# Enterprise Architecture Analysis

## State of the Art

Consensus on the state of EA Frameworks is that Frameworks really are only as useful as the models inside of them, with little to no benefit for implementing them wholesale as a prescriptive design.

TOGAF thanks to its reach and popularity, is useful to know for the more commonly used terminology (*‘a lengua franca’*); Despite its popularity, it is almost never to never implemented in full for it shares the same issues as every other EA framework: It's too rigid and static to be able to adequately address the changing business needs of an organization, and planners and organizers will always have limited information when designing the implementation roadmap. Another common criticism is that implementing a full framework ‘by the books’ takes too long, so much that by the time the technology is up and running, users will already have found an alternative, business needs would have changed, or the technology would already obsolete.

<https://www.bcs.org/articles-opinion-and-research/a-comparison-of-the-top-four-enterprise-architecture-frameworks/>

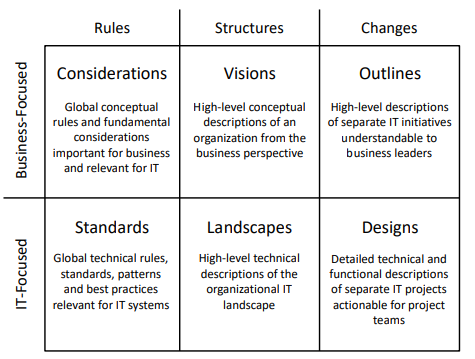
<https://www.reddit.com/r/EnterpriseArchitect/comments/128uuc6/for_those_who_have_read_the_practice_of/?rdt=49693>

*In fact, decades of research on information systems planning have long identified a broad spectrum of problems associated with business strategy as a basis for acting[16] . Strategy can be vague, ambiguous and interpreted differently by different people (e.g. “become number one” or “provide best services”). Strategy can be purely aspirational and consist of mere motivational slogans (e.g. “get closer to the customer” or “leverage synergies”). Strategy can comprise various objectives and indicators offering no actionable hints, especially for IT (e.g. “increase market share by X%” or “grow revenue to $Y”). Strategy can be market sensitive, deliberately obscure and surrounded by secrecy (e.g. partnerships, mergers and acquisitions). Strategy can be unstable, volatile and change its direction dramatically several times a year (e.g. management turnover, political churn or sales-data-driven decisions in retail). Strategy can concentrate on the business aspects that are changing, but neglect some of the more fundamental pillars that remain the same (e.g. new capabilities instead of core ones). Finally, formal business strategy can be simply missing altogether (e.g. privately held companies). - Svyatoslav Koustev : Enterprise Architecture is Based on Business Strategy, is it not?*

In the same article, Koustev posits Business Unit Plans as a better basis for EA planning, as those deal with more down to earth topics like operational problems (known issues, bottlenecks), ad hoc demands (customer needs, regulatory changes, competitor moves), permanent imperatives, and technical concerns; The proposed way of implementing EA into an organization is ‘bottom up’, taking and analyzing different unit business plans to identify common needs and redundant systems, before taking action on the company as a whole.

<https://kotusev.com/teaching_pack/>

CVSLOD Model (Considerations, Visions, Changes, Standards, Landscapes, Designs)



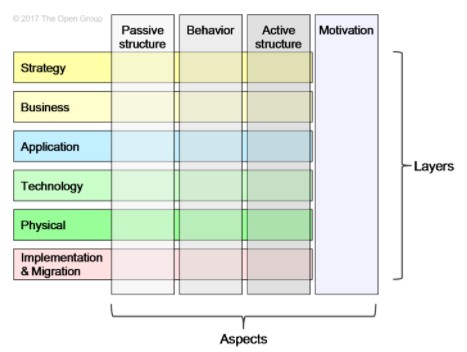
Where EA artifacts are classified as such:

* Considerations define business focused rules like principles and policies.
* Standards are IT focused rules like guidelines and tech reference models
* Visions are structures like roadmaps and capability models.
* Landscapes are IT structures like landscape diagrams and inventories.
* Outlines are business focused changes like options assessments and solution overviews.
* Designs are IT changes like solution designs.

https://www.bcs.org/media/3787/csvlod.pdf

# ArchiMate Modeling Language

The ArchiMate modelling language is an Open Group standardization of different models, that whose purpose is to allow clear visualization of business and technology processes and how the interact. The standard can be found in the Open Group website and is 207 pages long.

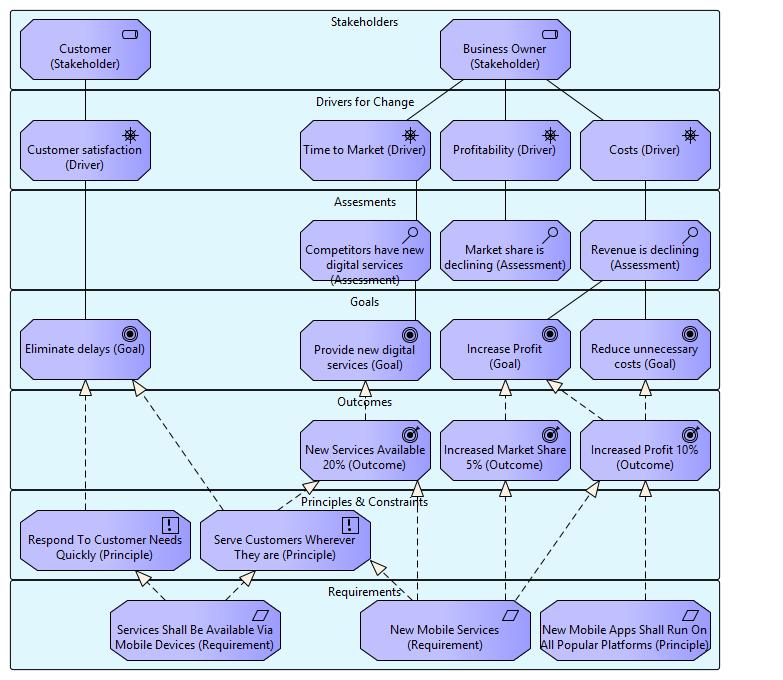


https://www.hosiaisluoma.fi/blog/useful-archimate-diagram-types/

### Motivation View (Goals view)

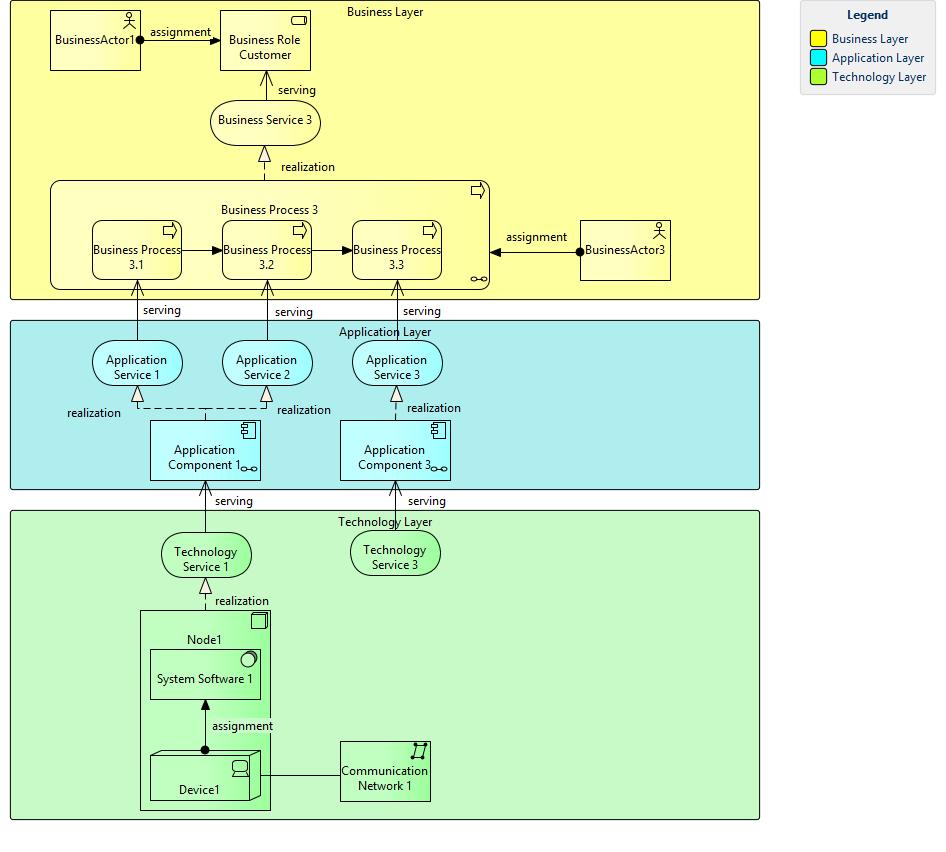
This diagram allows analyzing the *to whom, why* and *what* questions using the stakeholder, driver, assessment, and goal, outcome, principle, and requirement concepts.

* *To whom this development target is meaningful?*
* *Why is this important?*
* *What are the exact goals?*



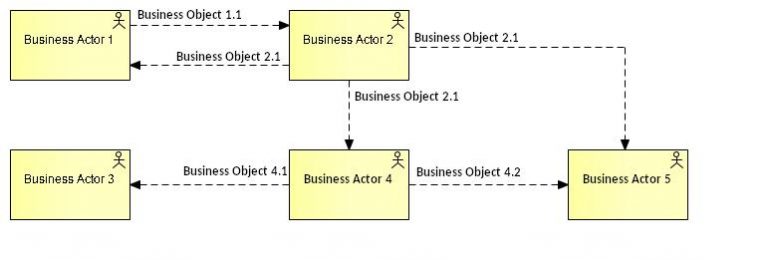
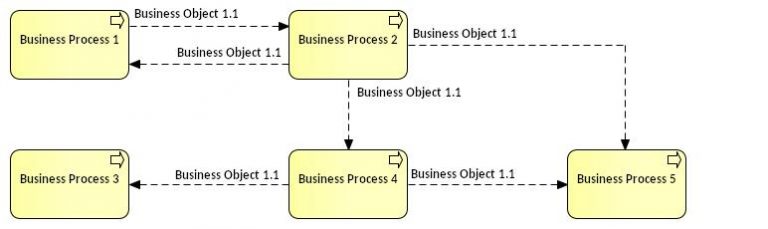
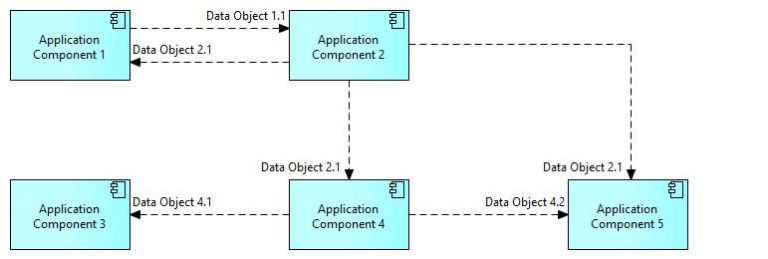
### Layered View

This diagram shows the interaction of different elements on different layers for a business process; This can be used for defining *what* services and elements are related to a specific business process. Layes may or may not be included depending on the target audience and what the purpose of the analysis is.

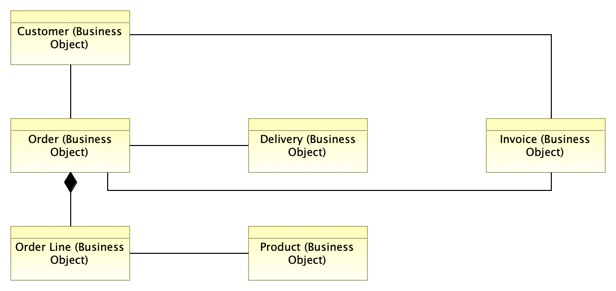


### Interaction View

This diagram can be used to define how information flows between different elements; What information flows in what direction and how actors interact with each other. This diagram can be applied to several layers and with many variations as illustrated:

* Business Actor Interaction View  
  
* Business Process Interaction View  
  
* Application Interaction View  
  *Note that the color of the bubbles means we are in the application layer*  
  

### Conceptual Data Model View



# ArchiMate 3.2 Standard

*This section is a TL; DR, notes, and reflections on the ArchiMate 3.2 standard document for my own future reference.*

*A ‘review tables’ sub-section is provided at the end of this section.*

## Overview

The language uses service orientation (the layer below offers services to layers above it) to define, distinguish, and relate the Technology, Application, and Business layers (tech -> app -> business -> stakeholders).

**Business Layer:** Depicts business services offered to customers, realized in business processes by business actors.

* **Application Layer:** Depicts application services that support the business, as well as the applications themselves.
* **Technology Layer:** Comprises both information and operational technology, including processing, storage, communication, physical technology, facilities, equipment, materials, distribution networks, etc.

The relationships between layers are formed by *serving* relationships that depict how elements in one layer are served by services in another (or the same) layer, and *realization* relationships where a lower lever service realizes an object in a higher layer (r*ealizes* in this context means that it implements the intent, behavior, or structure of the higher layer element).

**Abstractions**

* **Black box and white box:** Systems may be depicted as black boxes or white boxes, depending on whether the analysis of the system is internal (how the system works) or external (how the system relates to its environment).
* **System behavior and active structure:** Used to separate what the system *must do* and *how the system does it* from its components; The former is more commonly used in greenfield projects while the latter is used to analyze exiting systems.
* **Conceptual, logical, and physical:** Conceptual elements represent the information the business finds relevant, logical elements give logical structure to this information, and physical elements describe the storage of this information.

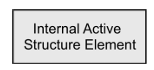
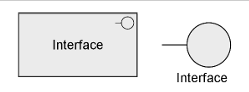
ArchiMate does not support a difference between types and instances; It is only concerned with types of objects rather than instances.

While there is no standardized set of colors in ArchiMate, it is common to use different colors to distinguish between business, application and technology layers using yellow, blue, and green respectively. Additionally, a letter can be placed in the top left corner of an element to denote whether they are a **M**otivation, **S**trategy, **B**usiness, **A**pplication, **T**echnology, **P**hysical, or **I**mplementation and migration.

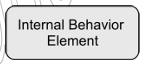
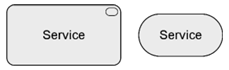
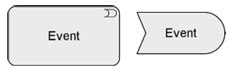
## Elements and Element Types

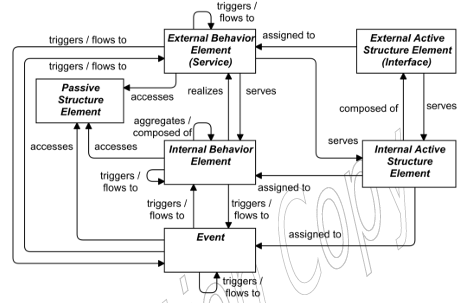
Additionally, the corners are used to distinguish between different element types:

### Structure elements

* Square corners
* Structure elements can further subdivide into active (ones that perform an action) and passive (ones that are acted upon), as well as external (outside the organization) and internal (inside the organization).
  + For example, an external active structure element would be another business that we react to (like a provider shipment), a customer request, or an interface that provides a service while concealing its internal structure.
  +  
  + 

### Behavior elements

* Round corners
* Behavior elements can also be subdivided into internal and external (also called services) elements, as well as *events* which represent a change in state.
*   
* **Motivation elements**: Diagonal corners



*The image above is a guideline and does not contain all permitted relationships*

Specialized behavior elements:

* **Process:** Represents a sequence of behaviors that achieves a specific result.
* **Function:** Collection of behavior based specific criteria, such as required resources, competencies, or location, and is managed, performed, or implemented as a whole.
* **Interaction:** An interaction represents a unit of collective behavior (a task) that must be performed by two+ internal active structure elements, either assigned directly or in collaboration.
* **[Structure] Collaboration:** Represents an aggregate of two+ internal structure elements working together to perform a collective behavior

### Motivation Elements

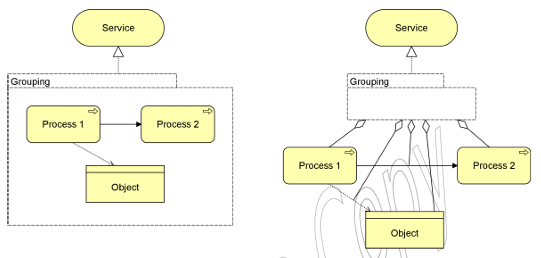
Motivation elements have the purpose of describing the *why,* and can be further subdivided into stakeholder, value, meaning, driver, assessment, goal, outcome, principle, and requirement and constraint.

Motivation elements are usually denoted using boxes with diagonal corners as such:



### Grouping

The grouping element is used to aggregate concepts, which can be elements and/or relationships. These are denoted by boxes with a dotted outline. Relationships that link to the group itself imply a relation to all or part of the group.



### Location

A location element is used to model physical places or positions where structure elements are located; This can be businesses actors, devices, components. A location plume is used to denote a location object.

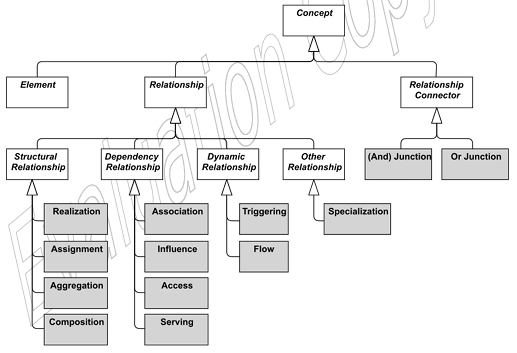


## Relationships

In addition to generic elements, the language defines a core set of generic relationships., each of which can connect a predefined set of source and target concepts (mostly elements, but sometimes they can also connect other relationships); The precise meaning for many of these relationships may differ depending on the source and destination concepts that they connect.

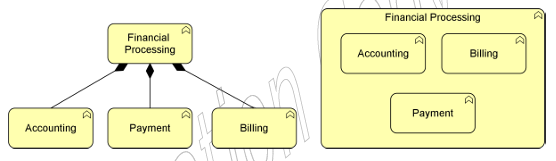
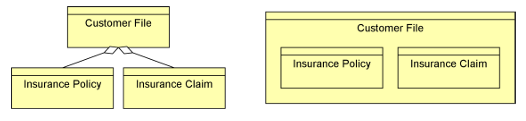
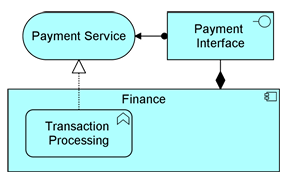
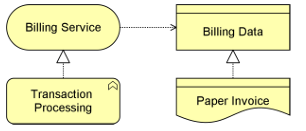
* **Structural Relationships:** model the static construction or composition of concepts of the *same type*.
* **Dependency Relationships:** model how elements are used to support other elements.
* **Dynamic Relationships:** Model behavioral dependencies between elements.
* **Other:** Describes relationships that do not fall into the other categories.

It is good practice to explicitly name or label every relationship that should be ambiguous or misunderstood.



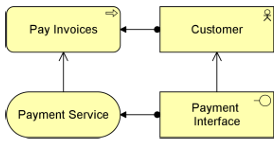
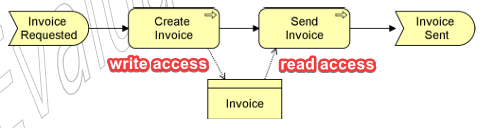
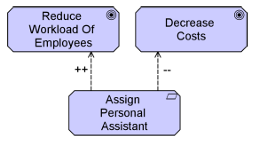
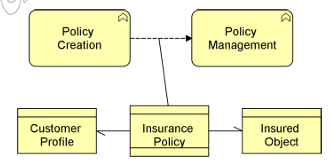
### Structural Relationships

Structural relationships represent the ‘static’ coherence within an architecture. For these relationships, the originator of the connector will always be an element, while the target may be another element or, in some cases, another relationship or relationship connector.

* **Composition Relationship:** The composition relationship represents that an element consists of other (one+) concepts. It notates that a whole or a part of the source is composed by the target and implies an existence dependency between the aggregated and aggregating elements.
  + 
* **Aggregation Relationship:** The aggregation relationship represents that an element combines one+ concept. Unlike the composition relationship, it does not imply an existence dependency between the elements.
  + 
* **Assignment Relationship:** The assignment relationship links active structure elements with units of behavior that are performed by them. It represents the allocation of responsibility, performance of behavior, storage, or execution.
  + The flow of the element is always:
    - Active Structure --> Behavior
    - Active Structure --> Passive structure
    - Behavior --> Passive Structure
  + *Example 4 includes the two ways to express the assignment relationship. The “Finance” application component is assigned to the “Transaction Processing” application function, and the “Payment Interface” is assigned to the “Payment Service”.*   
    
* **Realization Relationship:** Represents that an element plays a critical role in the creation, achievement, sustenance, or operation of a more abstract element. The interpretation of a realization relationship is that the *whole or part* of the source realizes the *whole* of the target. It is represented by a non-solid arrow with a dotted line.
  + 
  + 

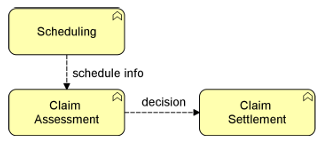
### Dependency Relationships

Dependency relationships represent how elements support, or are used, by other elements. There are four types of dependency relationships:

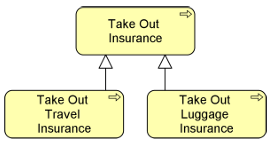
* **Serving** Re**lationships:** Indicates that an element provides its functionality to another element.
  + In the example, the customer is served to by the ‘payment interface’ interface, and the ‘pay invoices’ process is served by the ‘payment service’ service.
  + 
* **Access Relationships:** An access relationship represents the ability of a behavior or active structure to observe or act upon passive structure elements.
  + It indicates that the source ‘does something’ with the passive structure element (create a new object, read data, write or modify data, or delete object).
  + The relationship could also just indicate that the object is related with the behavior.
  + If the arrowhead is present, it indicates the creation, modification, or usage of the passive structure element. If the arrowhead points torwards the source, it may indicate a read (or read-write if both are present) access, this does not revert the flow of the relationship.
  + 
* **Influence Relationships:** Influence relationships represents that an element affects the implementation or achievement of some motivation element.
  + In general, it is used to say motivation element is realized to a certain degree, but not necessary in whole or as a requirement.
  + These are useful to trace a ‘motivational path’ to explain *why* we are doing these things.
  + These can be used to describe how elements can affect motivations ‘upwards’ as well as ‘sideways’ to describe how one motivation element may affect another.
  + Depending on the expected impact, the attribute values can be noted as {++,+,0,-,--} for a great positive impact, a positive impact, no impact, a negative impact, or a great negative impact.
  + 
  + 
* **Association Relationships:** An association relationship represents an unspecified relationship, or one that is not represented by another standard relationship.
  + These can be used when drawing a high-level model where relationships are denoted in a generic manner, for further refinement later.
  + An association is undirected by default but may be directed using a half arrow.
  +  

### Dynamic Relationships

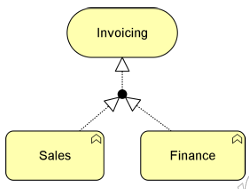
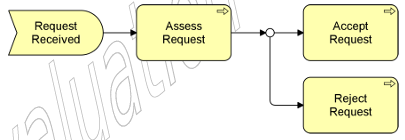
Dynamic relationships describe temporal dependencies between elements.

* **Triggering Relationship:** A triggering relationship represents a temporal or causal relationship between elements.
  + Can be interpreted to mean that a part or the whole of the source needs to be completed before the target may begin. This does not mean that the source *causes* the target, however.
  + It is represented by a solid arrowhead with a solid line.
  + 
* **Flow Relationship:**  A flow relationship is used to represent transfers from one element to another.
  + These transfers can represent the flow of goods, cash, information, etc. Between elements.
  + A flow relationship does not imply a causal relationship either.
  + It is represented by a solid arrow with a dotted lie.
  + 

### Other Relationships

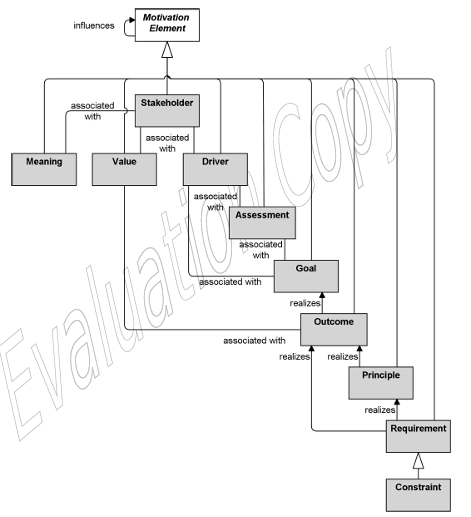
* **Specialization Relationship:** The specialization relationship represents that an element is a particular kind of another element.
  + It is represented by a hollow arrowhead with a solid line.
  + 

### Relationship Connectors

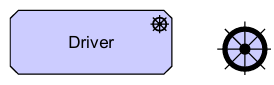
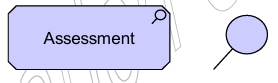
* **Junction:** A junction is not an actual relationship, but rather a relationship connector: It is used to connect relationships of the same type.
  + Junctions cannot be used to create connections that would otherwise not be allowed.
  + A junction can have one or many ingoing and outgoing relationships.
  + A junction can be an *AND* junction to express that all the elements need to participate in the relationship, or an *OR* relationship if only one needs to participate; AND and OR junctions are represented by a filled circle or a hollow circle respectively.
  + 
  + 
  + 

## Motivation Elements

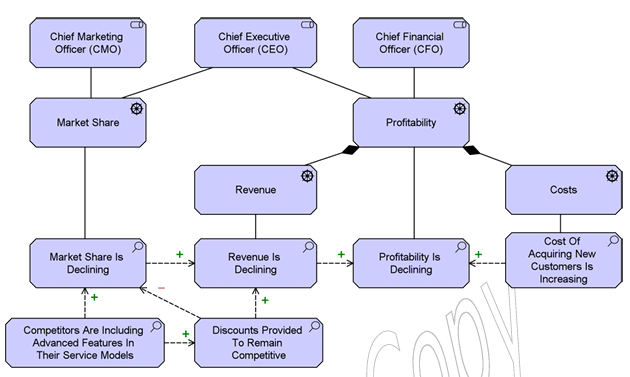
Motivation elements are used to model motivations and reasons that guide the design ot change of an enterprise architecture.



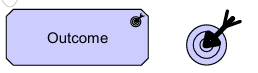
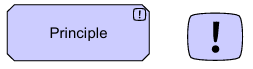
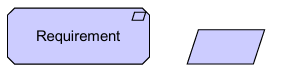
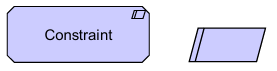
### Stakeholders, Drivers, and Assessments

* **Stakeholder:** An individual, group of people, or class thereof which influence motivation elements.
  + These can originate from inside or outside the company; Examples include the CEO, Editor Staff, or Floor Staff for internal stakeholders, and Municipal Govt, Clients, Competitors for external stakeholders.
  + Stakeholders are represented with a cylinder.
  + 
* **Driver:** A driver represents an external or internal condition that motivates the organization to define its goals and implement changes.
  + Drivers associated to stakeholders are deemed as ‘concerns’ of those stakeholders. Drivers may or may not be associated with a stakeholder.
  + Drivers are represented with a rudder.
  + 
* **Assessment:** Represents the result of an analysis of the situation with respect to a driver.
  + These can be used to identify strengths, weaknesses, opportunities and threats (four corners diagram) which can help in decision making.
  + These should be named with a very short sentence that encapsulates one of these aspects:
    - ‘Customers complain about service’
    - ‘Customers prefer to do their purchases online’
  + Assessments are represented by a magnifying glass.
  + 

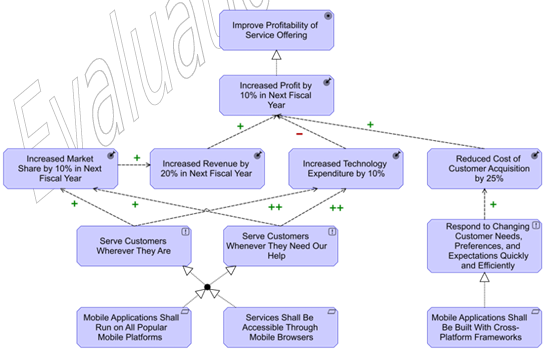
Here is an example of how these elements interact with each other; Note the use of influence connectors.



### Goal, Outcome, Principle, Requirement, and Constraint

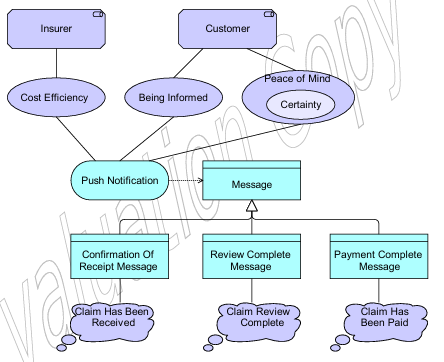
* **Goal:** Represents a desired end stare, direction, or statement of intent for an organization.
  + Goals can represent anything that a stakeholder may desire, such as a certain state of affairs or a produced value. Examples of goals can be:
    - ‘Increase revenue by 20%’
    - ‘Achieve a market share of 15%’
    - ‘Increase sales’
  + It is common to decompose the more nebulous and high-level goals into smaller scale goals for their analysis. ‘Increase profits’ --> (‘reduce costs’ + ‘increase revenue’)
  + It is also common to associate concrete outcomes with goals, describing both the quantitative and time related results that are desired: ‘Bring the prototype product to market by 10/2025 and have a production capability of 1,500 units’.
  + Goals are represented using a target.
  + 
* **Outcome:** An outcome represents a result, effect, or consequence of a situation.
  + Outcomes are high-level business-oriented results produced by an organization. These are also tangible, quantitative, and time related, and can be associated with assessments.
  + Simply put, a goal is what you want, and the outcome is what you achieve.
  + Unlike goals, outcomes can be used to model negative or unwanted effects to design for appropriate response measures.
  + An outcome is represented by a target with an arrow.
  + 
* **Principle:** A principle is a statement that defines intent for a general property that applies to any system.
  + ‘System’ in this definition refers to a group of functionally related elements, and may refer to any active structural element, behavior element, or passive structural element.
  + Principles are definitions of intended properties for a system and are bread in scope.
    - ‘Information management processes must comply with all relevant laws’
  + Prin ciples are represented by an exclamation mark inside a round square.
  + 
* **Requirement:** A statement of need that defines a property of a specific system.
  + In contrast to principles, requirements are narrower in scope and affect a limited number of elements.
  + Requirements are derived from goals and are the definition of required states that will bring the goal to fruition. These may be changes to existing elements or they may prescribe the creation of new systems.
    - Goal ‘improve portfolio management’ may be decomposed into the requirements ‘Introduce online tools for portfolio management’
  + Requirements are represented by a rhombus
  + 
* **Constraint:** Represents a limitation on aspects of the architecture, its implementation process, or its realization.
  + Constraints are used to describe limitations and restrictions in operations. These may be, but are not limited to, system restrictions (technological limits), implementation restrictions (budget or timing), or functional restrictions (legal constraints)
  + Constraints are represented by a rhombus with a dash on the left side.
  + 

An example of how these interact is presented below:



### Meaning and Value

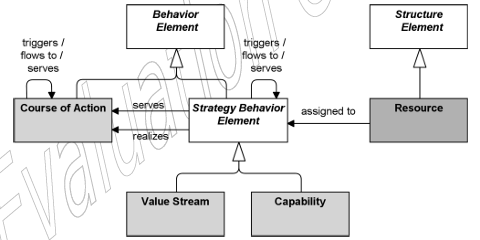
* **Meaning:** Represents knowledge or expertise in a concept in a particular context.
  + This is used to describe the meaning of passive structure elements, by describing the intent around that element.
  + What each element ‘means’ to different stakeholders may differ, so meanings may be associated with a stakeholder to represent that the meaning is describing the element from that stakeholders’ point of view.
  + Meanings are represented with a thought bubble.
  + 
* **Value:** Represents the relative worth, utility or importance of an object
  + Value is often represented as a monetary value (like the sell price to a customer), but it can also be used to represent non-monetary value (like convenience or security).
  + 



## Strategy Layer

The strategy levels and elements are used to model the strategic direction and choices of an enterprise and can be used to express how it seeks to generate stakeholder value, the capabilities it needs to generate said value, and the resources it needs to support said capabilities.

Strategy elements are intentionally simpler and smaller than other layers, and it's meant to be more abstract than the business layer.



### Structure Elements

* **Resource:** A resource represents an asset owned or controlled by an individual or organization.
  + Resources are analyzed in terms of strengths and weaknesses and considered when implementing strategies.
  + Resource limitations can be a very big consideration when choosing what goal, strategy, or project to implement.
  + Resources are often classified in tangible, intangible, and human assets. Tangible assets include financial assets (cash, borrowing capability) as well as physical assets (plant, equipment, merchandise), intangible assets include technology assets (patents, trade secrets), reputation assets, and culture assets, and human assets include skills and know-how the workers have, capacity for communication and collaboration, as well as worker motivation.
  + Resources are depicted using a sideways battery icon.  
    

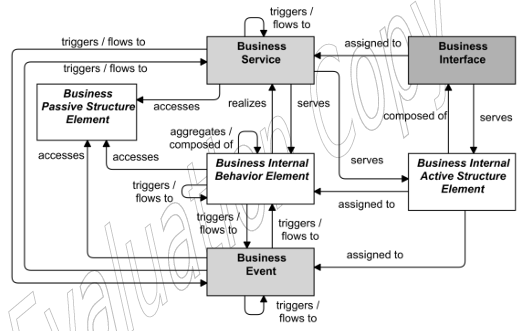
### Behavior Elements

* **Capability:** A capability represents an ability that an active structure element (organization, person, system, etc.) possesses.
  + Capabilities represent what the company can do (or be able to do in the future), and can be realized by various elements (people, processes, systems, etc.).
  + Capabilities can also have serving relationships.
  + Capabilities are classified as a behavior, expressed in high level terms, and are typically realized by a combination of organization, people, processes, information, and technology.
  + Capability names should represent ‘what we do’ rather than ‘how we do it’ (Risk management, market development, software development); Capabilities are represented using a stack of blocks.
  + 

* **Value Stream:** A value stream represents a sequence of activities that create an overall result for a customer, shareholder, or end user.
  + Value is *always* defined from the point of view of the relevant stakeholder (customer, user, client, etc).
  + Value streams can be defined at different levels of the organization (local, business unit, department, or enterprise level) and can be a composition of value adding activities.
  + While business processes and value streams seem alike, they are defined at different abstraction levels and serve different purposes: Business processes describe the time ordered sequence of behaviors required to create some result for an individual case and can used to describe alternative paths and decision points, while a value stream focuses on the value-creating behavior from the perspective of importance, worth or usefulness of what is being produced (it is not a description o the production process).
  + Value streams are represented by a right pointing wide arrow, and are usually named using a verb-noun construct in the active tense (‘Acquire Insurance Product’).
  + 
* **Course of Action:** A course of action represents an approach or plan for configuring capabilities and resources of the enterprise undertaken to achieve a goal.
  + It represents what the company has decided to do.
  + Courses of action can be classified by their scope into strategies (long term, large scale) or tactics (short term, small scale), although there is no strict distinction.
  + Courses of action are represented using an arrow pointing towards a target.
  + 

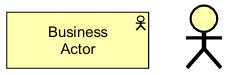
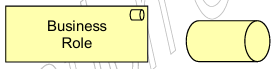
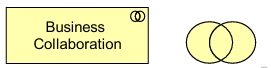
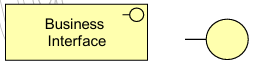
## Business Layer

The business layer is focused on describing the business level operations, services, and processes of an enterprise, including regular business processes like product acquisition, customer support, payroll management, and in general is relevant for the administrative and executive employees to make informed decisions in line with organizational goals.



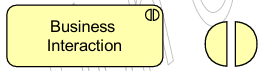
### Active Structure Elements

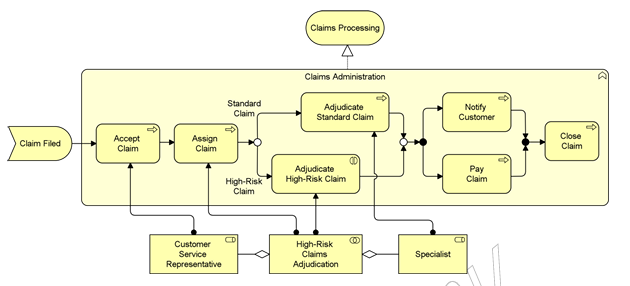
In the business layer active structure elements refer to the static structure of an organization, representing the entities and the relationships that make it up. Active entities are the subjects that perform behavior and can be either individual people or groups of people that have a permanent status within the organization.

* **Business Actor:** A business actor represents a business entity capable of performing a behavior.
  + Actors may include entities outside the organization (clients, providers, etc.)
  + A business actor may be assigned to a business role, and perform the behavior described to their assigned roles.
  + Business actors may be individual people or organizations.
  + Business actors are representing using a stick figure.
  + 
* **Business Role:** Represents the responsibility for performing a specific behavior, or the part played in a particular action or event.
  + Business actors assigned to a role are responsible for performing the corresponding behaviors are carried out, either by performing them or delegating them.
  + Business roles are represented with a sideways cylinder and are usually named with a noun.
  + 
* **Business Collaboration:** Represents an aggregate of two or more active internal structures that work together to perform a behavior.
  + It represents a collection of business roles that work together for a business process and may either be permanent or temporary.
  + It can be useful when modelling interaction between separate businesses or organizations.
  + Business interactions are represented using two overlapping circles and are either unnamed or named with a noun.
  + 
* **Business Interface:** A business interface represents a point of access where services are made available to the environment.
  + These can either be the result of a process inside the same business layer, or from the layer below, the application layer.
  + 

### Behavior Elements

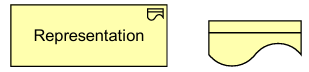
Business processes can be generally described as the actions performed by business active structures that fulfill a business function; While not strictly a behavior, business events are also included in this section as they occupy a similar place in diagrams.

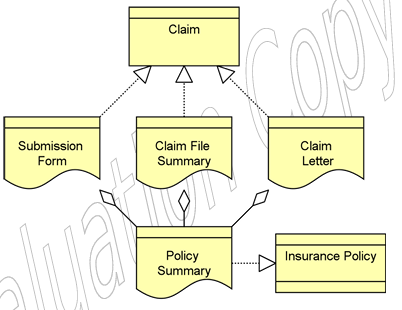
* **Business Process:** A business process represents a sequence of business behaviors that achieves a specific result, such as a defined set of products or business services.
  + It is classified as an internal behavior as the process itself is irrelevant to the consumer, only its products and services are.
  + A business process describes a ‘flow’ of activities, where a business function groups activities by required skills and resources.
  + A business process may trigger or be triggered by any other business behavior element, may access business objects, may realize business services, may access business services, and may be assigned to a business role to perform this process manually, or to an application process to perform the process automatically.
  + Business processes should be clearly named to indicate a predefined sequence of actions and are representing using an arrow pointing right.
  + 
* **Business Function:** Represents a collection of business behavior based on a chosen set of criteria and is managed or performed as a whole.
  + A business function may trigger or be triggered by any other business behavior, may access business objects, may realize business services, may be served by business, application, or technology services, and may be assigned to a business tole.
  + Business functions are represented with an upwards facing arrow.
  + 
* **Business Interaction:** A business interaction represents the collective behavior performed by a collaboration of business actors, roles, or other collaborations.
  + Business interactions may trigger or be triggered by any other business behavior element, may access business objects, may realize business services, may use internal business or application services. A collaboration o may be assigned to a business interaction.
  + Business interactions should be named with a verb in simple present tense and are represented using two half circles.
  + 
* **Business Event:** Represents a business-related change of state.
  + Business events are instantaneous and can originate from other busies processes or from the environment of the organization (from a customer, for example.)
  + Business events may trigger or be triggered by other business processes, functions or interactions, may access a business object, and may be composed of other business events.
  + Business events names should be a verb in perfect tense (‘order received’) and are represented using an arrow with a round head as shown bellow.
  + 
* **Business Service:** Represents an explicitly defined behavior that a business role, actor or collaboration exposes to its environment.
  + Services should provide a unit of behavior that is meaningful from the point of view of the environment,
  + Business services can be external customer-facing services or internal support services.
  + Business services are represented with a pill shape.
  + 



### Passive Structure Elements

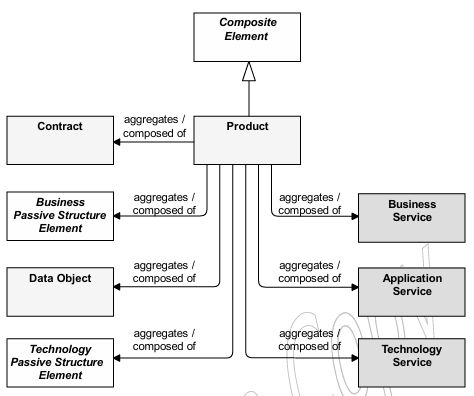
Passive structure elements are elements that are manipulated by behaviors (accessed, modified, written, read, etc.).

* **Business Object:** A business object represents a concept that is used within a particular business domain.
  + Business objects generally represent a *class* of objects and rarely an *instance* of an object.
  + Business objects are represented with a rectangle with a line at the top.
  + 
* **Contract:** A contract represents a specification of an agreements between a provider and a consumer that specifies the rights and obligations associated with the product and establishes functional and non-functional parameters for interaction.
  + Contracts may or may not be formal, representing both legal contracts as well as informal agreements.
  + Contracts are usually named with a noun and are represented by a rectangle with a line at the top and another at the bottom.
  + 
* **Representation:** A representation represents the perceptible form of the information carried by a business object
  + If relevant, representations can be classified by their medium or format (electronic, paper, audio, PDF, RTF, etc.).
  + Representations can be associated with Meanings, which carries the meaing of the representation in the point of view of a stakeholder.
  + Representations are represented using a ‘print’ icon with a line at the top.
  + 



### Composite Elements

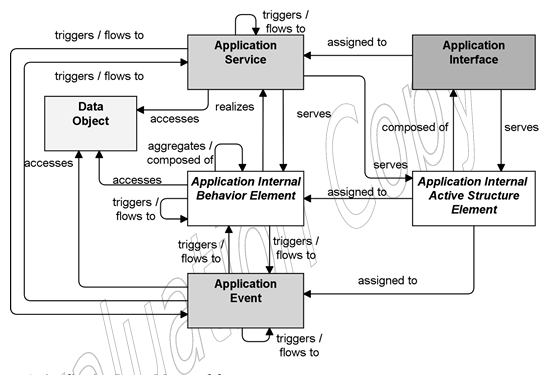
The only composite element in the business layer is the product, which aggregates services and passive structure elements across the layers.



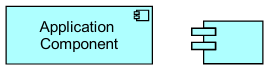
* **Product:** A product represents a coherent collection of services and/or passive structure elements, accompanied by a contract, offered in whole to customers.
  + Products may describe tangible products as well as intangible service-based information products.
  + A value may be associated with a product.
  + The name of the product is usually the name given in the interaction with the customer and is represented by a rectangle with a smaller rectangle in the top right corner.
  + 

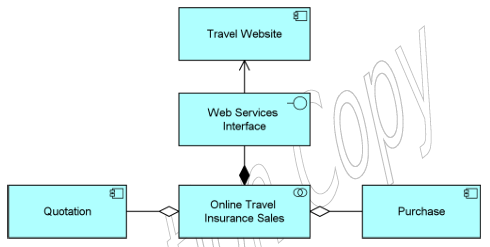
## Application Layer

The application layer is used to describe and model applications used by the company in normal business operations. It describes the behavior of the application and the flow of information and is serviced by the technology layer which describes the specific function of the implementation (the ‘metal’).



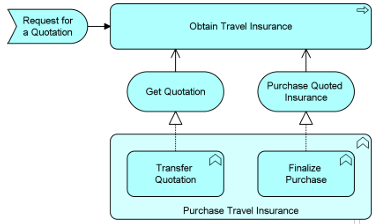
### Active Structure Elements

* **Application Component:** Represents an encapsulation of app functionality aligned to implementation structure.
  + An app component is a self-contained unit, independently deployable, reusable and replaceable.
  + An app component performs one or more app functions, has one or more app interfaces and may serve another app component via its interface.
  + 
* **Application Collaboration:** An aggregate of two+ app internal active structure elements that work together to perform collective application behavior.
  + Specifies which app components or collaborations cooperate to perform a task.
  + App collaborations are represented by two overlapping circles, using the collaboration symbol mentioned before.
* **Application Interface:** Represents a point of access where app services are made available to a user, another app component, or a node.
  + App interfaces represent a form of contract, specifying parameters, protocols, conditions and data formats.
  + App interfaces are represented using the aforementioned interface symbol.

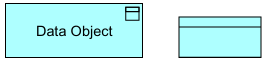


### Behavior Elements

* **Application Function:** Describes an automated behavior performed by an application component.
  + Describes an automated internal behavior done by an app component; If this this is exposed externally, it is done so via an app service.
  + An app function may realize one or more application services, and other services may serve an application function.
  + If an app component is assigned to a function, it means that the component performs the function.
  + Functions should be named with a verb ending in –ing and are represented with an upwards pointing arrow.
  + 
* **Application Interaction:** Represents a unit of collective behavior performed by a collaboration of two+ app components.
  + The details of this interaction may be detailed during the detailed application design using a UML interaction diagram or other applicable diagram.
  + App interactions are represented using the interaction icon described before, of two half circles next to each other.
* **Application Process:** Represents a sequence of application behaviors that achieves a specific result.
  + Describes the internal behavior required to realize a ser of services. It is designated an internal behavior because the user only interacts with the services offered and is not concerned with the process.
  + The name of an application process should clearly indicate a series of application behaviors using a verb-noun combination and are represented using the Process symbol of a right pointing arrow.
* **Application Event:** An application event represents an application state change.
  + Application functions and other app behavior may be triggered or interrupted by an app event.
  + Application behavior may raise evets that trigger other application behavior.
  + Events may originate from the organization environment, or internally from another organization application.
  + An application event may have a time attribute that denotes the moment(s) that the event is triggered, which can be used to model recurring tasks like a daily log.
  + Application events should be named in the present perfect tense and are represented the Event symbol of a pill with half an arrowhead.
* **Application Service:** An application service represents an explicitly defined exposed app behavior.
  + As mentioned before, app services are a strictly external definition to the app that performs the service.
  + An application service should provide a unit of behavior that is meaningful to is environment and useful to its users. An application interface may be assigned to an application service.
  + The name for an application service should be a verb ending in –ing, or otherwise explicitly mention the word ‘service’ (accounting service). Application services are represented usig a pill shape.

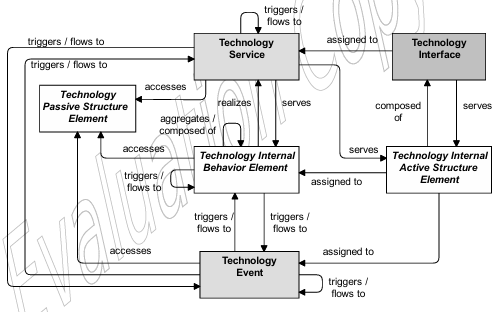


### Passive Structure Elements

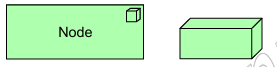
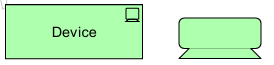
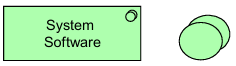
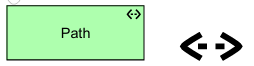
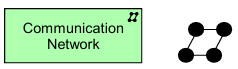
* **Data Object:** Represents structured data for automated processing.
  + A data object should be a self-contained piece of information relevant to the business, not just meaningful within the application layer (customer records, client data set, claims).
  + Data objects should represent a class of items of which one or multiple may exist.
  + Data objects should be named using nouns and are represented using a square box with a line at the top.
  + 

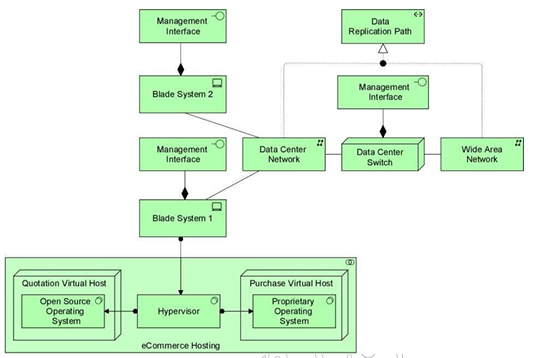
## Technology Layer

The technology layer is used to represent the underlying technology that supports the application layer, the ‘metal’ of the machine.



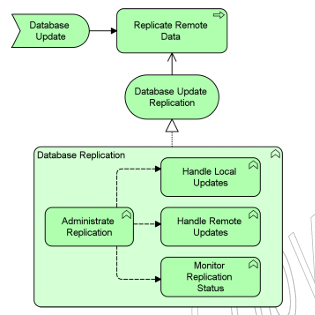
### Active Structure Elements

* **Node:** A node represents a computational or physical resource that hosts, manipulates, or interacts with other computational pr physical resources.
  + Nodes are the main active structure element for the technology layer and is used to model structural entities.
  + Nodes perform technology behavior and execute, process, store and process artifacts and materials.
  + Nodes are interconnected by paths which describe the communication between them.
  + Nodes may consist of sub-nodes.
  + The name of a node should be a noun. Nodes are represented suing a box shape.
  + 
* **Device:** A device represents a physical IT resource upon which system software and artifacts may be stored or deployed for execution.
  + Devices represent physical IT resources with processing capability.
  + Devices can be interconnected by communication networks and can be assigned to artifacts and software to model deployment.
  + A node can contain one or more devices.
  + Devices are named with according to their hardware and are represented with a computer symbol.
  + 
* **System Software:** Represents software that process or contributes to an environment for storing, executing, and using software or data deployed within it.
  + System software is usually combined with a device that represents the hardware environment it's on to form a general node.
  + System software may be composed of other system software.
  + System software is named according to its type of execution environment and is represented by a symbol of two discs overlapping.
  + 
* **Technology Collaboration:** Represents an aggregate of two+ technology internal active structure elements that work together to perform a unit of collective technology behavior.
  + Technology collaborations are represented using the collaboration symbol and are usually unnamed.
* **Technology Interface:** Represents a point of access where technology services are offered by a tech internal active structure element.
  + A single service may be exposed through different interfaces.
  + An interface represents a kind of ‘contract’ which may describe parameters, protocols, conditions, and data formats.
  + Technology interfaces are preferably named with a noun and are represented using the interface symbol.
* **Path:** Represents a link between two or more technology internal active structure elements through which these can exchange data, energy, or materials.
  + Paths are realized by one+ communication or distribution networks. The physical properties of the path are aggregated from the underlying networks.
  + A path connects two+ tech internal active structure elements.
  + Paths are represented using a symbol of a dotted arrow with two sides.
  + 
* **Communication Network:** A communication network represents a set of structures that connect devices or system software for transmission, routing, and reception of data.
  + Communication networks have properties such as bandwidth and latency; It represents the physical realization of the logical path between nodes.
  + Communication networks are represented using a depiction of four interconnected nodes.
  + 



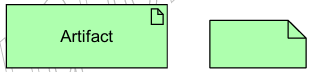
### Behavior Elements

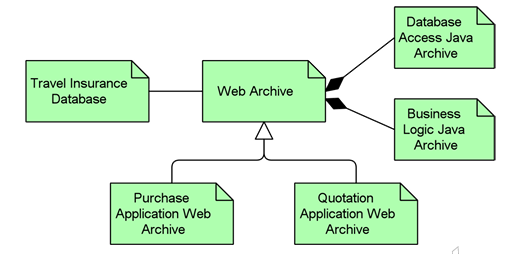
* **Technology Function:** Represents a collection of technology behavior that can be performed by a technology internal active structure element.
  + Tech functions are named using a verb ending in –ing, and are represented using the function symbol of an upwards pointing arrow.
* **Technolgy Process:** Represents a sequence of technology behaviors that achieves a specific result.
  + A technology active structure element may be assigned to a process to indicate that this element does this process.
  + Tech processes may realize, serve, and be served by technology services.
  + Technology processes should be named with a verb or verb-noun combination that clearly indicates the series of behaviors it encapsulates (boot up system) and are represented using the process symbol of a rightwards pointing arrow.
* **Technology Interaction:** Represents a unit of collaborative behavior performed by a technology collaboration.
  + The specific details of the interaction can be expressed with a UML interaction diagram.
  + The name of the interaction should clearly identify the behavior; Interactions are represented using the interaction symbol of two half circles facing each other.
* **Technology Event:** Represents a technology change of state.
  + Technology functions and behaviors may be triggered by, interrupted by, or raise events.
  + Events are always in instant change in state and are named accordingly in the present perfect tense; Events are represented using the event symbol of a half pill with a feathered end.
* **Technolgy Service:** Represents an explicitly defined exposed technology behavior.
  + Technology services expose the functionality of an internal active structure element to its environment.
  + Technology services should provide a meaningful unit of behavior to its environment.
  + Technology services should be named with a verb ending in –ing and are represented using a pill shape.



### Passive Structure Elements

Technology passive structure elements describe elements that are used and processed by the infrastructure; They are representations of informational and physical objects.

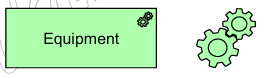
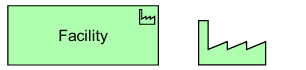
* **Artifact:** Represents a piece of data used or produced in a software development process, or by deployment and operation of an IT system.
  + Artifacts are typically used to model software end products such as source files, scripts, database tables, messages, documents, specifications, and model files.
  + An artifact instance (a copy) can be deployed in a device or system software.
  + Artifacts could represent a physical data component that realizes a data object.
  + Artifacts should be named after the file they represent (‘specs.jar’) and are represented by a file icon.
  + 



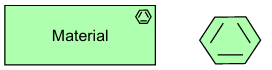
### Physical Elements

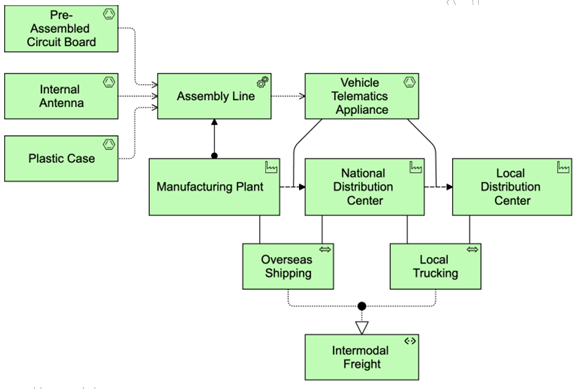
Physical elements are part of the technology layer, but they mostly focus on describing ‘real world’ equipment, objects, and systems.

#### Physical Active Structure Elements

* **Equipment:** Represents one+ physical machines, tools, or instruments that can create, use, store, move, or transform materials.
  + Equipment comprises all active processing elements that carry out physical processes which use, modify, or transform material.
  + Nodes can be used to model a combination of equipment and IT infrastructure elements, such as a production plant with its control systems.
  + Equipment elements are named using nouns and are represented by an interconnecting pair of gears.
  + 
* **Facility:** Represents a physical structure or environment.
  + Represents a physical resource that facilitates (houses) the use of equipment; It is typically used to model factories, buildings or outdoor constructions.
  + Facilities are interconnected by distribution networks.
  + Facilites can serve other facilities and active structure elements.
  + A facility can be compromised of other facilities.
  + Facilities should be named with a noun that refers to that facility and are represented using a factory symbol.
  + 
* Distribution Network: Represents a physical network used to transport materials or energy.
  + A distribution network connects two+ facilities and may realize one+ path.
  + Distribution networks may consist of sub-networks.
  + Distribution networks are represented by a solid line with two ends.
  + 

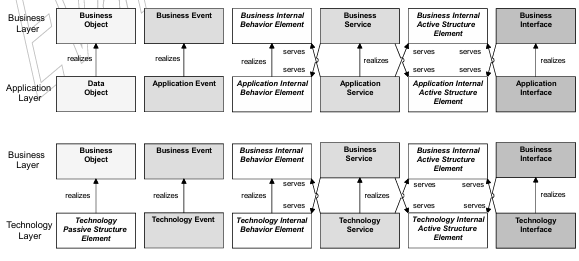
#### Physical Passive Structure Elements

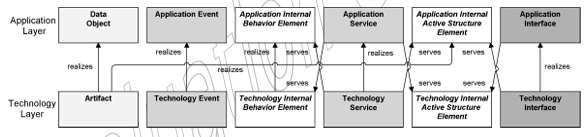
* **Material:** Materials represent tangible physical matter or energy.
  + Typically used to represent raw materials, physical products, and energy sources.
  + Material can be accessed by technology behavior elements to model how the material changes.
  + Material may be assigned to by equipment to model where the material is stored.
  + Material should be named with a noun, may be composed of other materials, and are represented by the molecular structure of benzene, a chemical used in the fabrication of plastics and nylon derived from petroleum.
  + 



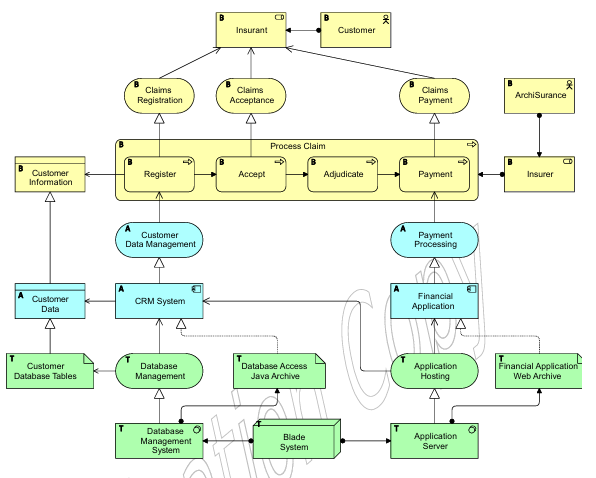
## Relationship Between Layers

The following diagrams represent the relationship between the layers.



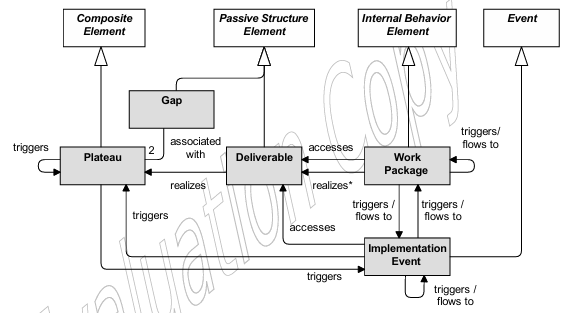


Below is an example of the interaction between layers.

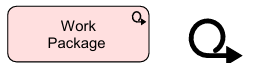
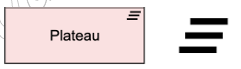
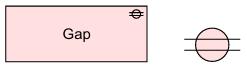


## Implementation and Migration Layer

The implementation and migration layer has the purpose of bridging the gap between a current state and the desired architecture.



### Implementation and Migration Elements

* **Work Package:** A work package represents a series of actions identified and designed to achieve a specific result within time and material constraints.
  + Work packages are definite in duration and are, usually, not used to represent recurring tasks.
  + Work packages should produce a well-defined set of results, modeled as goals, outcomes, or deliverables.
  + Within the Agile framework, work packages can be used to model sprints.
  + Work packages are represented by a looping arrow.
  + 
* **Deliverable:** Represents a precisely defined result of a work package.
  + These may be resulting products of any kind, or intangible outcomes such as organizational change.
  + Deliverables are represented using a print symbol.
  + 
* **Implementation Event:** Represents a state change in relation to implementation or migration.
  + Work packages may trigger, be triggered by, or raise events.
  + Events are instantaneous and do not have a duration.
  + Implementation events may include a time attribute to model project schedules and milestones.
  + Implementation events may access deliverables, trigger work packages or plateaus, and be composed of other implementation events.
  + Implementation events are represented with the event symbol.
* **Plateau:** Represents a relatively stale state of architecture that exists transiently.
  + Ca be used to represent ‘transition architectures or phases which bridge the gap between the current and desired state, as well as the current and desired state themselves.
  + Plateaus are represented by three horizontal lines stacked vertically, offset of one another.
  + 
* **Gap:** Represents a statement of difference between two plateaus.
  + Gaps are modeled using a symbol of a circle with two parallel lines that divide it into thirds.
  + 

## Views and Viewpoints

Viewpoints represent a subset of ArchiMate elements and relationships that are relevant t a particular stakeholder, for a particular purpose. They are part of the description of ‘who is this diagram for’ implicit in the design of every diagram.

The following is the list of 23 example viewpoints provided by the ArchiMate standard and is by no means a comprehensive or absolute list of all viewpoints possible.

### Composition

Viewpoints that define the internal composition and aggregation of elements.

Utilizes concepts from the three core layers: business, application, and technology.

#### Organization

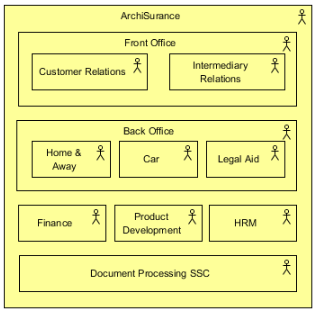
The organization viewpoint is used to analyze the organizational structure of an organization unit, such as a corporate, company, department, or a network of companies. It is commonly used in identifying competencies and responsibilities within an organization.

**Stakeholders:** Enterprise, process, and domain architects, managers, employees, and shareholders.

**Perspective:** Analyze the structure of the enterprise in terms of roles and departments.

**Concerns:** Identify competency, authority and responsibilities.

**Scope:** Single layer / single aspect



#### Information Structure

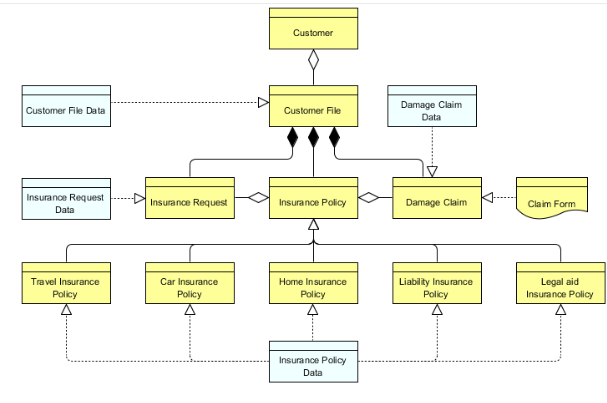
The information structure works like a traditional information model and is commonly created when developing an information system, This viewpoint shows how information used in business operations is mapped to underlying application systems.

**Stakeholders:** Domain and information architects

Concerns: Structure dependencies of the used data and information consisteny.

Purpose: Designing

Scope: Multiple layers, singe aspect-



#### Technology

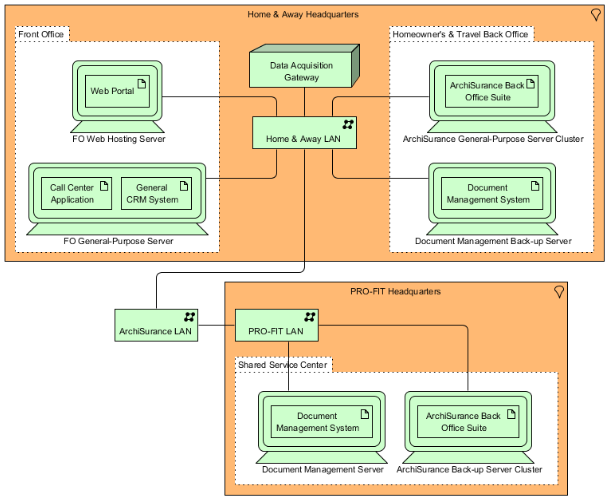
The technology viewpoint shows how the application layer is supported by the technology layer elements like physical hardware, networks, and system software.

**Stakeholders:** Infrastructure architects, operational managers.

**Concerns**: Stability, security, dependencies, costs of the infrastructure.

**Purpose:** Design

**Scope:** Single layer, multiple aspect.



#### Layered

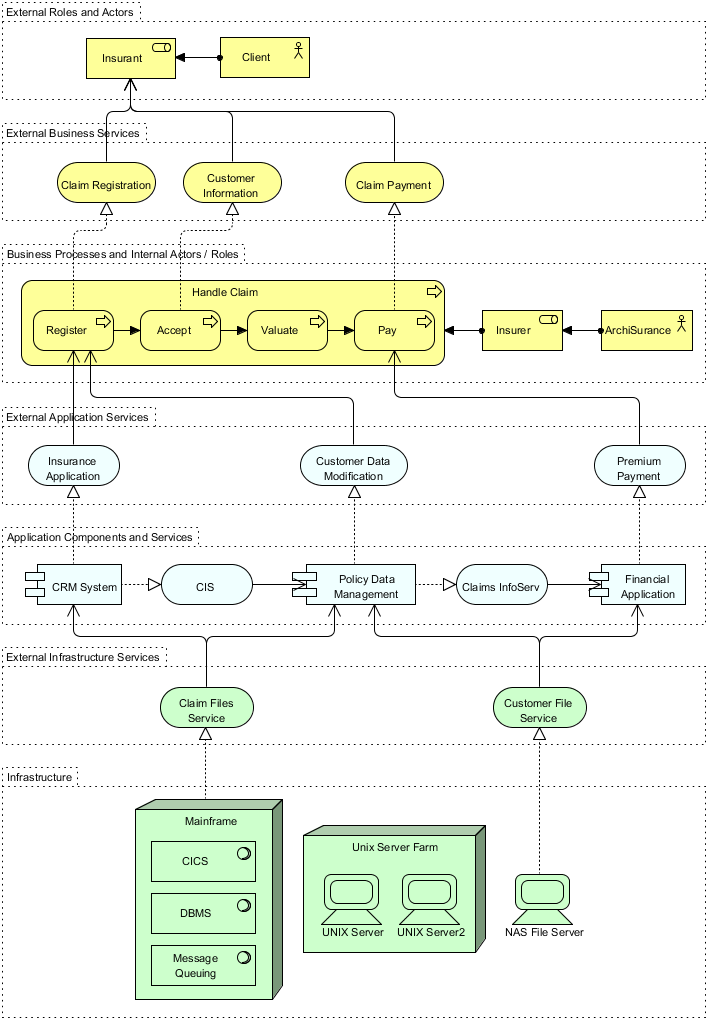
Presents a ‘birds' eye’ view of all the layers, with the purpose to reveal the relations and interactions between the layers by means of realizations and services.

**Stakeholders:** Enterprise, process, application, infrastructure, and domain architects.

**Concerns:** Consistency, reduction of complexity, impact of change, flexibility.

**Purpose:** Designing, deciding, informing

**Scope:** Multiple layers, multiple aspects.



#### Physical

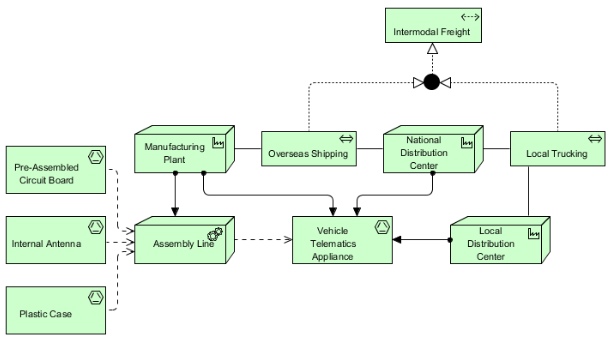
The physical viewpoint shows equipment, facilities, distribution networks and what active elements are assigned to this equipment. This is useful in analyzing how the IT services and applications are related to the underlying equipment.

**Stakeholders:** Infrastructure architects, operational managers.

**Concerns:** Relationships and dependencies of the physical environment and its relation to IT infrastructure.

**Purpose:** Design

**Scope:** Multiple layers, multiple aspects.



### Support

Viewpoints that show how elements are supported by other elements, often those in the layers inmediatly below them.

#### Product

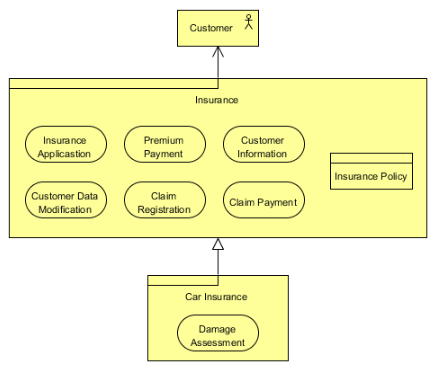
The product viewpoint is centered around the value our services or products give to the end customers It shows the composition of the product in terms of the constituting services, contracts and other agreements involved; An interface may also be used to express the terms in which the product is offered.

Stakeholders: Product developers, product managers, process and domain architects.

Concerns: Product development, value offered analysis.

Purpose: Design, decision making.

Scope: Multiple layers, multiple aspects



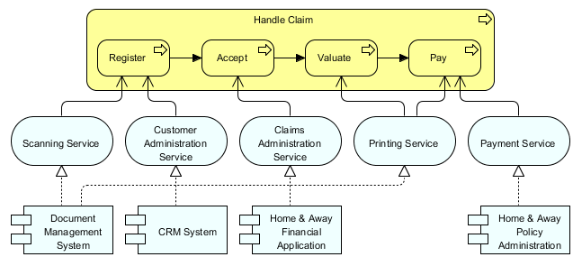
#### Application Usage

Visualizes how applications work together to support business processes and how apps are used by other applications. It can be useful in identifying the services needed to support a business process and other applications or in designing business processes around the available services.

**Stakeholder:** Enterprise, process, and application architects, operational managers.

**Concerns:** Consistency and completeness, reduction of complexity.

**Scope:** Multiple layers, multiple aspects.



#### Technology Usage

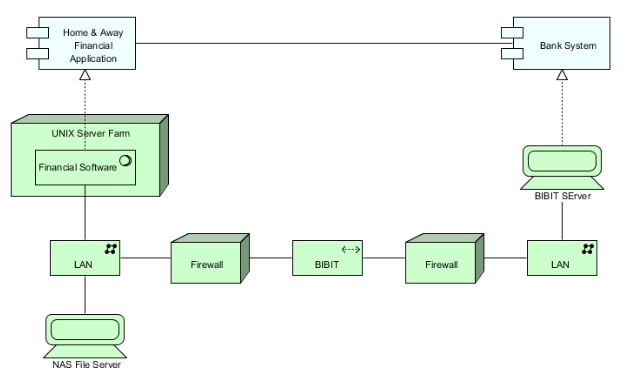
Shows how applications are supported by the underlying software and hardware technology. Normally used when analyzing performance or scalability.

Stakeholders: Application, infrastructure architects, operational managers.

Concerns: Dependencies, performance, scalability.

Purpose: Design

Scope: Multiple layers, multiple aspects.



### Cooperation

Shows how elements cooperate with other elements in their layer, typically across aspects.

#### Business Process Cooperation

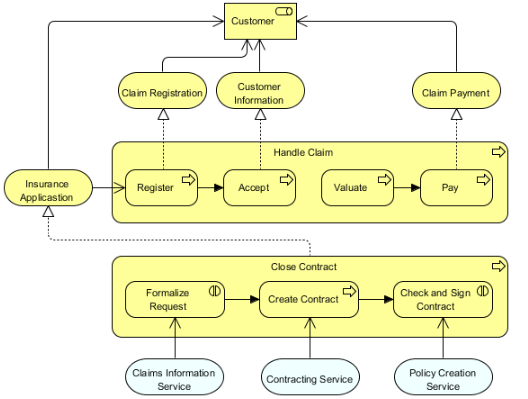
Used to model the main flow of business of an enterprise and can be used to create a high-level design of business processes. Can also be used to map the realization of business processes to business services and business functions.

**Stakeholders:** Process and domain architects, operational managers.

**Concerns:** Dependencies between business processes, consistency and completeness, responsibilities.

**Purpose:** Design, decision making

**Scope:** Multiple layers, multiple aspects.



#### Application Cooperation

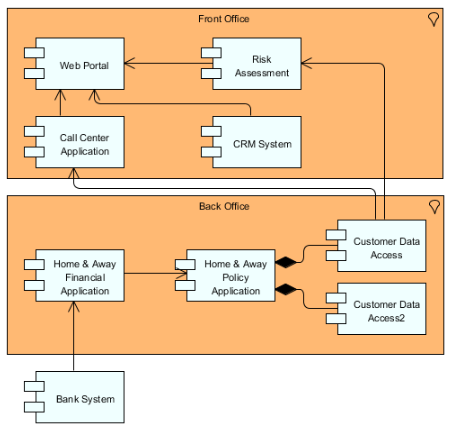
Similarly to the business cooperation viewpoint, the application cooperation viewpoint models the flow of information between application components, as well as the services these require.

**Stakeholders:** Enterprise, process, application and domain architects

**Concerns:** Relationships and dependencies between applications, choreography of services, consistency and completeness, reduction of complexity.

**Purpose:** Design

**Scope:** Multiple layers, multiple aspect.



### Realization

These are viewpoints whose focus is to show how elements realize other elements, typically form one layer down upwards.

#### Service Realization

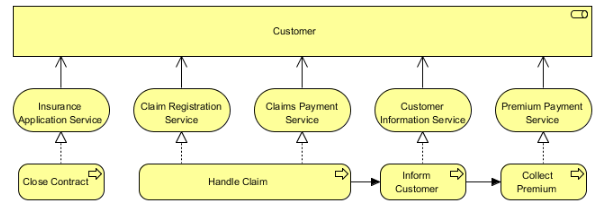
Shows how business services are realized by the underlying application components and business processes.

**Stakeholders:** Process and domain architects, product and operational managers.

**Concerns:** Added value of business processes, consistency and completeness, responsibilities.

**Purpose:** Design and decision making.

**Scope:** Multiple layers and multiple aspects.



#### Implementation and Deployment

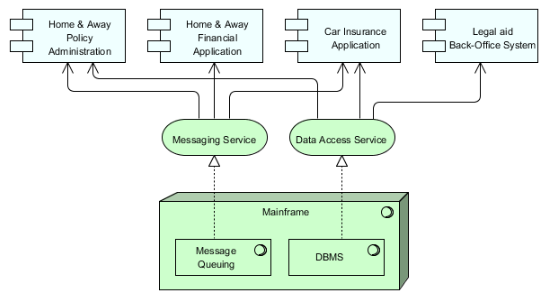
Shows the realization of applications on the infrastructure, mapping applications and components onto artifacts and the information used by these applications onto storage infrastructure.

Stakeholders: Application and domain architects

Concerns: Structure of application platforms and how they relate to supporting technology.

Purpose: Design and decision making.

Scope: Multiple layers, multiple aspects.



### Motivation

Viewpoints used to model the motivational aspects of the architecture.

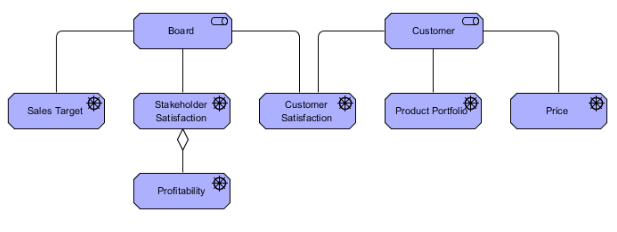
#### Stakeholder

Used to mode stakeholders, internal and external drives of change, and the assessment of these drivers. May also be used to model links to the initial goals that address these concerns.

Stakeholders: Stakeholders, business managers, enterprise and ICT architects, business analysts, requirement managers.

Concerns: Design, decision making, informing.

Scope: Motivation layer.



#### Goal Realization

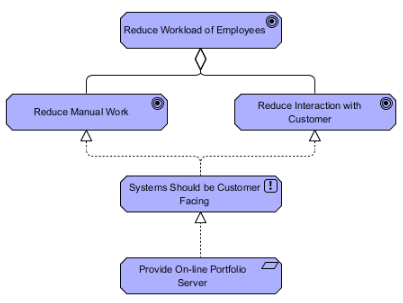
Models the refinement of high-level goals into smaller, more specific goals and these into actionable requirements and constraints.

**Stakeholders:** Stakeholder, business managers, enterprise and ICT architects, business analysts, requirements managers.

**Concerns:** Architecture mission, strategy and tactics, motivation.

**Purpose:** Design, decision making.

**Scope:** Motivation.



#### Requirements Realization

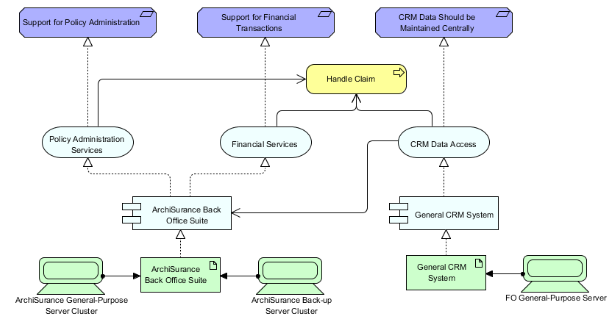
Shows the realization of requirements by core elements.

**Stakeholders:** Enterprise and ICT architects, business analysts, requirement managers.

**Concerns:** Architecture strategy and tactics, motivation.

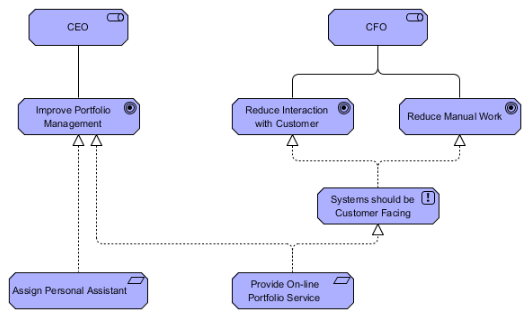
**Purpose:** Design, decision making, informing.

**Scope:** Motivation.



#### Motivation

Presents a complete or partial overview of the motivation aspect by relating processes, applications, services and objects with stakeholders, their primary goals, and principles derived.



### Strategy

Viewpoints used for describing the strategic aspect of the enterprise in its high-level direction and make-up.

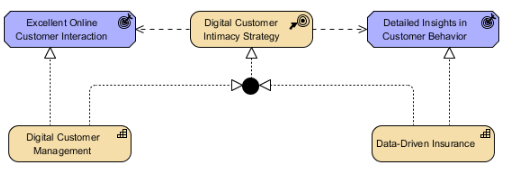
#### Strategy

**Stakeholders:** CxOs, business managers, enterprise and business architects.

**Concerns:** Strategy development

**Purpose:** Design, decision making.

**Scope:** Strategy.



#### Capability Map

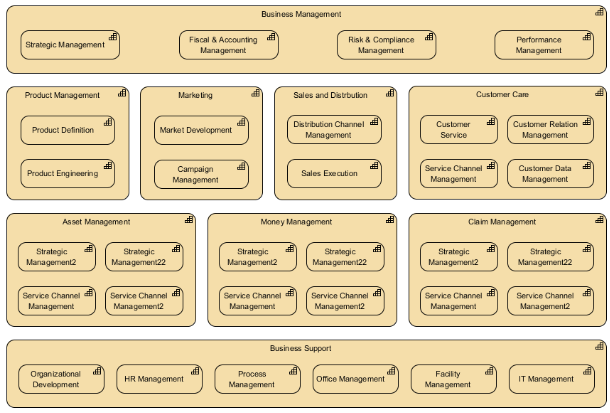
Creates an overview of the capabilities of the enterprise; Typically, it shows two to three levels of capability across the entire enterprise.

Stakeholders: Business managers, enterprise and business architects.

Concerns: Architecture strategy and tactics, motivation.

Purpose: Design, decision making.

Scope: Strategy.



#### Outcome Realization

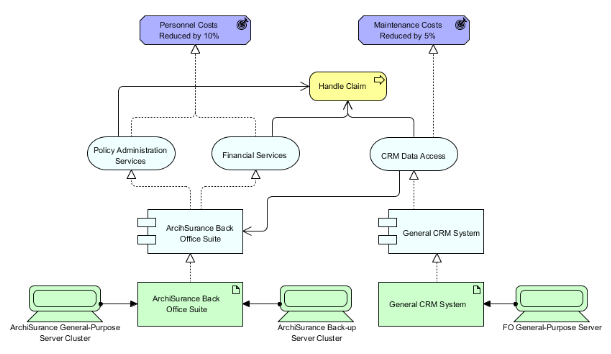
Used to show how the highest-level business-oriented results are produced y teh capabilities and underlying core concepts.

**Stakeholders:** Business managers, enterprise and business architects.

**Concerns:** Business oriented results.

**Purpose:** Design, decision making.

**Scope:** Strategy.



#### Resource Map

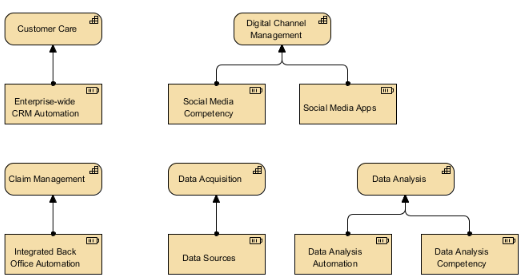
Shows a structured overview of the resources of the enterprise; These typically involve two to three layers of resources across the entire enterprise.

**Stakeholders:** Business managers, enterprise and business architects.

**Concerns:** Architecture strategy and tactics, motivation.

**Purpose:** Design, decision making.

**Scope:** Strategy.



### Implementation and Migration

Implementation and migration viewpoints are defined for modeling the management of architecture as it changes from the baseline to the target architecture.

#### Project

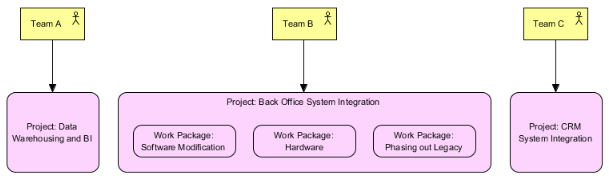
The project viewpoint is used to show the management of the architecture as it changes, modeling the project to be managed, governed, and delivered during said migration.

Stakeholders: Operational managers, enterprise and ICT architects, employees, shareholders.

**Concerns:** Architecture vision and policies, motivation.

**Purpose:** Decision making, informing.

**Scope:** Implementation and migration layer.



#### Migration

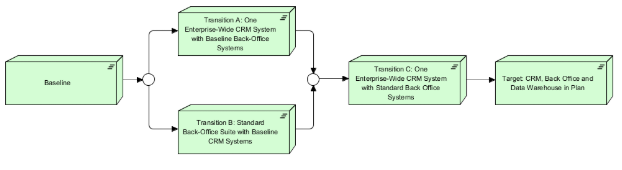
Used to model the transition from an existing architecture to a desired architecture by their stages.

Stakeholders: Enterprise, process, application, infrastructure, and domain architects, employees, and shareholders.

Concerns: Model history.

Purpose: Design, decision making informing.

Scope: Implementation and migration.



#### Implementation and Migration

Relates the programs, projects, and project activities to the infrastructure that they affect.

Stakeholders: Operational managers, enterprise and ICT architects, employees, shareholders.

Concerns: Architecture vision and policies, motivation.

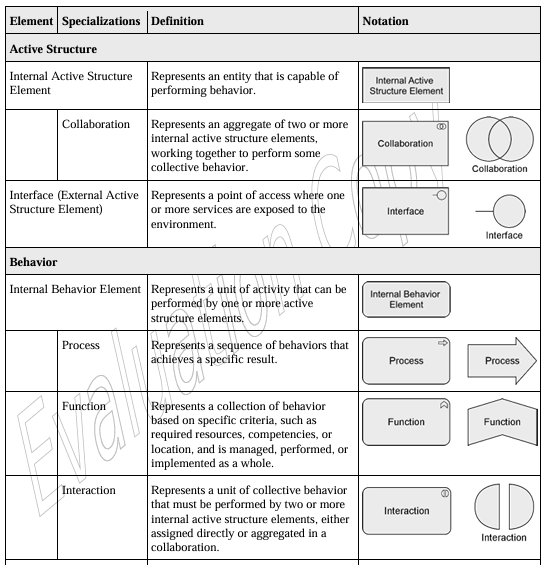
Purpose: Decision making, informing.

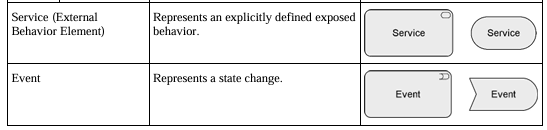
Scope: Multiple layers, multiple aspects.

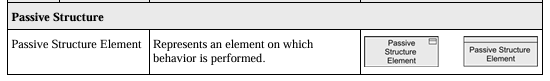


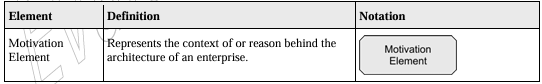
## Review Tables

### General Elements

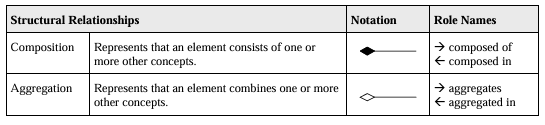


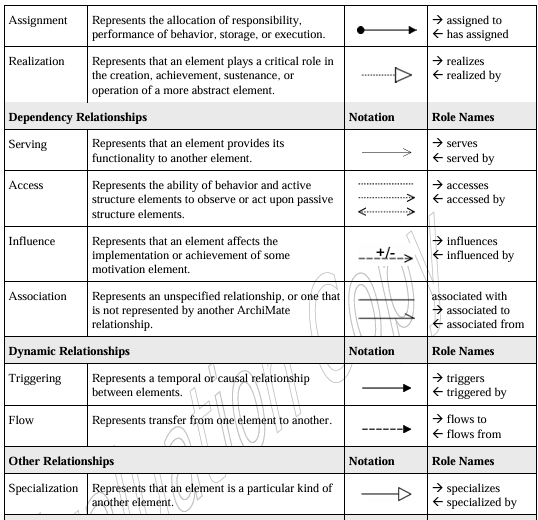


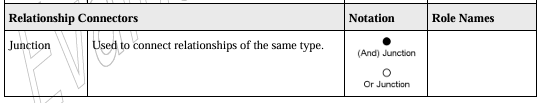




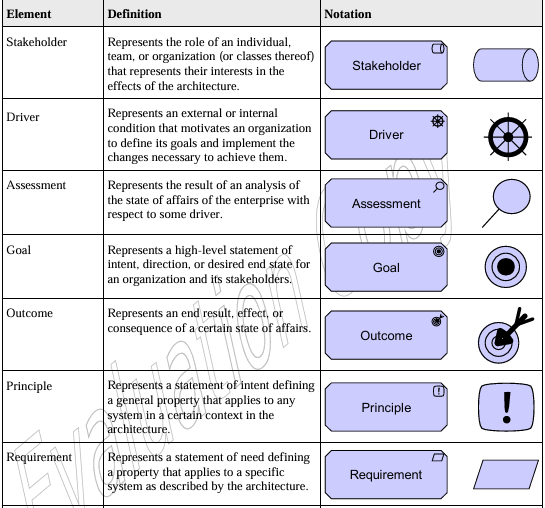
### Relationships

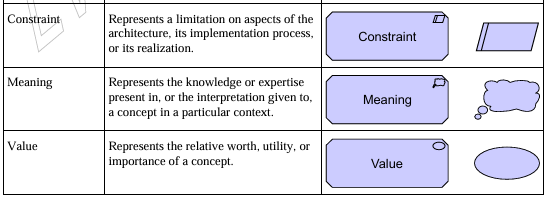




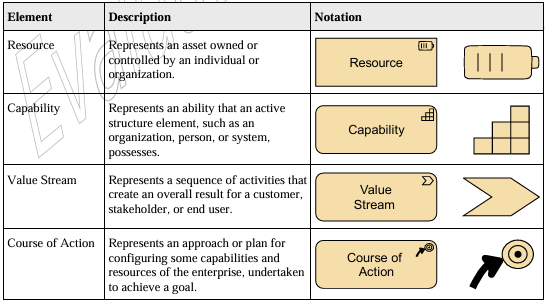


### Motivation Elements

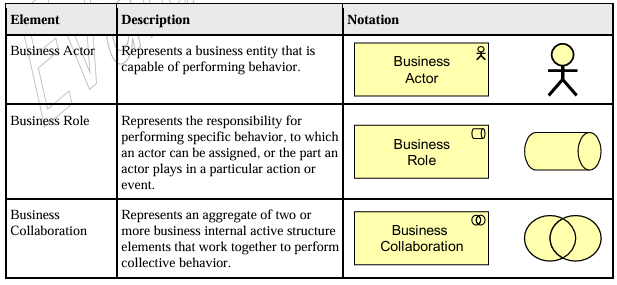


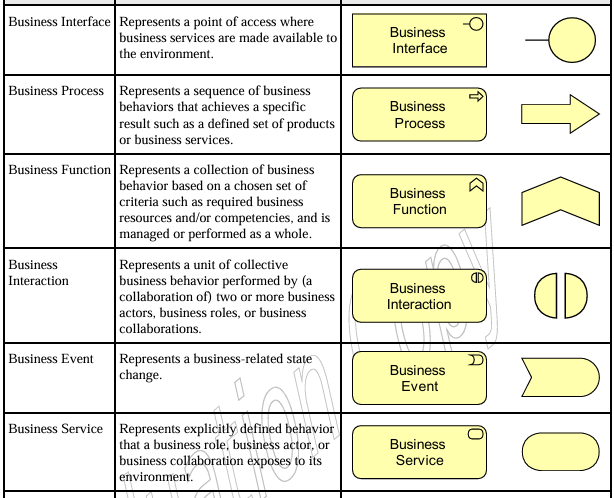


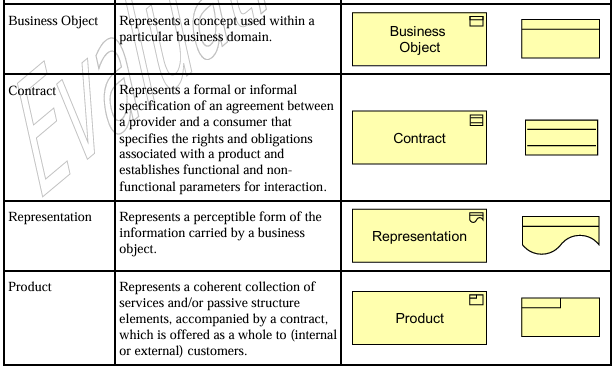
### Strategy Layer Elements



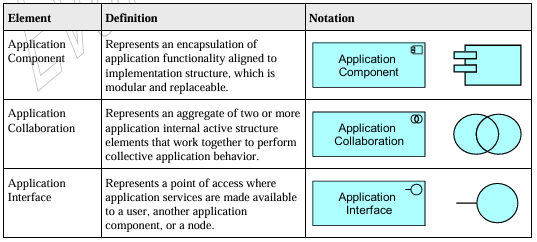
### Business Layer Elements

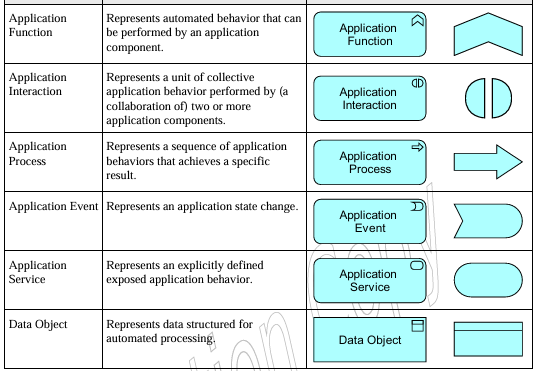




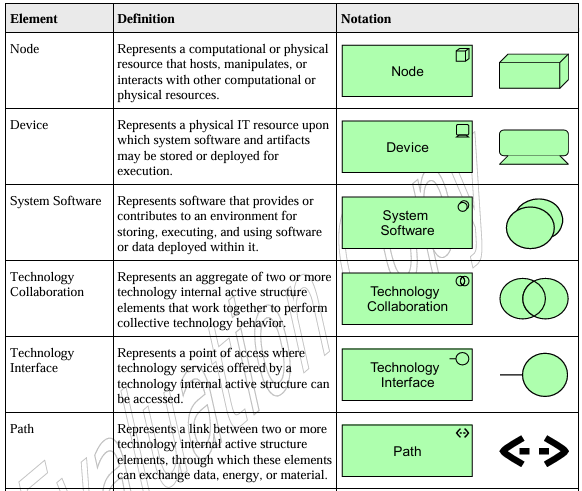


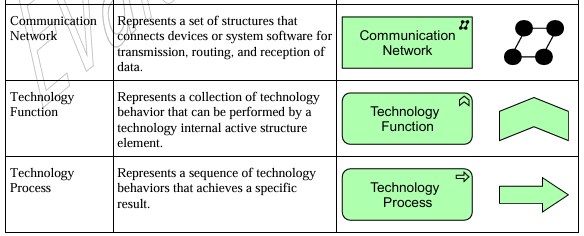
### Application Layer

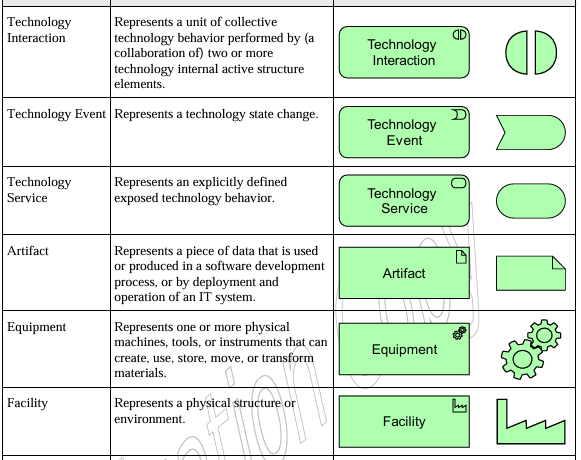


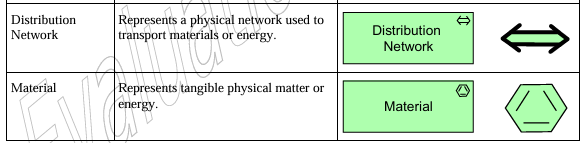


### Technology Layer

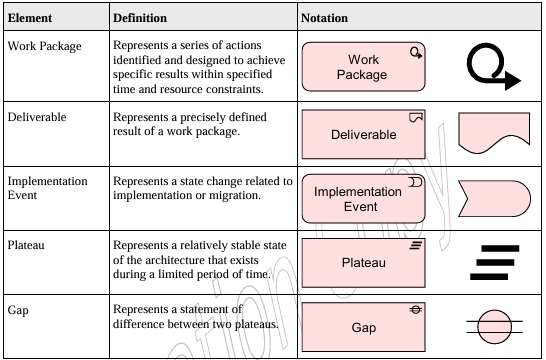




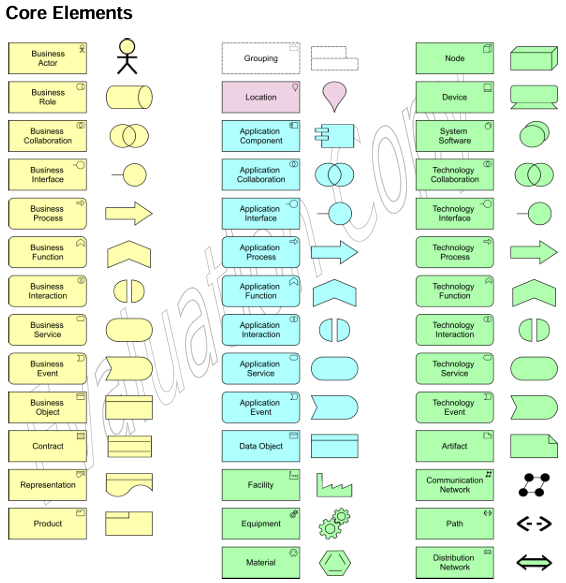


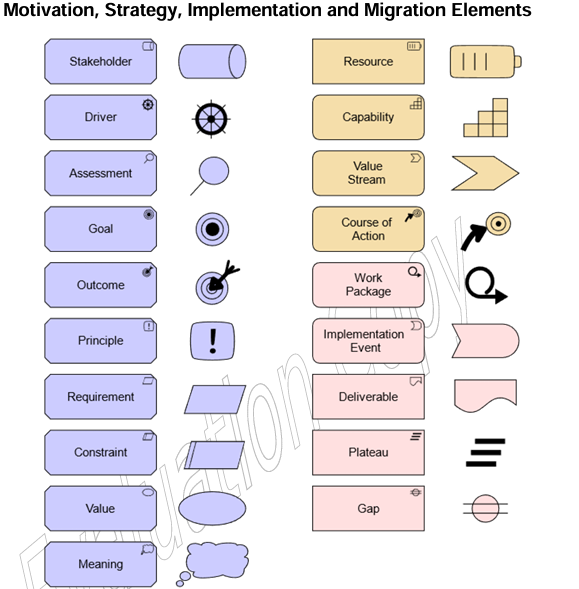


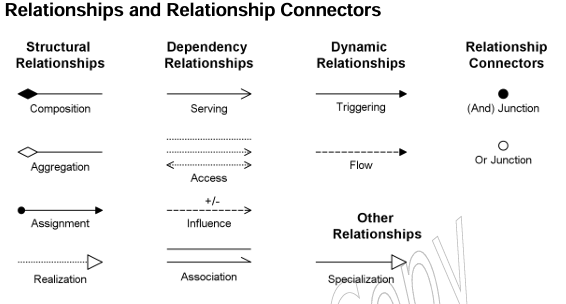
## Implementation and Migration Elements



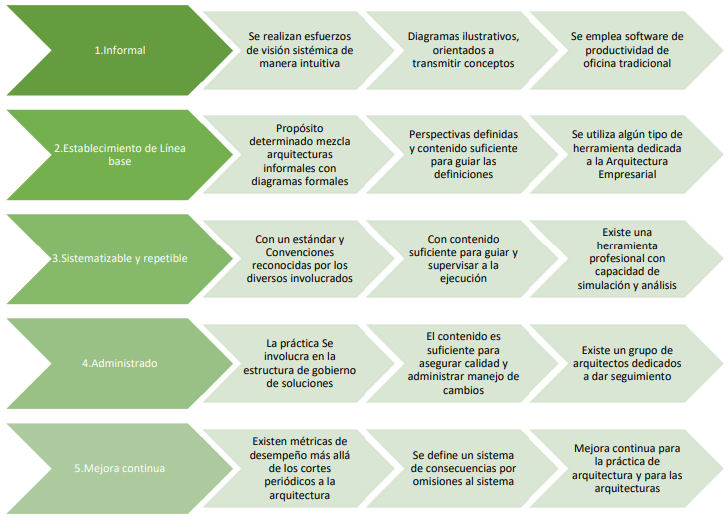
## Summary



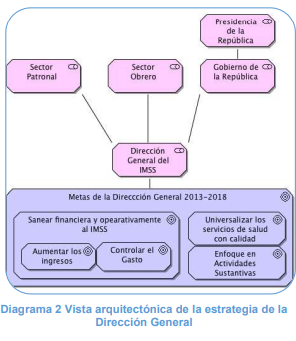


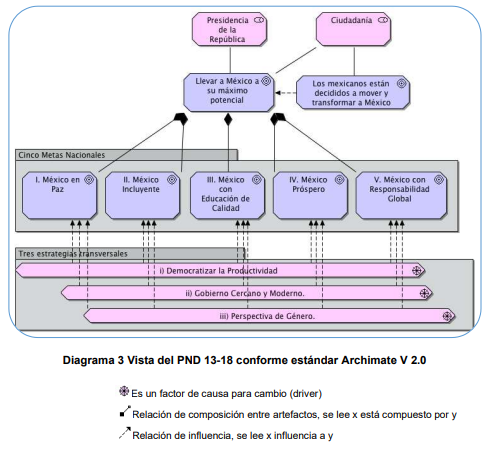


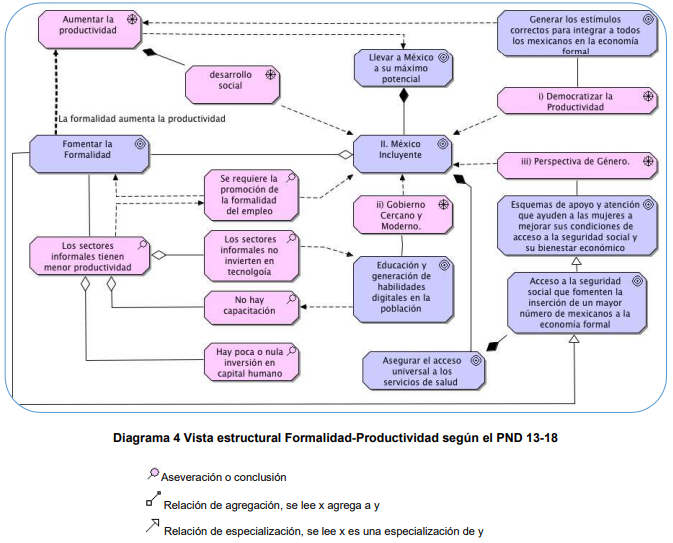
## Notas about the IMMSS



-- Motivadores de Iniciativa --







- Technology systems in IMSS have so far been developed as needs arise without a standardized development system; This has caused the various systems to have limited intercompatiility and large costs incurred by the integration shuffle.

* **SINDO:** Sistema Integral de Derechos y Obligaciones. ; Año 1982; Lenguaje: Cobol; Tecnología: Mainframe; Histórico de 90 millones de mexicanos y 3 mil millones de registros
* **SISCOB:** Sistema de Cobranza; Año 1997; Lenguaje: Cobol; Tecnología: Mainframe; Recibe pagos de 18 mil millones de pesos mensuales
* **IDS:** IMSS desde su Empresa; Año 2002; Lenguaje: Java; Tecnología: Windows/Oracle; Registro de 1.6 millones de movimientos mensuales de altas, bajas y cambios derivadas de las relaciones obrero patronales.
* **SIMF:** Sistema de Información de Medicina Familiar; Año 2002; Lenguaje: Java; Tecnología: Windows/IBM; Registro de casi 50 millones de expedientes clínicos electrónicos y soporta la operación de atención de Medicina Familiar con casi 300 mil consultas diarias en 1,229 Unidades Médicas.
* **PREI:** Planeación de Recursos Institucionales; Año 2003; Lenguaje y tecnología: PeopleSoft/Oracle; Registro y operación transaccional de Contabilidad, Activo Fijo, Control de Compromisos, Presupuestos, Cuentas por Pagar, Tesorería, Inversiones Financieras.
* **SAI:** Sistema Abasto Institucional; Año 1999; Lenguaje: Delphi; Tecnología: Oracle; Control de abasto institucional de medicamentos con casi 665 mil transacciones diarias a nivel nacional.
* **SIPARE:** Sistema de Pago Referenciado; Año 2013; Lenguaje: Java; Tecnología: Oracle; Pago de cuotas obrero patronales por Internet con 267,268 patrones dados de alta y 584,429 líneas de capturas generadas.

*El mantenimiento y soporte de este ambiente resulta altamente costoso e ineficiente, otra característica de este ambiente es que su diseño e implantación se ha enfocado en el desarrollo de sistemas autónomos, generando en las distintas capas del “subsuelo” bancos de información aislados que están orientados a atender necesidades muy específicas de cada área del Instituto, y soportando los modelos vigentes de atención y operación del IMSS.*

