CSCI427/927 Systems Development

Enterprise Service Architecture

Acknowledgement: Materials on these slides are adapted from TOGAF Specification (http://www.opengroup.org/togaf) and *Modeling Enterprise Architecture with TOGAF* by Gilbert Raymond, Philippe Desfray

What is Enterprise Architecture?

- What is an Enterprise?
  - any **collection of organizations** that has a common set of goals and/or a single bottom line.
  - E.g. 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
- What is an Architecture?
  - A formal **description of a system**, or a **detailed plan** of the system at component level to guide its implementation.
  - The **structure of components**, their **inter-relationships**, and the principles and guidelines governing their design and evolution over time.

Architecture description

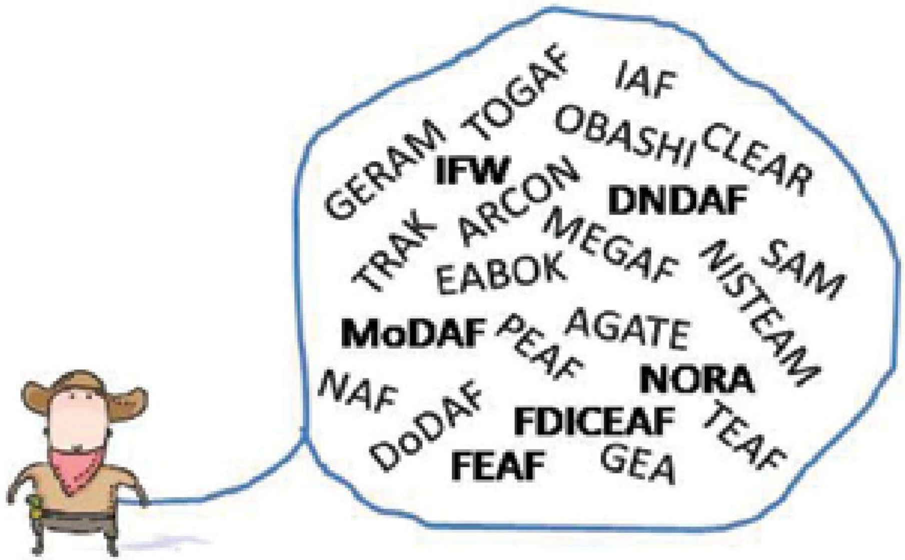
- An *architecture description*
  - is a **formal description of an information system**, organized in a way that supports reasoning about the **structural properties** of the system.
  - It defines the **components or building blocks** that make up the overall information system, and provides a plan from which products can be procured, and systems developed, that will work together to implement the overall system.
  - It thus enables you to manage your overall IT investment in a way that meets the needs of your business.

Architecture framework

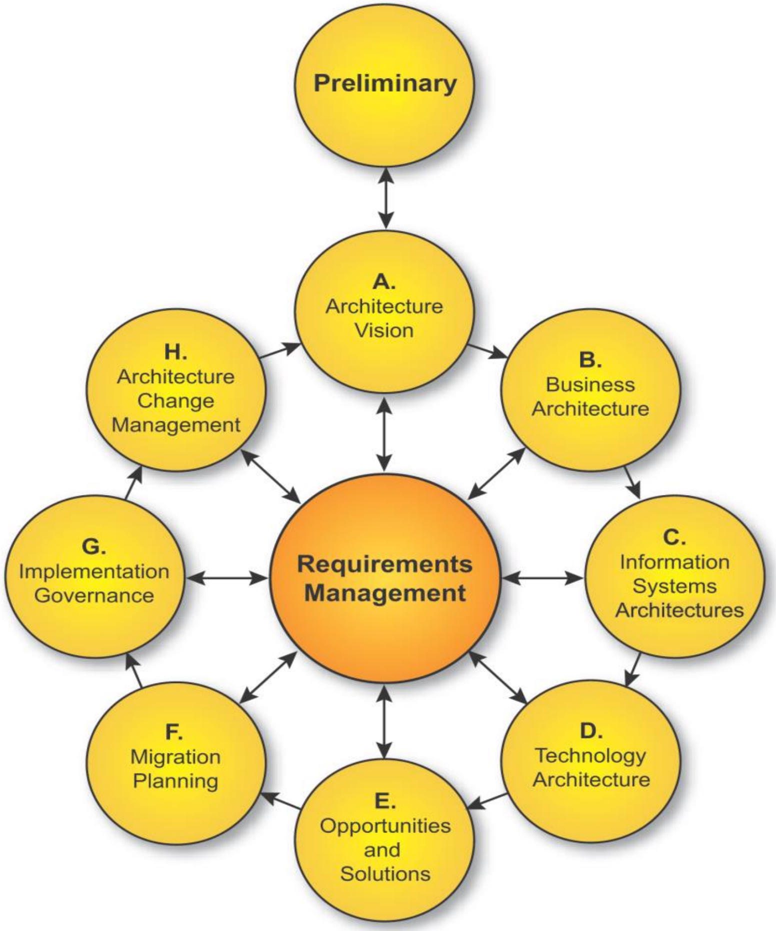
- An *architecture framework*
  - **is a tool** which can be used for developing a broad range of different architectures.
  - It should **describe a method** for designing an **information system** in terms of a set of building blocks, and for showing how the building blocks fit together.
  - It should contain **a set of tools** and **provide a common vocabulary**.
  - It should also include a list of **recommended standards** and compliant products that can be used to implement the building blocks.

Enterprise Architecture Frameworks

- **TOGAF**
- Others
  - Zachman
  - Federal Enterprise Architecture Framework
  - Etc.



TOGAF

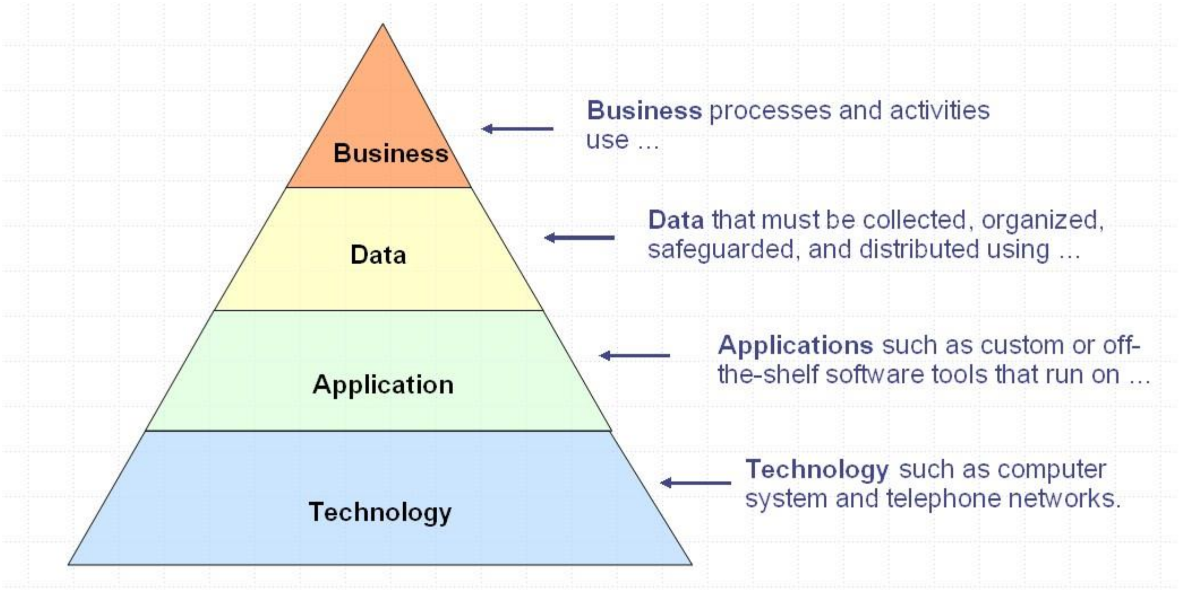


THE Open GROUP

http://www.opengroup.org

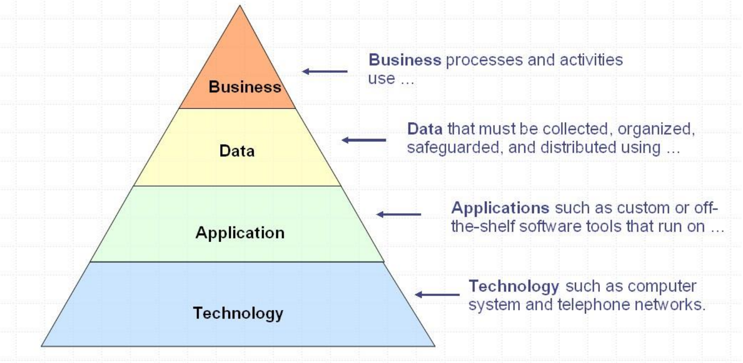
The Open Group Architecture Framework (TOGAF) is a framework – a detailed method and a set of supporting tools – for developing an enterprise architecture

Architecture Domains


Business
Data
Application
Technology
Business processes and activities use …
Data that must be collected, organized, safeguarded, and distributed using …
Applications such as custom or off-the-shelf software tools that run on …
Technology such as computer system and telephone networks.

- Business architecture
- Data architecture
- Application architecture
- Technology architecture

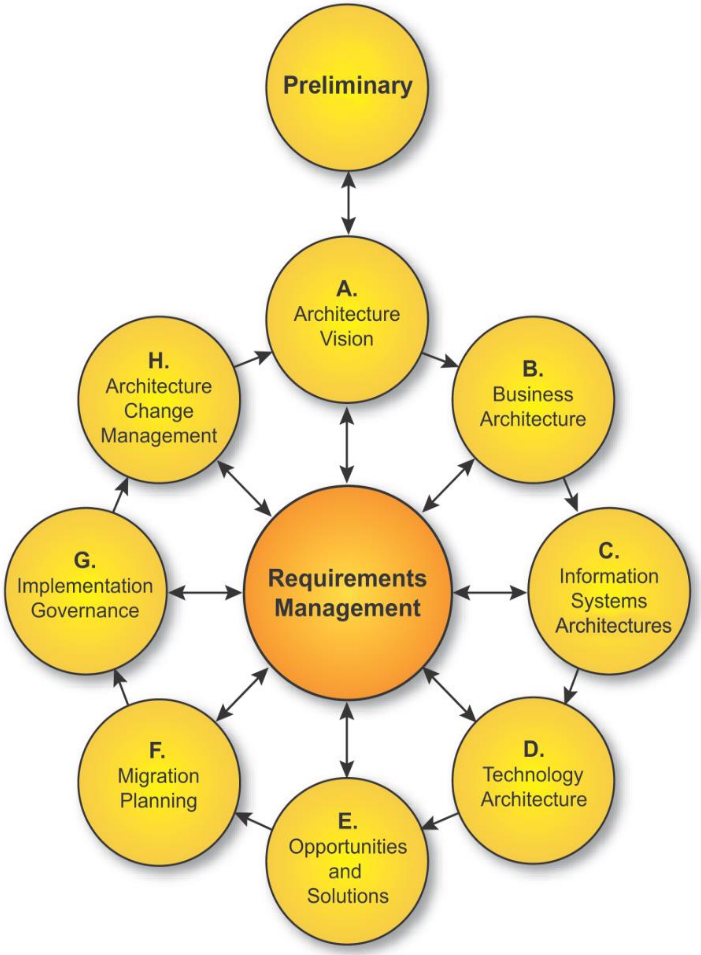
Architecture Domains


Business
Data
Application
Technology
Business processes and activities use ...
Data that must be collected, organized, safeguarded, and distributed using ...
Applications such as custom or off-the-shelf software tools that run on ...
Technology such as computer system and telephone networks.

- **Business Architecture**
  - Defines business **strategies**, governance, **organization**, and key **business processes**
- **Data architecture**
  - Describes the **structure** of an organization's logical and physical **data** assets and data management resources
- **Application architecture**
  - Describes how individual **application systems** to be deployed, their interactions, and their relationships to the core business processes of the organization.
- **Technology architecture**
  - Describes the logical **software and hardware capabilities** that are required to support the deployment of business, data, and application systems.
  - Includes IT **infrastructure**, middleware, networks, processing, communications, etc.

TOGAF Arch. Develop. Method

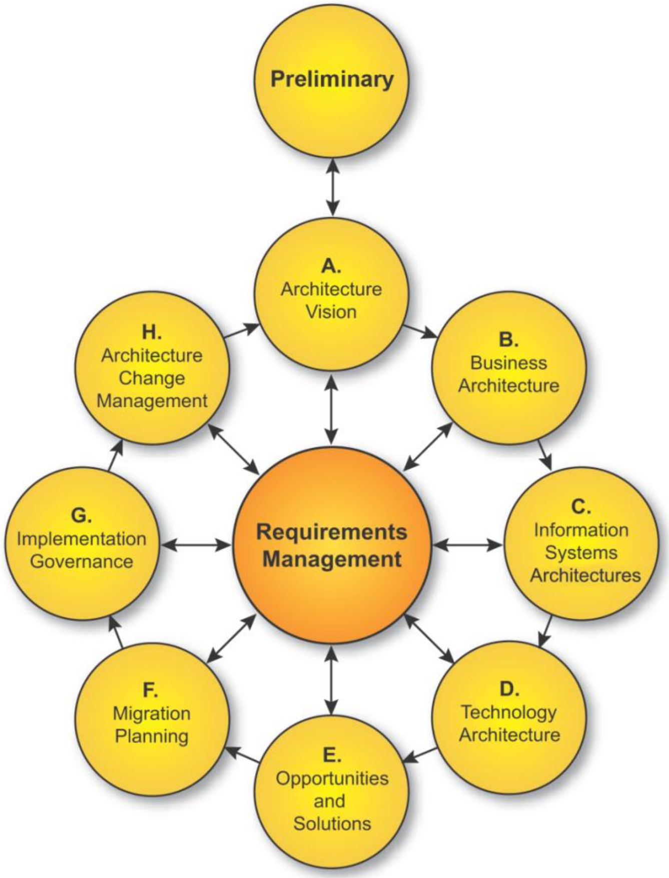
- The TOGAF *Architecture Development Method* (ADM) provides a **tested and repeatable** *process for developing architectures.*
- The ADM includes establishing an architecture framework, developing architecture content, transitioning, and governing the realization of architectures.
- All of these activities are carried out within an **iterative cycle** of **continuous** architecture definition and realization that allows organizations to transform their enterprises in a controlled manner in response to business goals and opportunities.


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

TOGAF Arch. Develop. Method

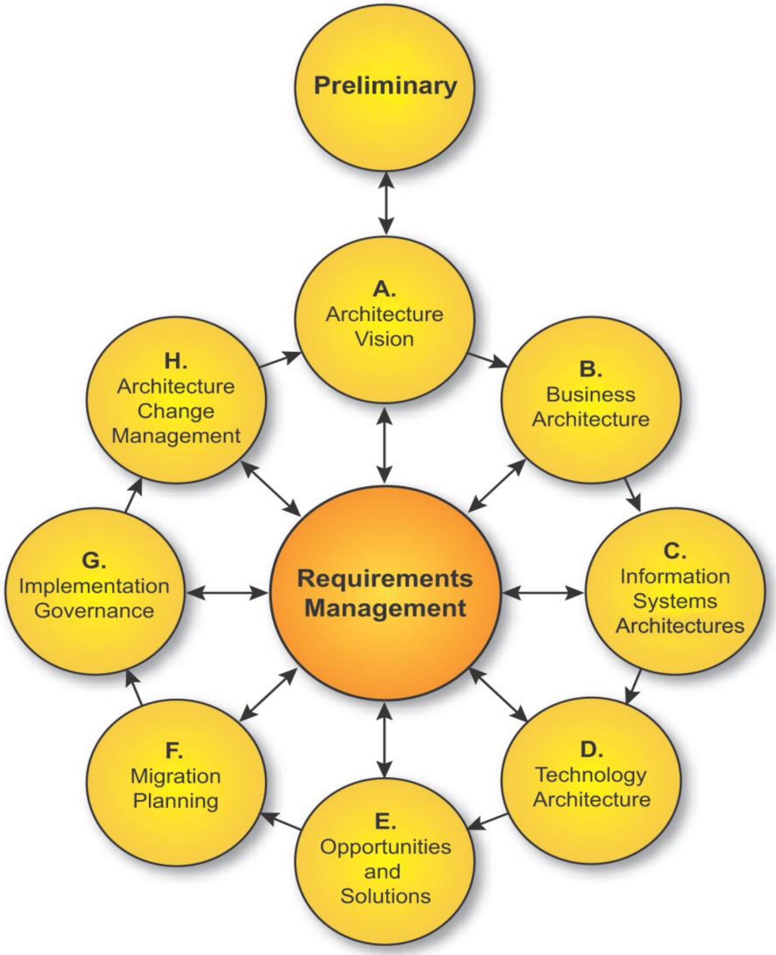
Phases within the ADM are as follows:

- The **Preliminary Phase** describes the preparation and initiation activities required to create an **Architecture Capability** including customization of TOGAF and definition of **Architecture Principles**.
- **Phase A: Architecture Vision** describes the initial phase of an architecture development cycle. It includes information about **defining the scope** of the architecture development initiative**, identifying the stakeholders**, **creating the Architecture Vision**, and obtaining approval to proceed with the architecture development.
- **Phase B: Business Architecture** describes the development of a Business Architecture to support the agreed Architecture Vision.


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

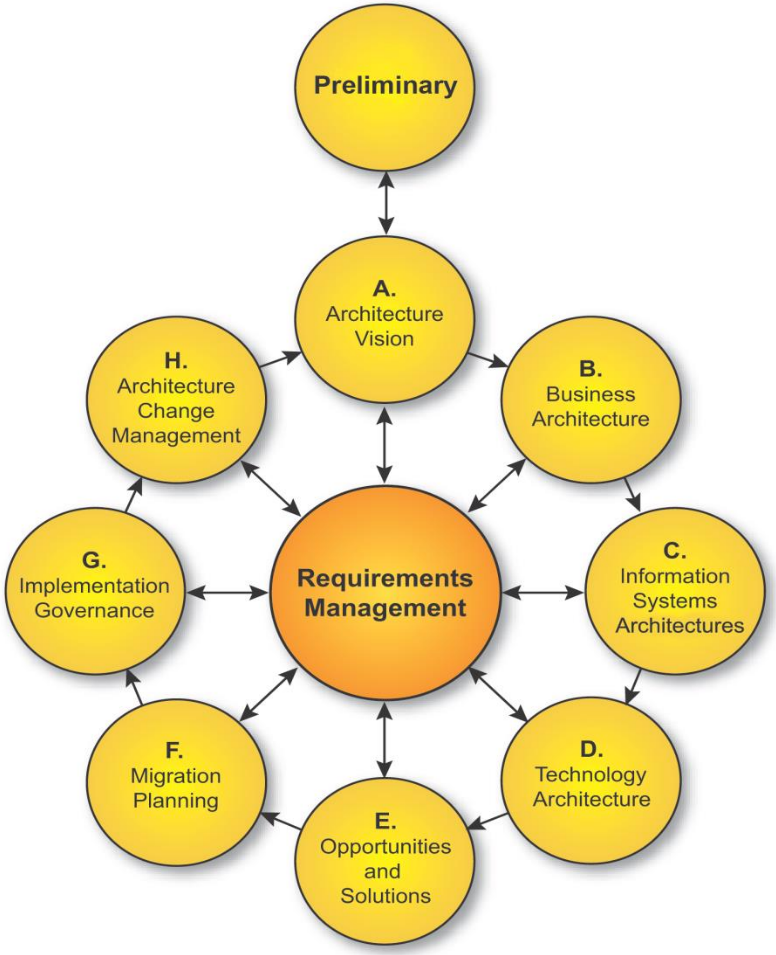
TOGAF Arch. Develop. Method

- **Phase C: Information Systems Architectures** describes the development of Information Systems Architectures to support the agreed Architecture Vision.
- **Phase D: Technology Architecture** describes the development of the Technology Architecture to support the agreed Architecture Vision.
- **Phase E: Opportunities & Solutions** conducts **initial implementation planning** and the identification of delivery vehicles for the architecture defined in the previous phases.
- **Phase F: Migration Planning** addresses how to **move from the Baseline to the Target Architectures** by finalizing a detailed Implementation and Migration Plan.


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

TOGAF Arch. Develop. Method

- **Phase G: Implementation Governance** provides an architectural oversight of the implementation.
- **Phase H: Architecture Change Management** establishes procedures for managing change to the new architecture.
- **Requirements Management** examines the process of managing architecture requirements throughout the ADM.


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

References

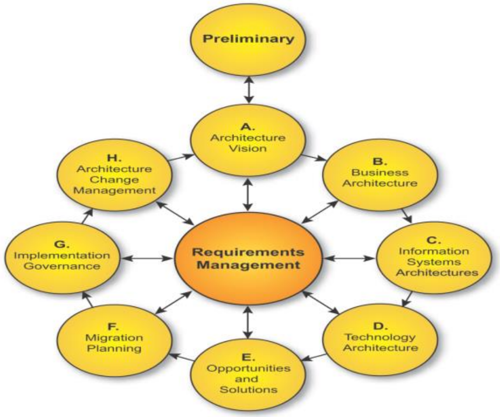
- TOGAF at The Open Group https://www.opengroup.org/togaf
- Modeling Enterprise Architecture with TOGAF by Gilbert Raymond, Philippe Desfray

ArchiMate

Acknowledgement: Materials on these slides are adapted from ArchiMate Specification (www.opengroup.org/archimate)

What is ArchiMate?

- ArchiMate enterprise architecture (EA) modelling language.
- It is an OpenGroup standard.
- ArchiMate language, complements TOGAF in that it provides a vendor-independent set of concepts, including a graphical representation.


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

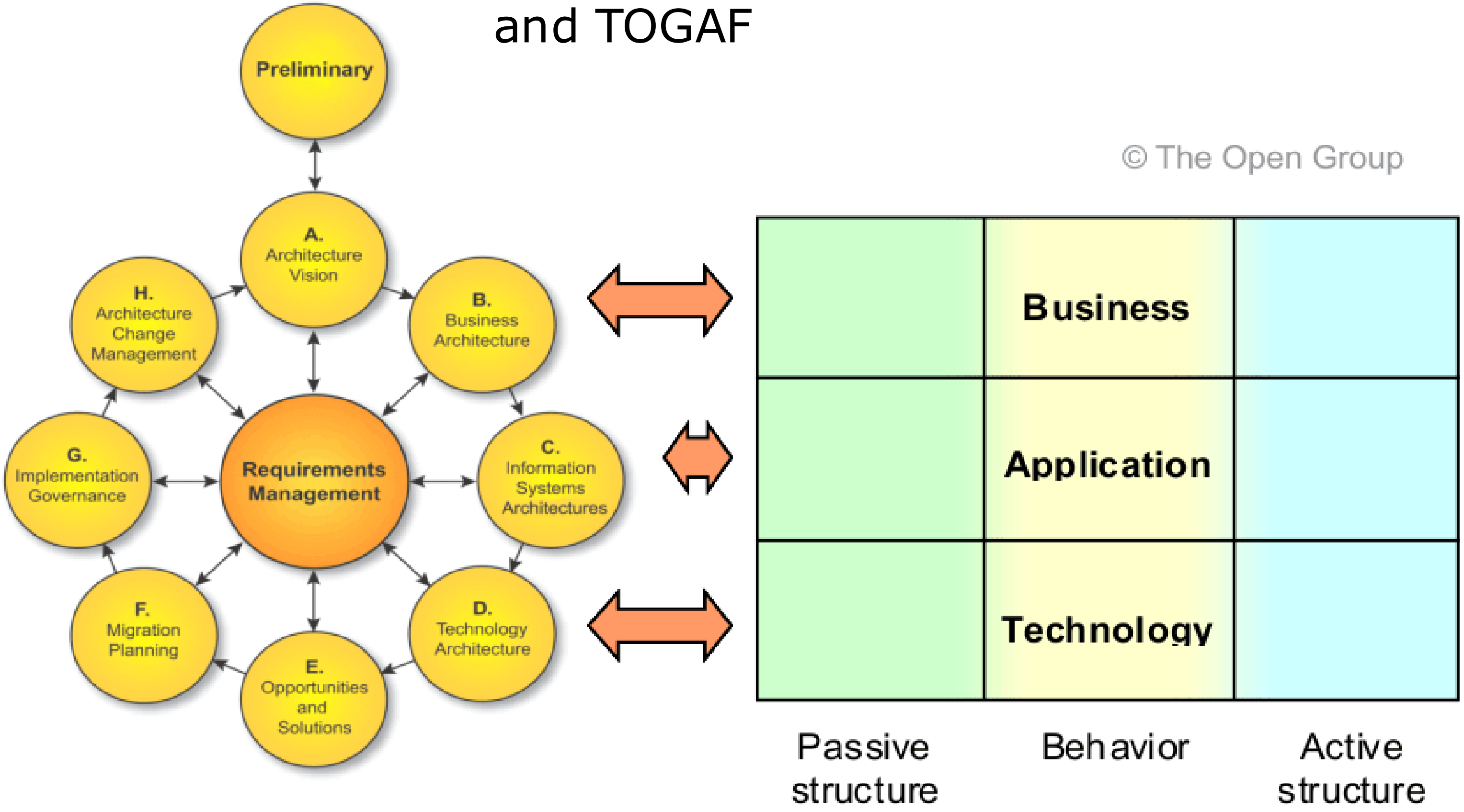
THE Open GROUP

ArchiMate

- The ArchiMate language defines **3 main layers**
  - *Business Layer* offers **products** and **services** to external customers, which are realized in the organization by **business processes** performed by business actors.
  - *Application Layer* supports the business layer with **application services** which are realized by (software) applications.
  - *Technology Layer* offers **infrastructure services** (e.g., processing, storage, and communication services) needed to run applications, realized by computer and communication hardware and system software.

ArchiMate & TOGAF

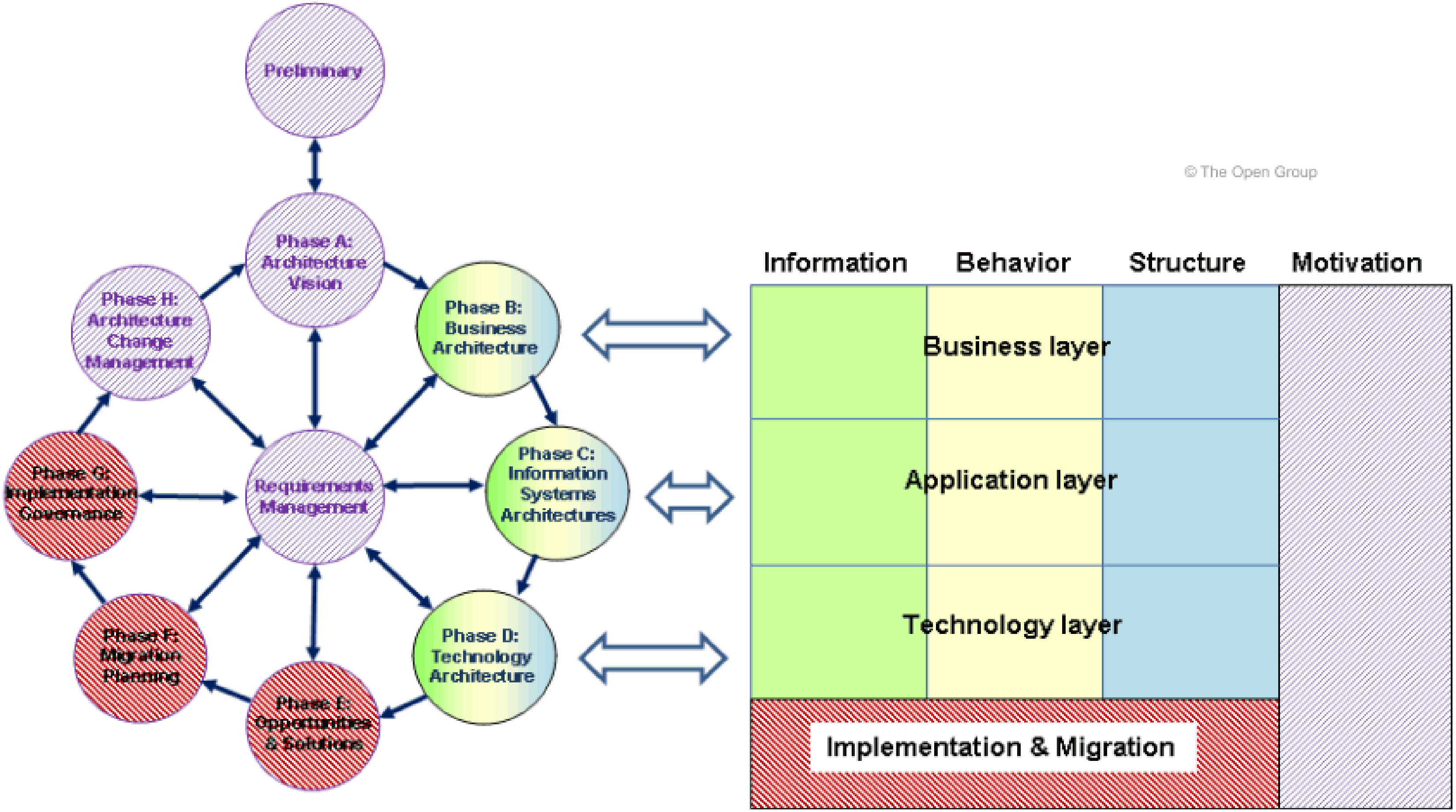
Correspondence between ArchiMate and TOGAF



© The Open Group

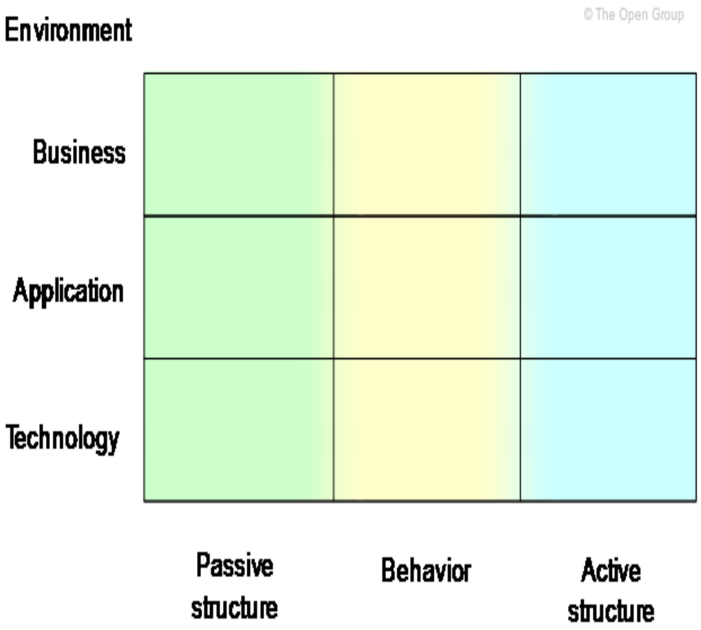
TOGAF ADM **ArchiMate**

ArchiMate & TOGAF (cont.)


Preliminary
Phase A: Architecture Vision
Phase B: Business Architecture
Phase C: Information Systems Architectures
Phase D: Technology Architecture
Phase E: Opportunities & Solutions
Phase F: Migration Planning
Phase G: Implementation Governance
Phase H: Architecture Change Management
Requirements Management
© The Open Group
Information
Behavior
Structure
Motivation
Business layer
Application layer
Technology layer
Implementation & Migration

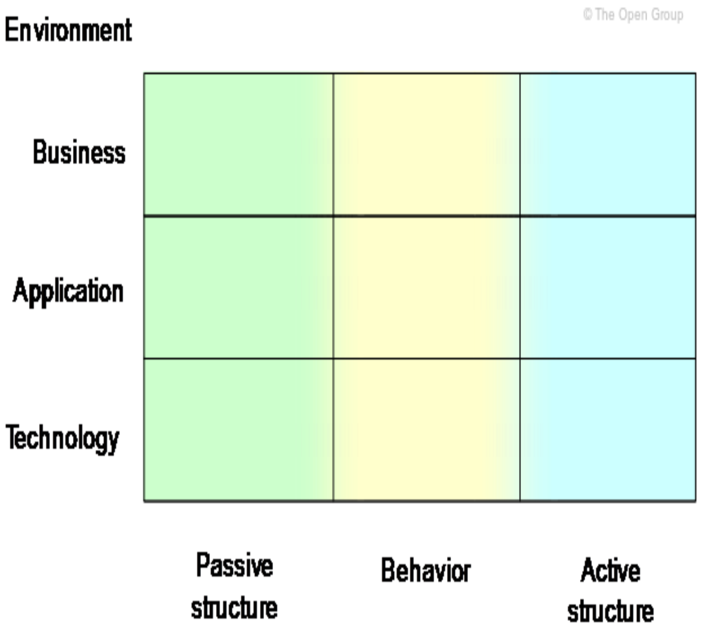
ArchiMate

- Modeling of EA from **3 different viewpoints**:
  - **Active structure aspect** represents the **structural concepts** (the business actors, application components, and devices that display actual behavior; i.e., the “subjects” of activity).
  - **Behavior aspect** represents the behavior (processes, functions, events, and services) performed by the actors. Behavioral concepts are assigned to structural concepts, to show who or what displays the behavior.



ArchiMate

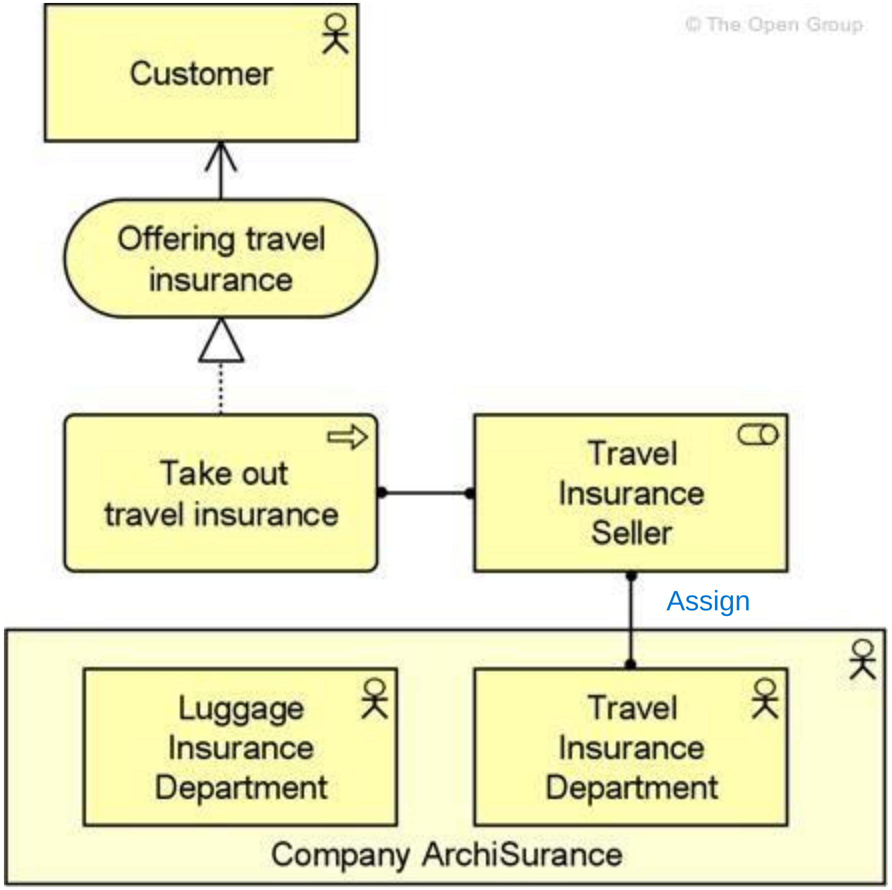
- Modeling of EA from 3 different viewpoints:
  - **Passive structure aspect** represents the **objects on which** behavior is performed. These are usually **information objects** in the business layer and **data objects** in the application layer, but they may also be used to represent **physical objects**.



Business Layer

- Active Structure Concepts
  - The **static structure** of an organization, in terms of the **(active) 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 Actor, Business Role, Business Collaboration, Business Interface, and Location

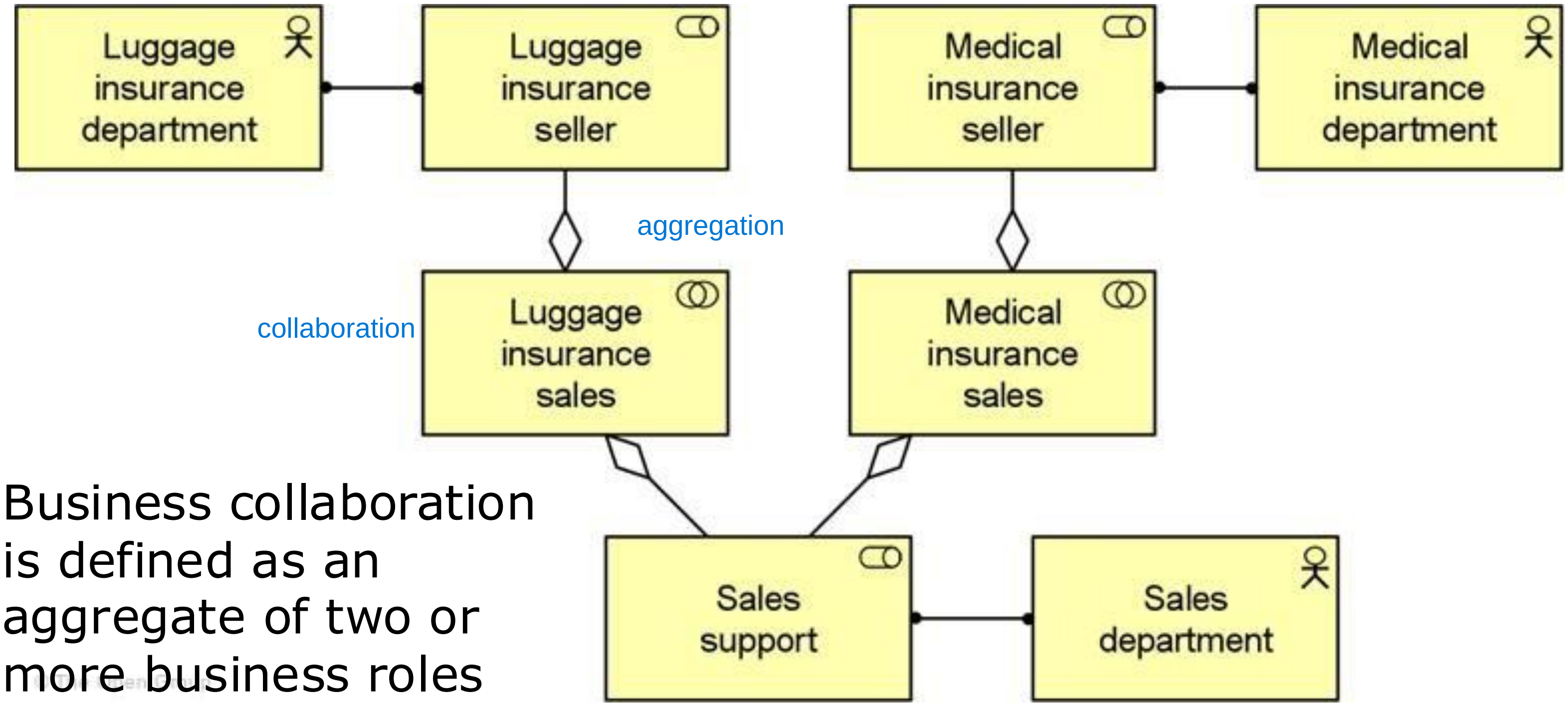
Business actor & business role


© The Open Group
Customer
Offering travel insurance
Take out travel insurance
Travel Insurance Seller
Assign
Luggage Insurance Department
Travel Insurance Department
Company ArchiSurance

A business actor is defined as an organizational entity that is capable of performing behavior.

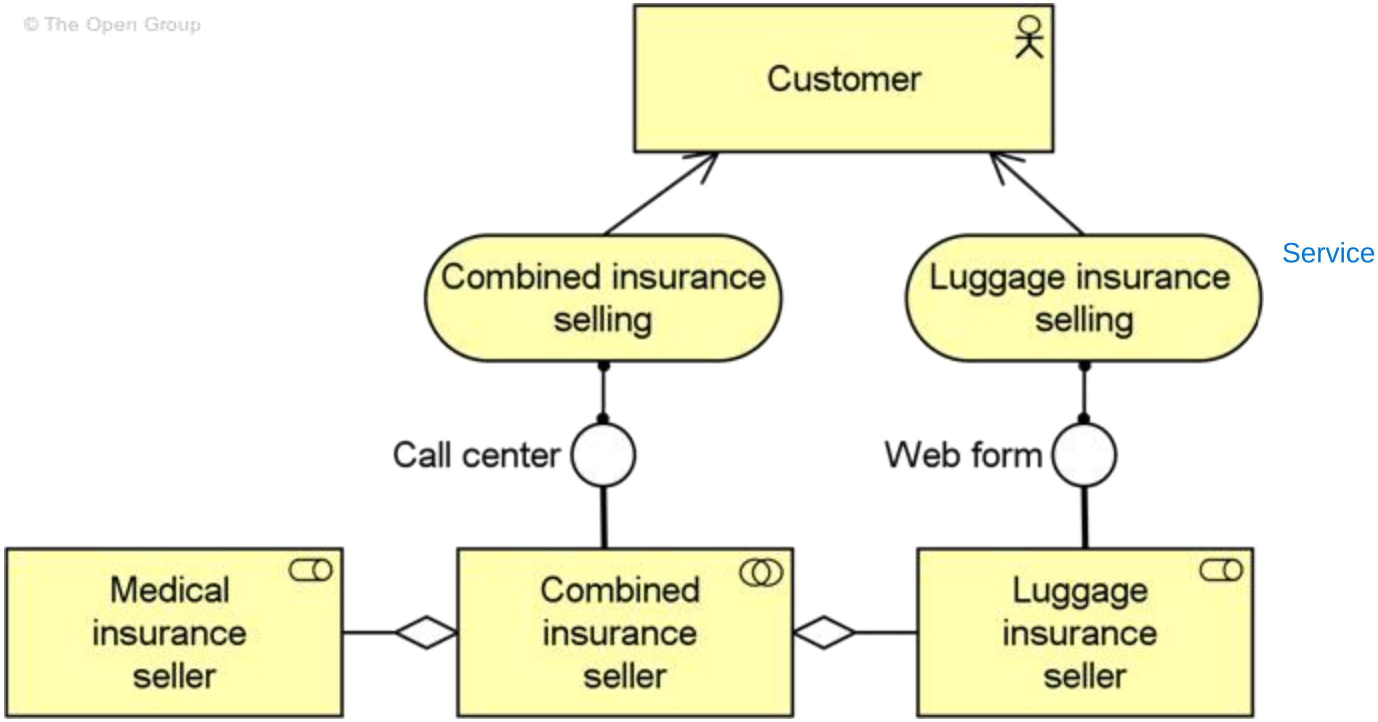
A business role is defined as the responsibility for performing specific behavior, to which an actor can be assigned.

Business collaboration



Business collaboration is defined as an aggregate of two or more business roles that work together to perform collective behavior.

Business interface

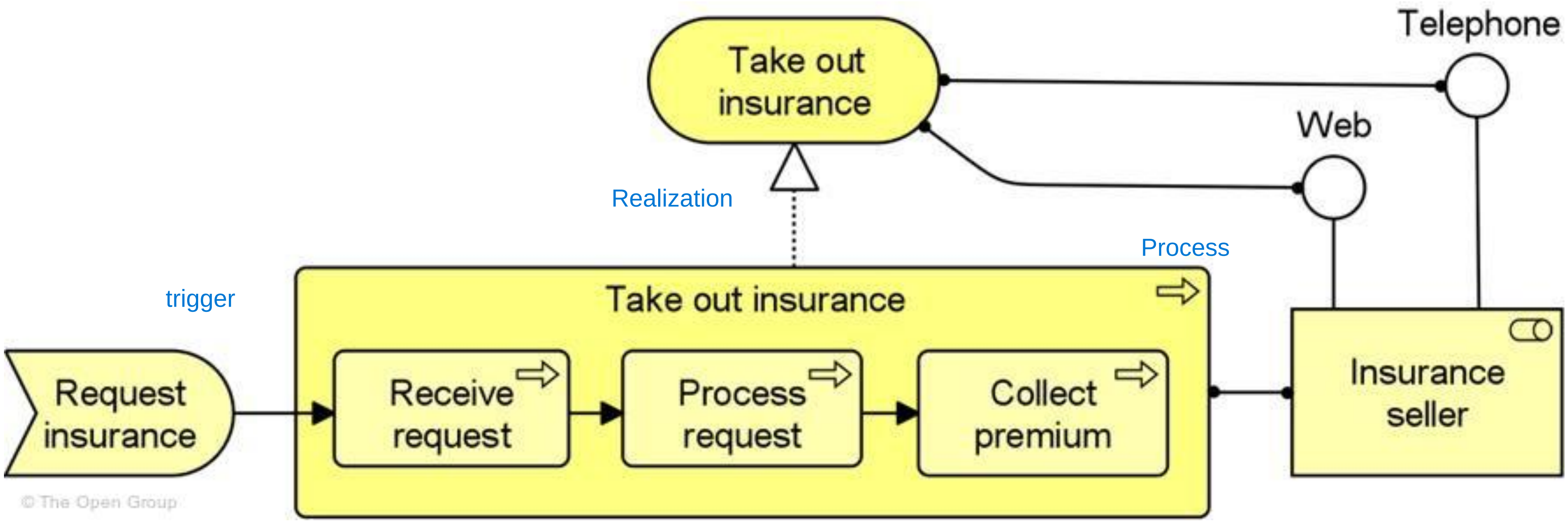


A business interface exposes the functionality of a business service to other business roles (provided interface), or expects functionality from other business services (required interface)

Business Layer

- Behavioral Concepts
  - Distinction between “external” and “internal” behavior of an organization.
  - 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
  - Business Process, Business Function, Business Interaction, Business Event, and Business Service

Business Process



Realization

Process

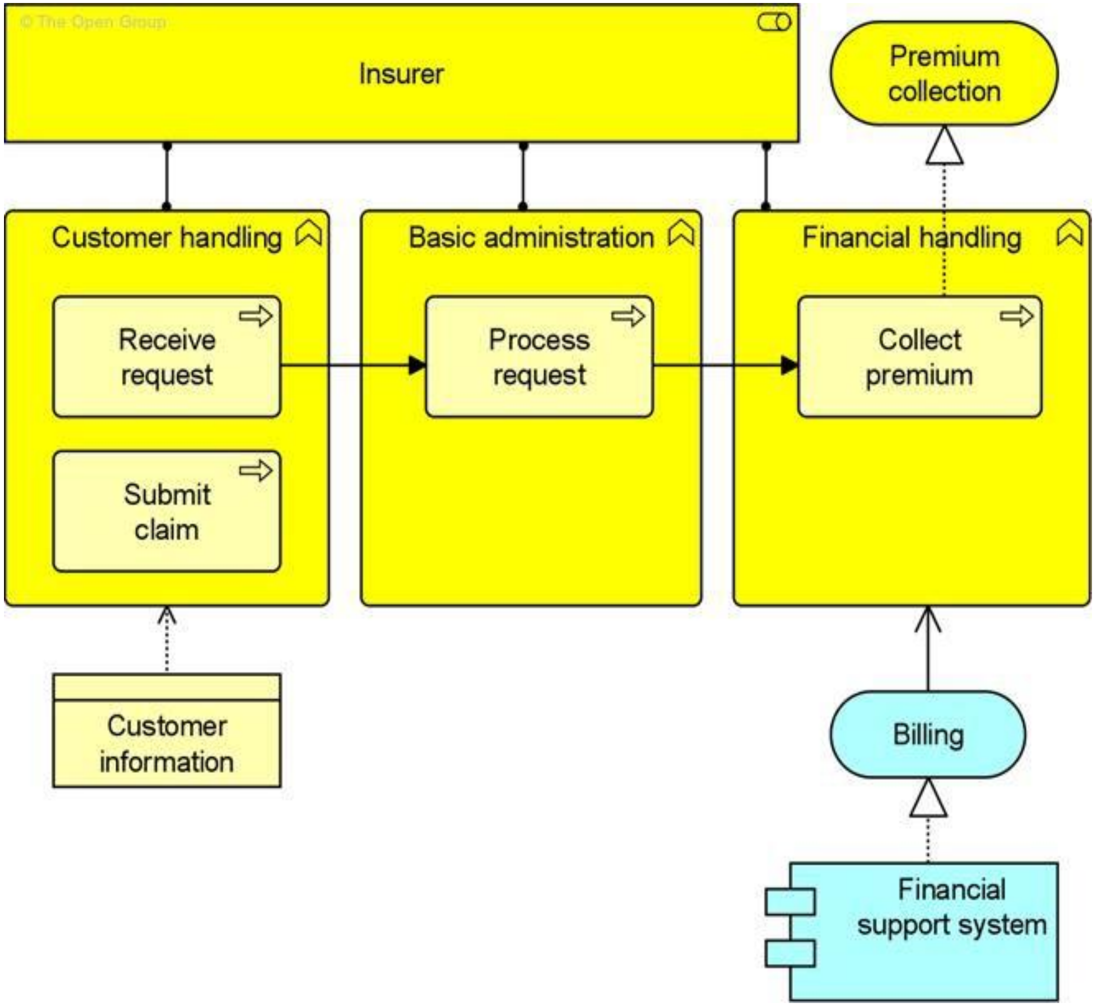
trigger

© The Open Group

A business process describes the internal behavior performed by a business role that is required to produce a set of products and services

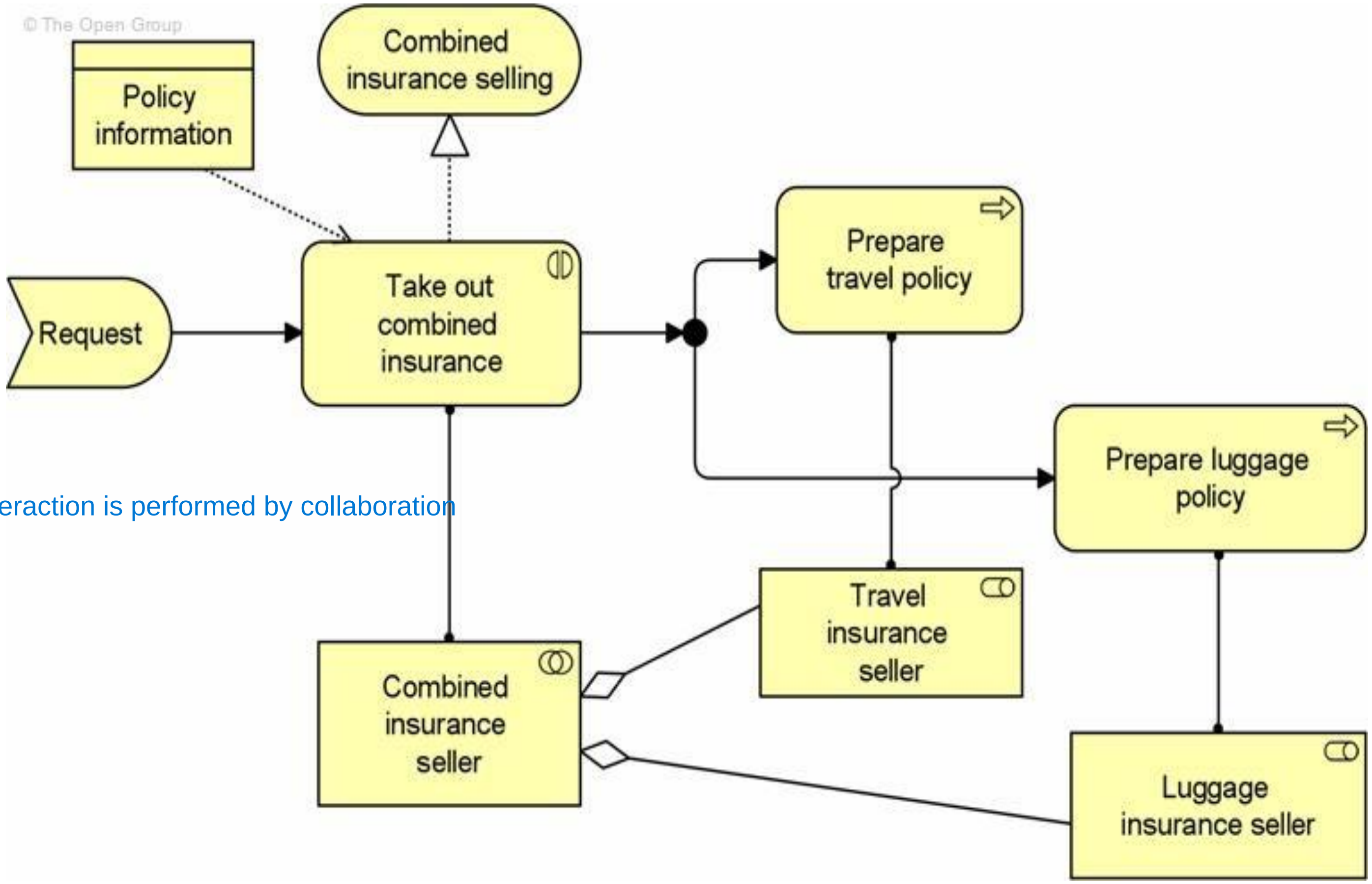
Business Function

Function is a group of process



A business function 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.

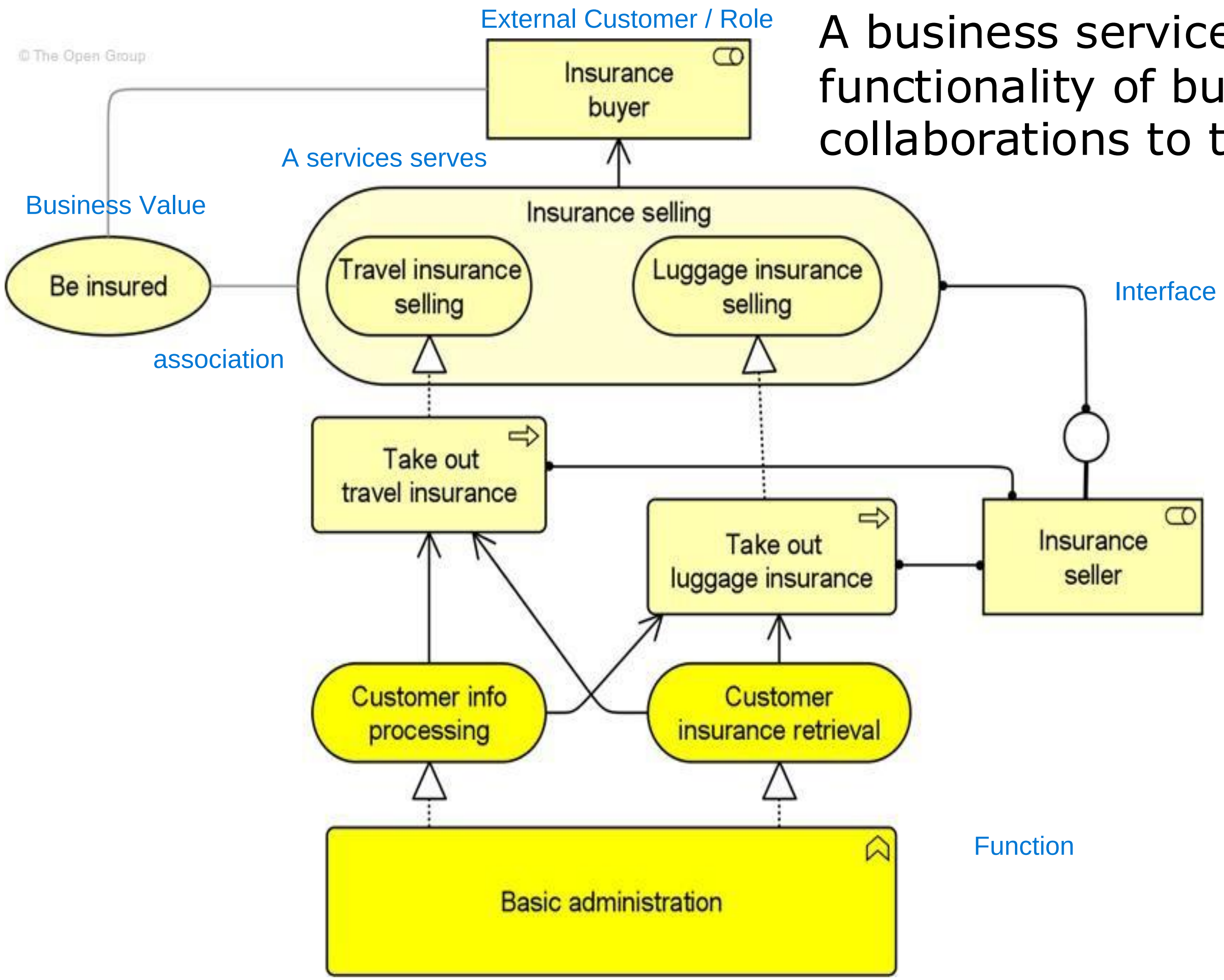
Business Interaction



interaction is performed by 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.**

Business Service



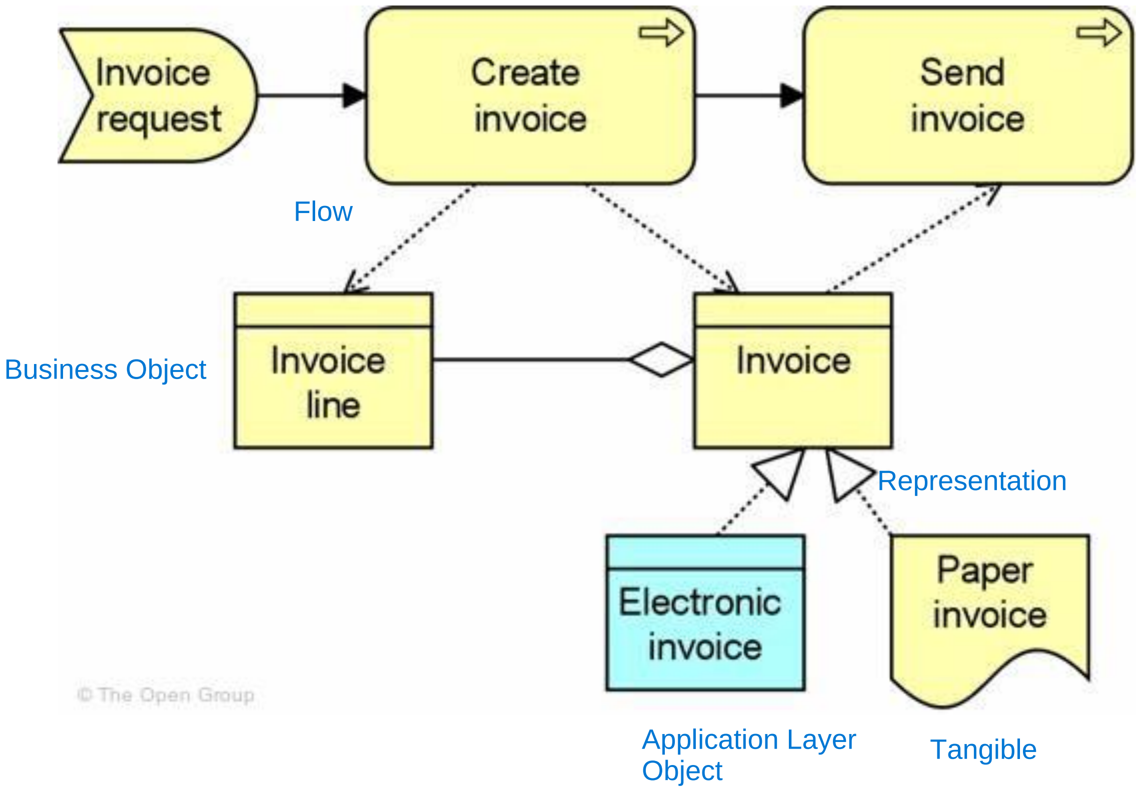
A business service exposes the functionality of business roles or collaborations to their environment.

Business services can be external, customer-facing services (e.g., a travel insurance service) or internal support services (e.g., a resource management service).

Business Layer

- Passive Structure Concepts
  - Model the passive entities 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.
  - Business Object, Representation, Meaning, Value, Product, Contract

Passive Structure Objects



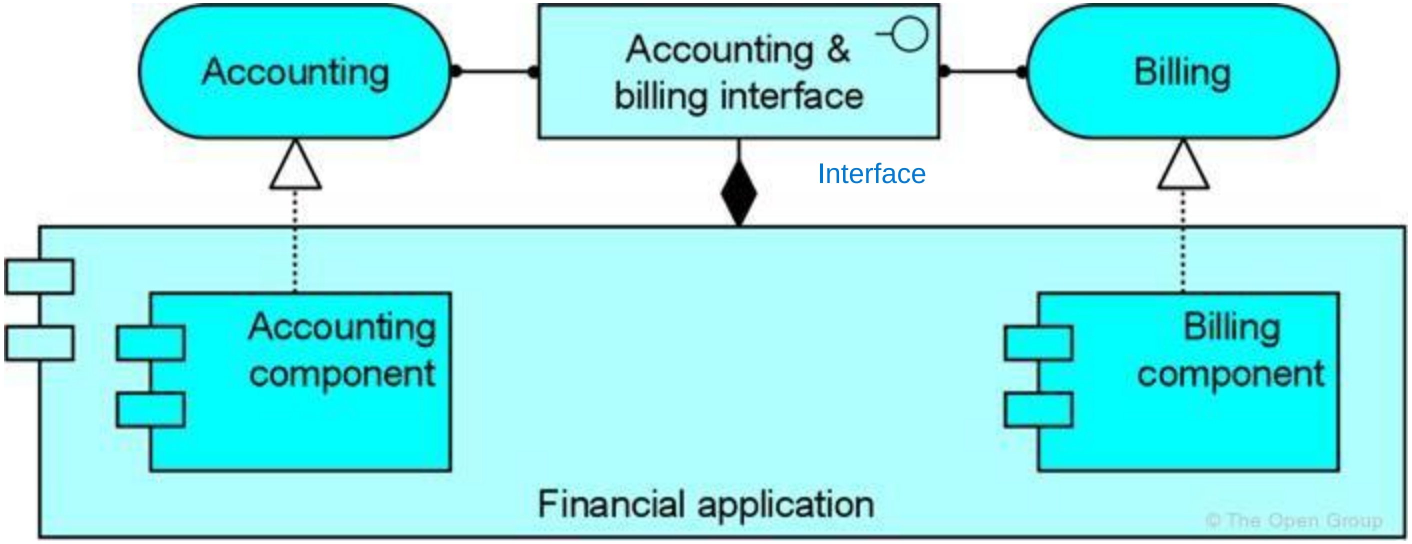
Business objects represent the important “informational” or “conceptual” elements in which the business thinks about a domain

Representations (for example, messages or documents) are the perceptible carriers of information that are related to business objects.

Application Layer

- Active Structure concepts:
  - The main active structure concept for the application layer is the **application component.**
  - The inter-relationships of components are an essential ingredient => **application collaboration.**
  - Application interface
    - Application-to-application interface
    - Application-to-business interface

Application component



Interface

Encapsulation

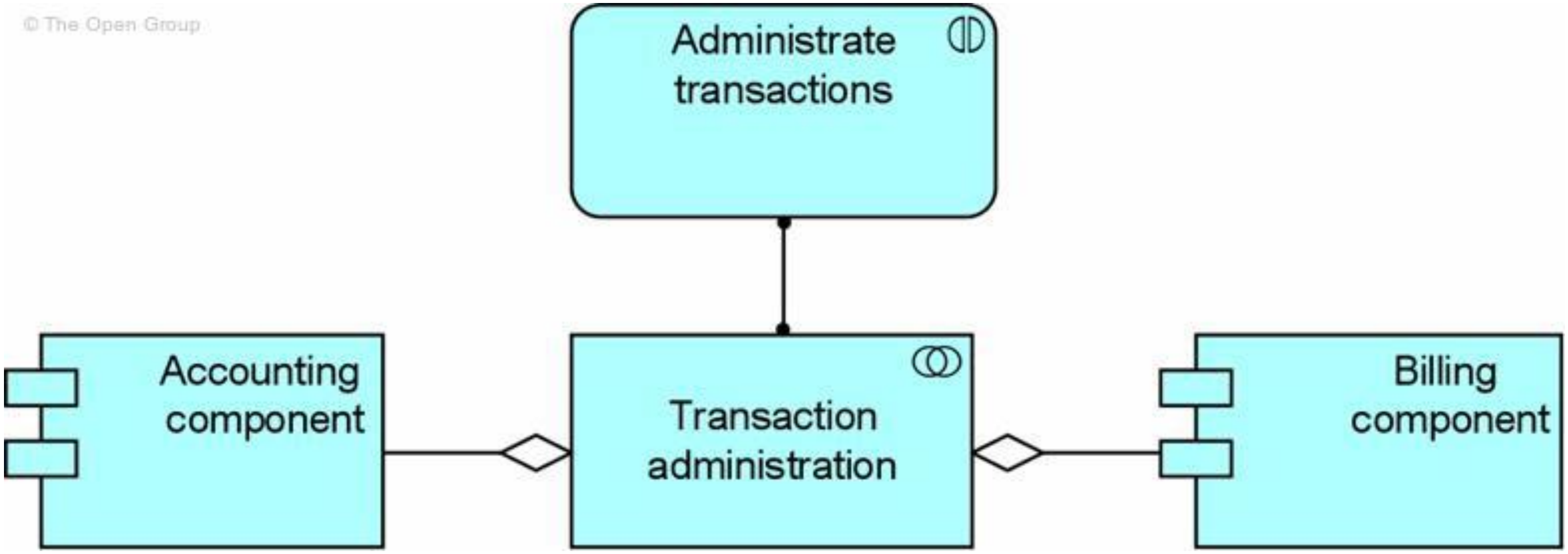
An application component is a **self-contained unit** of functionality.

An application component may be assigned to one or more **application functions**, **business processes**, or **business functions**.

An application component has one or more **application interfaces**, which expose its functionality.

Application interfaces of other application components may be used by an application component.

Application collaboration

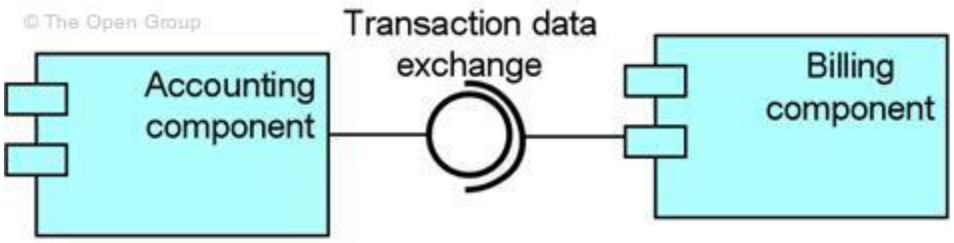


An application collaboration specifies which **components co-operate** to perform some task.

An application collaboration may be assigned to one or more **application interactions** or **business interactions**, which model the associated behavior.

An **application interface** may be used by an application collaboration, and an application collaboration may be composed of application interfaces.

Application interface



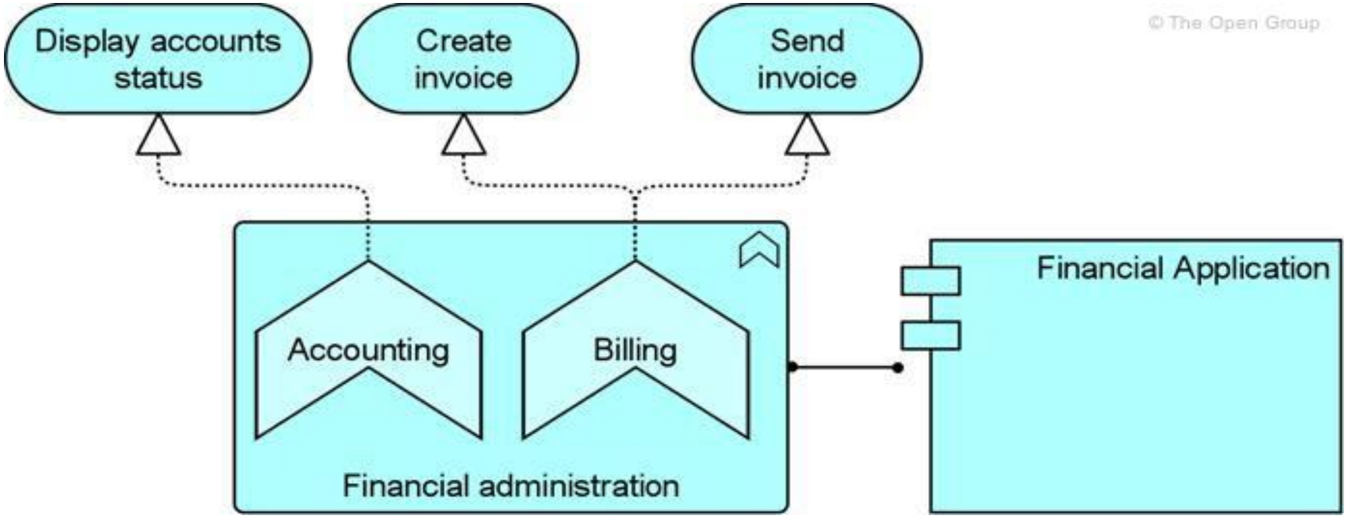
An application interface specifies how the functionality of a component can be accessed by other components (**provided interface**), or which functionality the component requires from its environment (**required interface**)

An application interface can be assigned to **application services** or **business services**, which means that the interface exposes these services to the environment.

Application Layer

- Behaviour concepts
  - Described in a way that is very similar to business layer behavior.
  - Make a distinction between the external behavior of application components in terms of *application services*, and the internal behavior of these components; i.e., *application functions* that realize these services
  - Application service, application function, application interaction

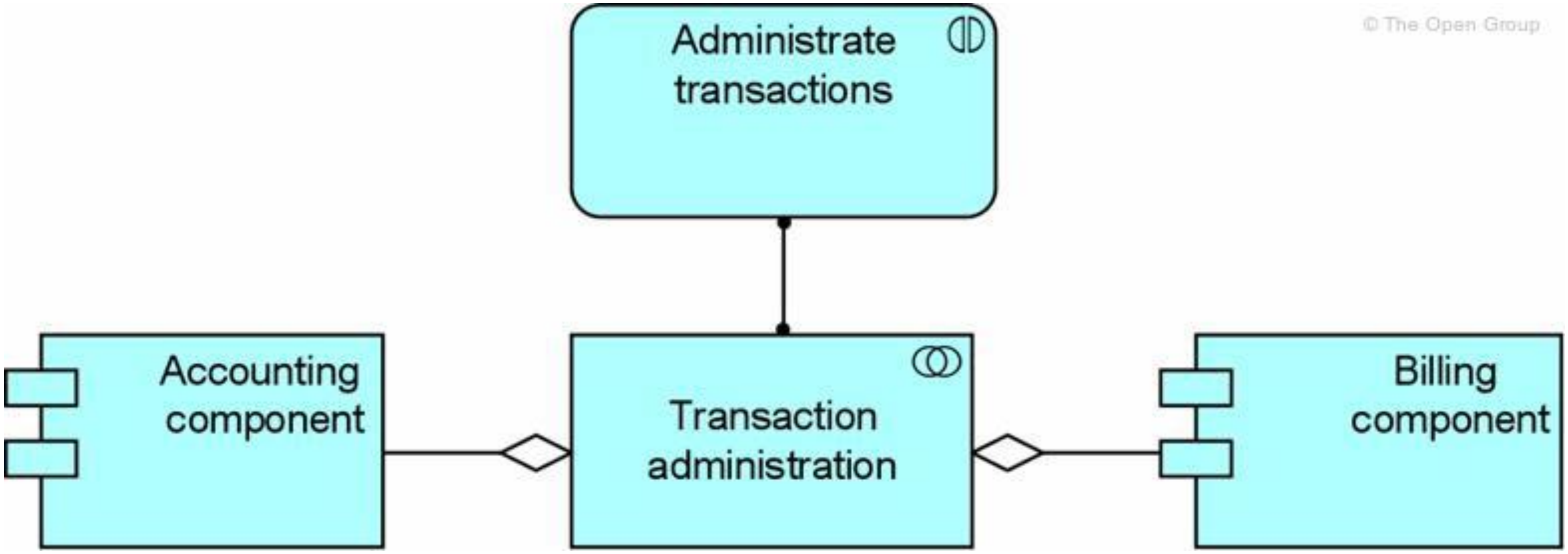
Application function



An application function describes the **internal behavior** of an application component.

An application function may realize one or more **application services**. Application services of other application functions may be used by an application function.
An application function may access **data objects**.
An **application component** may be assigned to an application function (which means that the application component performs the application function).

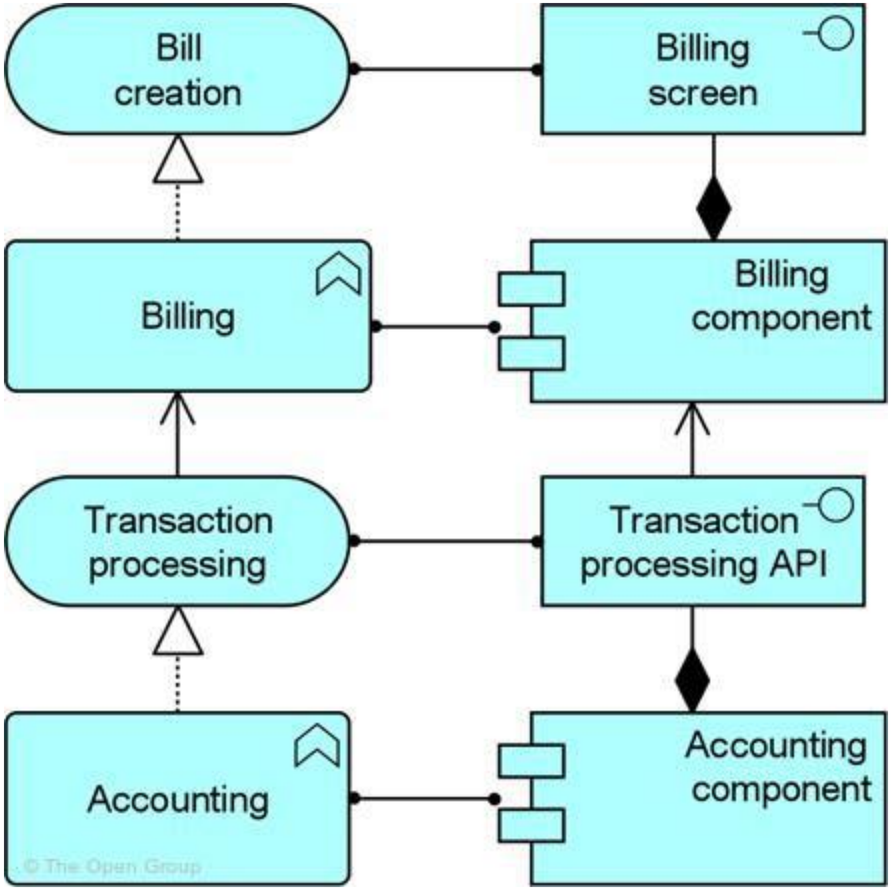
Application interaction



An application interaction describes the collective behavior that is performed by the components that participate in an **application collaboration**

An application collaboration may be assigned to an application interaction. An application interaction may realize an **application service**. Application services and infrastructure services may be used by an application interaction. An application interaction may access **data objects**.

Application service



An application service exposes the functionality of components to their environment (i.e. **external behaviour**)

An application service may be used by **business processes**, **business functions**, **business interactions**, or **application functions**.

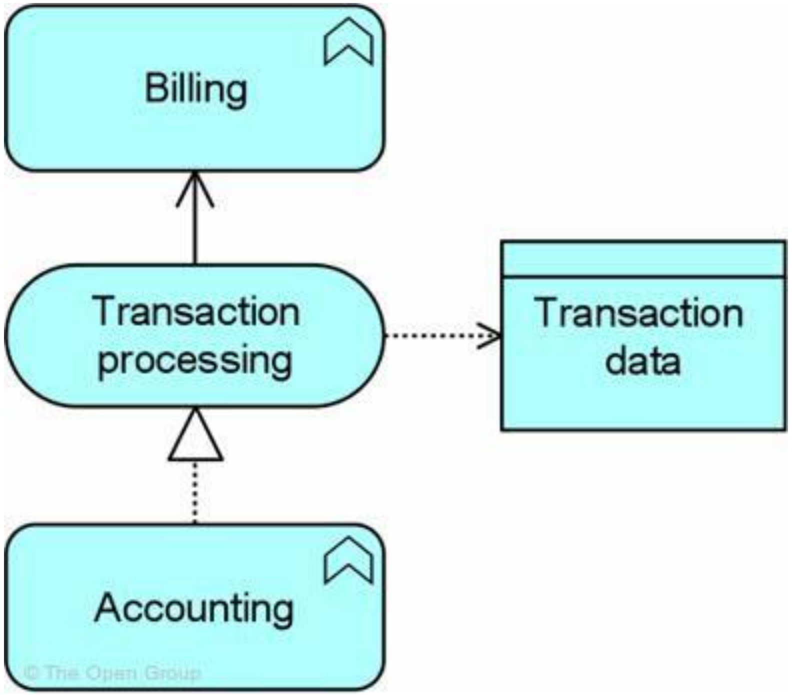
An **application function** may realize an application service. An **application interface** may be assigned to an application service.

An application service may access **data objects**.

Application Layer

- Passive Structure Concepts
  - *Data object* 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 concept in the business layer.

Data object



A data object can be accessed by an **application function**, **application interaction**, or **application service**.

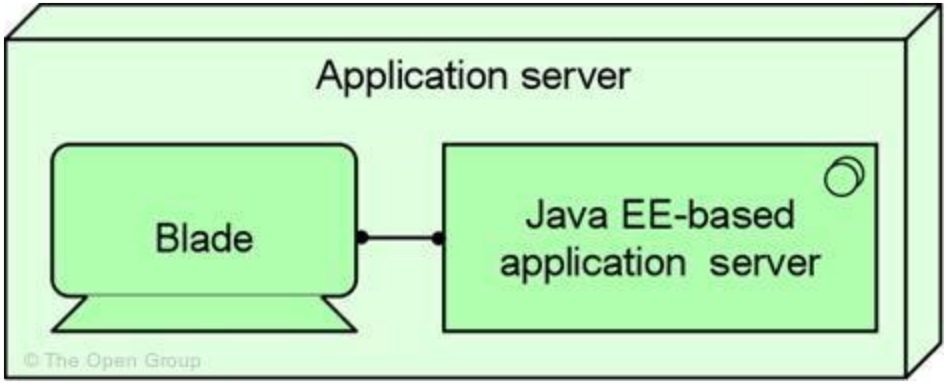
A data object may realize a **business object**, and may be realized by an **artifact**.

A data object may have association, specialization, aggregation, or composition relationships with other data objects.

Technology Layer

- Active Structure Concepts
  - The main active structure concept for the technology layer is the *node*. This concept is used to model structural entities in this layer.
  - A *node* strictly models the structural aspect of a system: its behavior is modeled by an explicit relationship to the behavioral concepts.
  - Node, Device, System Software, Infrastructure Interface, Communication Path

Node



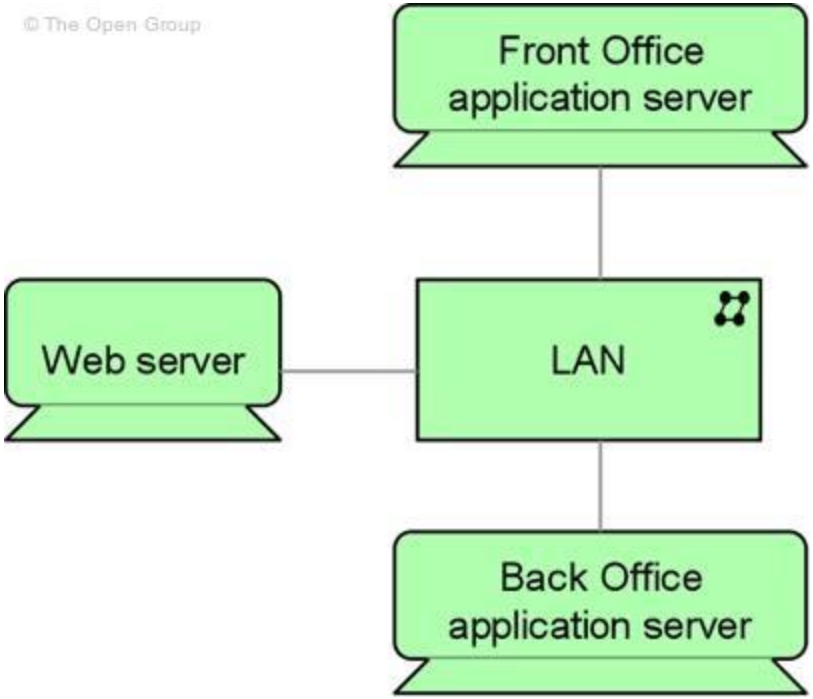
Nodes are active processing elements that execute and process artifacts, which are the representation of components and data objects.

Nodes are, for example, used to *model application servers*, *database servers*, or *client workstations*.
A node is often a combination of a **hardware device** and **system software**, thus providing a complete execution environment.

Nodes can be interconnected by **communication paths**.
Artifacts can be assigned to (i.e., deployed on) nodes.
The name of a node should preferably be a noun. A node can consist of **sub-nodes**.

Artifacts deployed on a node may either be drawn inside the node or connected to it with an assignment relationship.

Device

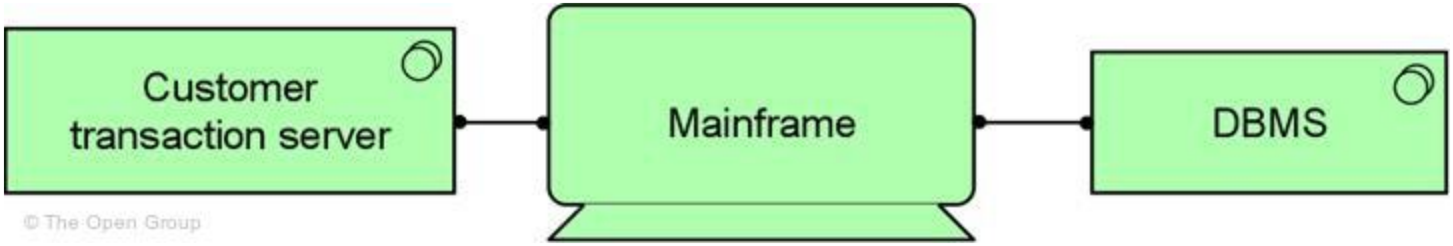

© The Open Group
Front Office application server
Web server
LAN
Back Office application server

A device is a **specialization of a node** that represents a physical resource with processing capability.

Devices can be interconnected by **networks**. **Artifacts** can be assigned to (i.e., deployed on) devices. **System software** can be assigned to a device.

A node can contain one or more devices.

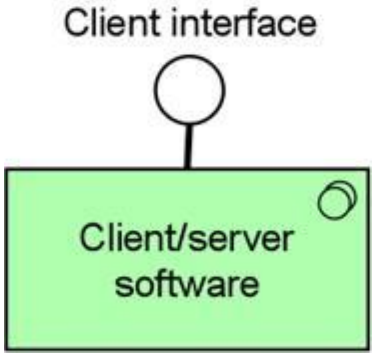
System software



System software is a **specialization of a node** that is used to model the software environment in which artifacts run, e.g. *an operating system, a JEE application server, a database system, or a workflow engine*.

System software can be assigned to a **device**. **Artifacts** can be assigned to (i.e., deployed on) system software. A node can contain system software.

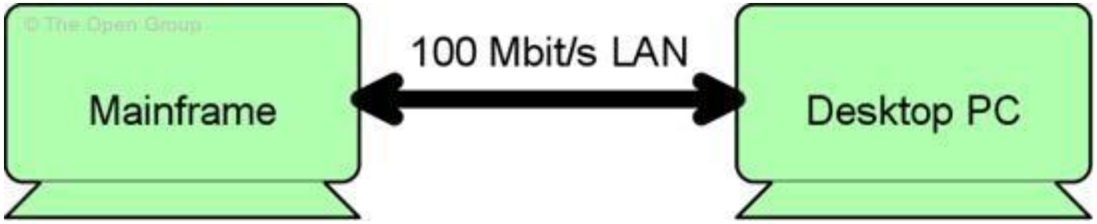
Infrastructure interface



An infrastructure interface specifies how the infrastructure services of a node can be accessed by other nodes (**provided interface**), or which functionality the node requires from its environment (**required interface**).

An **infrastructure service** can be assigned to an infrastructure interface, which exposes the service to the environment.

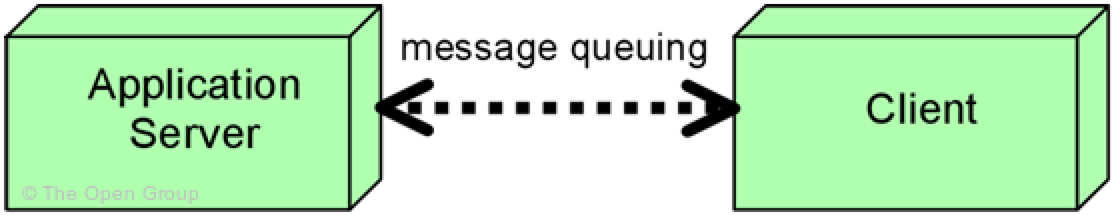
Network



A network represents the **physical communication infrastructure**. This may comprise one or more fixed or wireless network links.

A network connects two or more **devices**. A network realizes one or more **communication paths**.

Communication path



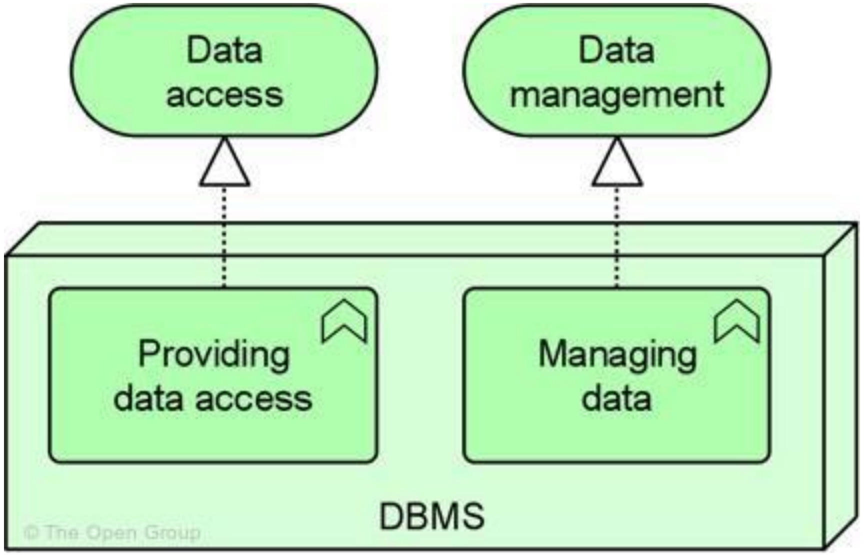
A communication path is used to model the logical communication relations between nodes. It is realized by one or more **network**s, which represent the physical communication links.

A communication path connects two or more nodes. A communication path is realized by one or more networks. A communication path is atomic.

Technology layer

- Behavioral Concepts
  - Behavior elements in the technology layer are similar to the behavior elements in the other two layers.
  - Also here, we make a distinction between the external behavior of nodes in terms of *infrastructure services*, and the internal behavior of these nodes; i.e., *infrastructure functions* that realize these services.

Infrastructure function

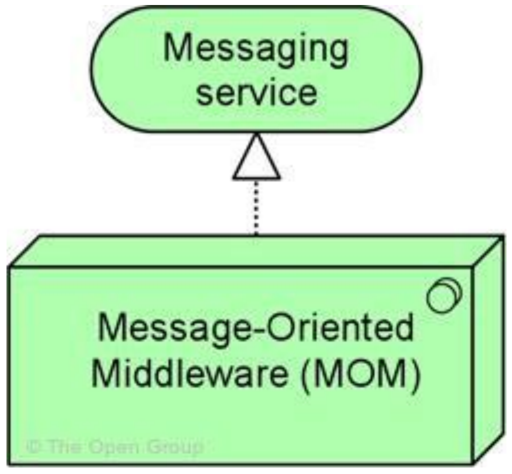


An infrastructure function describes the **internal behavior** of a node; for the user of a node that performs an infrastructure function, this function is invisible.

An infrastructure function may realize **infrastructure services**. Infrastructure services of other infrastructure functions may be used by an infrastructure function.

An infrastructure function may access **artifacts**. A **node** may be assigned to an infrastructure function (which means that the node performs the infrastructure function).

Infrastructure service



An infrastructure service exposes the functionality of a node to its environment.

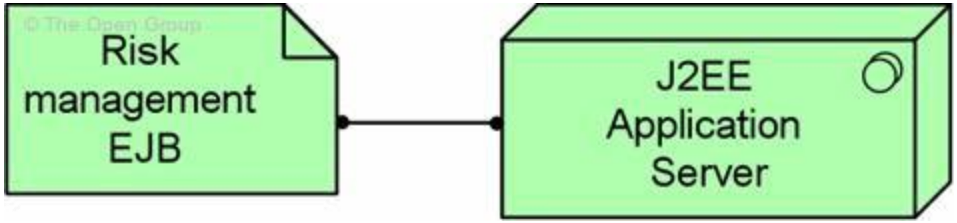
This functionality is accessed through one or more infrastructure **interfaces**. It may require, use, and produce **artifacts**.

An infrastructure service may be used by **application components** or **nodes**. An infrastructure service is realized by a node

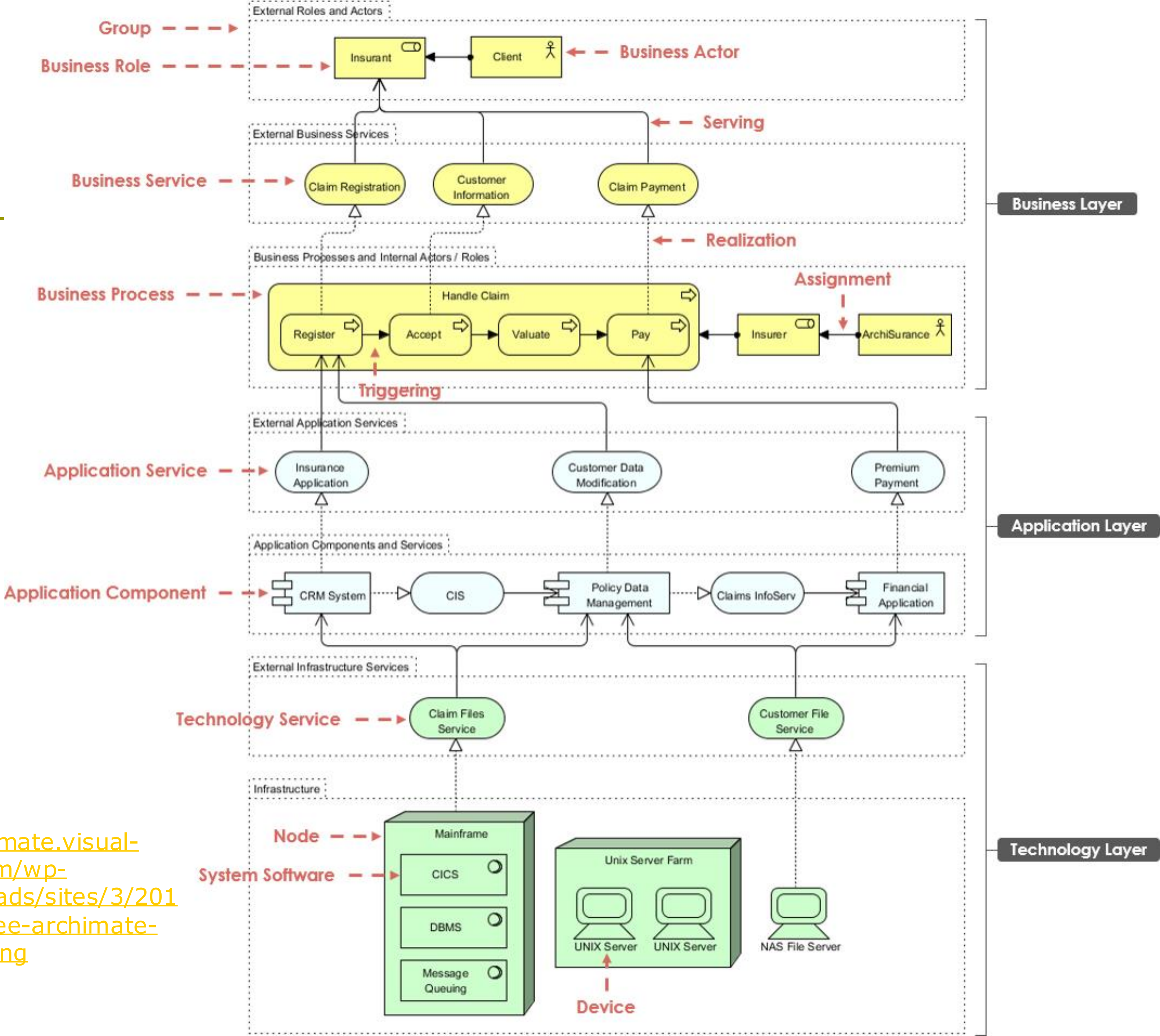
An infrastructure service may consist of sub-services.

Technology layer

- Passive Structure Concepts
  - 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 typically used to model (software) products such as source files, executables, scripts, database tables, messages, documents, specifications, and model files



Example



Source: https://archimate.visual-paradigm.com/wp-content/uploads/sites/3/2018/03/the-three-archimate-core-layers.png

References and Tools

- ArchiMate official doc: https://pubs.opengroup.org/architecture/archimate3-doc/index.html
- Software tool: **Archi** is a free, open source, cross-platform tool and editor to create **ArchiMate** models https://www.archimatetool.com/

ArchiMate vs. BSRL, BPMN, SoaML

Language	Purpose
ArchiMate	includes a richer set of elements to provide a comprehensive description of the enterprise architecture (EA), including the business layer, application layer & technology layer.
SoaML	provides the support for modeling services; a detailed description of the application service the application layer.
BPMN	provides a standard business process modelling; suitable for formally describing the business process in the business layer.
BSRL	a contract language for the (external) business services in the business layer.