



# *Alignment of Business and IT - ArchiMate*

*Dr. Barbara Re*

# What is ArchiMate?

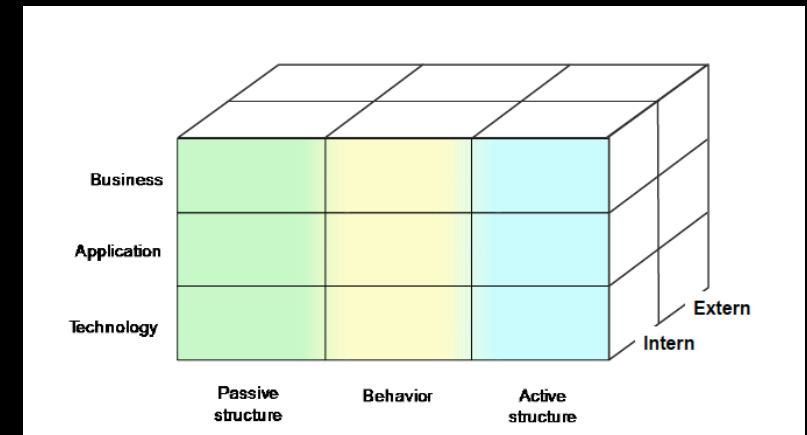
- ArchiMate is a modelling technique ("language") for describing enterprise architectures.
- It presents a clear set of concepts within and relationships between architecture domains, and offers a simple and uniform structure for describing the contents of these domains.
- ArchiMate distinguishes itself from other languages such as Unified Modeling Language (UML) and Business Process Modeling Notation (BPMN) by its well defined metamodel, and wider enterprise modelling scope.

# What is ArchiMate?

- ArchiMate offers a common language for describing the construction and operation of business processes, organizational structures, information flows, IT systems, and technical infrastructure.
- This insight helps the different stakeholders to design, assess, and communicate the consequences of decisions and changes within and between these business domains.

# Layers

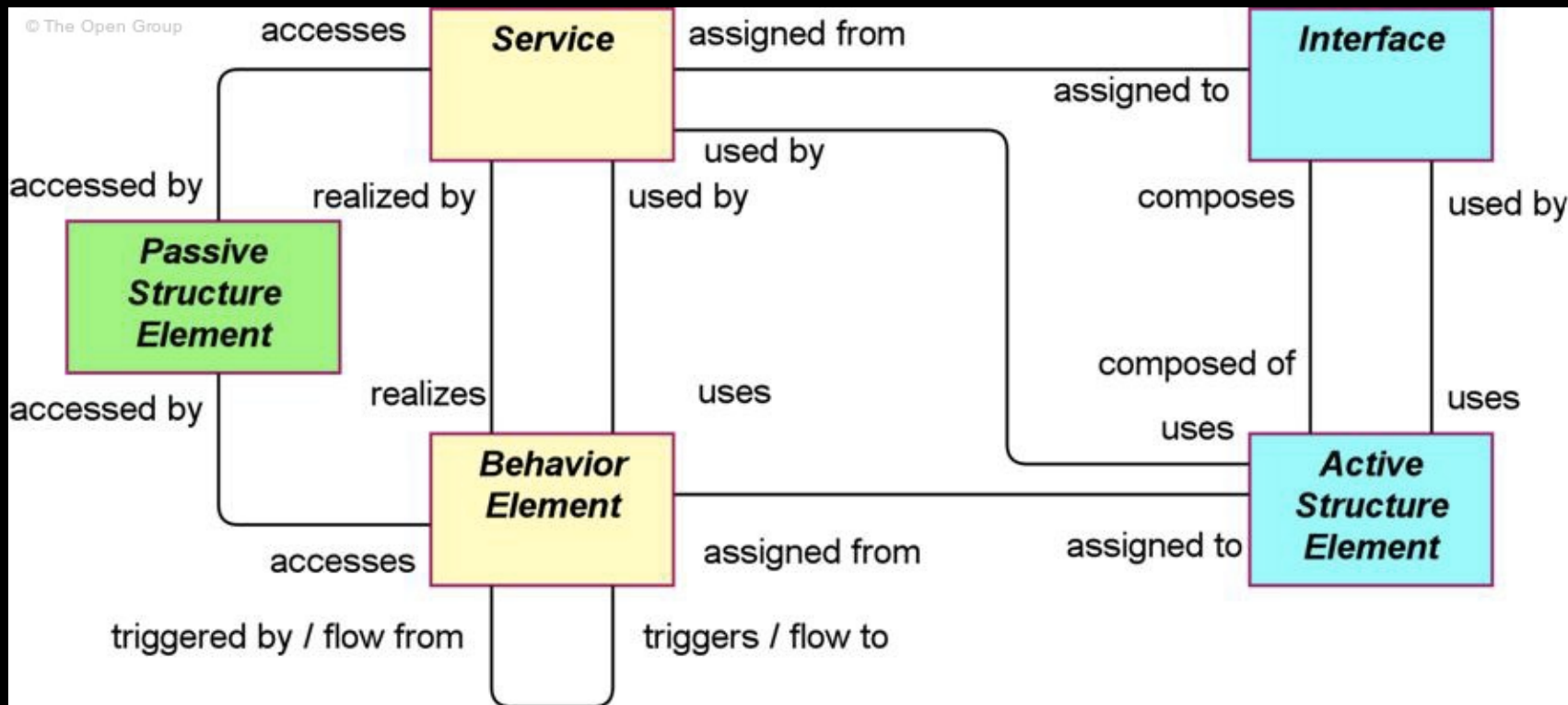
- A layered view provides a natural way to look at service-oriented models. The higher layers use services that are provided by the lower layers. ArchiMate distinguishes three main layers:
  - ◆ The **Business layer** offers products and services to external customers, which are realized in the organization by business processes performed by business actors and roles.
  - ◆ The **Application layer** supports the business layer with application services which are realized by (software) application components.
  - ◆ The **Technology layer** offers infrastructural services (e.g., processing, storage and communication services) needed to run applications, realized by computer and communication hardware and system software.
  
- The general structure of models within the different layers is similar. The same types of concepts and relationships are used, although their exact nature and granularity differ.



# Dimensions of ArchiMate

- Three main types of elements:
  - ◆ **Active structure** element: an entity that is capable of performing behavior.
  - ◆ **Behavior** element: a unit of activity performed by one or more active structure elements.
  - ◆ **Passive structure** element: an object on which behavior is performed.
  
- External vs. internal behavior and view on systems:
  - ◆ **Service**: externally visible **behavior** of a system. A service is a unit of **functionality** that a system exposes to its environment, while hiding internal operations
  - ◆ **Interface**: external **view** on service provider. An interface is **a point of access** where one or more services are made available to the environment

# Core Concepts and Relations



*Passive structure*

*Behavior*

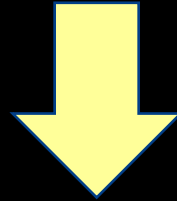
*Active structure*

*external*

*internal*

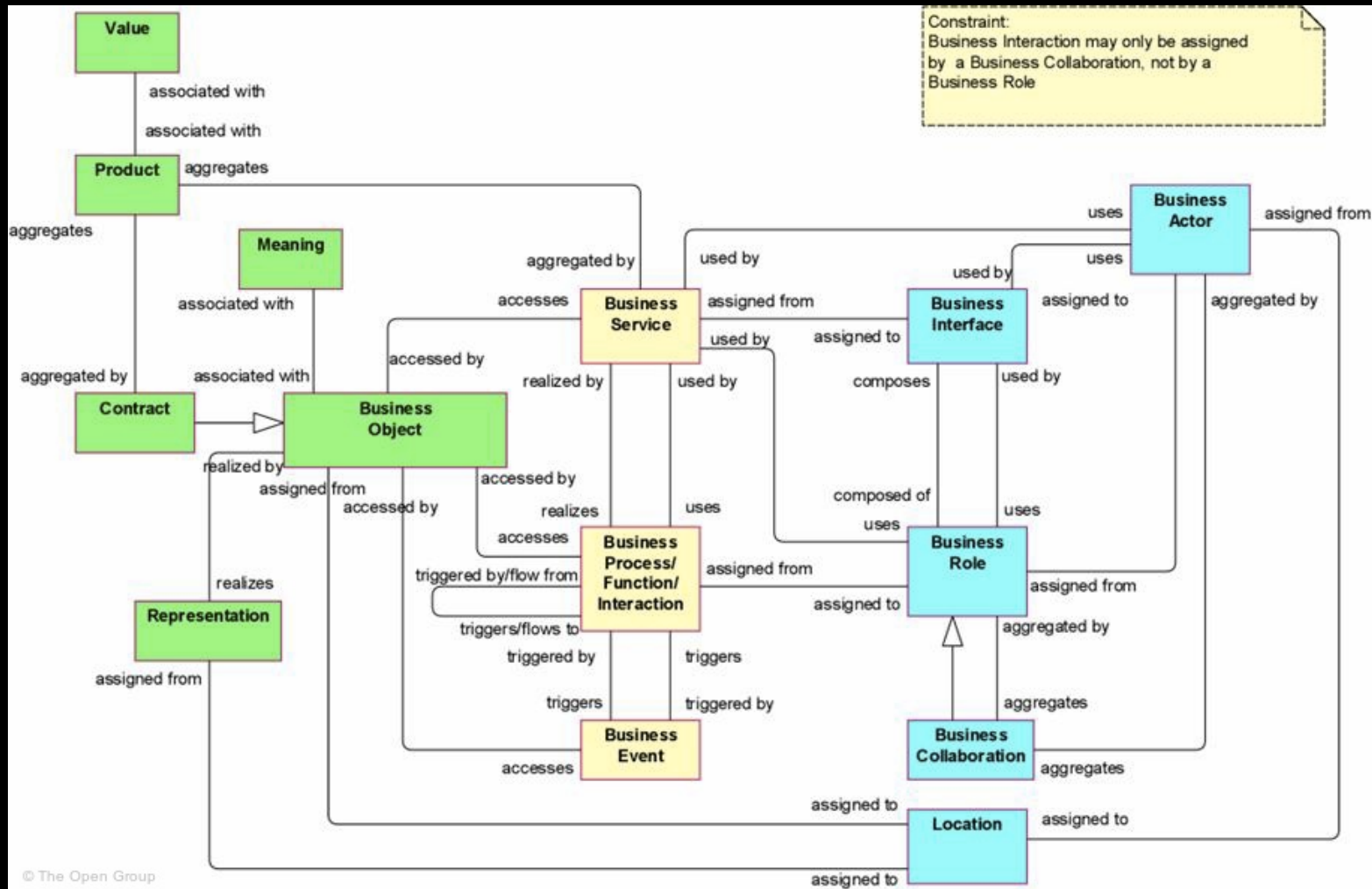
# Welcome to ArchiMate® 2.1, *an Open Group Standard*

- Description and Example are taken from



<http://pubs.opengroup.org/architecture/archimate2-doc/toc.html>

# Business Layer Meta-Model



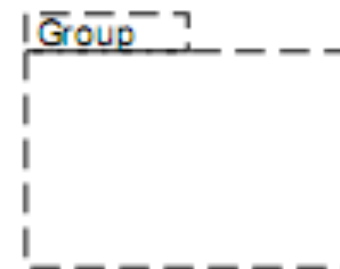
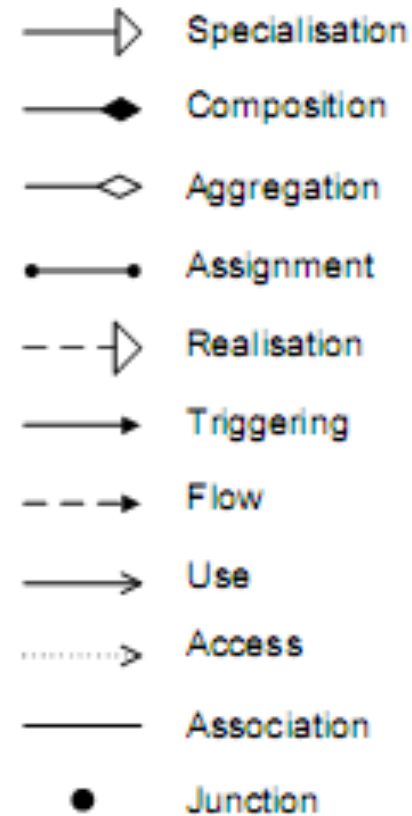


# Notations

- Every concept and relation should have a precise graphical notation, with a sufficient resemblance the 'standard' ArchiMate notation. The notation in the Visio stencils can be used as a guideline
- Optionally, multiple notations may exist for a single concept.
- It should be possible to denote composition, aggregation and assignment both with their 'line' notation and with nesting.

# Relations



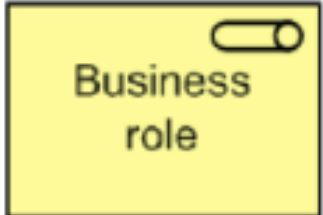

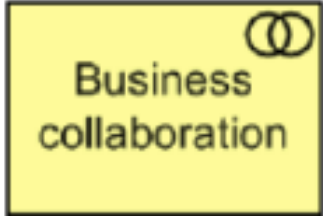
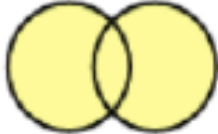

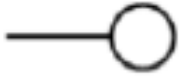


- The following relation types should be supported:
  - ◆ Structural relations:
    - composition\*
    - aggregation
    - assignment
    - used by
    - realisation
    - access
    - association
  - ◆ Dynamic relations:
    - triggering
    - flow
  - ◆ Other relations:
    - grouping
    - junction
    - specialisation\*



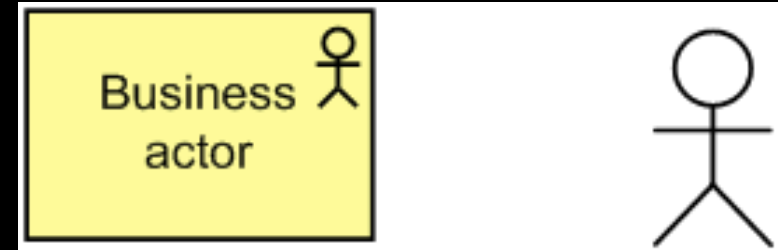
# Business Layer Active Structure Concepts

- The active structure aspect at 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 Layer Active Structure Concepts - Relevant Elements

Business actor	An organizational entity that is capable of performing behavior.		
Business role	The responsibility for performing specific behavior, to which an actor can be assigned.		
Business collaboration	An aggregate of two or more business roles that work together to perform collective behavior.		
Business interface	A point of access where a business service is made available to the environment.		
Location	A conceptual point or extent in space.		

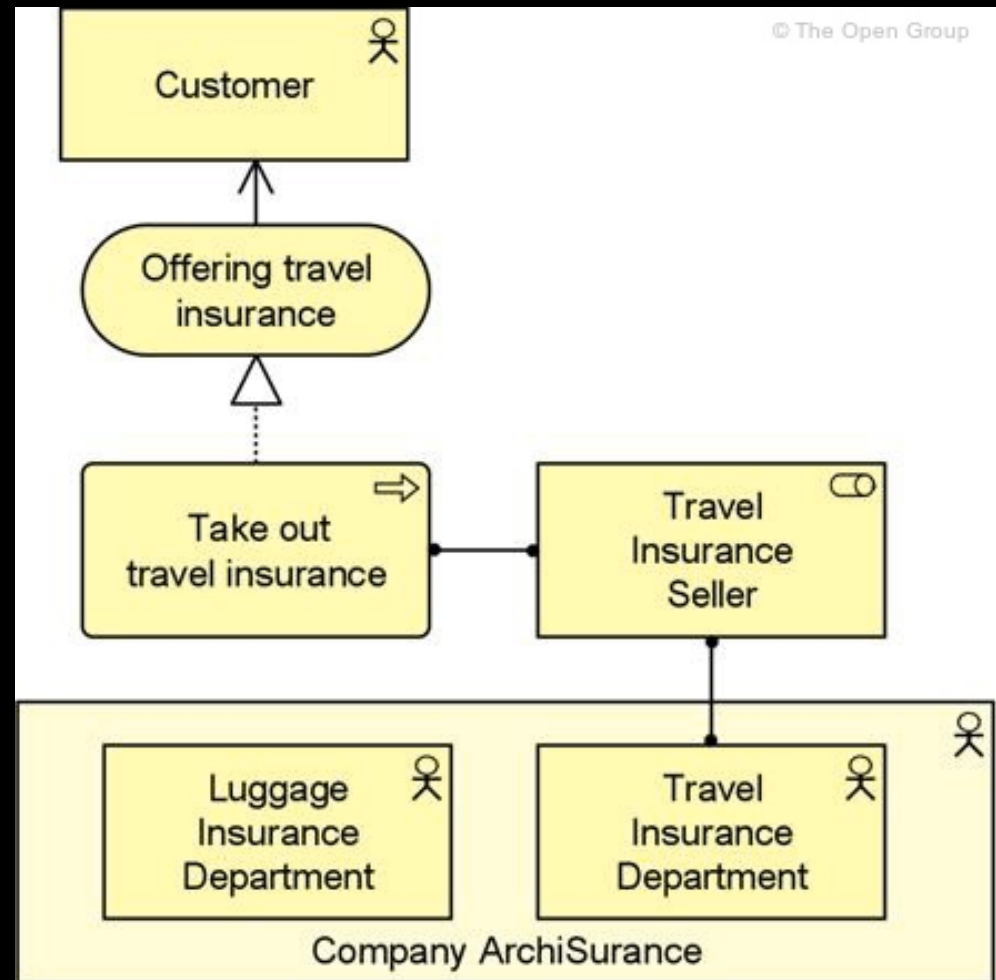
# Business Actor



- A business actor performs the behavior assigned to (one or more) business roles.
- A business actor is an organizational entity as opposed to a technical entity; i.e., it belongs to the business layer.
- Actors may, however, include entities outside the actual enterprise; e.g., customers and partners.
  - ◆ Examples of business actors are humans, departments, and business units.
- A business actor may be assigned to one or more business roles.
- The name of a business actor should preferably be a noun.

# Business Actor Example

- The company **ArchiSurance** is modeled as a business actor that is composed of two departments.
- The Travel insurance seller role is assigned to the travel department.
- In this role, the travel department performs the Take out insurance process, which offers a service that is accessible via the business interface assigned to this role.



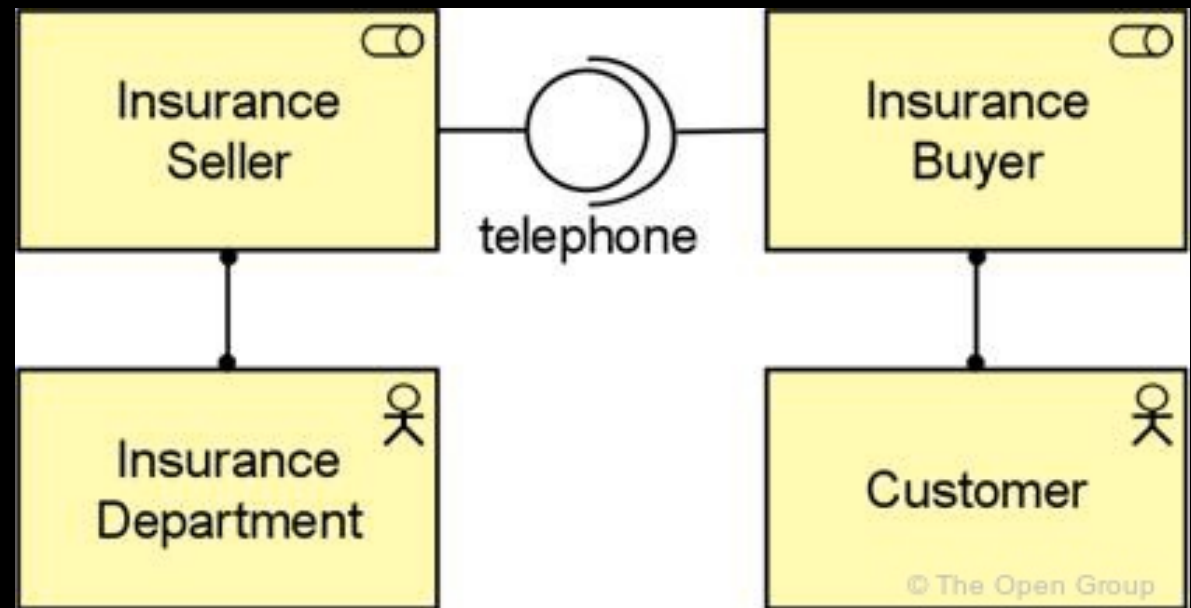
# Business Role



- A business role is defined as the responsibility for performing specific behavior, to which an actor can be assigned.
- Business processes or business functions are assigned to a single business role with certain responsibilities or skills. A business actor that is assigned to a business role ultimately performs the corresponding behavior.
- In addition to the relation of a business role with behavior, a business role is also useful in a (structural) organizational sense; for instance, in the division of labor within an organization.

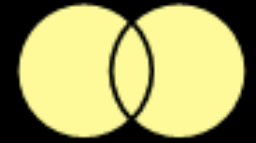
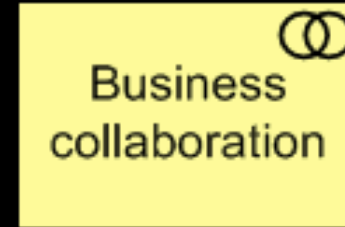
## Business Role Example

- In the model below, the business role Insurance Seller is fulfilled by the Insurance Department actor and has telephone as a provided interface. The business role Insurance Buyer is fulfilled by the Customer actor, and has telephone as a required interface.



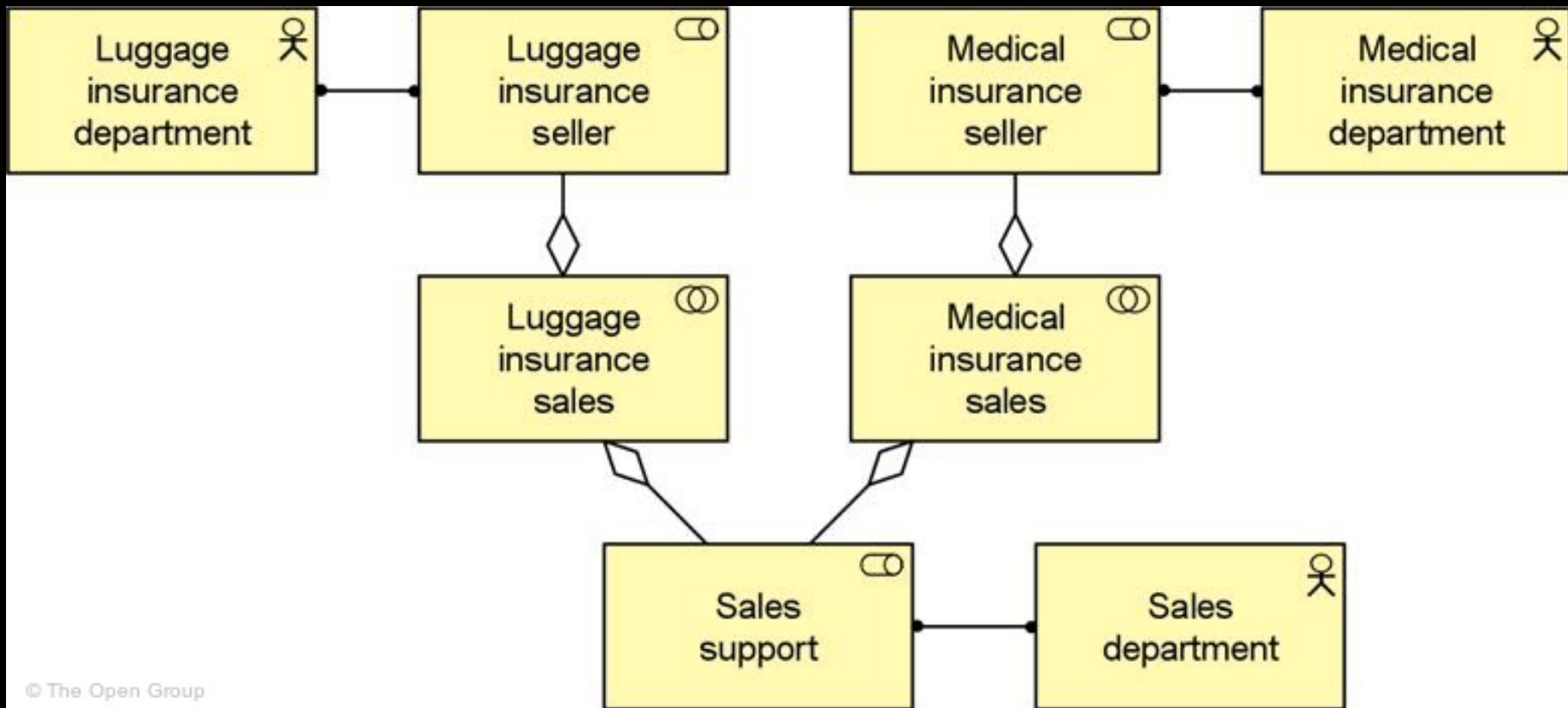


# Business Collaboration

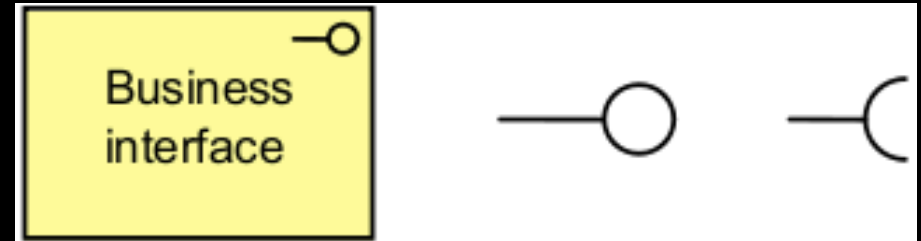


- Business collaboration is defined as an aggregate of two or more business roles that work together to perform collective behavior.
- Unlike a department, which may also group roles, a business collaboration does not have an official (permanent) status within the organization; it is specifically aimed at a specific interaction or set of interactions between roles. However, a business collaboration can be regarded as a kind of “virtual role”, hence its designation as a specialization of role. It is especially useful in modeling B2B interactions between different organizations.

# Business Collaboration Example



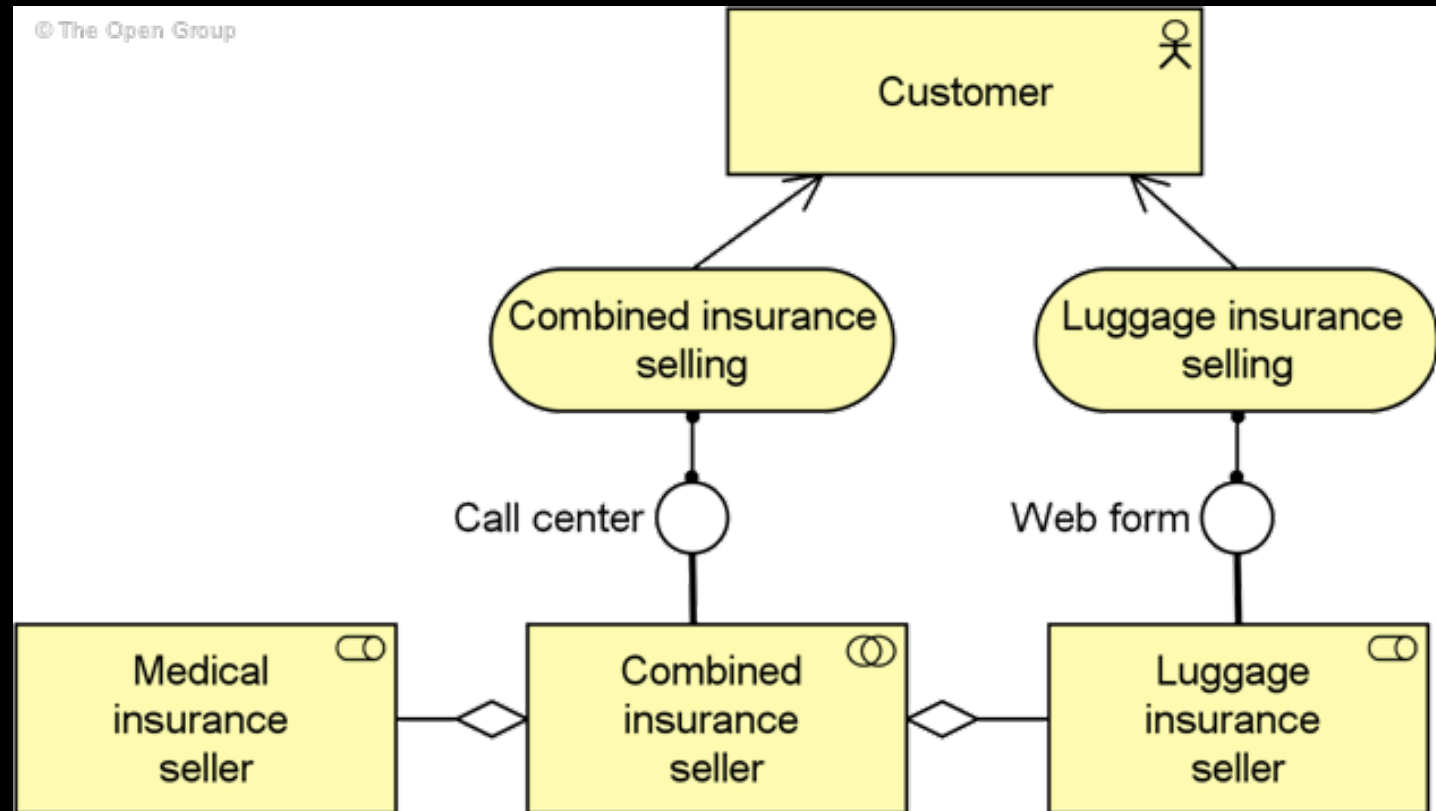
# Business Interface



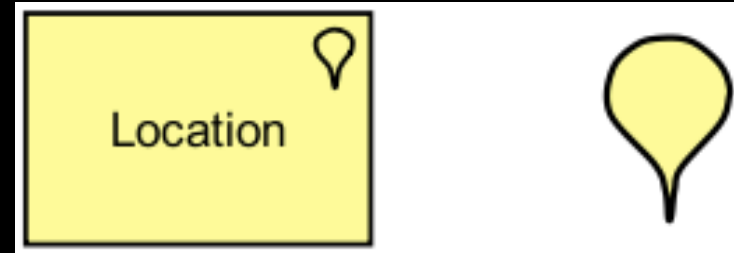
- A business interface is defined as a point of access where a business service is made available to the environment.
- A business interface exposes the functionality of a business service to other business roles (provided interface), or expects functionality from other business services (required interface). It is often referred to as a channel (telephone, internet, local office, etc.). The same business service may be exposed through different interfaces.

# Business Interface Example

- In the model below, the business services provided by the Luggage insurance seller and its collaboration with the Medical insurance seller are exposed by means of a web form and call center business interface, respectively.



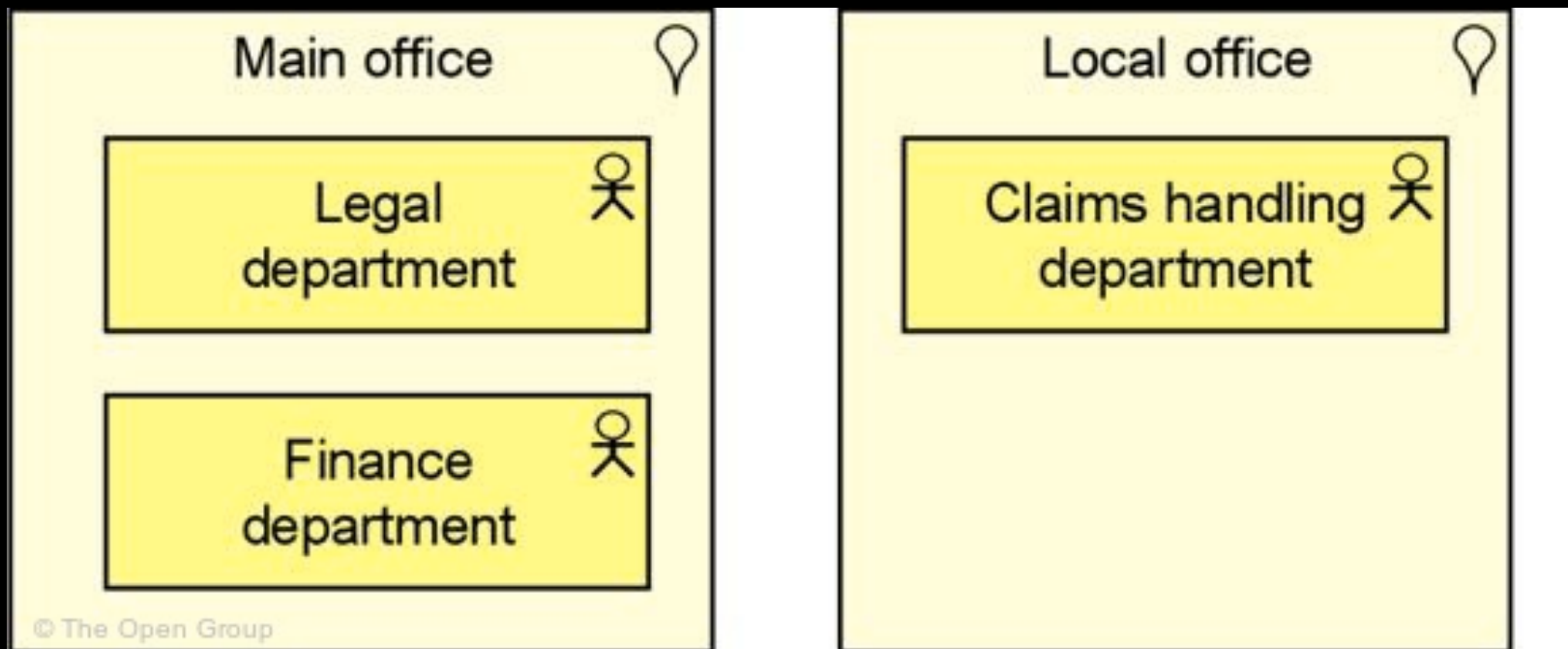
# Location



- A location is defined as a conceptual point or extent in space.
- The location concept is used to model the distribution of structural elements such as business actors, application components, and devices. This is modeled by means of an assignment relationship from location to structural element. Indirectly, a location can also be assigned to a behavior element, to indicate where the behavior is performed.

# Location Example


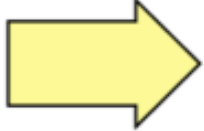








- The model below shows that the departments of an insurance company are distributed over different locations. The Legal and Finance departments are centralized at the main office, and there are claims handling departments at various local offices throughout the country.



# Business Layer Behavioral Concepts

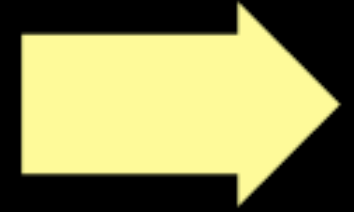
- Based on service orientation, a crucial design decision for the behavioral part of our metamodel is the distinction between “internal” and “external” behavior of an organization.
- Several types of internal behavior elements that can realize a service are distinguished.
  - ◆ Although the distinction between the two is not always sharp, it is often useful to distinguish a *process view* and a *function view* on behavior; two concepts associated with these views, *business process* and *business function*, are defined.
- The externally visible behavior is modeled by the concept *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.
- A *business interaction* is a unit of behavior similar to a business process or function, but which is performed in a collaboration of two or more roles within the organization.
- A *business event* is something that happens (externally) and may influence business processes, functions, or interactions.

# Business Layer Behavioral Concepts - Relevant Elements

Business process	A behavior element that groups behavior based on an ordering of activities. It is intended to produce a defined set of products or business services.		
Business function	A behavior element that groups behavior based on a chosen set of criteria (typically required business resources and/or competences).		
Business interaction	A behavior element that describes the behavior of a business collaboration.		
Business event	Something that happens (internally or externally) and influences behavior.		
Business service	A service that fulfills a business need for a customer (internal or external to the organization).		



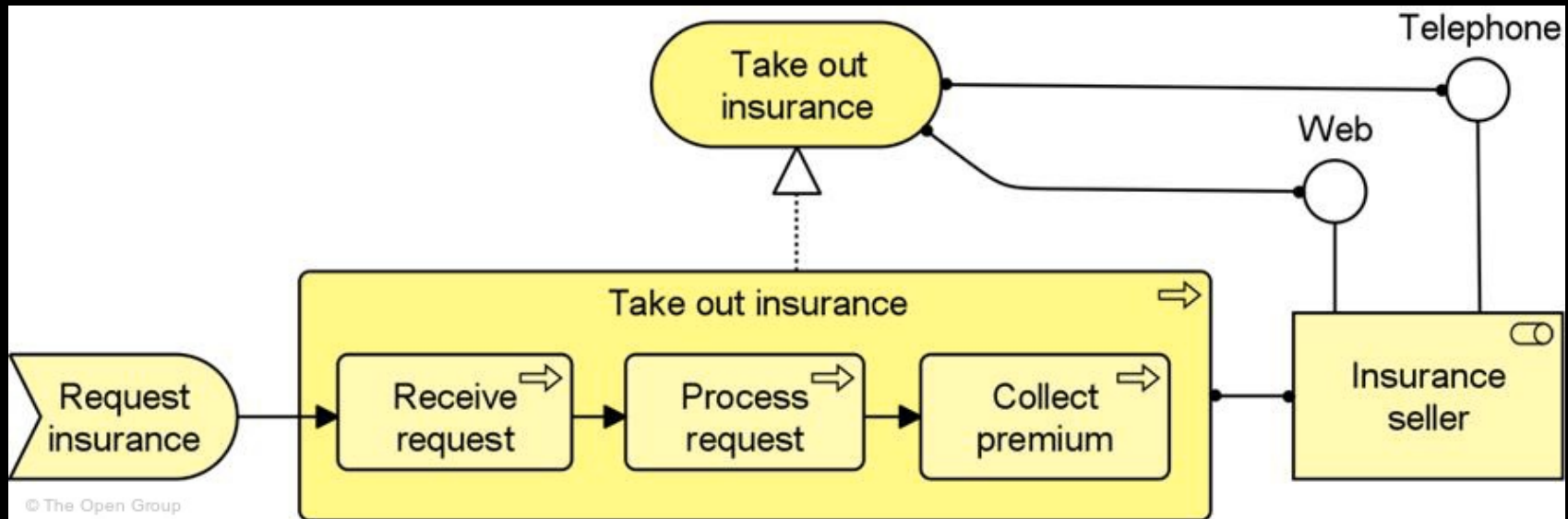
# Business Process



- A business process is defined as a behavior element that groups behavior based on an ordering of activities. It is intended to produce a defined set of products or business services.
- A business process describes the internal behavior performed by a business role that is required to produce a set of products and services. For a consumer the products and services are relevant and the required behavior is merely a black box, hence the designation “internal”.
- There is a potential many-to-many relationship between business processes and business functions. Informally speaking, processes describe some kind of “flow” of activities, whereas functions group activities according to required skills, knowledge, resources, etc.
- In an ArchiMate model, the existence of business processes is depicted. It does not, however, list the flow of activities in detail. During business process modeling, a business process can be expanded using a business process design language; e.g., BPMN .

# Business Process Example

The Take out insurance process is composed of three sub-processes. For clarity, the sub-processes are drawn in the overall process (structuring). Each sub-process triggers the next sub-process. The event Request for Insurance triggers the first sub-process. A particular role, in this case an insurance seller, is assigned to perform the required work. The process itself realizes an Insurance selling service. The Receive request sub-process is triggered by request insurance.

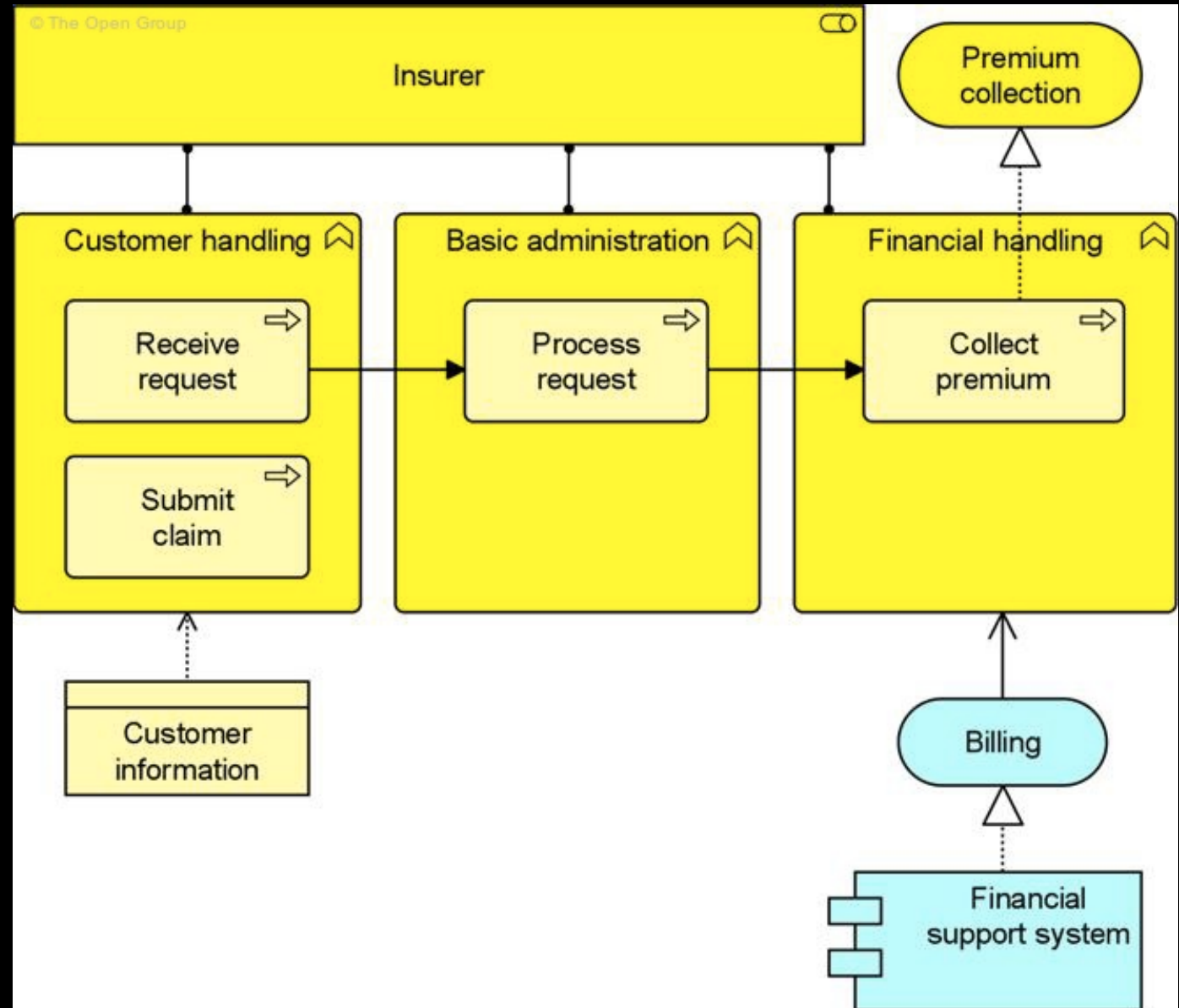


# Business Function



- A business function is defined as a behavior element that groups behavior based on a chosen set of criteria (typically required business resources and/or competences).
- Just like a business process, a business function also describes internal behavior performed by a business role. However, while a business process group's behavior is based on a sequence or "flow" of activities that is needed to realize a product or service, a business function typically groups behavior based on required business resources, skills, competences, knowledge, etc.
- There is a potential many-to-many relation between business processes and business functions.
- Complex processes in general involve activities that offer various functions. In this sense a business process forms a string of business functions.
- In general, a business function delivers added value from a business point of view. Organizational units or applications may coincide with business functions due to their specific grouping of business activities.

# Business Function Example

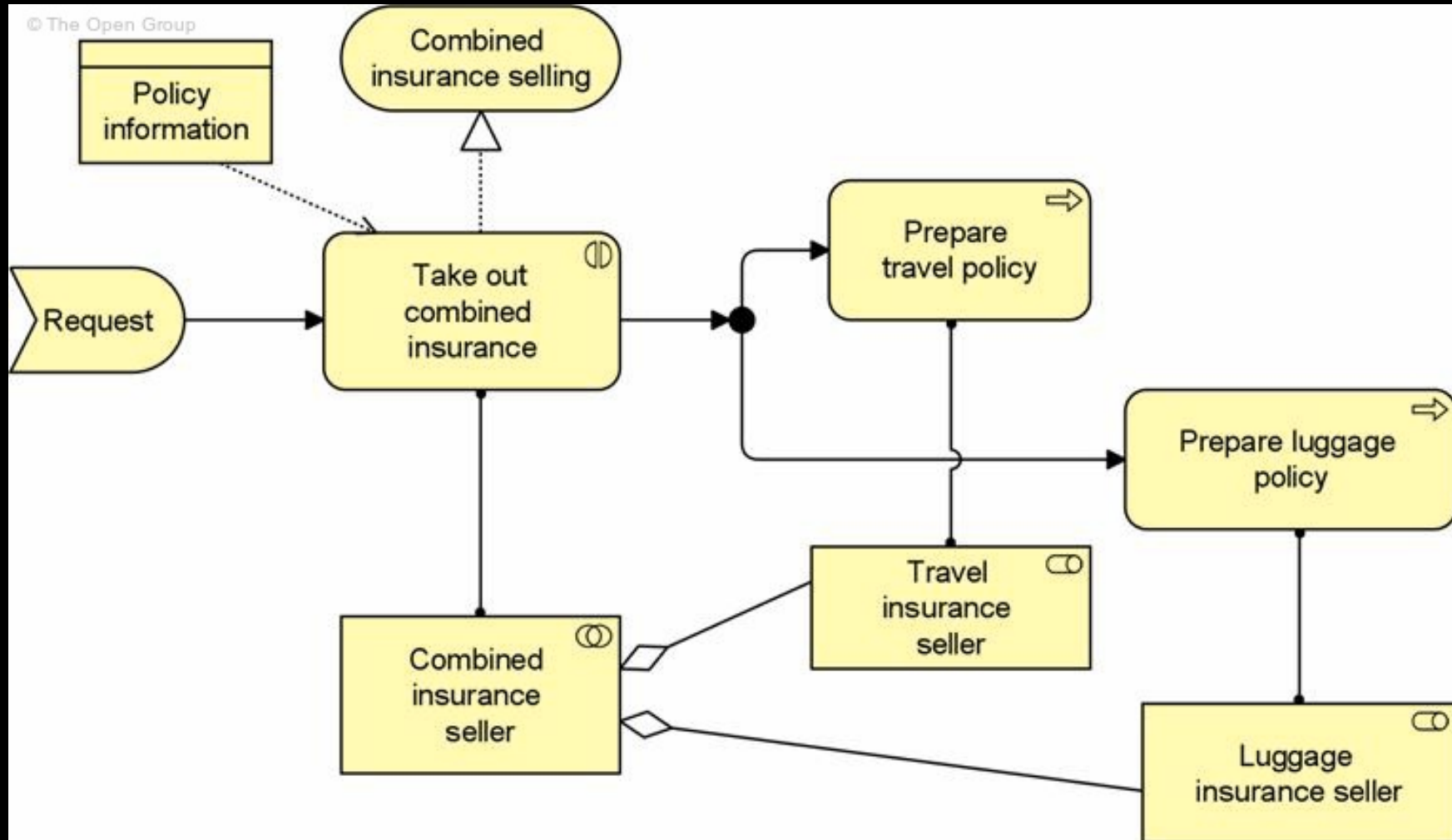


# Business Interaction



- A business interaction is defined as a behavior element that describes the behavior of a business collaboration.
- A business interaction is similar to a business process/function, but while a process/function may be performed by a single role, an interaction is performed by a collaboration of multiple roles. The roles in the collaboration share the responsibility for performing the interaction.
- A business interaction may be triggered by, or trigger, any other business behavior element (business event, business process, business function, or business interaction).
- A business interaction may access business objects.
- The name of a business interaction should preferably be a verb in the simple present tense.

# Business Interaction

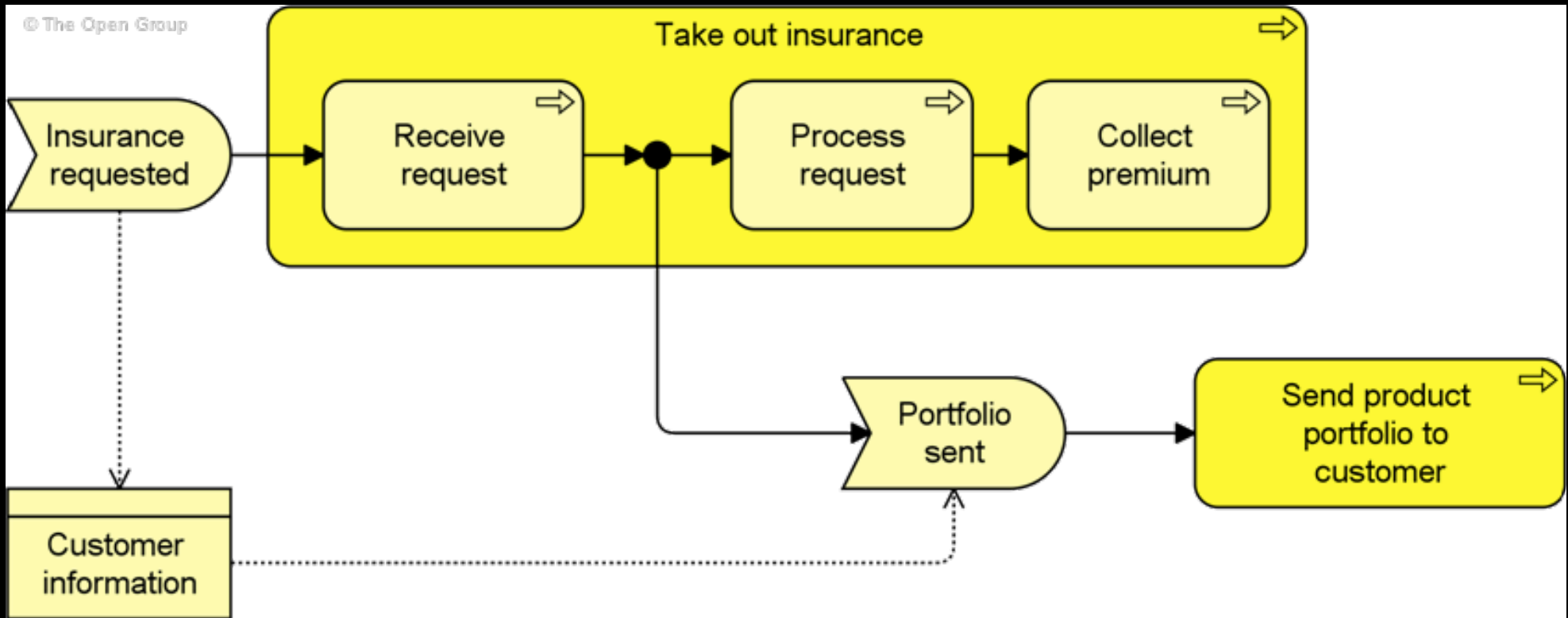


# Business Events



- A business event is defined as something that happens (internally or externally) and influences behavior.
- A business event is instantaneous: it does not have duration
- Events may originate from the environment of the organization (e.g., from a customer), but also internal events may occur generated by, for example, other processes within the organization.
- The name of a business event should preferably be a verb in the perfect tense; e.g., “claim received”.

# Business Events Example



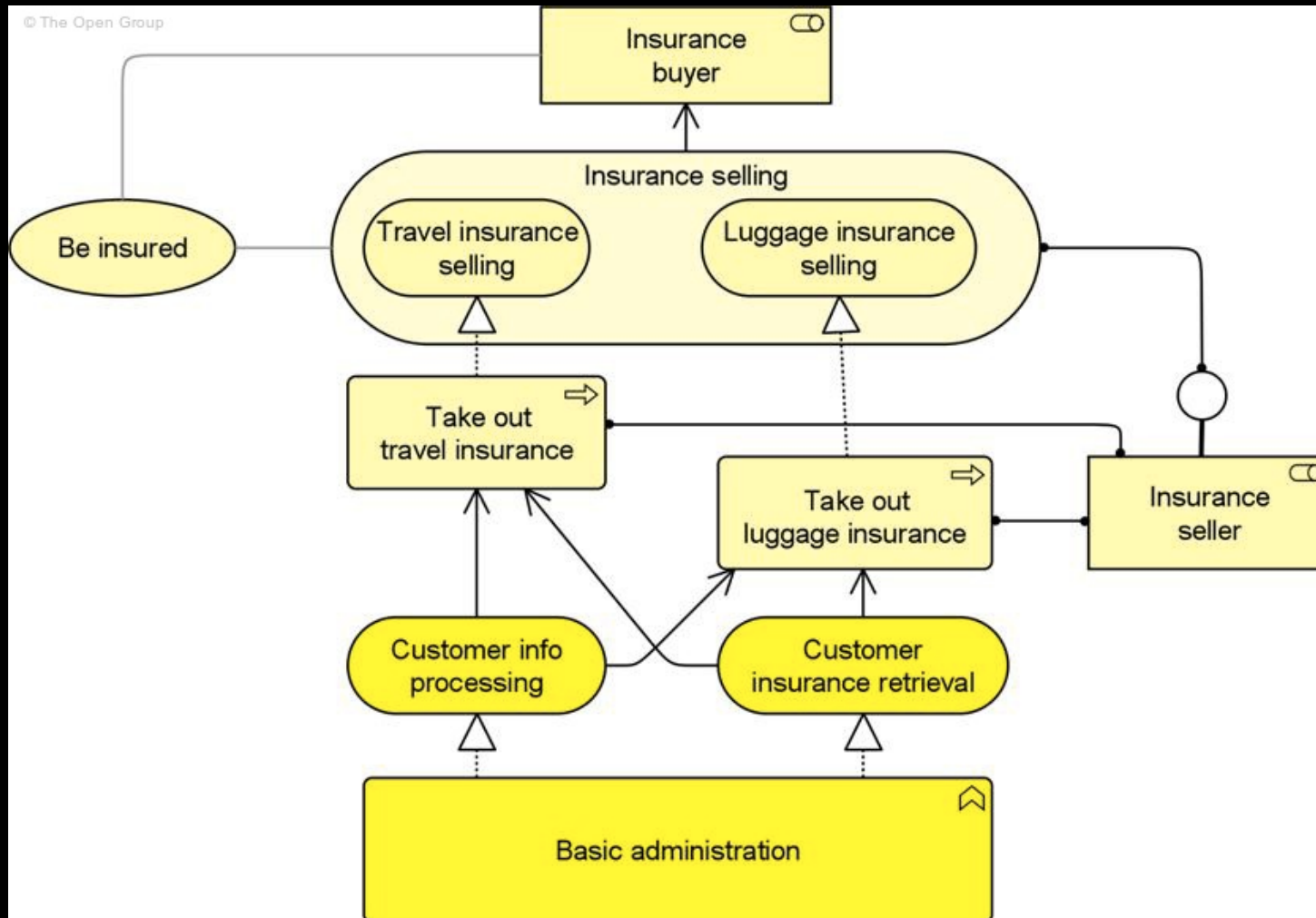


# Business Service



- A business service is defined as a service that fulfills a business need for a customer (internal or external to the organization). A business service should provide a unit of functionality that is meaningful from the point of view of the environment.
- A business service exposes the functionality of business roles or collaborations to their environment. This functionality is accessed through one or more business interfaces.
- A business service is realized by one or more business processes, business functions, or business interactions that are performed by the business roles or business collaborations, respectively.
- It may access business objects.
- The name of a business service should preferably be a verb ending with “-ing”; e.g., “transaction processing”. Also, a name explicitly containing the word “service” may be used.

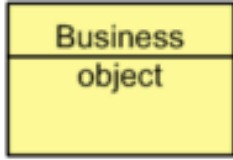


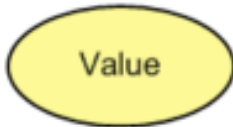
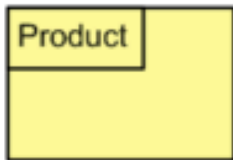
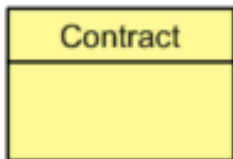
# Business Service Example



# Business Layer Passive Structure Concepts

- In the passive structure aspect at the business layer, we model the passive entities (business objects) that are manipulated by behavior
- The passive entities represent the important concepts in which the business thinks about a domain.

# Business Layer Passive Structure Concepts - Relevant Elements

Business object	A passive element that has relevance from a business perspective.	
Representation	A perceptible form of the information carried by a business object.	
Meaning	The knowledge or expertise present in a business object or its representation, given a particular context.	
Value	The relative worth, utility, or importance of a business service or product.	
Product	A coherent collection of services, accompanied by a contract/set of agreements, which is offered as a whole to (internal or external) customers.	
Contract	A formal or informal specification of agreement that specifies the rights and obligations associated with a product.	

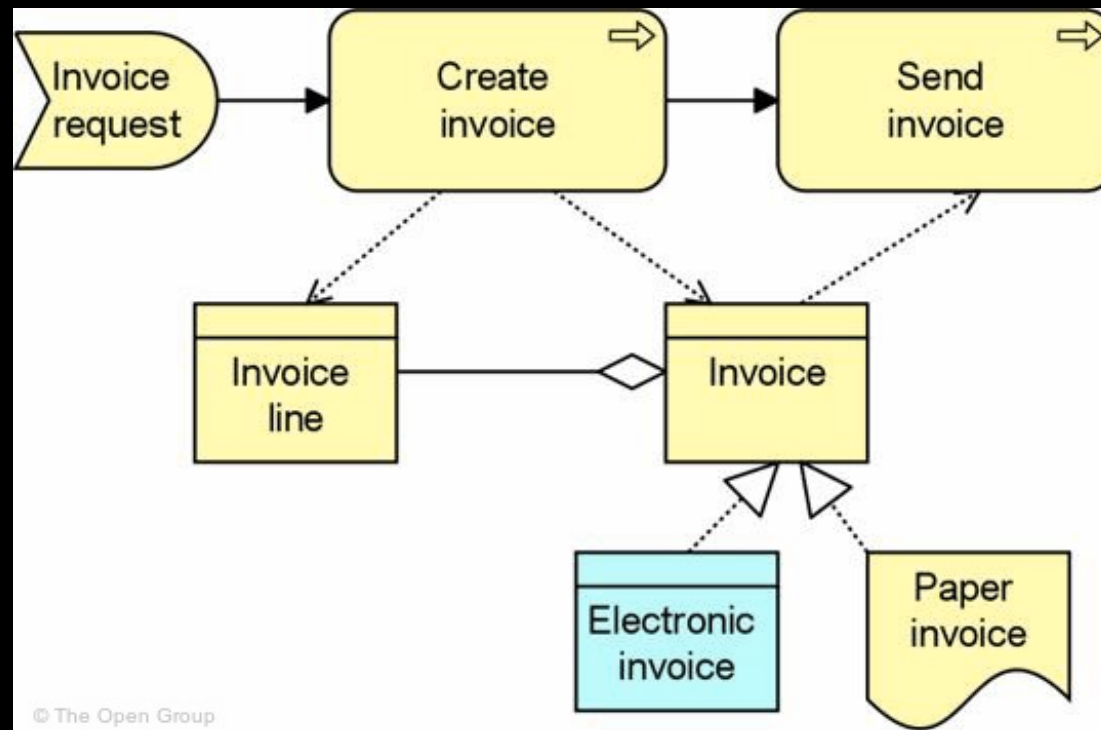
# Business Object

Business
object

- **A business object is defined as a passive element that has relevance from a business perspective.**
- At the enterprise architecture abstraction level, it is more common to model types rather than instances.
- Business objects may be accessed by a business process, function, business interaction, business event, or business service. A business object may have association, specialization, aggregation, or composition relationships with other business objects.
- The name of a business object should preferably be a noun.

# Business Object Example

The model shows a business object Invoice, which aggregates (multiple) business objects Invoice line. Two possible realizations of this business object exist: an Electronic invoice (data object) and a Paper invoice (representation). The business process Create invoice creates the invoice and the invoice lines, while the business process Send invoice accesses the business object Invoice.

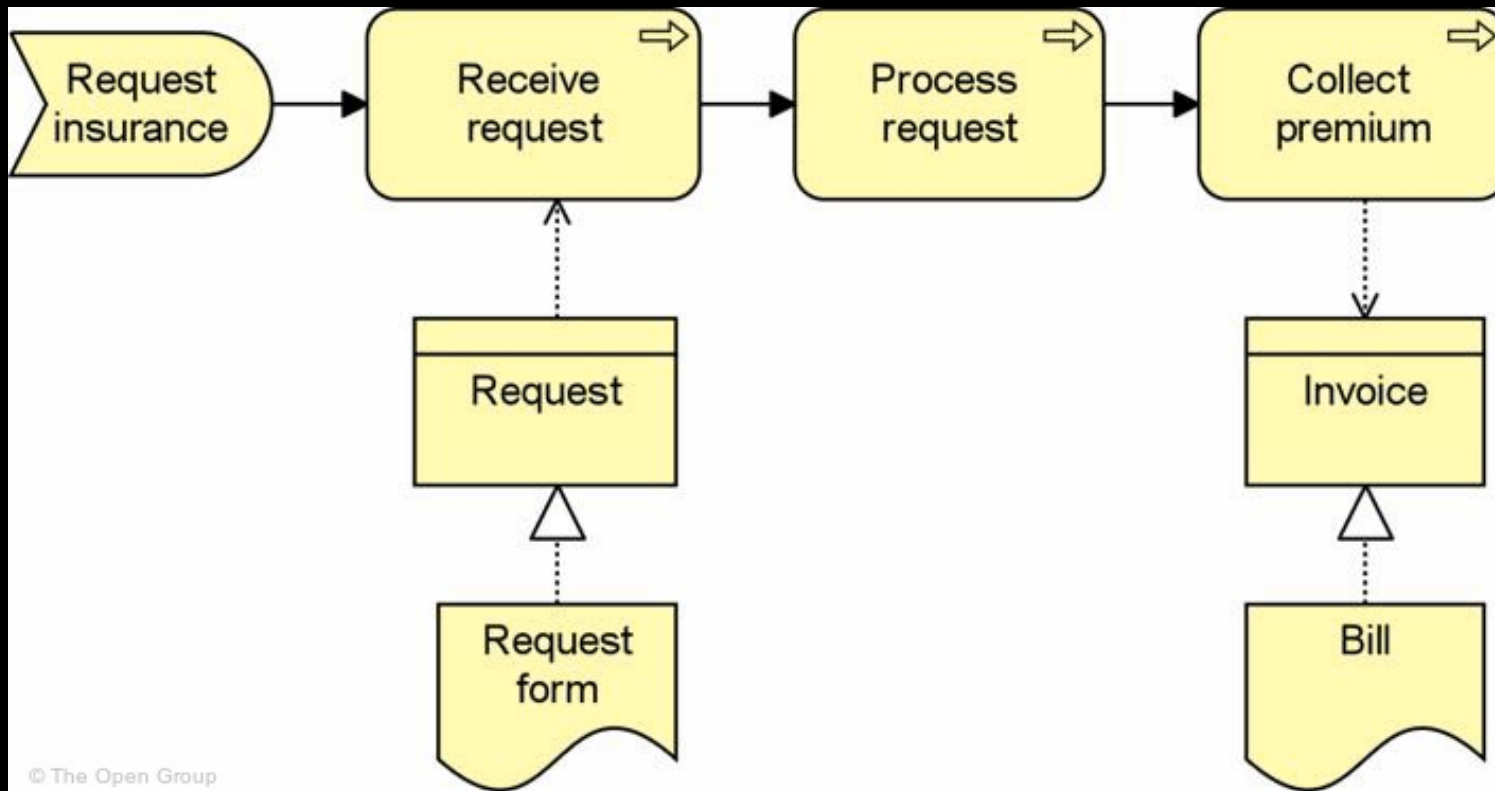


# Representation

- A representation is defined as a perceptible form of the information carried by a business object.
- Representations (for example, messages or documents) are the perceptible carriers of information that are related to business objects.
- A single business object can have a number of different representations. Also, a single representation can realize one or more specific business objects.
- A representation may realize one or more business objects.
- The name of a representation is preferably a noun.

# Representation Example

- The model below shows the business object Request for insurance, which is realized (represented) by a (physical) request form. The Invoice business object is realized (represented) by a paper bill.





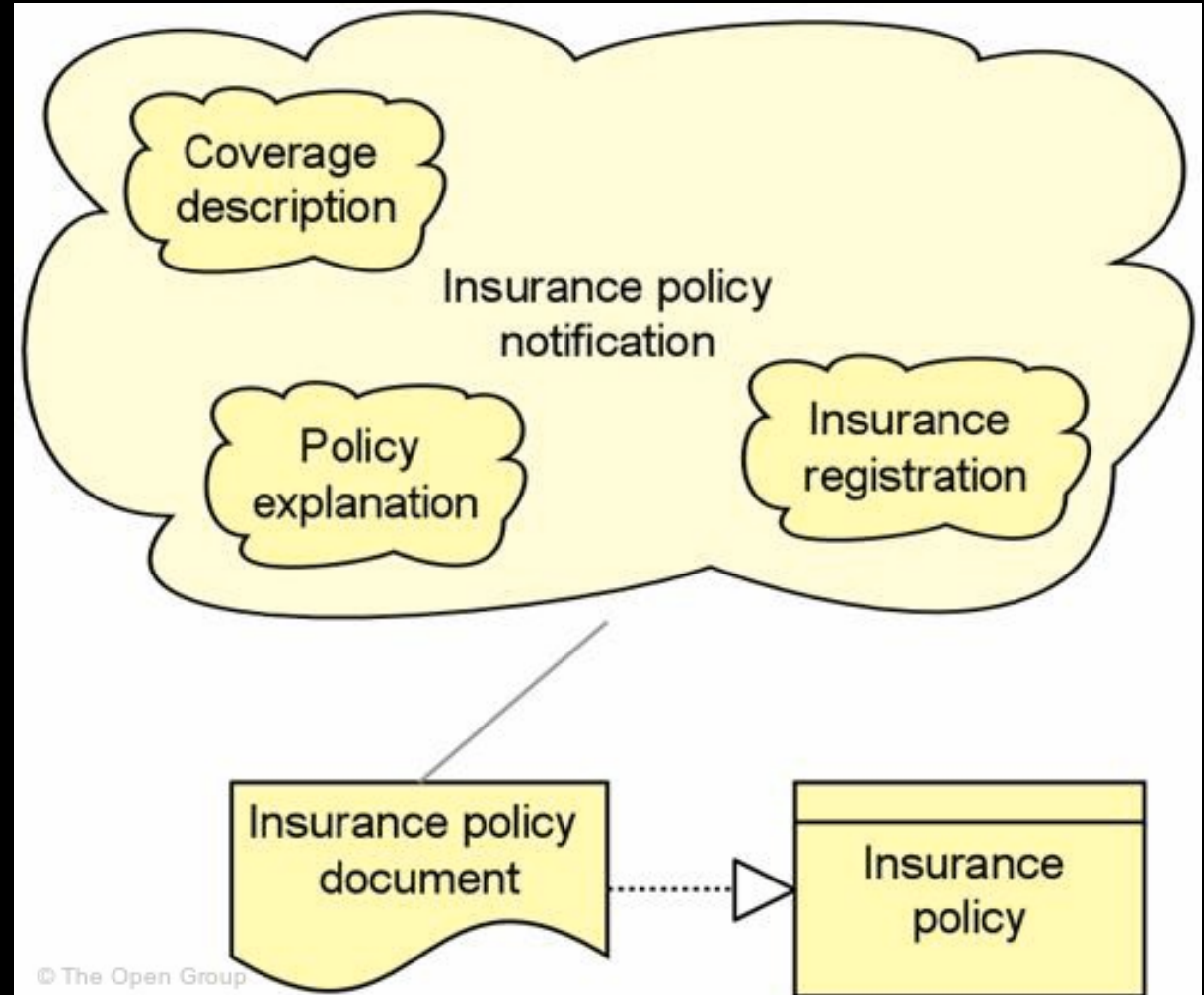
# Meaning



- Meaning is defined as the knowledge or expertise present in a business object or its representation, given a particular context.
- A meaning is the information-related counterpart of a value: it represents the intention of a business object or representation (for example, a document, message; the representations related to a business object). It is a description that expresses the *intent* of a representation; i.e., how it informs the *external user*.
- A meaning can be associated with a representation that carries this meaning. The name of a meaning should preferably be a noun or noun phrase.

# Meaning Example

- The model shows an Insurance policy document that is the representation of an Insurance policy, which is a business object. The meaning related to this document is the Insurance policy notification, which consists of a Policy explanation, an Insurance registration, and a Coverage description.



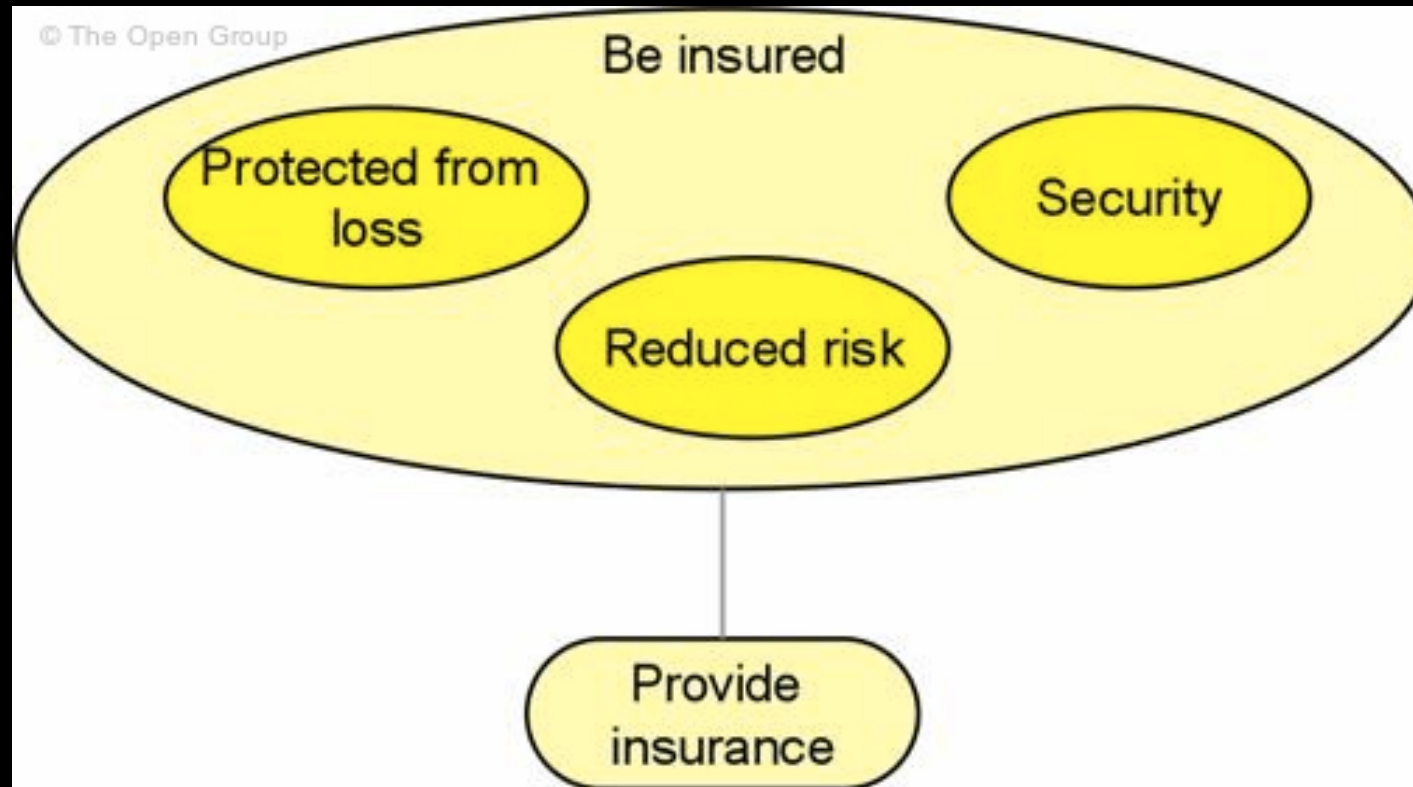
# Value

Value

- The *value* of a product or service is that which makes some party appreciate it, possibly in relation to providing it, but more typically to acquiring it.
- Value may apply to what a party gets by selling or making available some product or service, or it may apply to what a party gets by buying or obtaining access to it. Value is often expressed in terms of money, but it has long since been recognized that non-monetary value is also essential to business; for example, practical/functional value (including the *right* to use a service), and the value of information or knowledge.

## Example of Value

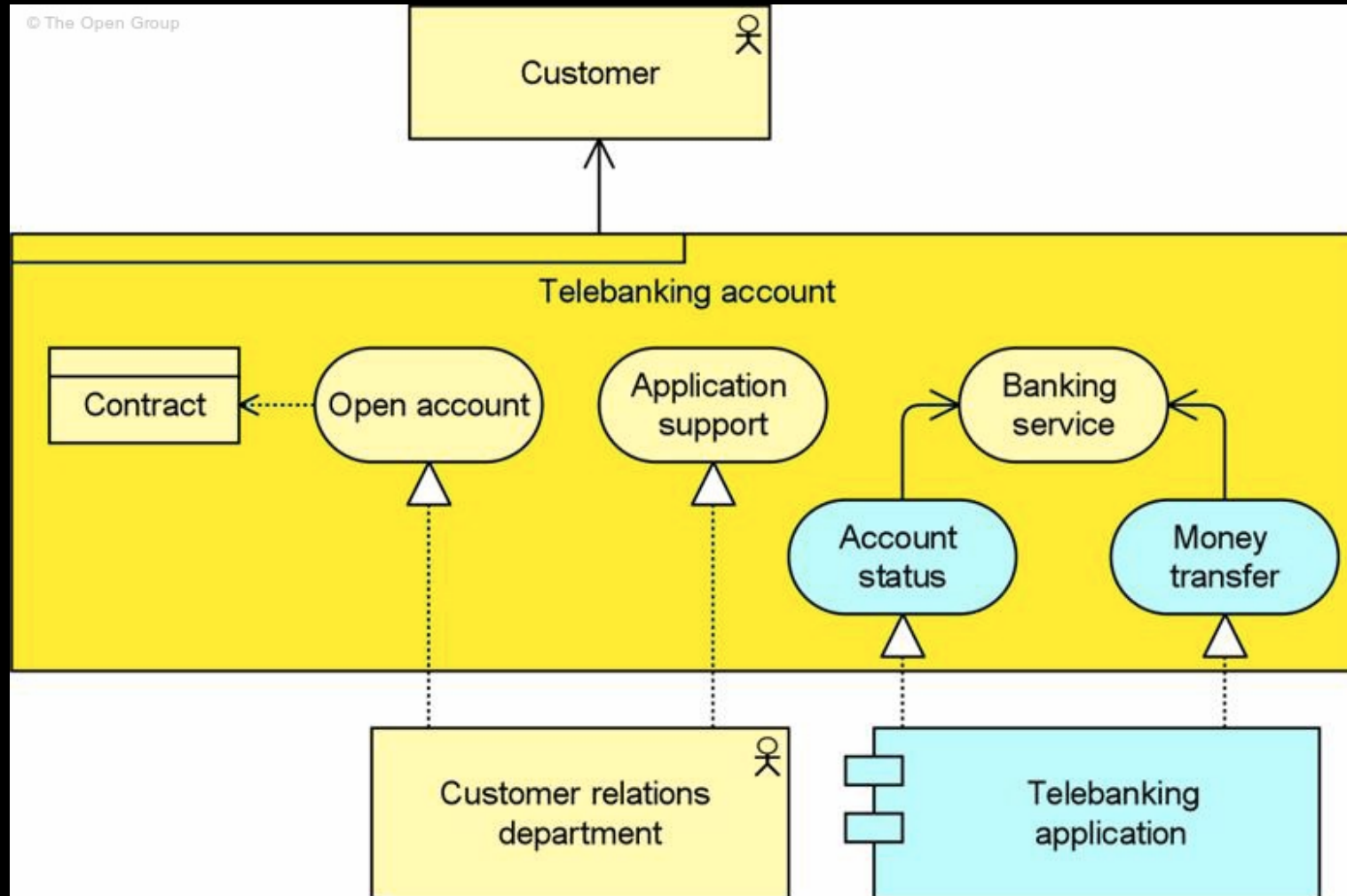
- In the model the value Be Insured is the highest-level expression of what the service Provide Insurance enables the client to do; three “sub-values” are distinguished that are part of what Be Insured amounts to.



# Product

- A product is defined as a coherent collection of services, accompanied by a contract/set of agreements, which is offered as a whole to (internal or external) customers.
- This definition describes financial, services-based, or information products that are common in information-intensive organizations, rather than physical products.

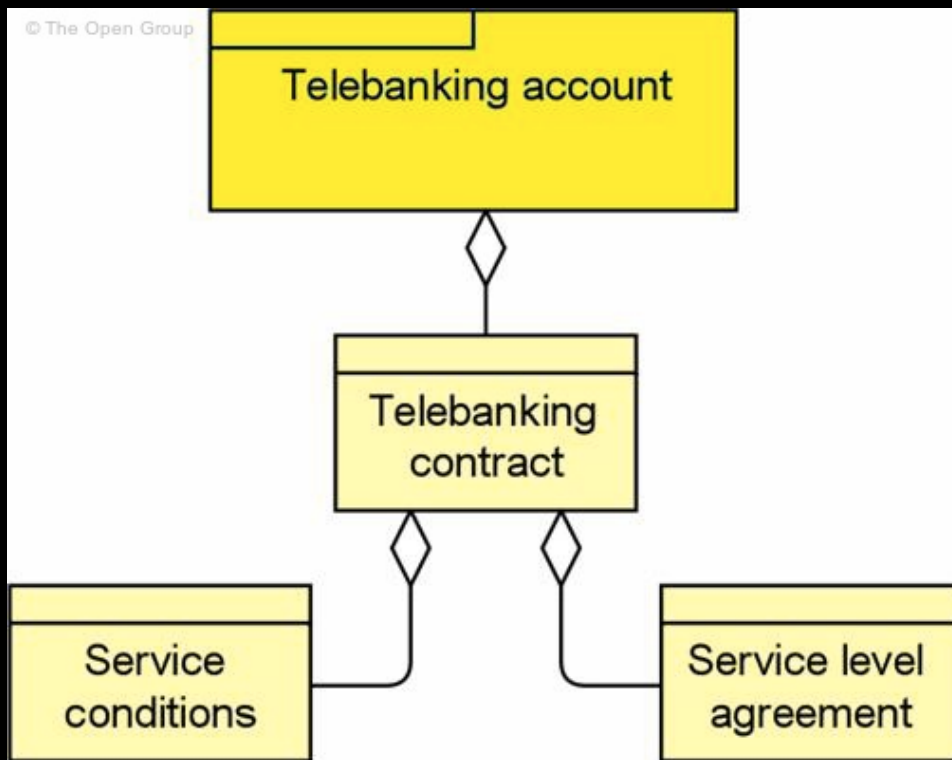
# Product Example



# Contract

- We define a *contract* as a formal or informal specification of agreement that specifies the rights and obligations associated with a product.
- The relationships that apply to a business object also apply to a contract.
- In addition, a contract may have an aggregation relationship with a product.
- The name of a contract is preferably a noun.

# Contract Example

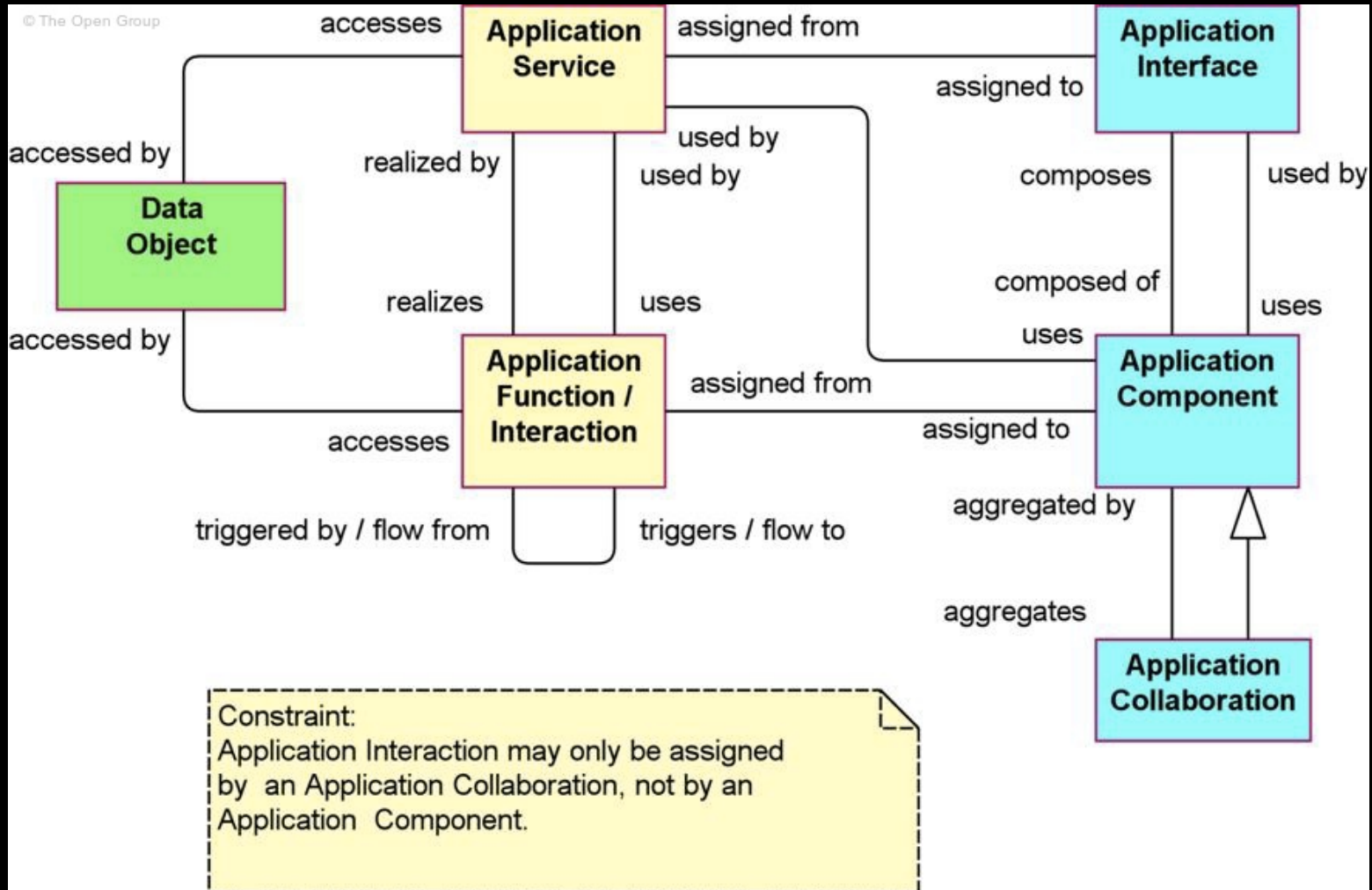


- The model shows a Telebanking contract associated with the product Telebanking account. The contract consists of two parts (subcontracts): the Service Conditions and a Service Level Agreement.



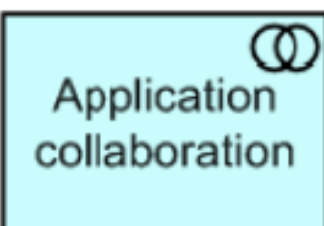
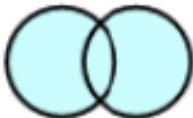


■



# Application Layer Meta-Model

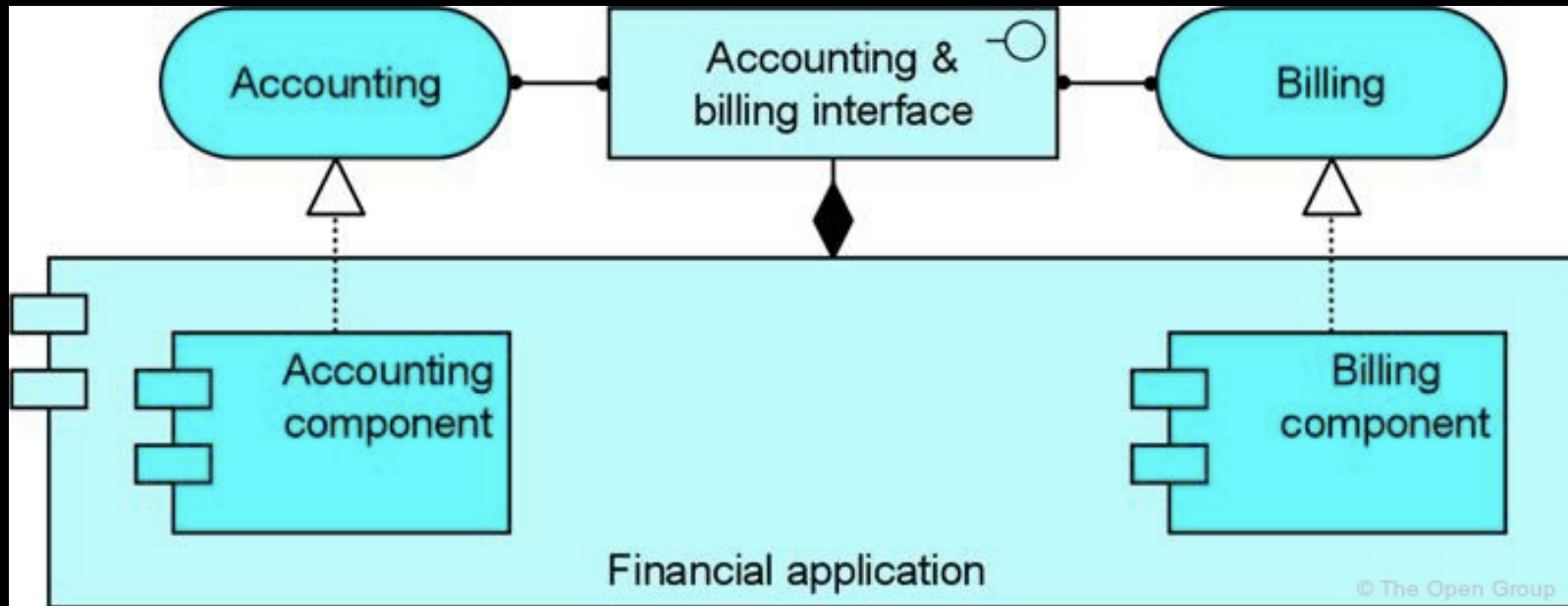


# Application Layer Active Structure Concepts - Relevant Elements

<p>Application component</p>	<p>A modular, deployable, and replaceable part of a software system that encapsulates its behavior and data and exposes these through a set of interfaces.</p>	 <p>Application component</p>	
<p>Application collaboration</p>	<p>An aggregate of two or more application components that work together to perform collective behavior.</p>	 <p>Application collaboration</p>	
<p>Application interface</p>	<p>A point of access where an application service is made available to a user or another application component.</p>	 <p>Application interface</p>	

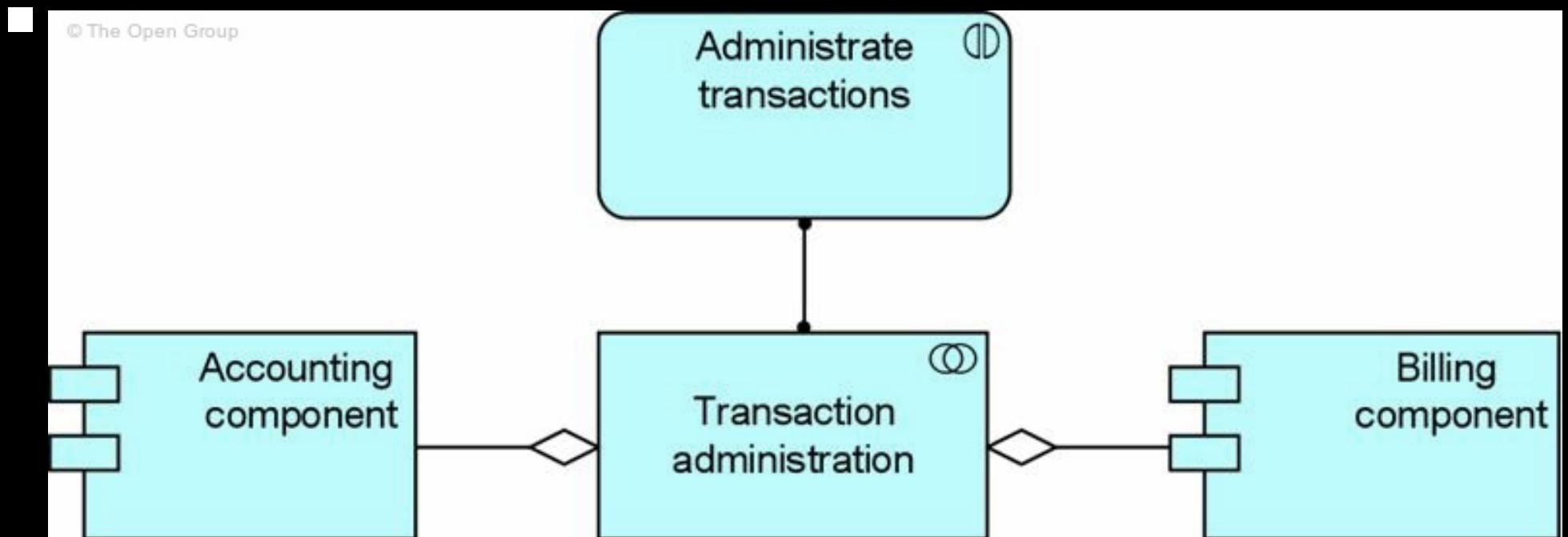
# Application Component Example

In the model a financial application is depicted as an application component consisting of two subcomponents for accounting and billing, each of which offers an application service to the environment. These services are accessible through a shared accounting & billing application interface, which is part of the financial application.



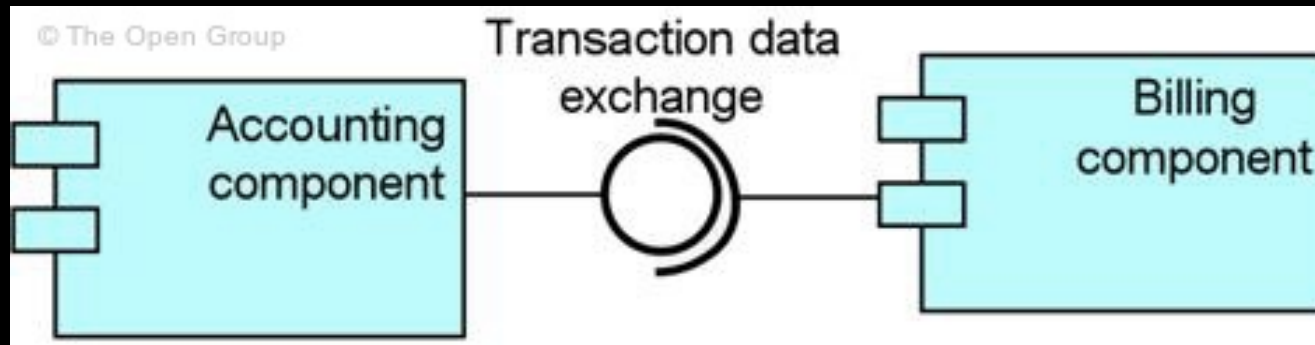
# Application Collaboration

- In the model two components collaborate in transaction administration: an Accounting component and a Billing component. This collaboration performs the application interaction Administrate transactions.








# Application Interface

- In the model, an Accounting component is shown that provides an application interface for Transaction data exchange, and a Billing component that requires such an interface.

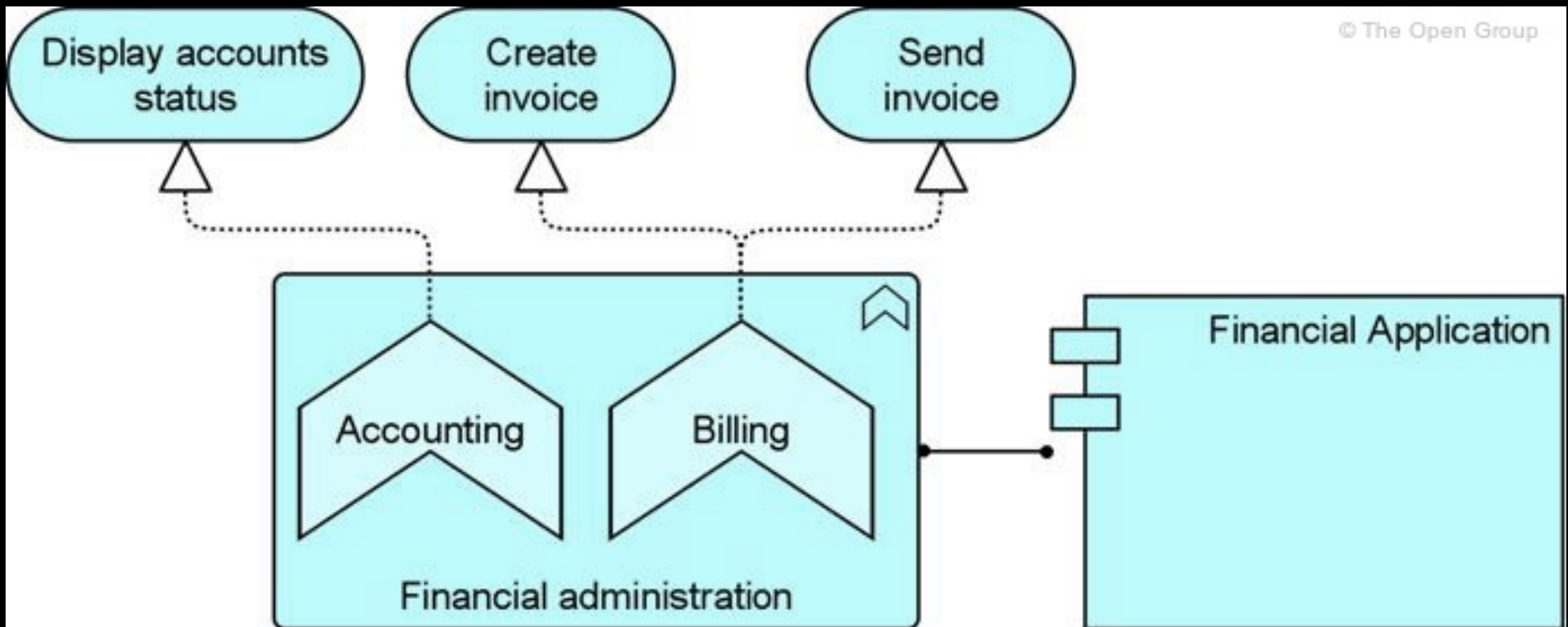


# Application Layer Behavioral Structure Concepts - Relevant Elements

<p>Application function</p>	<p>A behavior element that groups automated behavior that can be performed by an application component.</p>		
<p>Application interaction</p>	<p>A behavior element that describes the behavior of an application collaboration.</p>		
<p>Application service</p>	<p>A service that exposes automated behavior.</p>		

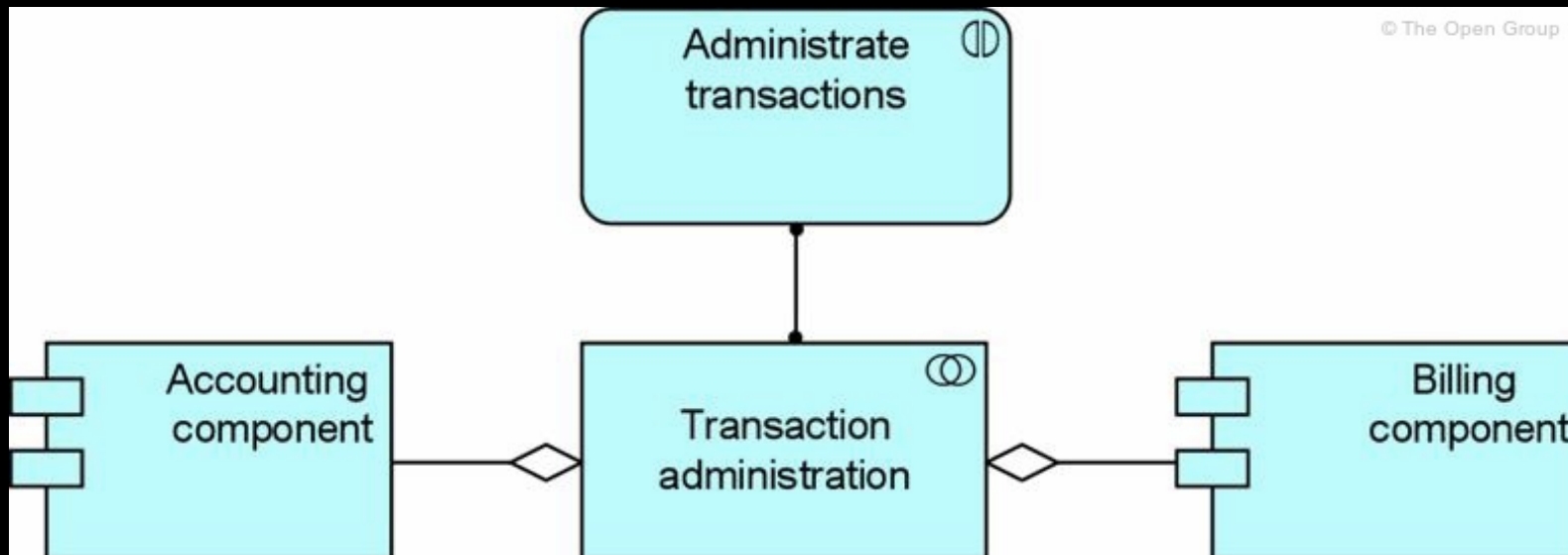
# Application Function Example

In the model the internal behavior of the Financial application component is modeled as an application function consisting of two sub-functions. These application functions realize the application services that are made available to the users of the application.



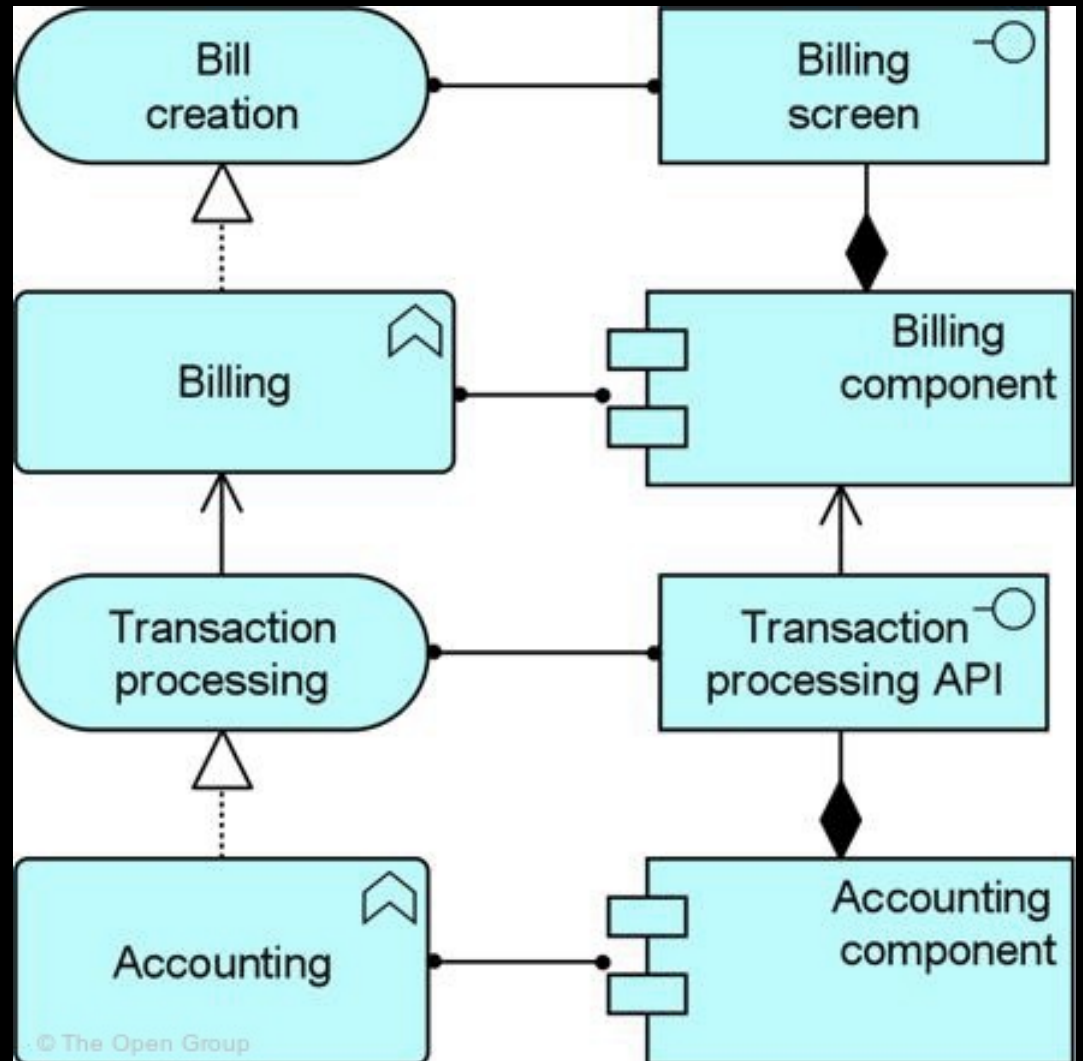
# Application Interaction Example

- In the model, an Accounting component and a Billing component of a financial system co-operate to compose an *administrative transactions* interaction. This is modeled as an application interaction assigned to the collaboration between the two components.

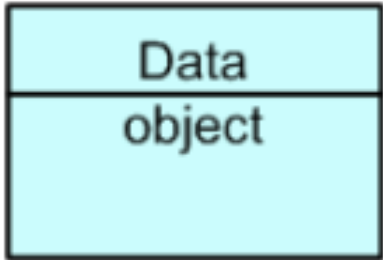




# Application Service Example



# Application Layer Passive Structure Concepts - Relevant Elements

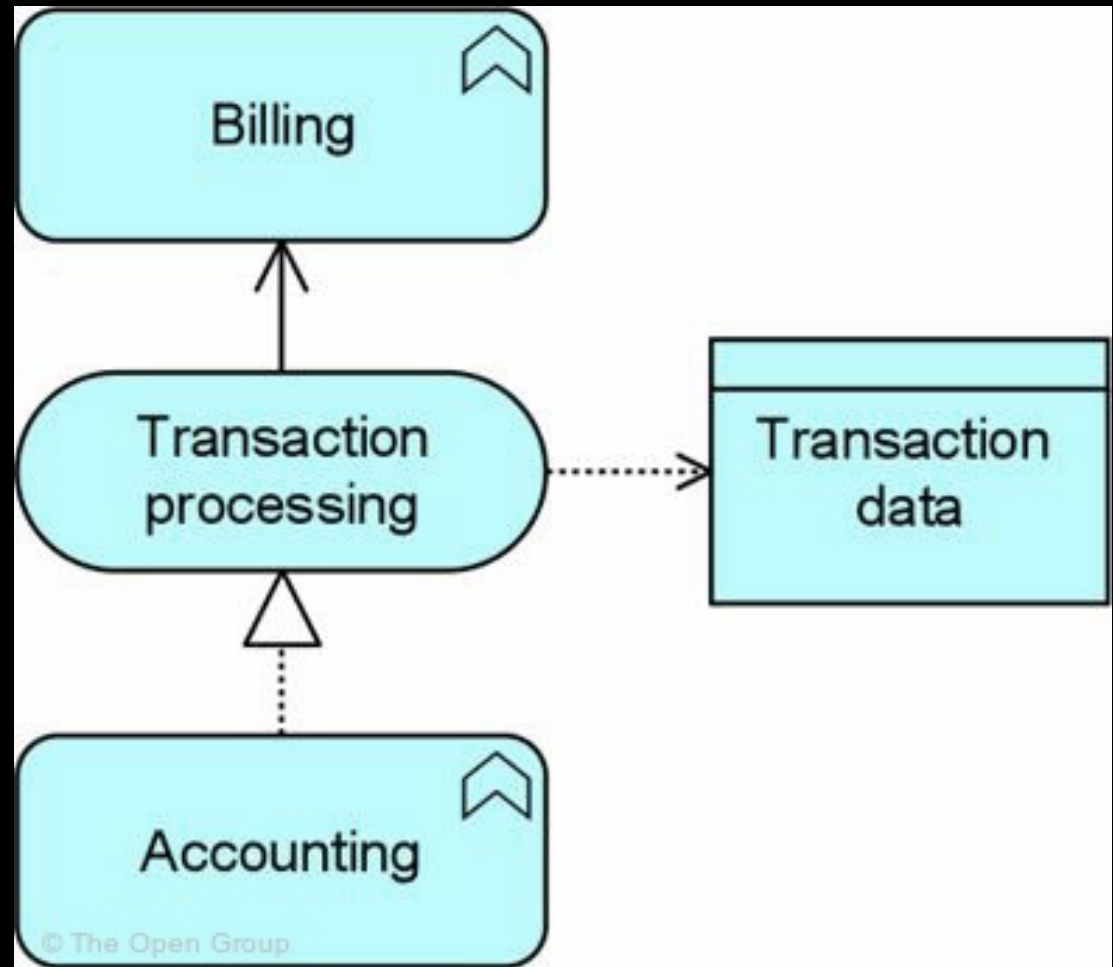
Data object	A passive element suitable for automated processing.	
-------------	--	---

# Data Object

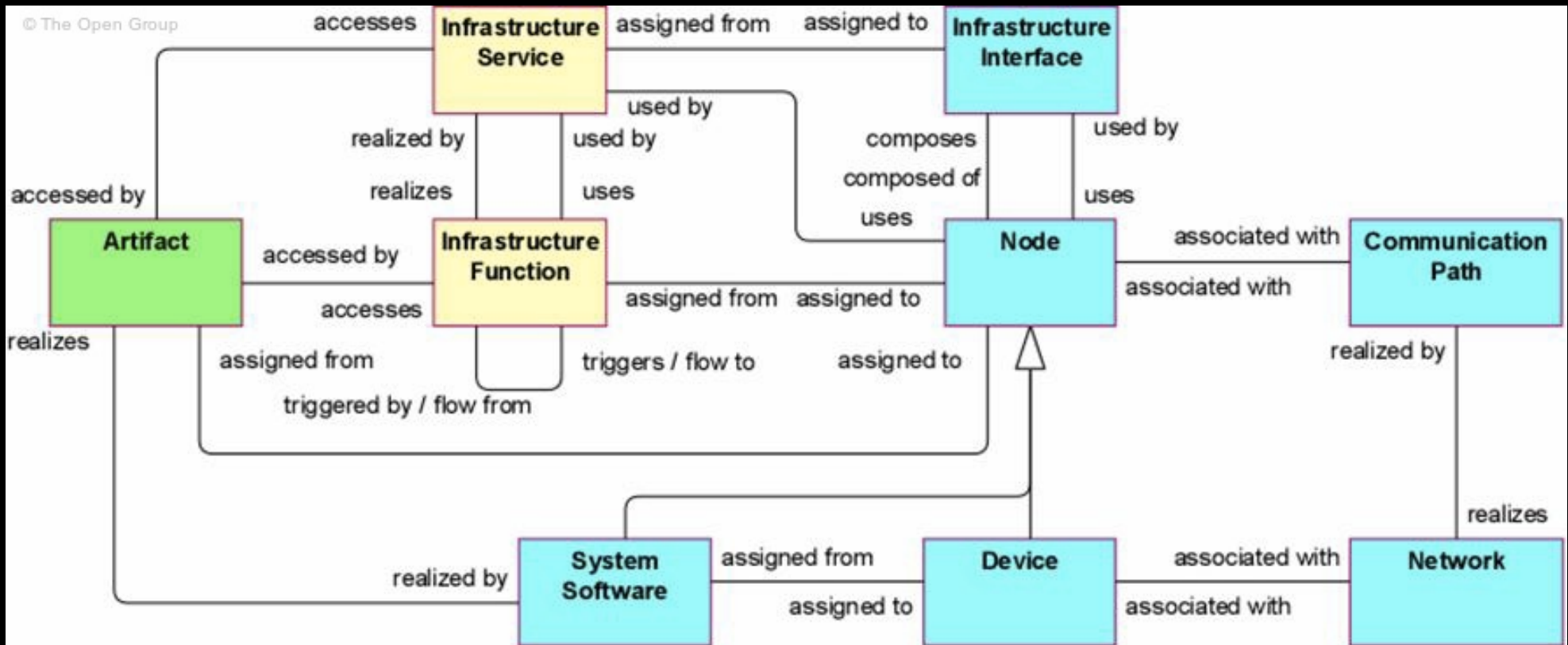
- A data object is defined as a passive element suitable for automated processing.
- An application function operates on a data object. A data object may be communicated via interactions and used or produced by application services. It should be a self-contained piece of information with a clear meaning to the business, not just to the application level. Typical examples of data objects are a customer record, a client database, or an insurance claim.
- A data object can be accessed by an application function, application interaction, or application service. A data object may realize a business object, and may be realized by an artifact. A data object may have association, specialization, aggregation, or composition relationships with other data objects. The name of a data object should preferably be a noun.

## Data Object Example

- In the model below, two application functions co-operate via an application service, in which a data object holding Transaction data is exchanged.



# Technology Layer Meta-Model

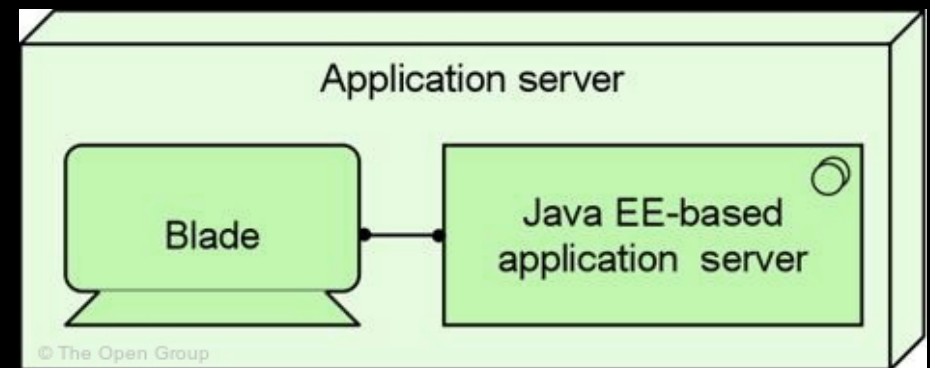
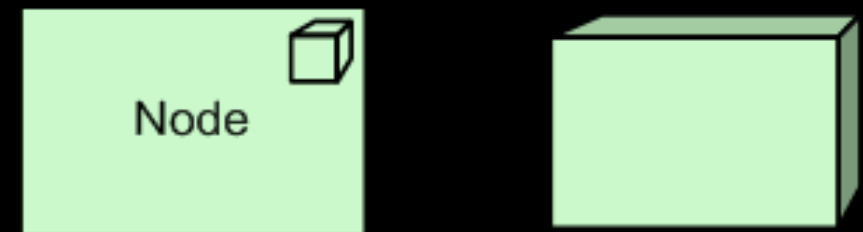


# Technology Layer Active Structure Concepts - Relevant Elements

- The main active structure concept for the technology layer is the *node*. This concept is used to model structural entities in this layer. It is identical to the node concept of UML 2.0. It strictly models the structural aspect of a system: its behavior is modeled by an explicit relationship to the behavioral concepts.
- An *infrastructure interface* is the (logical) location where the infrastructure services offered by a node can be accessed by other nodes or by application components from the application layer.
- Nodes come in two flavors: *device* and *system software*, both taken from UML 2.0. A *device* models a physical computational resource, upon which artifacts may be deployed for execution. *System software* is an infrastructural software component running on a device. Typically, a node consists of a number of sub-nodes; for example, a device such as a server and system software to model the operating system.
- The inter-relationships of components in the technology layer are mainly formed by the communication infrastructure. The *communication path* models the relation between two or more nodes, through which these nodes can exchange information. The physical realization of a communication path is modeled with a *network*; i.e., a physical communication medium between two or more devices (or other networks).

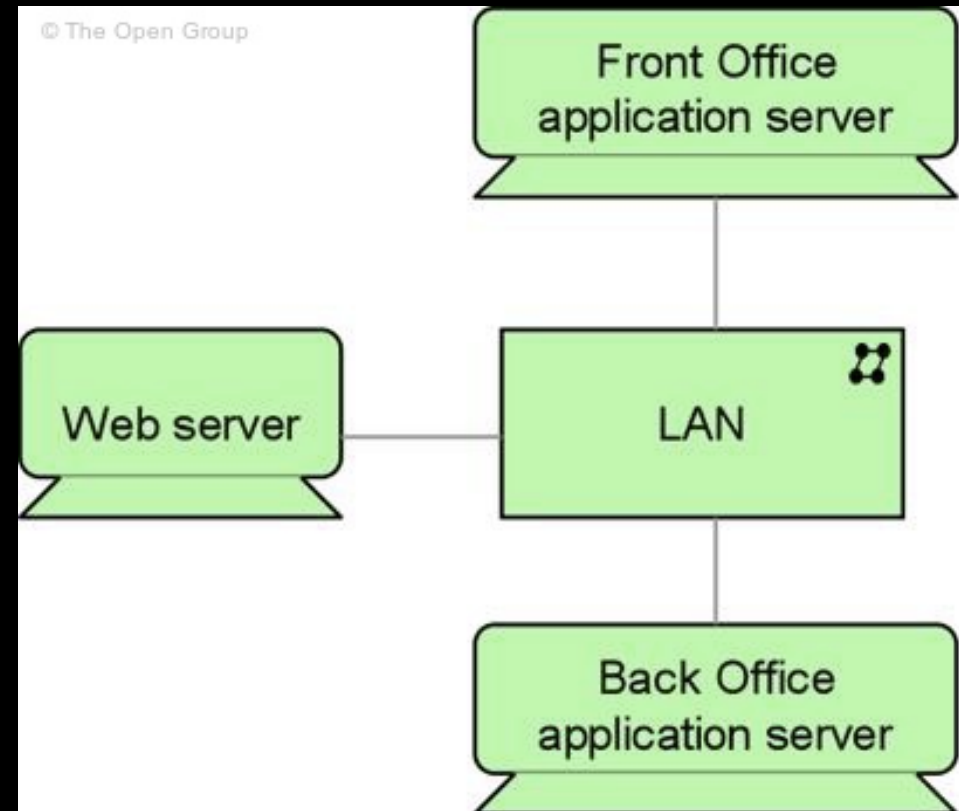
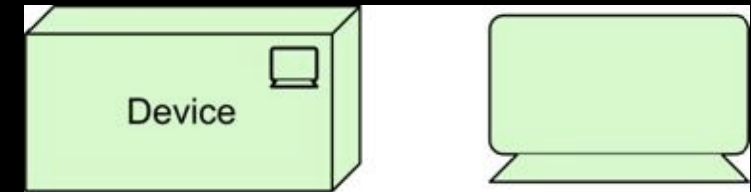
# Node and Example

- A node is defined as a computational resource upon which artifacts may be stored or deployed for execution.
- In the model, we see an Application Server node, which consists of a Blade device and Java EE-based application server system software.



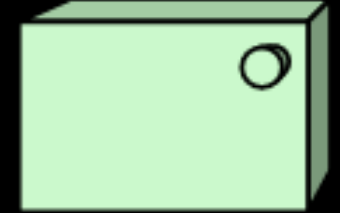
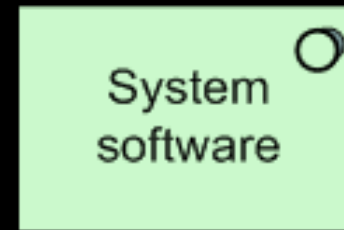
## Device and Example

- A device is defined as a hardware resource upon which artifacts may be stored or deployed for execution.
- The model shows an example of a number of servers, modeled as devices, interconnected through a local area network (LAN).





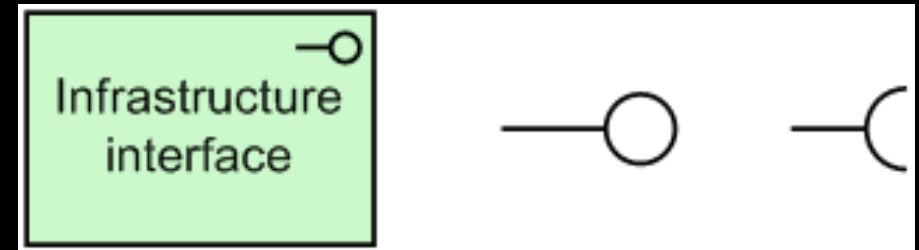
# System Software and Example



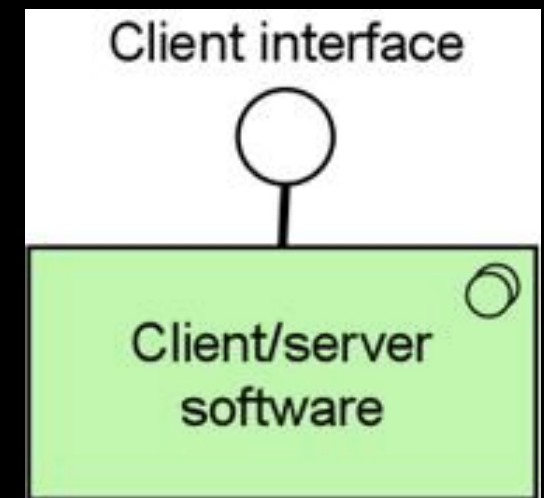
- System software represents a software environment for specific types of components and objects that are deployed on it in the form of artifacts.
- In the model below, we see a mainframe device that deploys two system software environments: a customer transaction server and a database management system (DBMS).



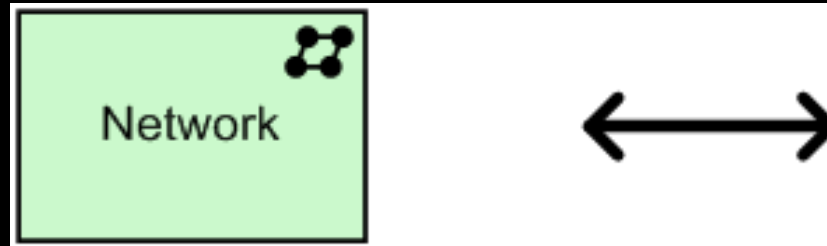
# Infrastructure Interface and Example



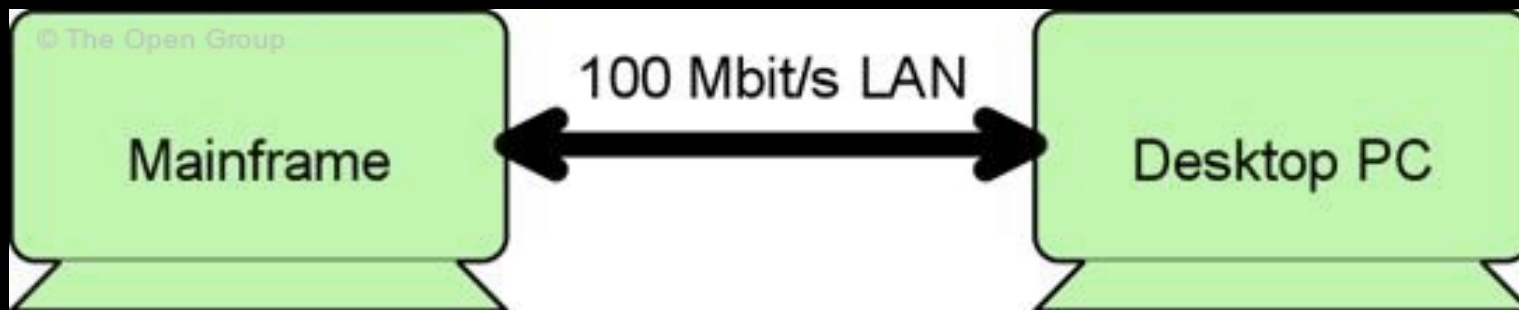
- An infrastructure interface is defined as a point of access where infrastructure services offered by a node can be accessed by other nodes and application components.
- In the model, we see a client infrastructure interface exposed, which is part of the client-server system software.



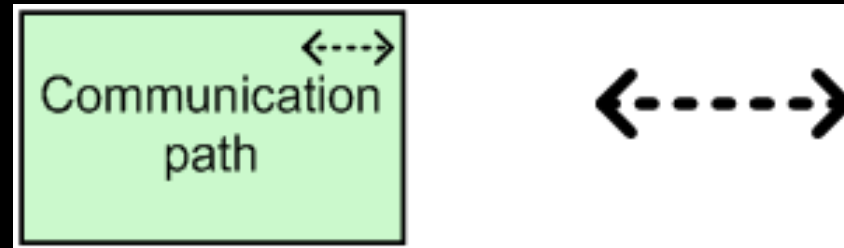
# Network and Example



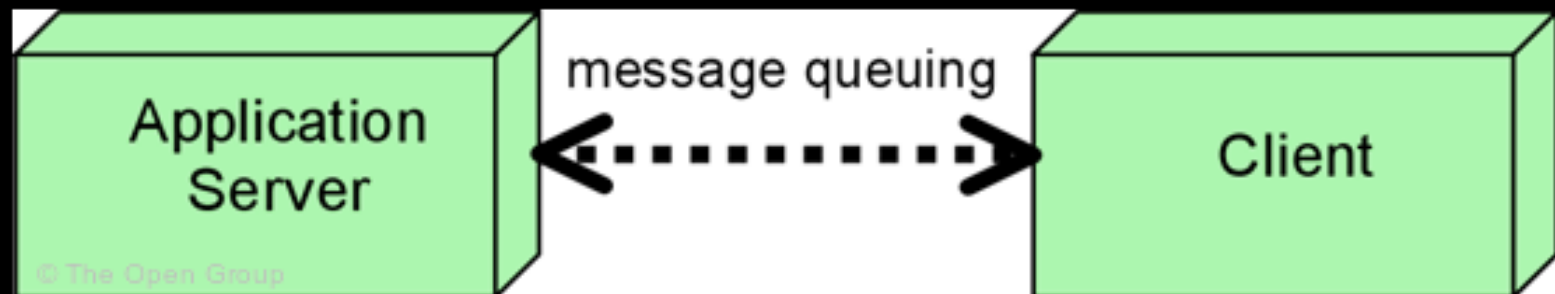
- A network is defined as a communication medium between two or more devices.
- In the model below, a 100 Mb/s LAN network connects a mainframe and PC device.



# Communication Path and Example



- A communication path is defined as a link between two or more nodes, through which these nodes can exchange data.
- In the model, we see a communication path “message queuing” between an Application Server and a Client.

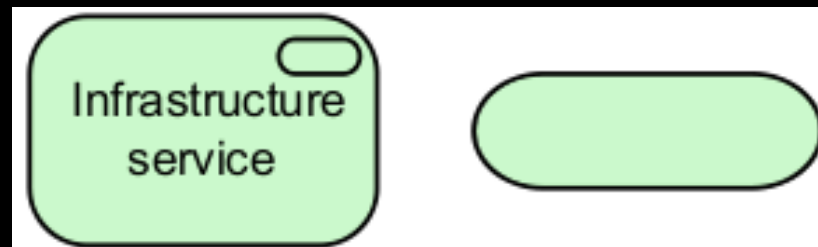


# Technology Layer Behavioural Structure Concepts - Relevant Elements

## ■ Infrastructure Function

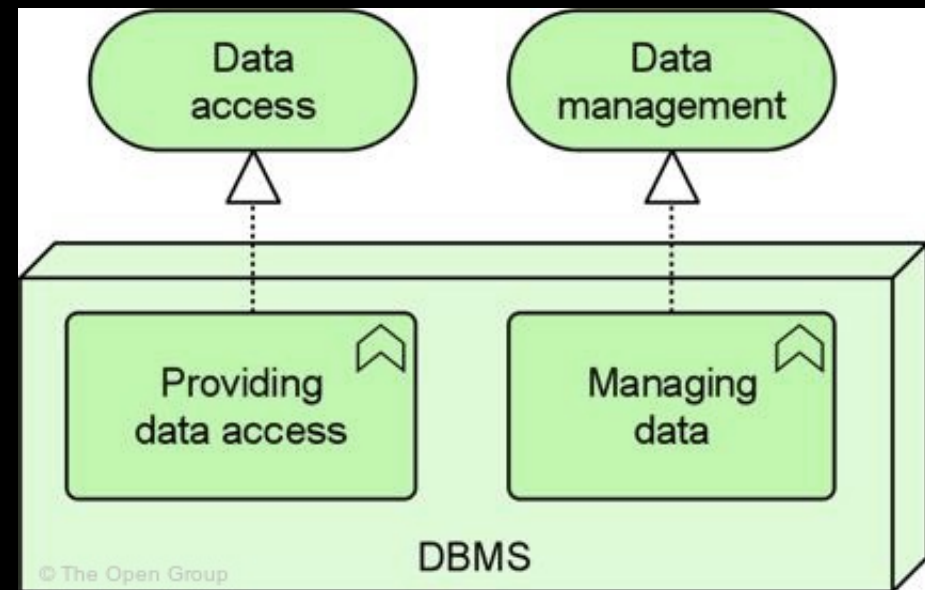


## ■ Infrastructure Service



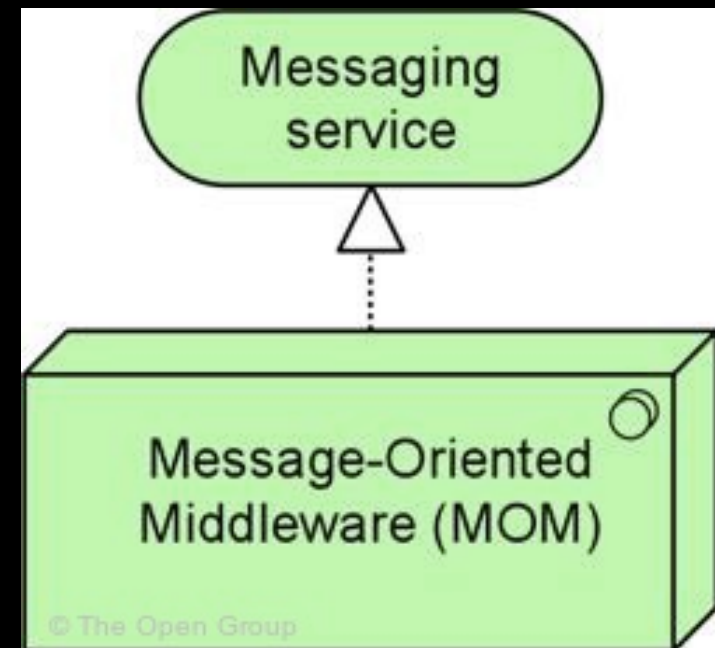
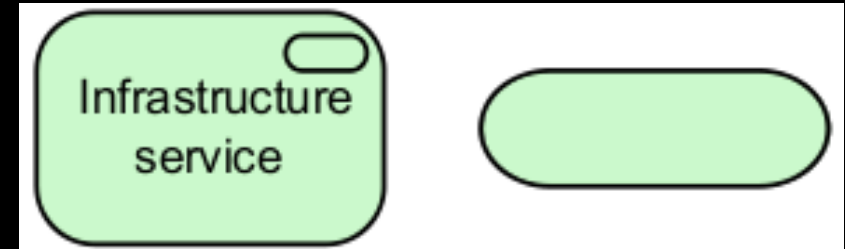
# Infrastructure Function and Example

- An infrastructure function is defined as a behavior element that groups infrastructural behavior that can be performed by a node.
- In the model, the database management system (DBMS) node performs two infrastructure functions: providing data access (realizing a data access service for application software), and managing data (realizing a data management service for database administration).



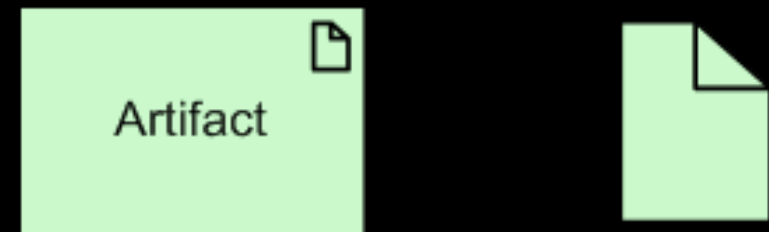
# Infrastructure Service and Example

- An infrastructure service is defined as an externally visible unit of functionality, provided by one or more nodes, exposed through well-defined interfaces, and meaningful to the environment.
- In the model below, we see a Messaging service realized by Message-Oriented Middleware (MOM) system software.

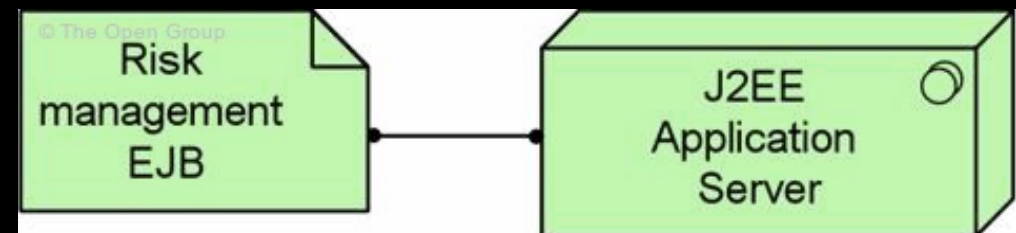


# Technology Layer Passive Structure Concepts - Relevant Elements (Artifact)

- 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

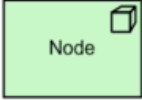
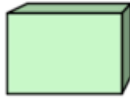

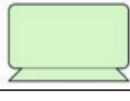

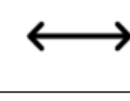
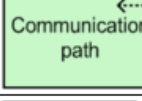
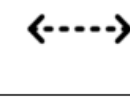
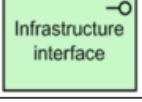
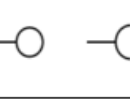
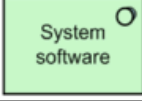
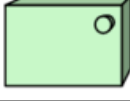
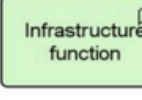

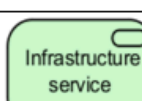
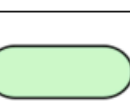

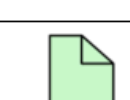


- In the example, we see an artifact Risk management EJB, which represents a deployable unit of code, assigned to (deployed on) an application server

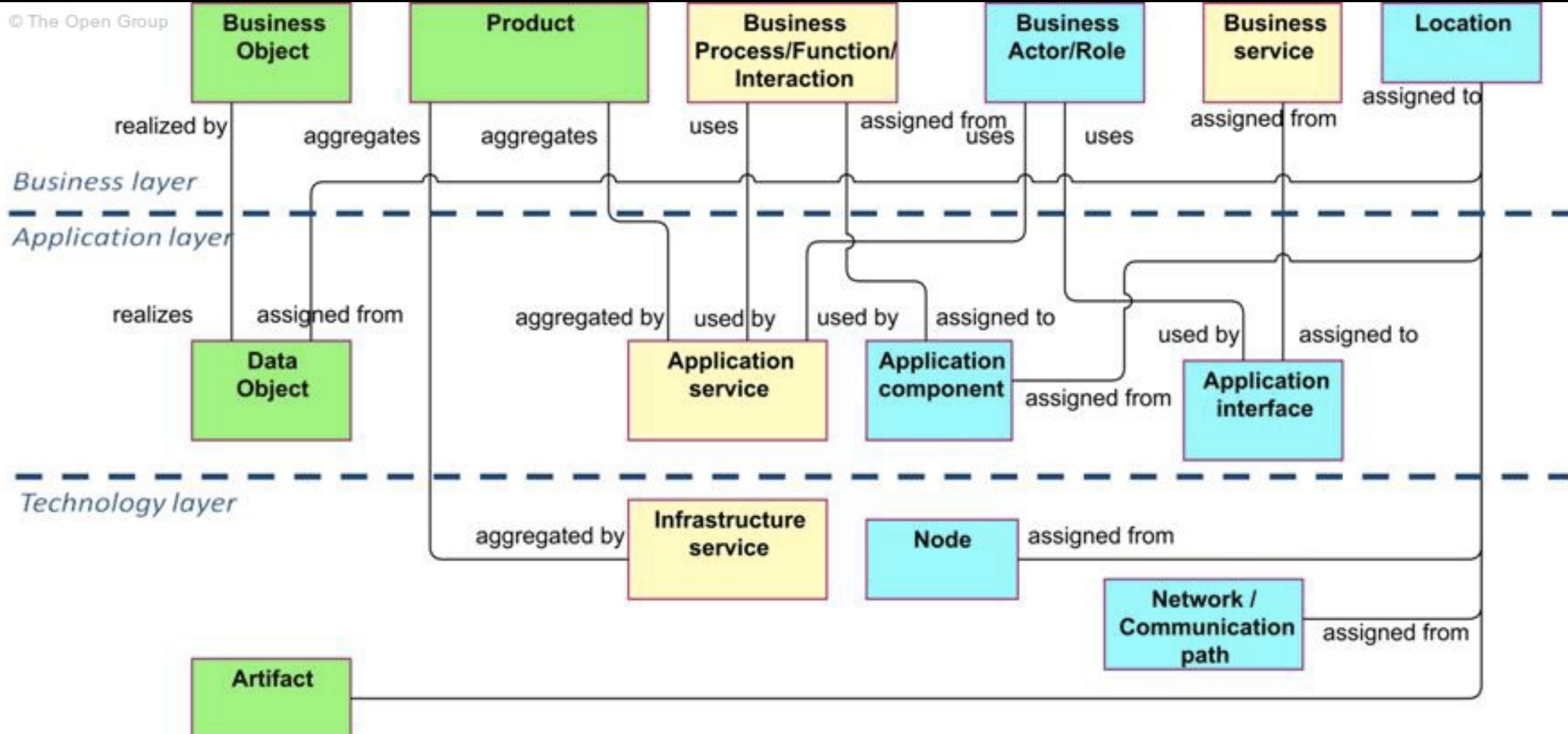




# Technology Layer - Summary

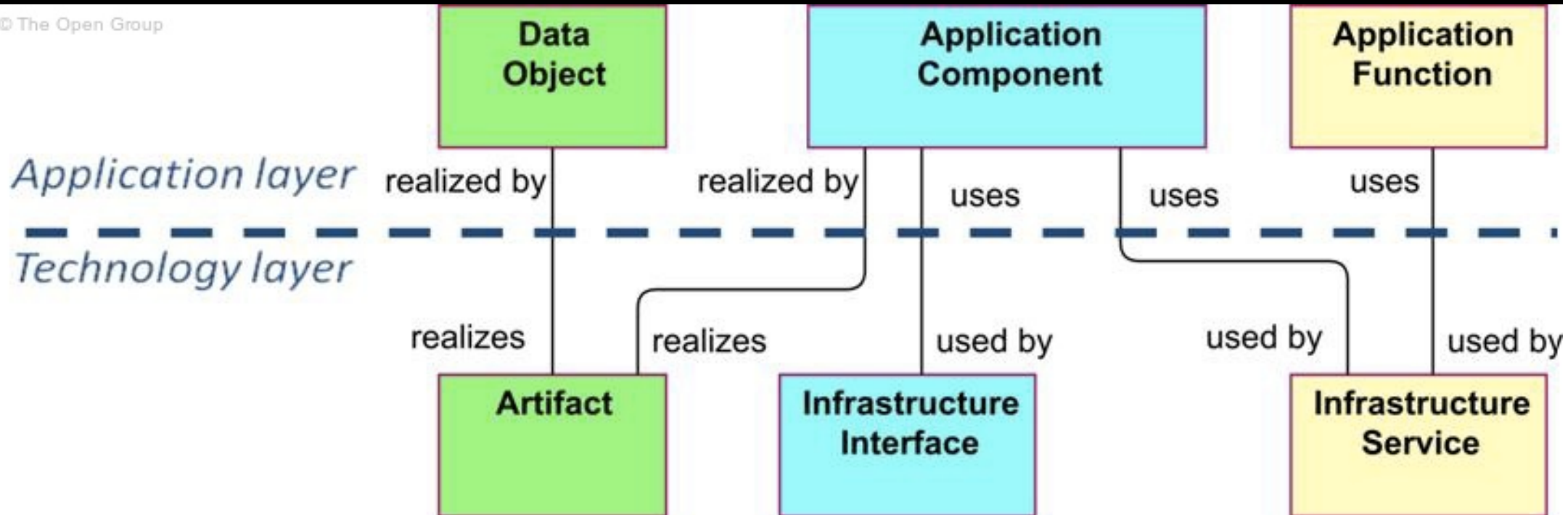
Node	A computational resource upon which artifacts may be stored or deployed for execution.		
Device	A hardware resource upon which artifacts may be stored or deployed for execution.		
Network	A communication medium between two or more devices.		
Communication path	A link between two or more nodes, through which these nodes can exchange data.		
Infrastructure interface	A point of access where infrastructure services offered by a node can be accessed by other nodes and application components.		
System software	A software environment for specific types of components and objects that are deployed on it in the form of artifacts.		
Infrastructure function	A behavior element that groups infrastructural behavior that can be performed by a node.		
Infrastructure service	An externally visible unit of functionality, provided by one or more nodes, exposed through well-defined interfaces, and meaningful to the environment.		
Artifact	A physical piece of data that is used or produced in a software development process, or by deployment and operation of a system.		

# Cross-Layer Dependencies: Business Layer and Lower Layers Alignment



# Cross-Layer Dependencies: Application-Technology Alignment

© The Open Group



# Architecture Viewpoints

The following are examples of stakeholders and concerns as a basis for the specification of viewpoints:

- *End user*: For example, what are the consequences for his work and workplace?
- *Architect*: What is the consequence for the maintainability of a system, with respect to corrective, preventive, and adaptive maintenance?
- *Upper-level management*: How can we ensure our policies are followed in the development and operation of processes and systems? What is the impact of decisions (on personnel, finance, ICT, etc.)?
- *Operational manager*: Responsible for exploitation or maintenance: For example, what new technologies are there to prepare for? Is there a need to adapt maintenance processes? What is the impact of changes to existing applications? How secure are my systems?
- *Project manager*: Responsible for the development of new applications: What are the relevant domains and their relationships? What is the dependence of business processes on the applications to be built? What is their expected performance?
- *Developer*: What are the modifications with respect to the current situation that need to be done?

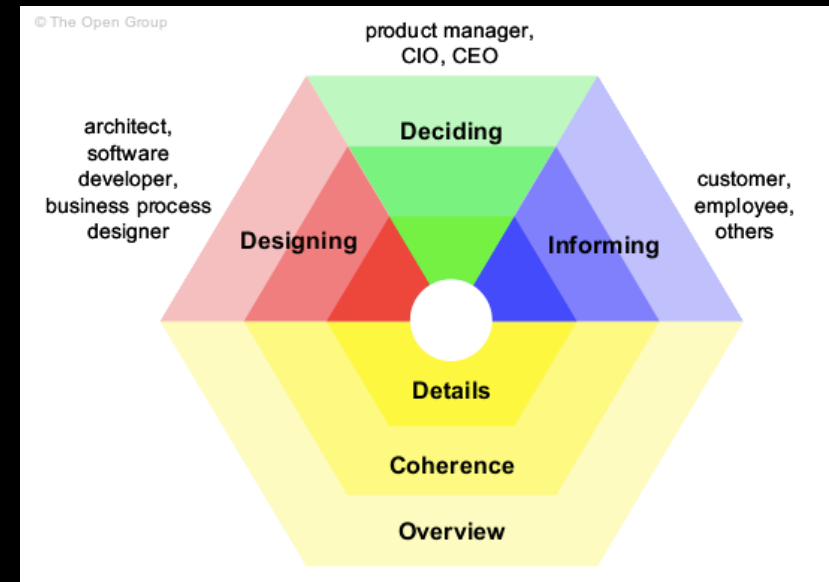
# Viewpoints

- Introductory Viewpoint
- Organization Viewpoint
- Actor Co-operation Viewpoint
- Business Function Viewpoint
- Business Process Viewpoint
- Business Process Co-operation Viewpoint
- Product Viewpoint
- Application Behavior Viewpoint
- Application Co-operation Viewpoint
- Application Structure Viewpoint
- Application Usage Viewpoint
- Infrastructure Viewpoint
- Infrastructure Usage Viewpoint
- Implementation and Deployment Viewpoint
- Information Structure Viewpoint
- Service Realization Viewpoint
- Layered Viewpoint
- Landscape Map Viewpoint

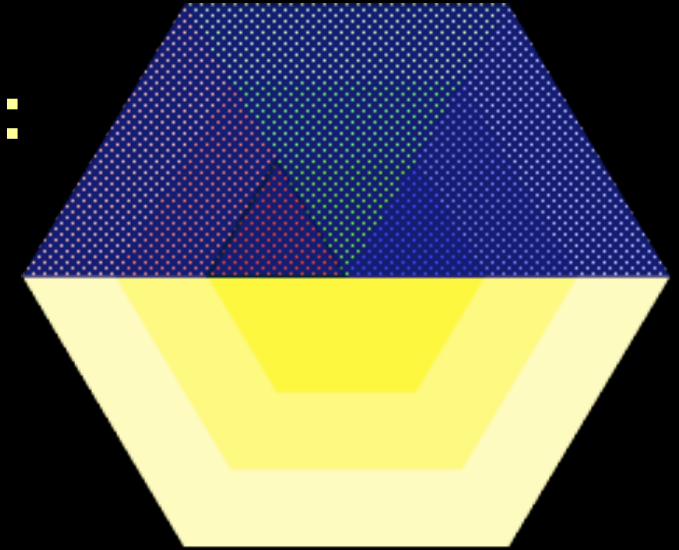
# Viewpoint Classification

	Typical Stakeholders	Purpose	Examples
<b>Details</b>	software engineer, process owner	design, manage	UML class diagram, BPMN process diagram
<b>Coherence</b>	operational managers	analyze dependencies, impact of-change	views expressing relationships like “use”, “realize”, and “assign”
<b>Overview</b>	enterprise architect, CIO, CEO	change management	landscape map

	Typical Stakeholders	Purpose	Examples
<b>Designing</b>	architect, software developer, business process designer	navigate, design, support design decisions, compare alternatives	UML diagram, BPMN diagram, flowchart, ER diagram
<b>Deciding</b>	manager, CIO, CEO	decision-making	cross-reference table, landscape map, list, report
<b>Informing</b>	employee, customer, others	explain, convince, obtain commitment	animation, cartoon, process illustration, chart

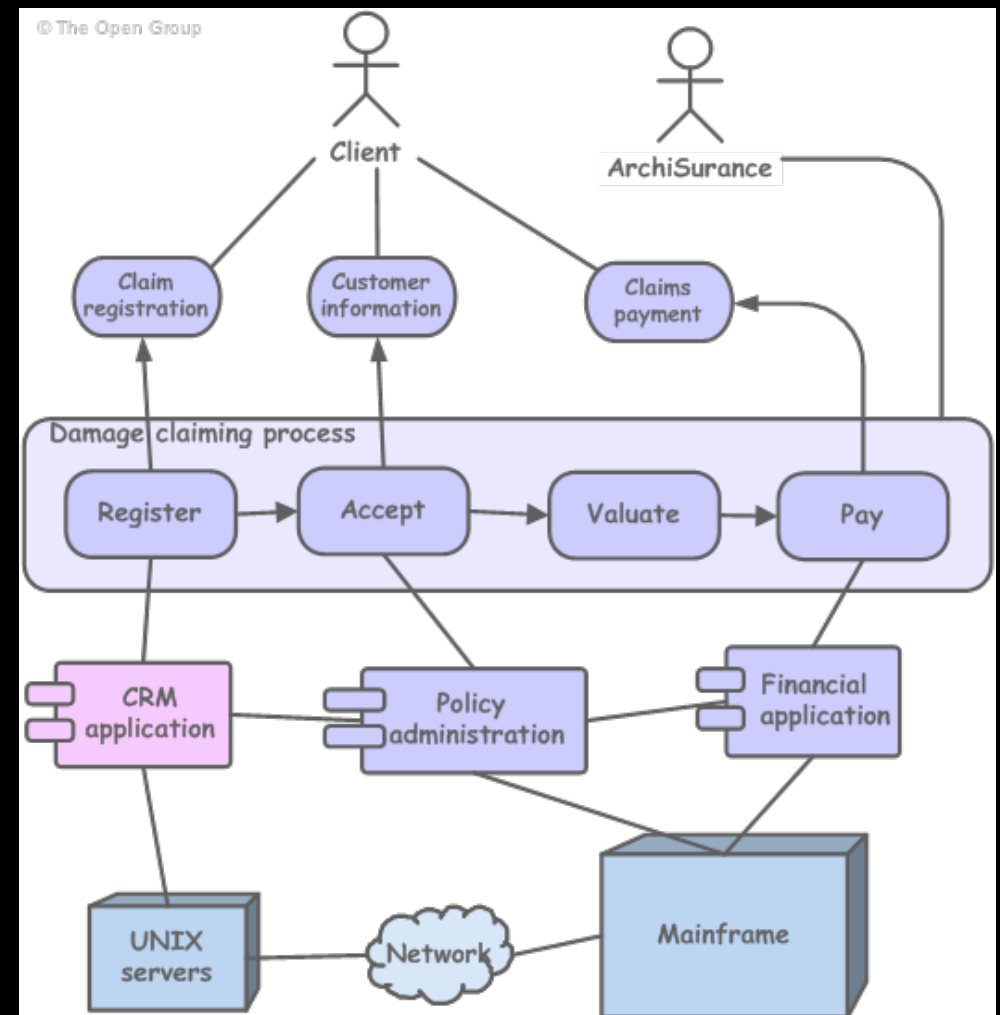
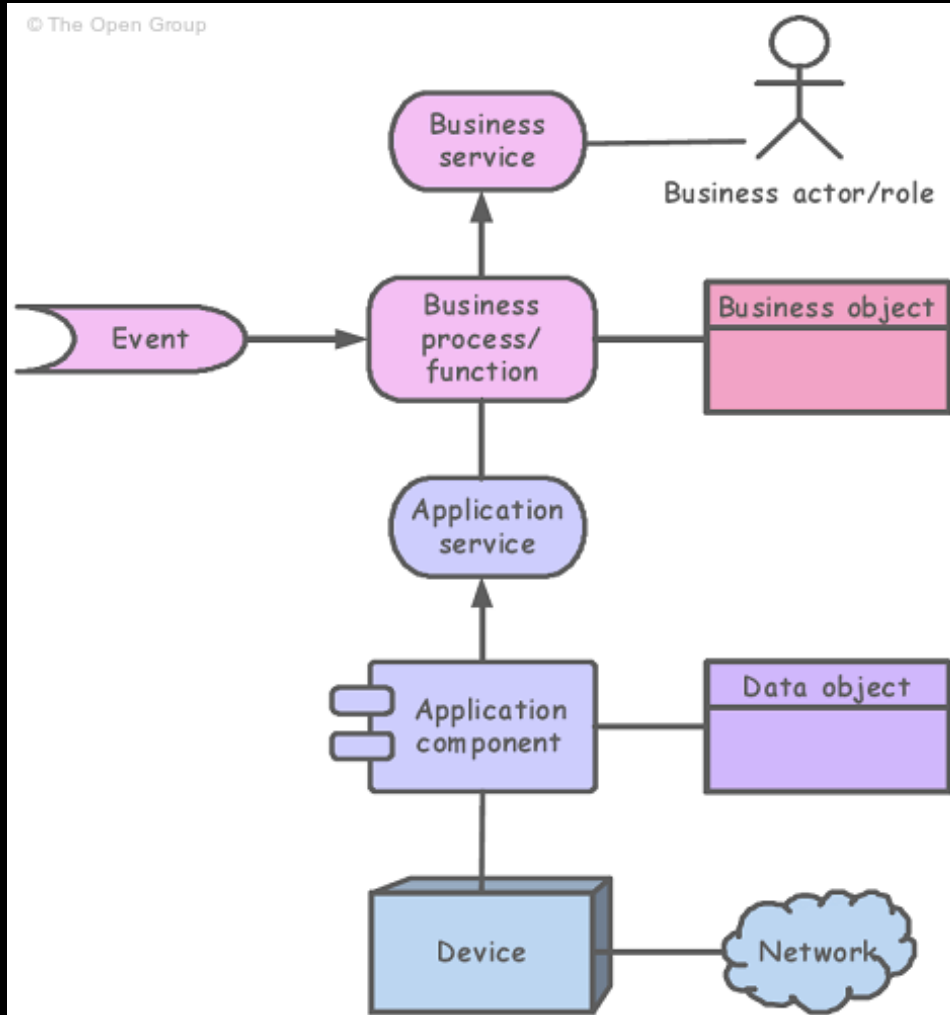


# Introductory Viewpoint Description:



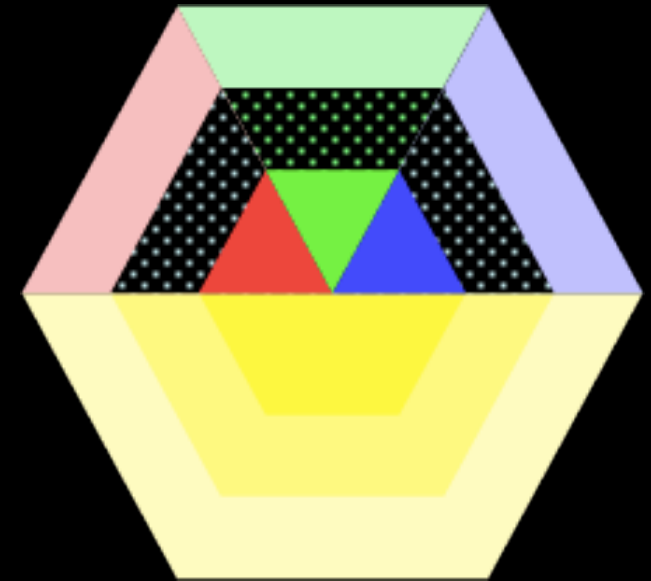
<b>Introductory Viewpoint</b>	
<b>Stakeholders</b>	Enterprise architects, managers
<b>Concerns</b>	Make design choices visible, convince stakeholders
<b>Purpose</b>	Designing, deciding, informing
<b>Abstraction Level</b>	Coherence, Overview, Detail
<b>Layer</b>	Business, Application, and Technology layers (see also Figure 4)
<b>Aspects</b>	Active structure, behavior, passive structure (see also Figure 4)

# Introductory Viewpoint Description: Concepts/Relationships and Example



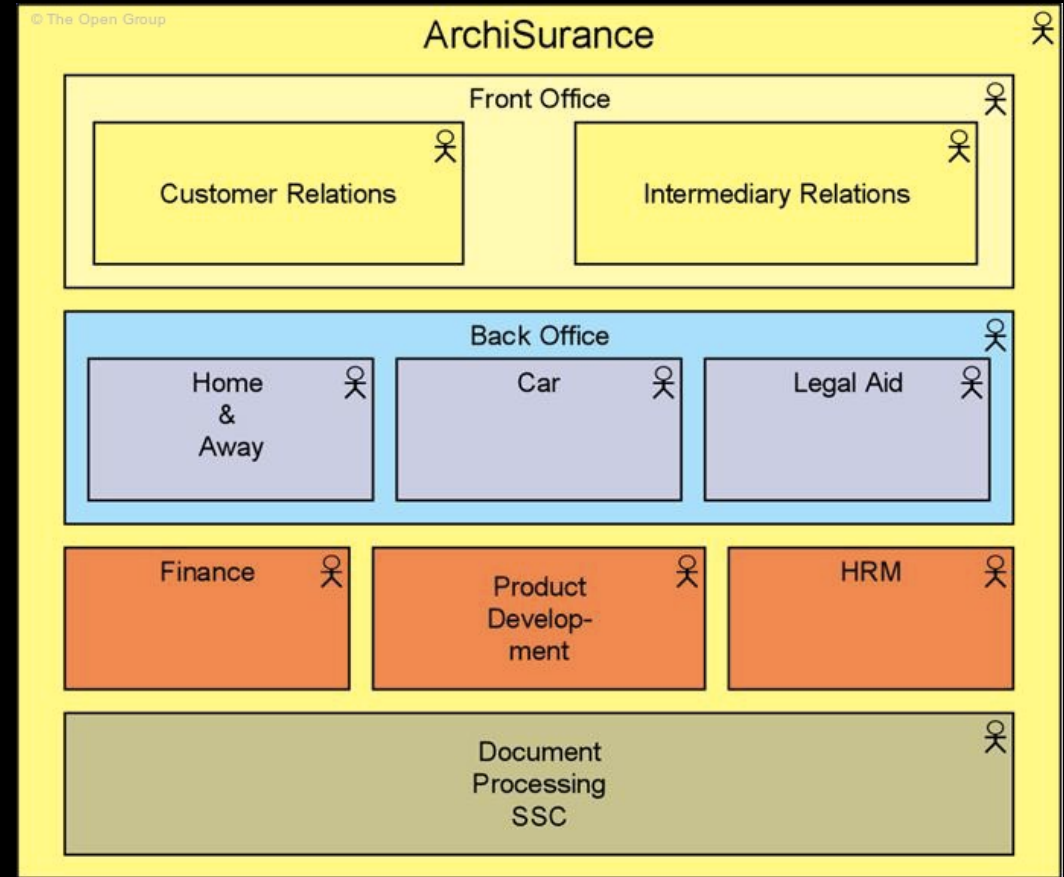
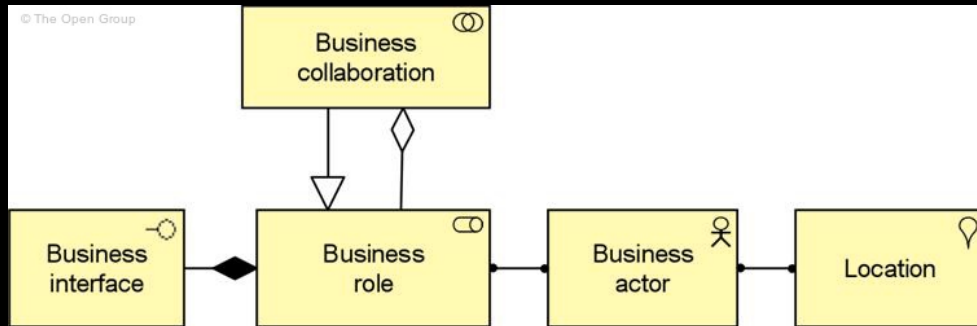


# Organization Viewpoint Description

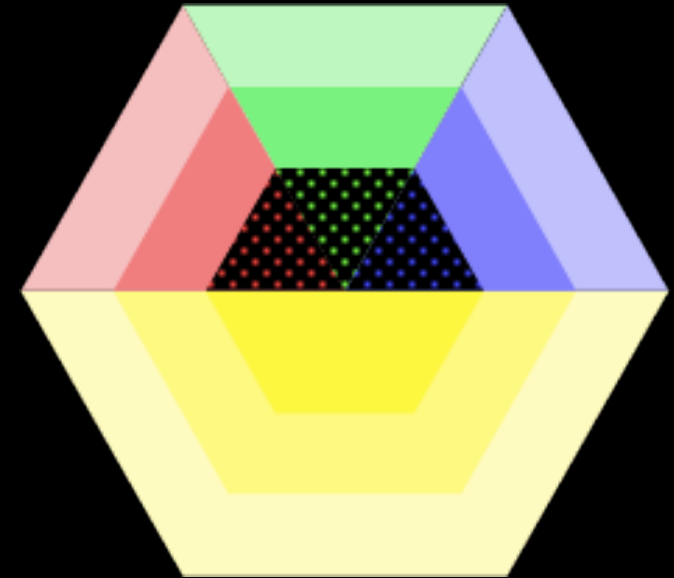


Organization Viewpoint	
<b>Stakeholders</b>	Enterprise, process and domain architects, managers, employees, shareholders
<b>Concerns</b>	Identification of competencies, authority, and responsibilities
<b>Purpose</b>	Designing, deciding, informing
<b>Abstraction Level</b>	Coherence
<b>Layer</b>	Business layer (see also Figure 4)
<b>Aspects</b>	Active structure (see also Figure 4)

# Organization Viewpoint Description Concepts/Relationships and Example

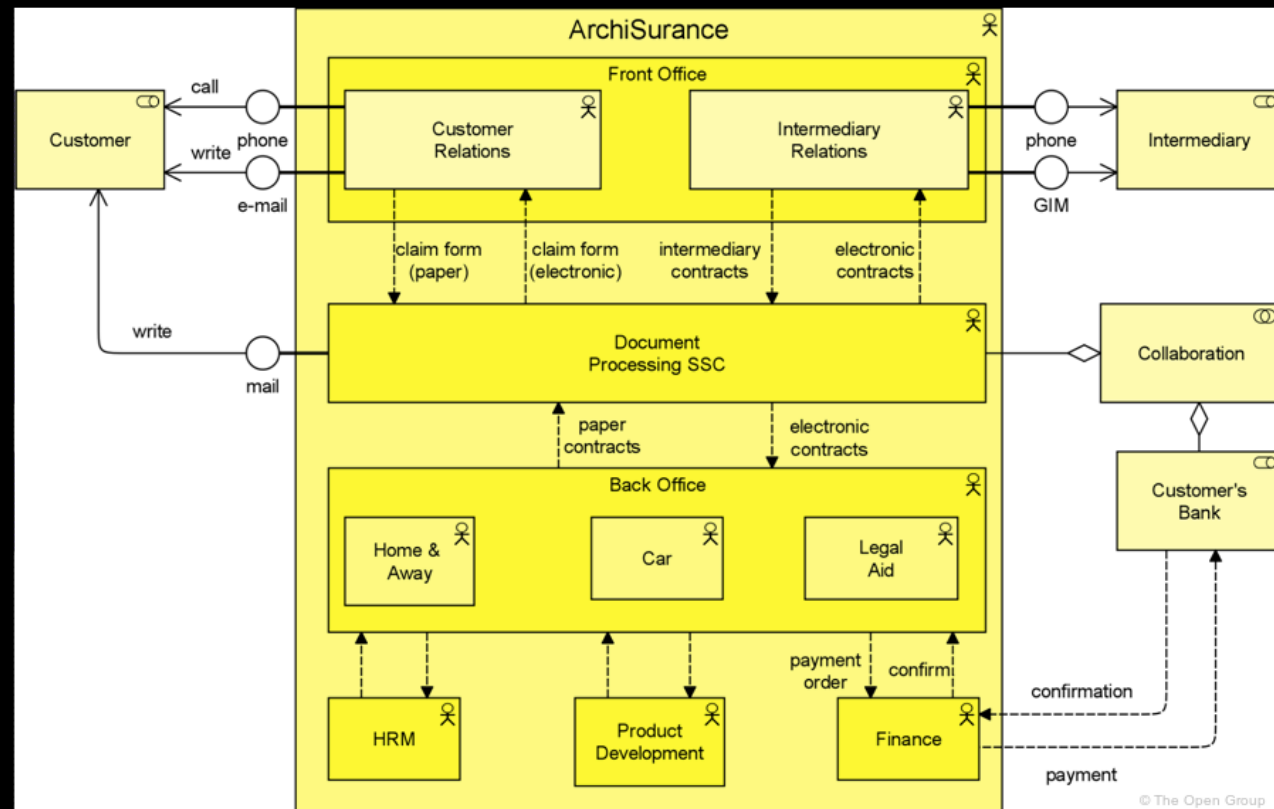
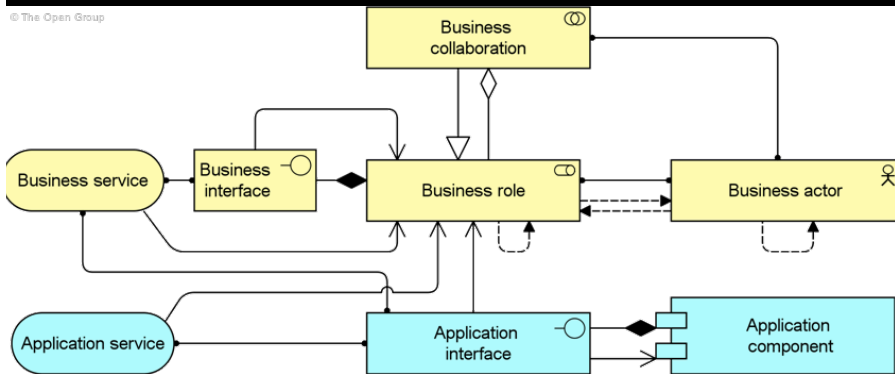


# Actor Co-operation Viewpoint



Actor Co-operation Viewpoint	
<b>Stakeholders</b>	Enterprise, process, and domain architects
<b>Concerns</b>	Relationships of actors with their environment
<b>Purpose</b>	Designing, deciding, informing
<b>Abstraction Level</b>	Detail
<b>Layer</b>	Business layer (application layer) (see also Figure 4)
<b>Aspects</b>	Active structure, behavior (see also Figure 4)

# Actor Co-operation Viewpoint: Concepts/Relationships and Example



## More details

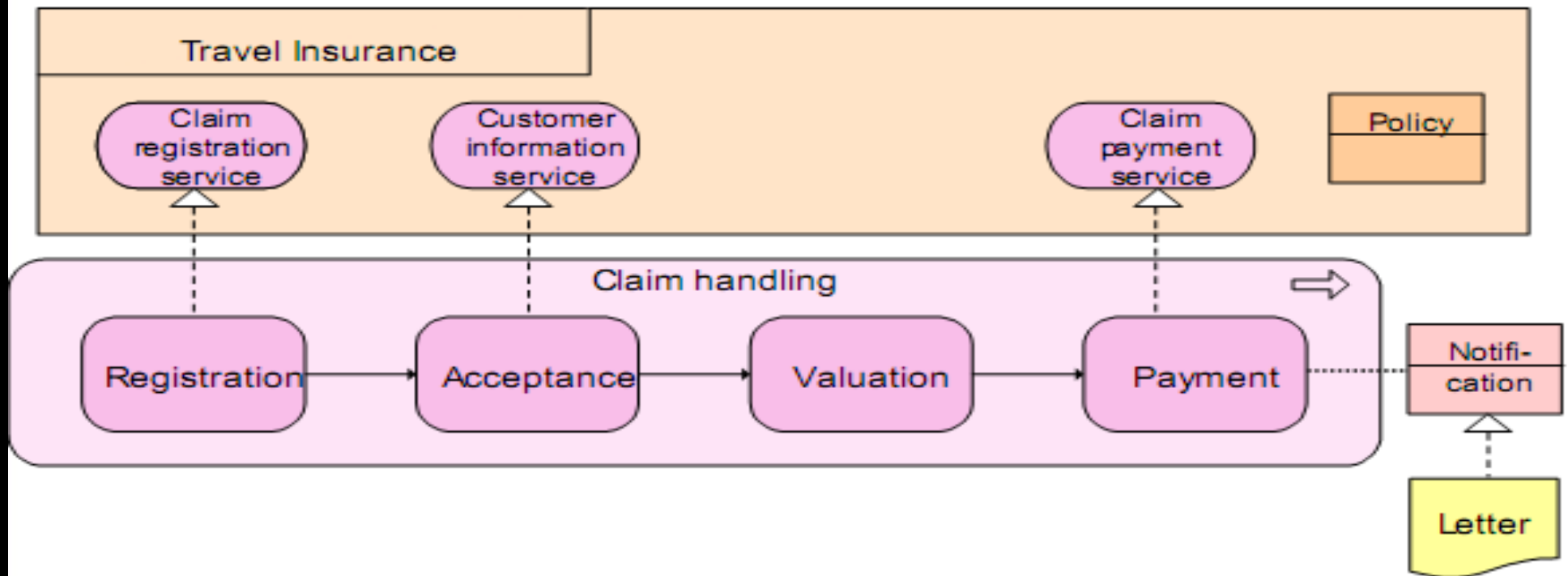
- [http://pubs.opengroup.org/architecture/archimate2-doc/chap08.html#\\_Toc371945226](http://pubs.opengroup.org/architecture/archimate2-doc/chap08.html#_Toc371945226)



***Demo***

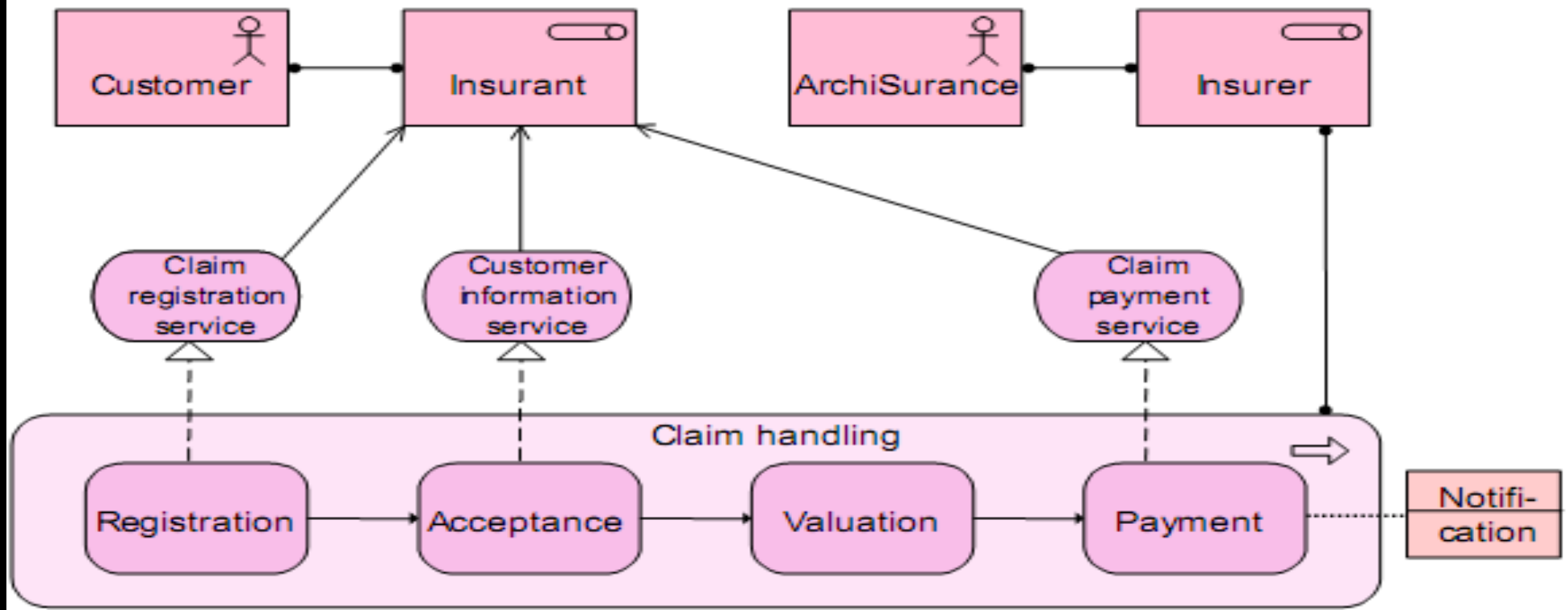
# Demo

## Product and Process



# Demo

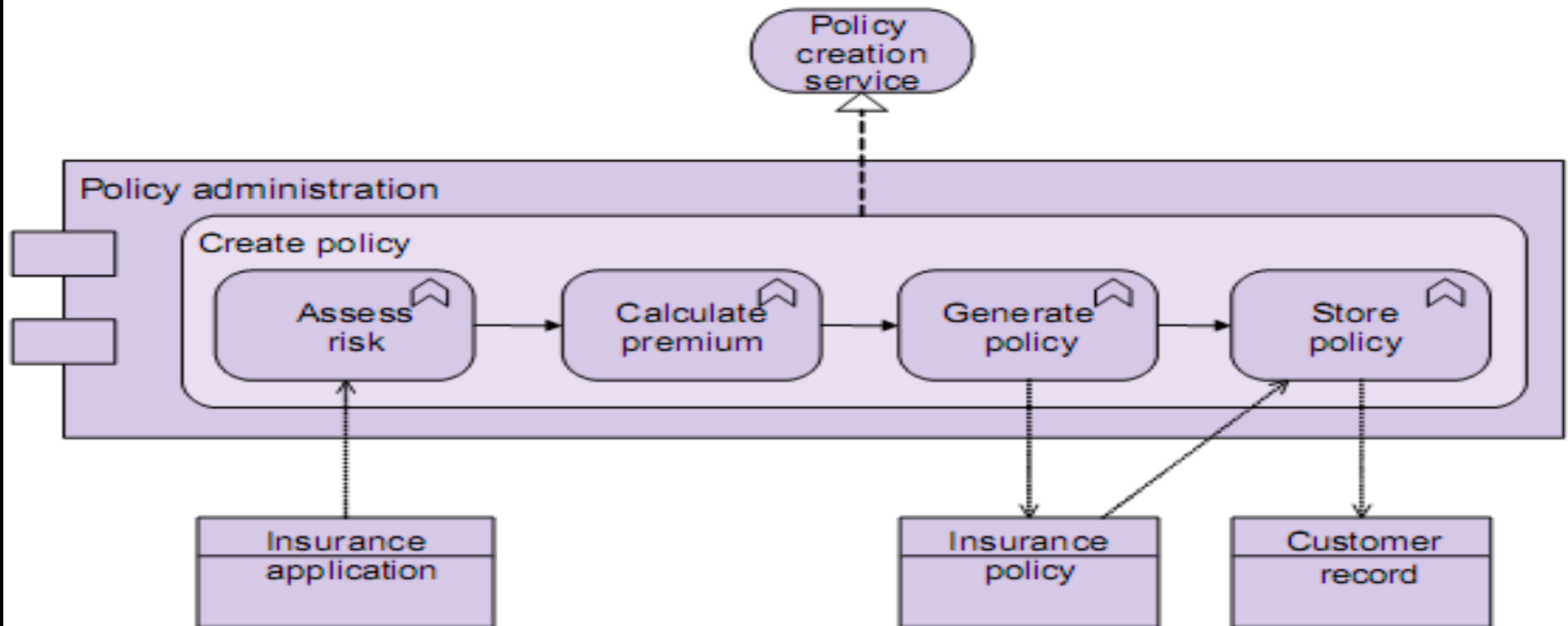
## Actors, Roles, Processes





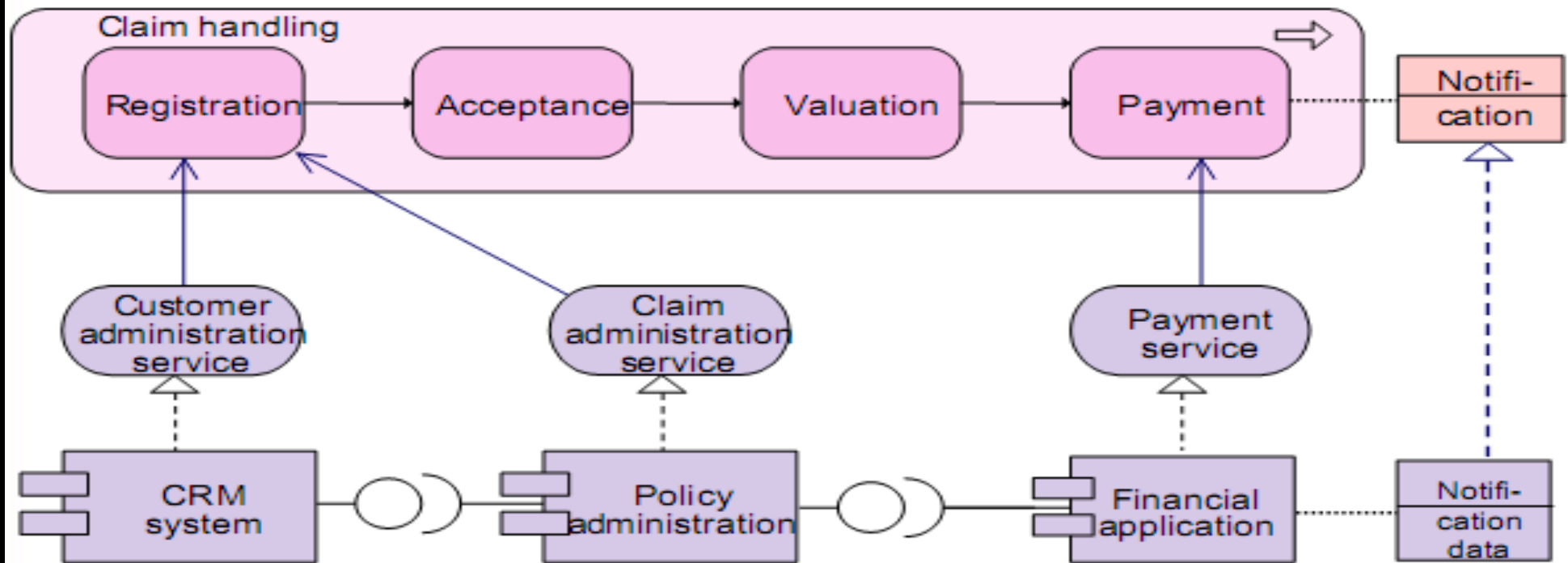
# Demo

## Application Behavior



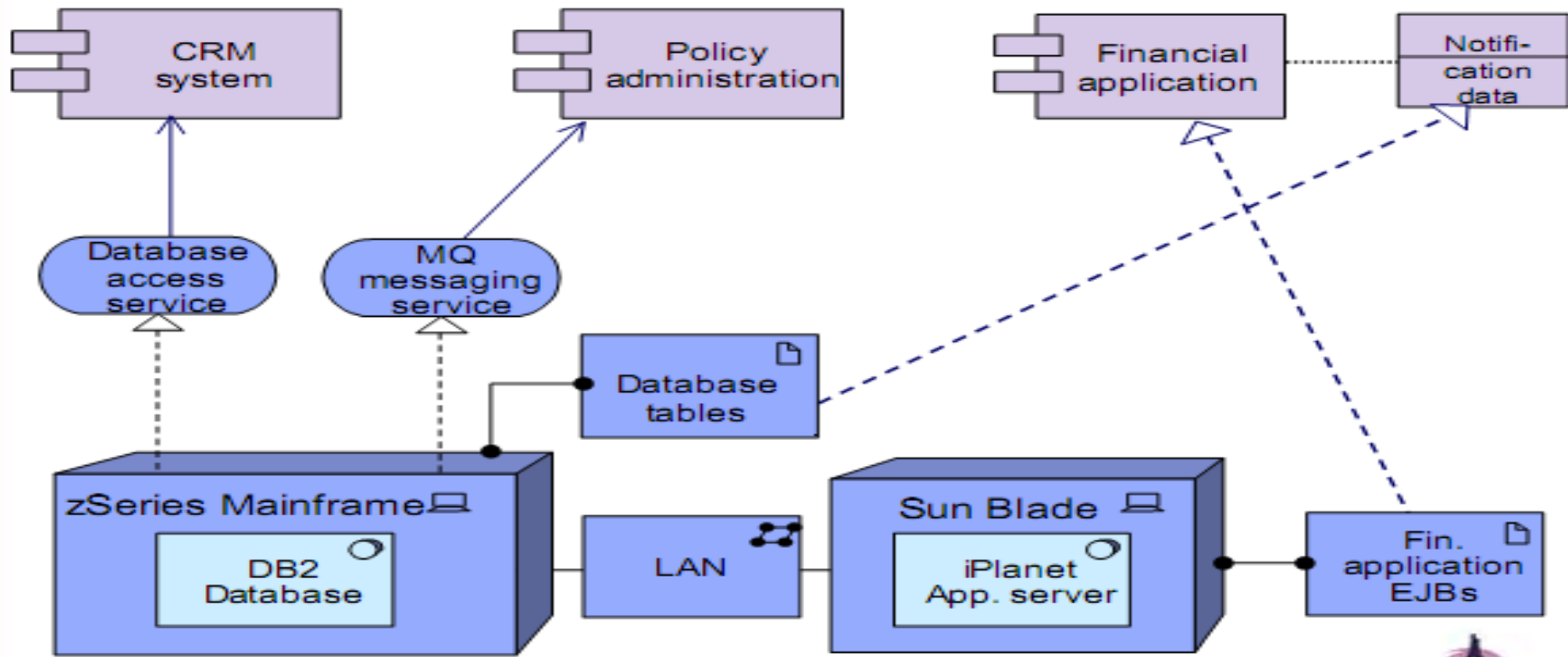
# Demo

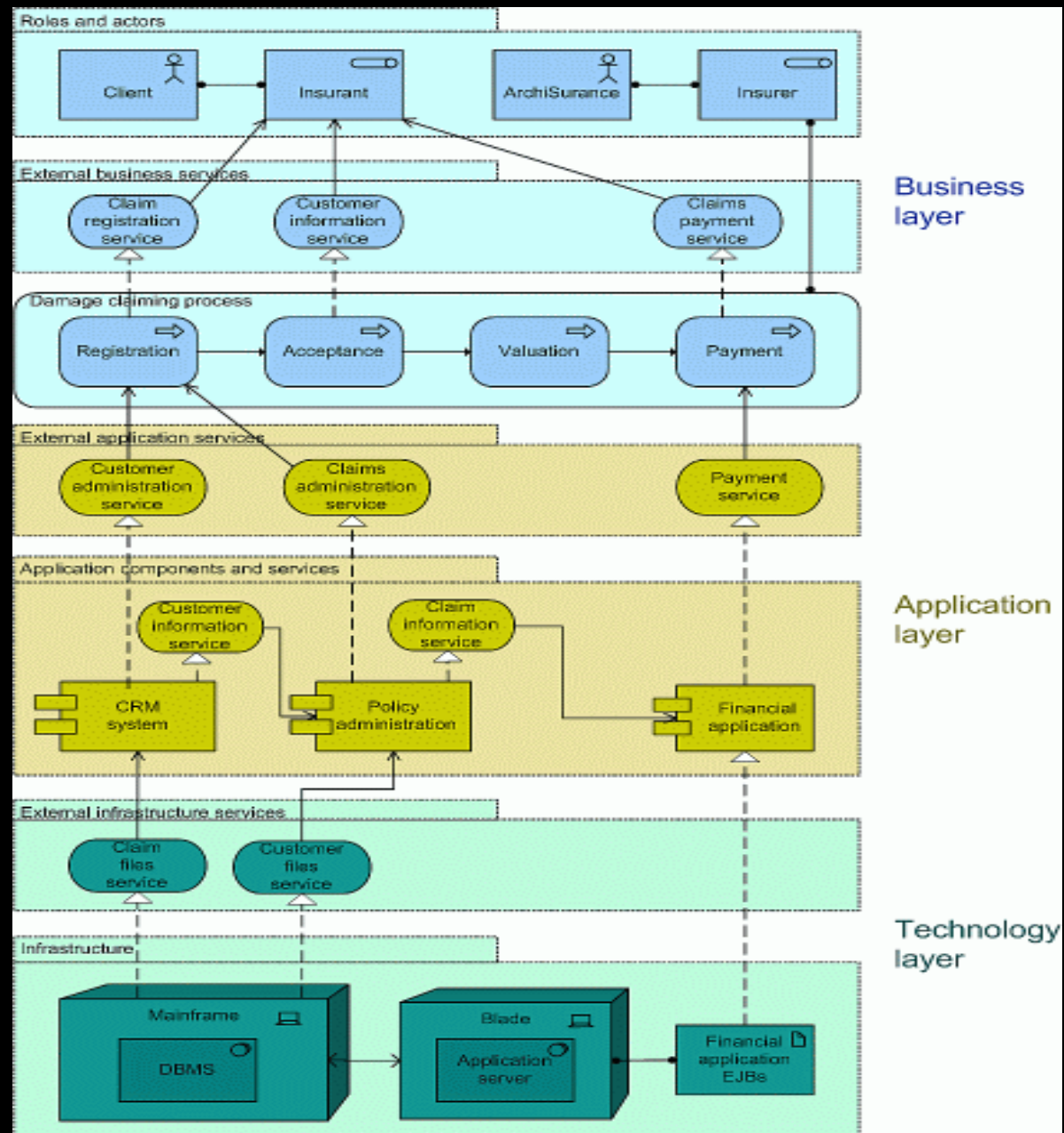
## Application Usage



# Demo

## Deployment





# More Information?

Most ArchiMate results are available through the website:

[www.archimate.org](http://www.archimate.org)

See also The Open Group's ArchiMate Forum:

[www.opengroup.org/archimate](http://www.opengroup.org/archimate)

