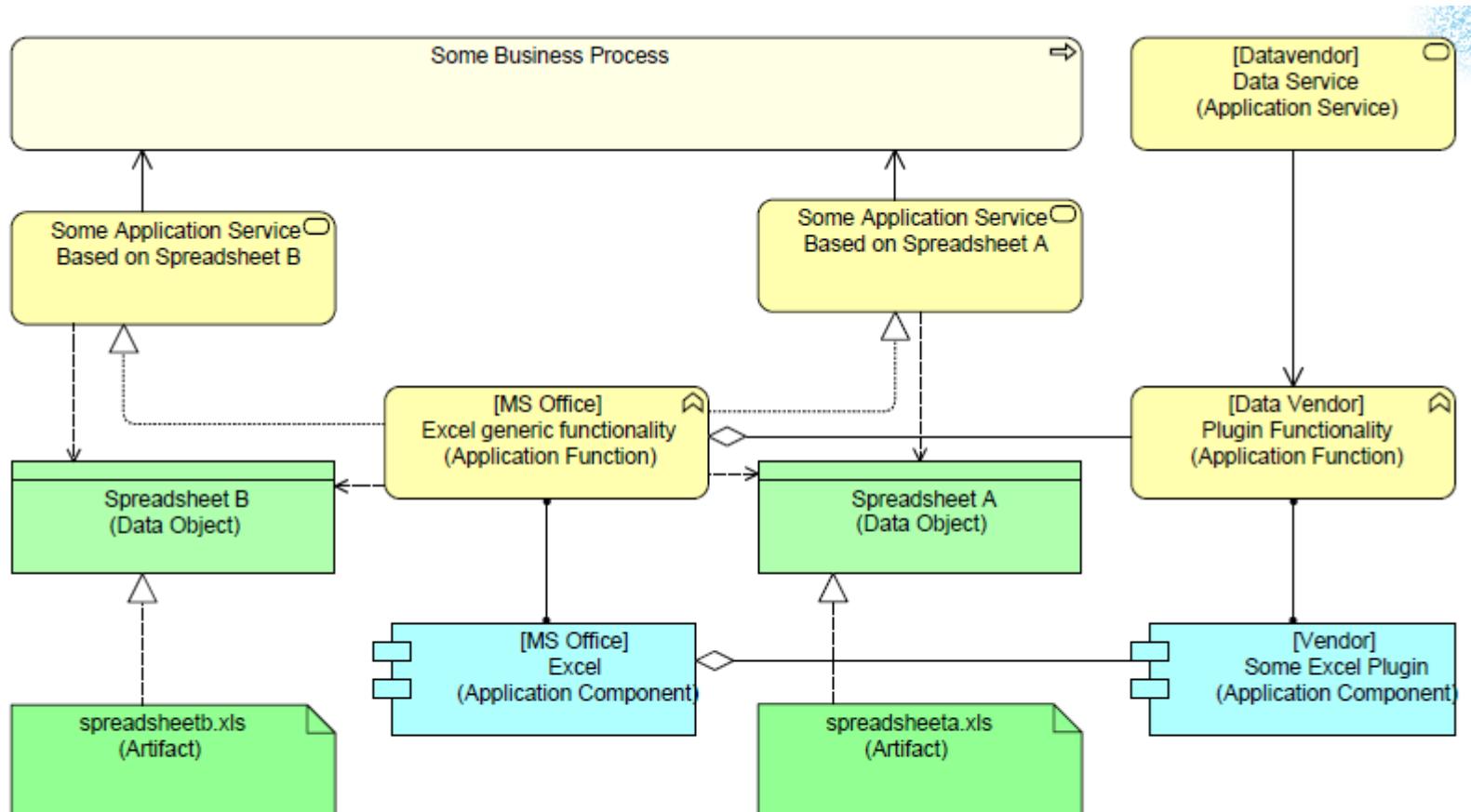
View 64. Deploying a database: using a Node to make an abstract device

Example – Deploying a database...

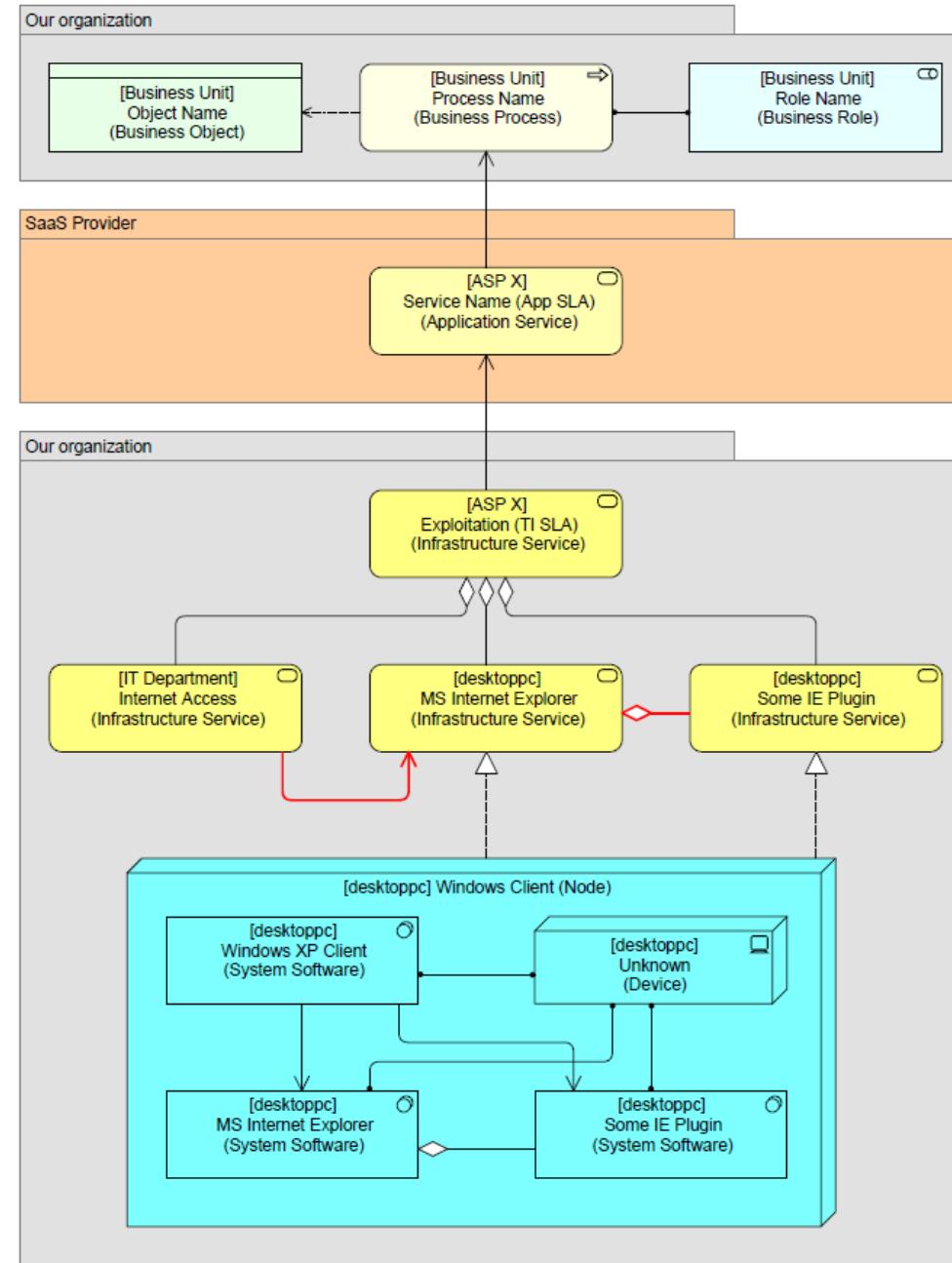


View 65. Deploying a database: Nesting details in a Node

Example - Excel as an Application Component...



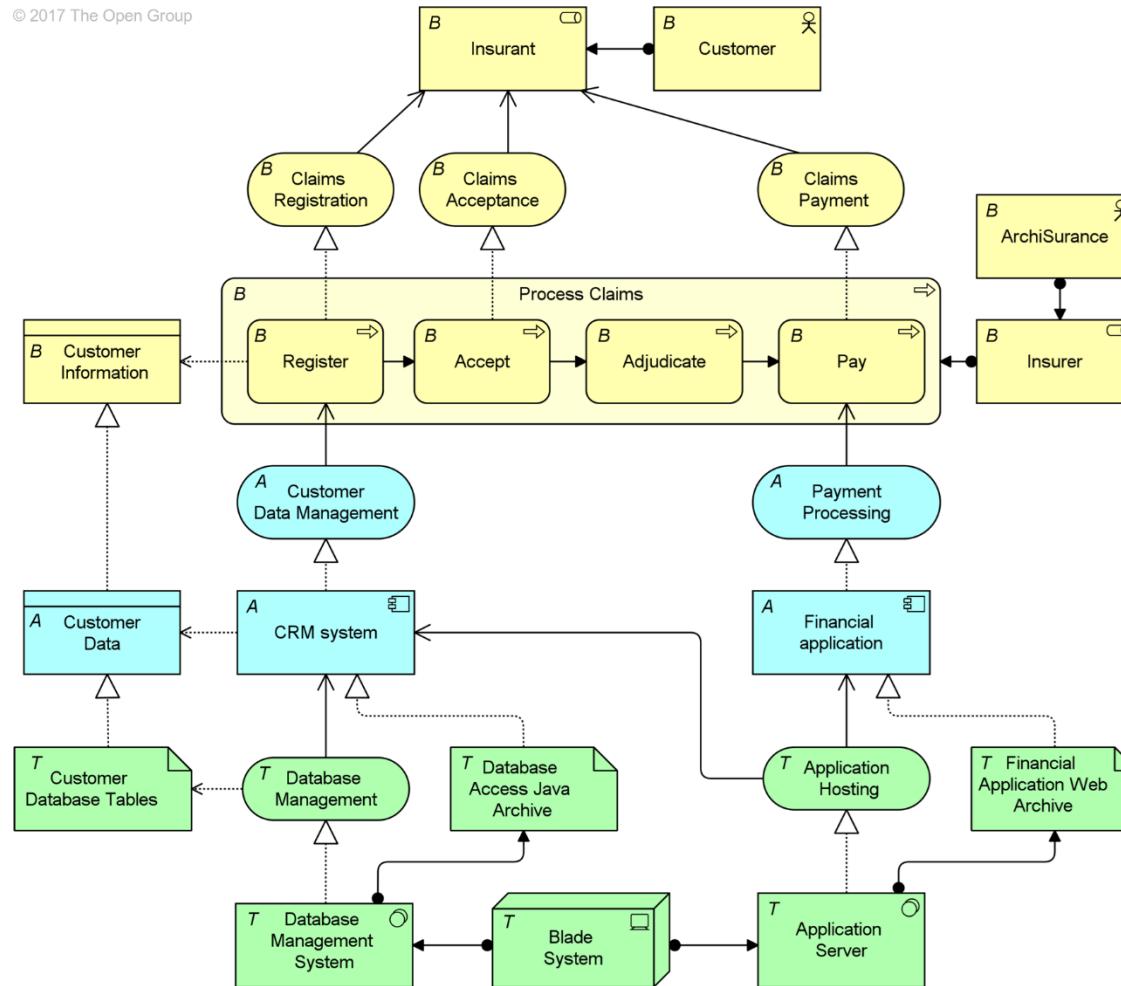
View 67. Excel as an Application Component



View 84. Deployment Pattern: SaaS Provider uses our Infrastructure

Example...

© 2017 The Open Group



Example 33: Cross-Layer Relationships