Content Framework :

Architects executing the ADM will produce a number of output as a result of their efforts such as

- Process flow
- Architecture requirements
- Project plans
- Project compliance assessment etc.

The Content framework provides a structural model for architectural content that allows the major work products that an architect creates to be consistently defined, structured and presented.

The Content Framework provided here is intended to allow the TOGAF framework to be used as standalone framework for architecture within an enterprise. However, other content framework exists (Such as Zachman framework)

The architecture content framework uses the following three categories to describe type of architecture work product within the context of use:

- Deliverable : is a work product that is contractually specified and in turn formally reviewed, agreed and signed off by the stakeholders.

  Deliverable represent the output of the project and those deliverables that are in documentation form will typically be archived at completion of a project, or transitioned into the Architecture Repository as a reference model, standard or snapshot of the Architecture landscape at a point in time.

- Artifact : is an architectural work product that describes an aspect of the architecture.

  Artifact are generally classified as catalogs (list of things), metrices (showing relationship between things) and diagram (picture of things).

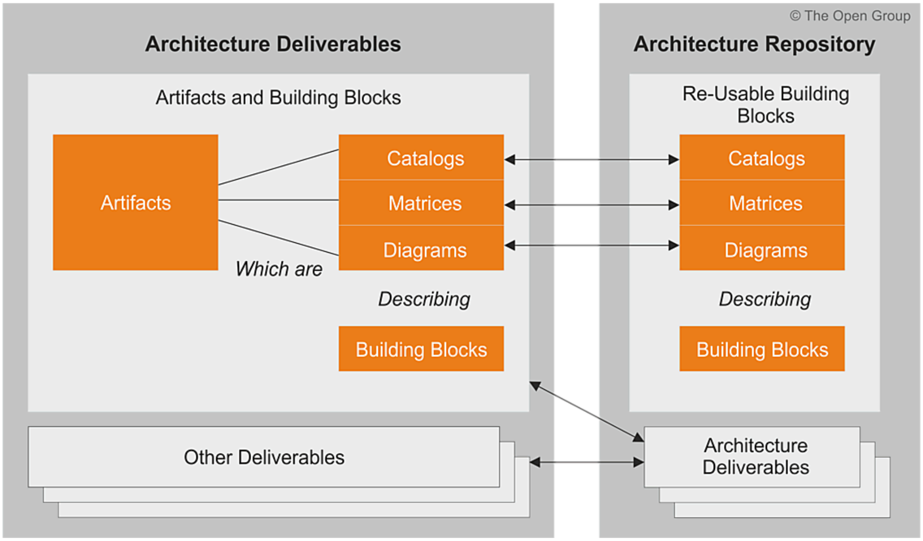
  Example include a requirement catalog, business interaction matrix, and use case diagram.
  An architectural deliverables may contain many artifacts and artifacts will form the content of the Architecture Repository.

- Building block : represents (potentially re-usable) components of enterprise capability that can be combined with other building blocks to deliver architectures and solutions.

  Building blocks can be defined at various levels of detail, depending on what stage of architecture development has been reached. For instance, at an early stage, a building block can simply consist of a name or an outline description. Later on, a building block may be decomposed into multiple supporting building blocks and may be accompanied by full specification. Building block can relate to "architecture" or "solution".

Architecture building block : typically describe required capability and shape the specification of SBBs for example a customer service capability may be required within an enterprise, supported by many SBBs such as processes, data and application software.

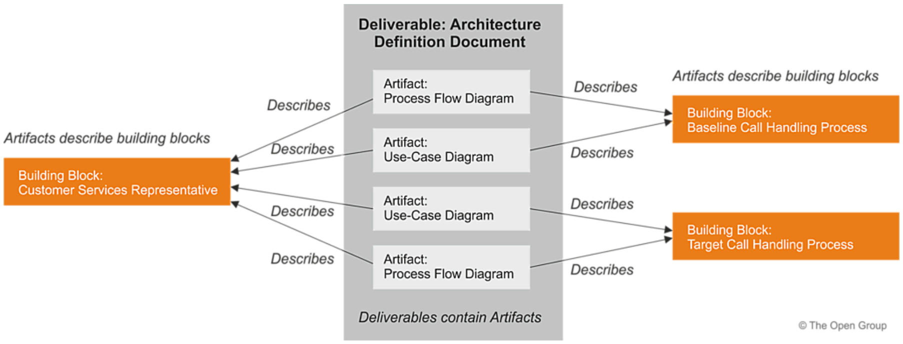
Solution building block (SBB) : represents components that will be used to implement requirement capability for example a network is a building block that can be described through complementary artifacts and the put to use to realized solutions for enterprise.



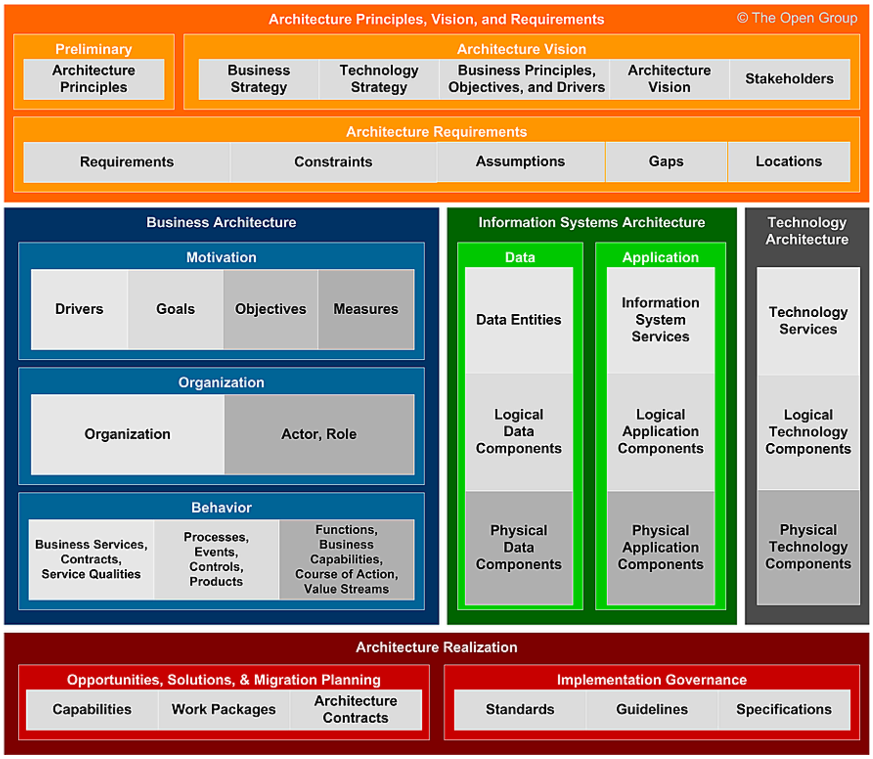
For example, an Architecture Definition Document is a deliverable that documents an Architecture Description. This document will contain a number of complementary artifacts that are architecture views of the building blocks relevant to the architecture. For example, a process flow diagram (an artifact) may be created to describe the target call handling process (a building block). This artifact may also describe other building blocks, such as the actors involved in the process (e.g., a Customer Services Representative).

Ex; Statement of architecture work is a deliverable because it is an contract and this can contain Network diagram of an enterprise, stakeholder matrix which is a artifact . Deliverable can contain many artifact.

Artifacts are used to describe Building blocks


Deliverable: Architecture Definition Document
Artifact: Process Flow Diagram
Artifact: Use-Case Diagram
Artifact: Use-Case Diagram
Artifact: Process Flow Diagram
Deliverables contain Artifacts
Artifacts describe building blocks
Building Block: Customer Services Representative
Describes
Describes
Describes
Describes
Artifacts describe building blocks
Building Block: Baseline Call Handling Process
Building Block: Target Call Handling Process
Describes
Describes
Describes
Describes
© The Open Group

Content Metamodel : Provides a definition of all the types of building blocks that may exist within an architecture, showing how these building block can be described and relate to one another.


Architecture Principles, Vision, and Requirements
© The Open Group
Preliminary
Architecture Principles
Architecture Vision
Business Strategy
Technology Strategy
Business Principles, Objectives, and Drivers
Architecture Vision
Stakeholders
Architecture Requirements
Requirements
Constraints
Assumptions
Gaps
Locations
Business Architecture
Motivation
Drivers
Goals
Objectives
Measures
Organization
Organization
Actor, Role
Behavior
Business Services, Contracts, Service Qualities
Processes, Events, Controls, Products
Functions, Business Capabilities, Course of Action, Value Streams
Information Systems Architecture
Data
Data Entities
Logical Data Components
Physical Data Components
Application
Information System Services
Logical Application Components
Physical Application Components
Technology Architecture
Technology Services
Logical Technology Components
Physical Technology Components
Architecture Realization
Opportunities, Solutions, & Migration Planning
Capabilities
Work Packages
Architecture Contracts
Implementation Governance
Standards
Guidelines
Specifications

The ADM will address a business need through a process of visioning, architecture definition, transformation planning and architecture governance.

30: Content Meta model

The TOGAF ADM provides a process lifecycle to create and manage architecture within an enterprise.

At each phase within the ADM a discussion of inputs, outputs and steps describes a number of architectural work product or artifacts, such as process and application.

The content metamodel provided here defines a formal structure for these term to ensure consistency within the ADM and also provide guidance for organizations that wish to implement their architecture within an architecture tool.

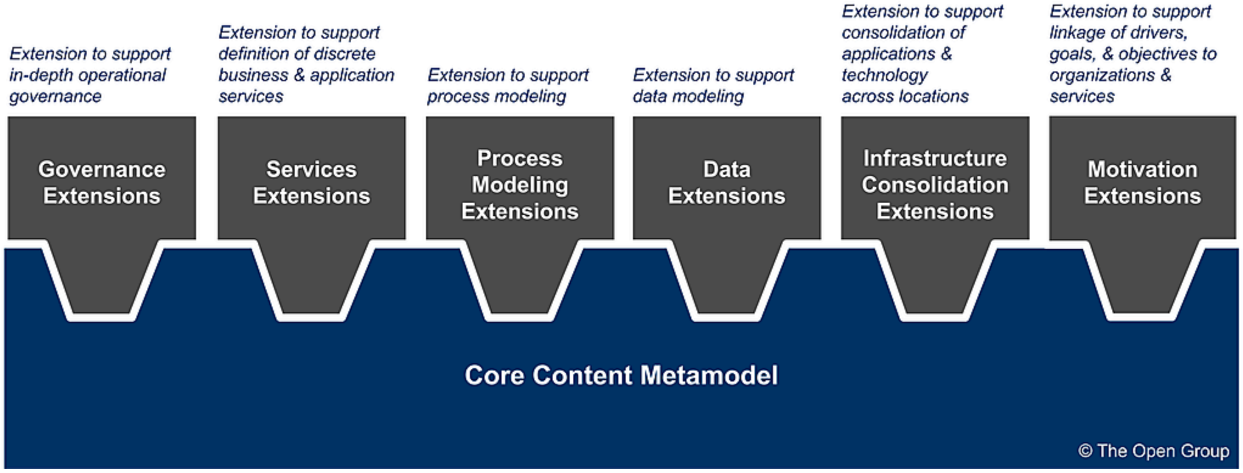
Content metamodel :

- Core content metamodel concepts identifies key concepts within the core content metamodel including :
  - Core and extension content
  - Formal and informal modeling
  - Core metamodel entities.

Core and Extension Context : Provides an introduction to the way in which the TOGAF framework employs a basic core metamodel and then applies a number of extension modules to address specific architectural issues in more detail.

Core Metamodel entities : introduces the core TOGAF metamodel entities, showing the purpose of each entity and key relationship that support architectural traceability.

Metamodel must provide a basic model with the minimum feature set and then support the inclusion of optional extensions during engagement tailoring.


Extension to support in-depth operational governance
Extension to support definition of discrete business & application services
Extension to support process modeling
Extension to support data modeling
Extension to support consolidation of applications & technology across locations
Extension to support linkage of drivers, goals, & objectives to organizations & services
Governance Extensions
Services Extensions
Process Modeling Extensions
Data Extensions
Infrastructure Consolidation Extensions
Motivation Extensions
Core Content Metamodel
© The Open Group

The core metamodel provides a minimum set of architectural content to support traceability across artifacts. Additional metamodel concepts to support more specific or more in depth modeling are contained within a group of extensions that logically cluster extension, catalogs, matrices and diagram allowing focus in areas of specific interest and focus.

All extension modules are optional and should be selected during the Preliminary phase of ADM to meet the needs of the organization.

Core Metamodel entities : The content metamodel uses the terminology discussed within the TOGAF ADM as the basis for formal metamodel. The following core term are used :

- **Actor** : A person, organization or system that is outside the consideration of Architecture model but interacts with it.
- **Application component** : An encapsulation of application functionality that is aligned to implementation structure.
- **Business capability** : a particular ability that a business may posses or exchange to achieve a specific purpose.
- **Course of action** : Direction and focus provided by strategic goal and objectives, often to deliver the value proposition characterized in the business model.
- **Date Entity** : an encapsulation of data that is recognized by business expert as a discrete concept.

Data entity can be tied to application, repository and services and may be structured according to implementation consideration.

- **Function** : delivers business capability closely aligned to an organization but not explicitly governed by the organization.
- Information system service : the automated element of a business service.

Information system service may deliver or support part or all of one or more business service.

- **Organization unit** : Self contained unit of resources with goals, objectives and measures.

  Organization unit may include external parties and business partner organization.

- **Role** : an actor consumes a role to perform a task.
- **Technology component** : an encapsulation of technology infrastructure that represents a class of technology product or specific technology product.
- **Technology service** : a technical capability required to provide enabling infrastructure that supports the delivery of application.
- **Value stream** : a representation of and end-to-end collection of value-adding activities that create an overall result for customer, stakeholder or end-user.

Key relationship concepts related to core metamodel entities are described below:

- **Process should normally be used to describe flow** : A Process is a flow of interaction between function and services and can not by physically deployed

All process should describe the flow of execution for a function and therefore deployment of a process is through the function it support. I.e. an application implements a function that has a process not an application implements a process.

- **Function describes unit of business capability at all levels of granularity** - function encapsulate terms such as value chain, process area, capability, business function etc.

- **Business service support organizational objectives and are defined at a level of granularity consistent with level of governance needed.**

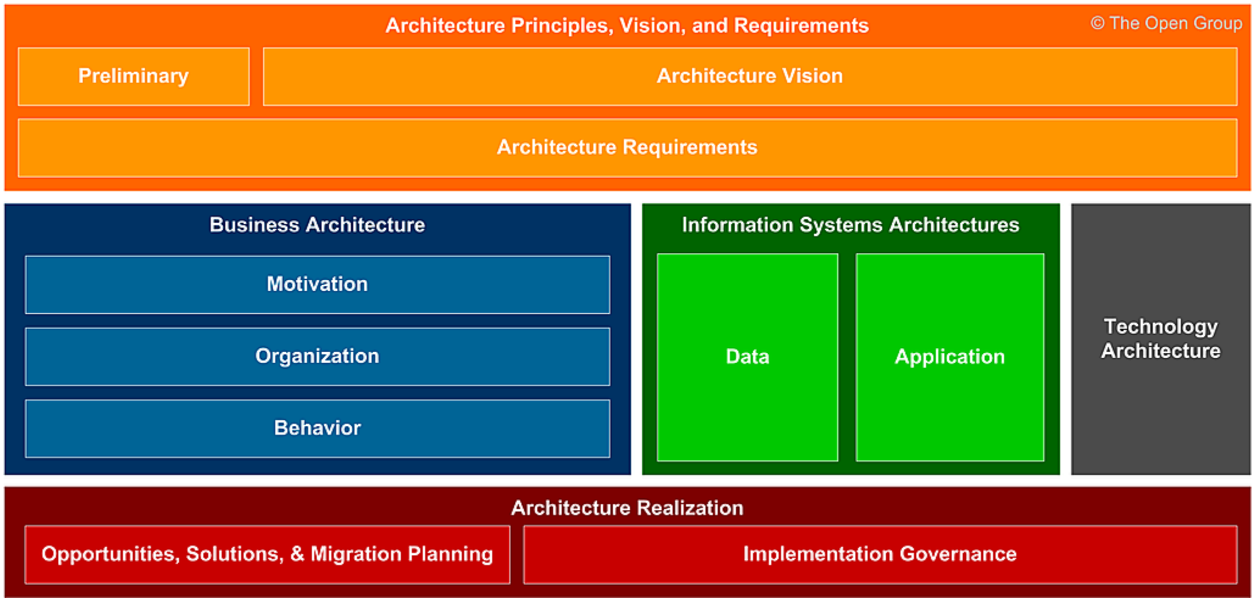
A Business service operates as a boundary for one or more functions. The granularity of business services is dependent on the focus and emphasis of the business (as reflected by its drivers, goals, and objectives). A service in Service-Oriented Architecture (SOA) terminology (i.e., a deployable unit of application functionality) is actually much closer to an application service, application component, or technology component, which may implement or support a business service.

- **Business services are deployed onto application components** : Business service may be realized by business activity that does not relate to IT or may be realized through IT.

Application components can be hierarchically decomposed and may support one or more business service. It is possible for a business service to be supported by multiple application components, but this is problematic from governance standpoint and is symptomatic of business service that too coarse grained or application component that are too fine grained.

- **Application components are deployed onto technology component.**

The Content Metamodel defines a set of entities that allow architectural concepts to be captured, stored, filtered, queried and represented in a way that supports consistency, completeness and traceability.


Architecture Principles, Vision, and Requirements
© The Open Group
Preliminary
Architecture Vision
Architecture Requirements
Business Architecture
Motivation
Organization
Behavior
Information Systems Architectures
Data
Application
Technology Architecture
Architecture Realization
Opportunities, Solutions, & Migration Planning
Implementation Governance

Architecture principles, Vision and Requirements : Entities are intended to capture the surrounding context of formal architecture model, including general architecture principles, strategic context that form input for architecture modeling and requirements generated from the architecture.

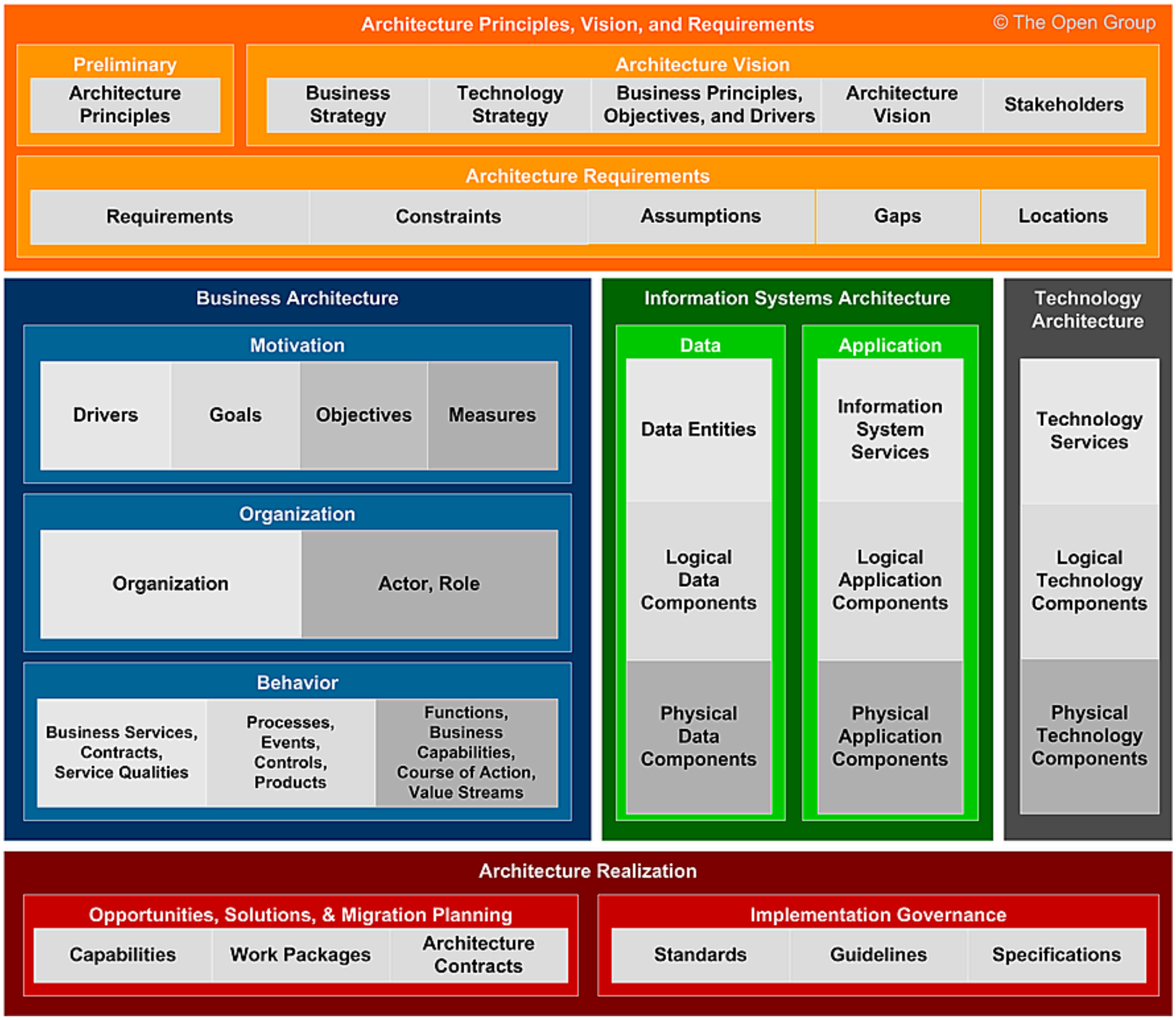
Note : Architecture context is typically collected in the Preliminary and Architecture Vision phases.

Business architecture : Entities capture architecture models of business operations, looking specifically at factors that motivate the enterprise, how enterprise is organizationally structured and also what business capabilities the enterprise has.

Information system architecture : Entities capture architecture model of IT systems, looking at application and data in line with TOGAF ADM Phases

Technology architecture : Entities capture procured technology assets that are used to implement and realize information system solutions.

Architecture Realization : Entities capture change roadmaps showing transition between architecture states and binding statement that are used to steer and govern an implementation of the architecture.



© The Open Group

Architecture Principles, Vision, and Requirements

Preliminary
- Architecture Principles

Architecture Vision
- Business Strategy
- Technology Strategy
- Business Principles, Objectives, and Drivers
- Architecture Vision
- Stakeholders

Architecture Requirements
- Requirements
- Constraints
- Assumptions
- Gaps
- Locations

Business Architecture

Motivation
- Drivers
- Goals
- Objectives
- Measures

Organization
- Organization
- Actor, Role

Behavior
- Business Services, Contracts, Service Qualities
- Processes, Events, Controls, Products
- Functions, Business Capabilities, Course of Action, Value Streams

Information Systems Architecture

Data
- Data Entities
- Logical Data Components
- Physical Data Components

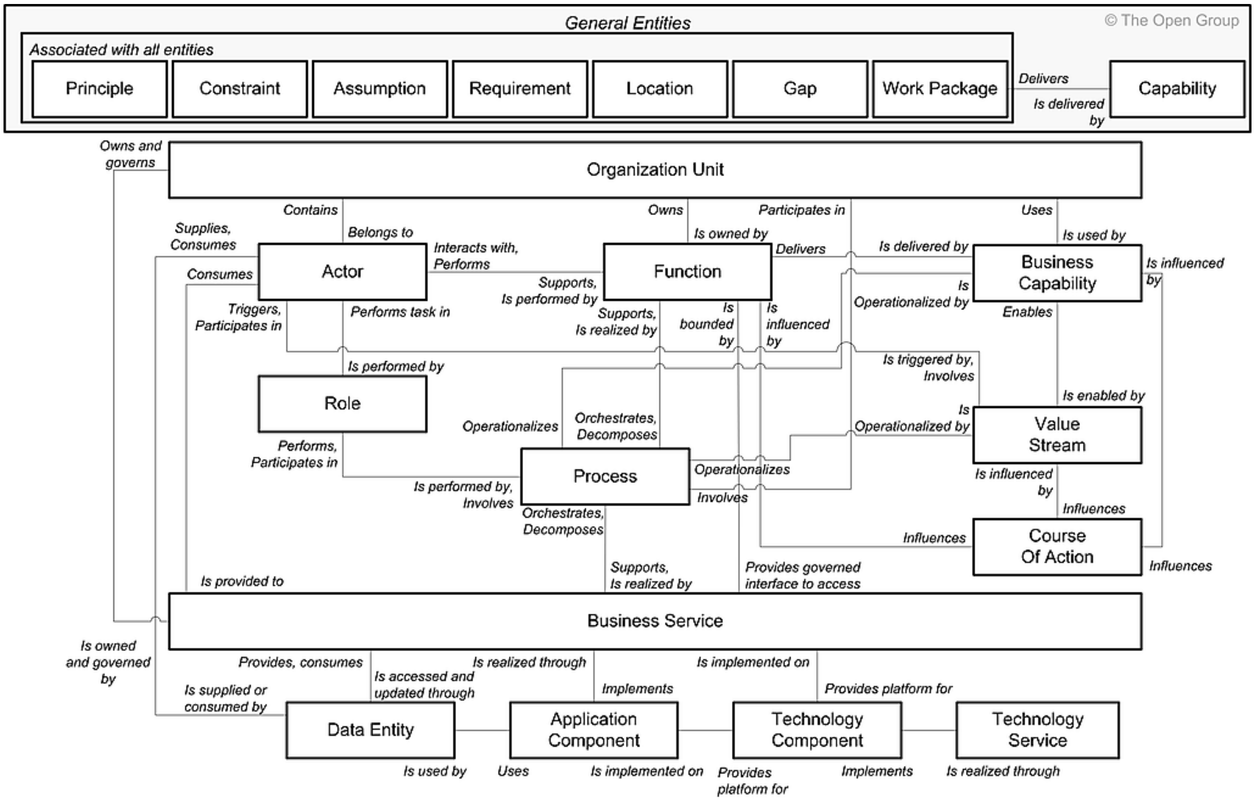
Application
- Information System Services
- Logical Application Components
- Physical Application Components

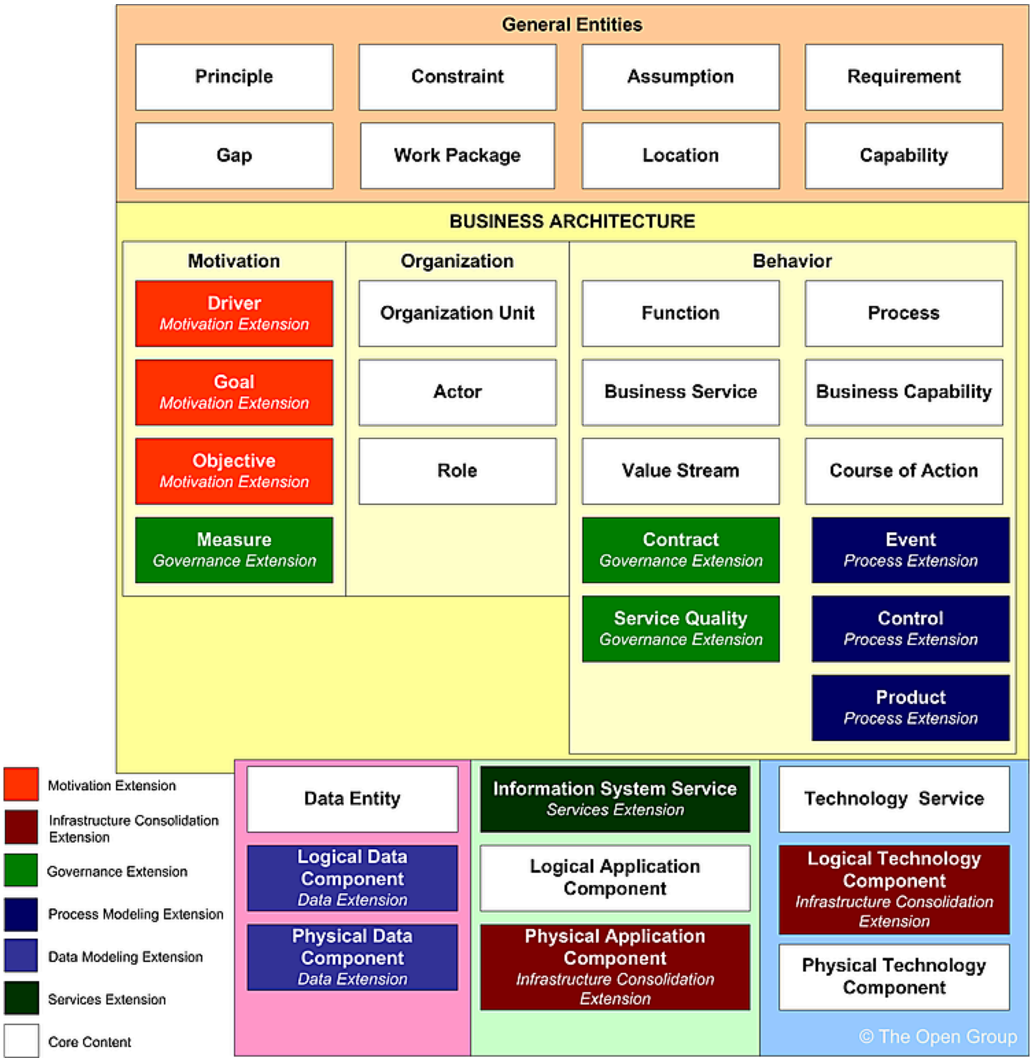
Technology Architecture
- Technology Services
- Logical Technology Components
- Physical Technology Components

Architecture Realization

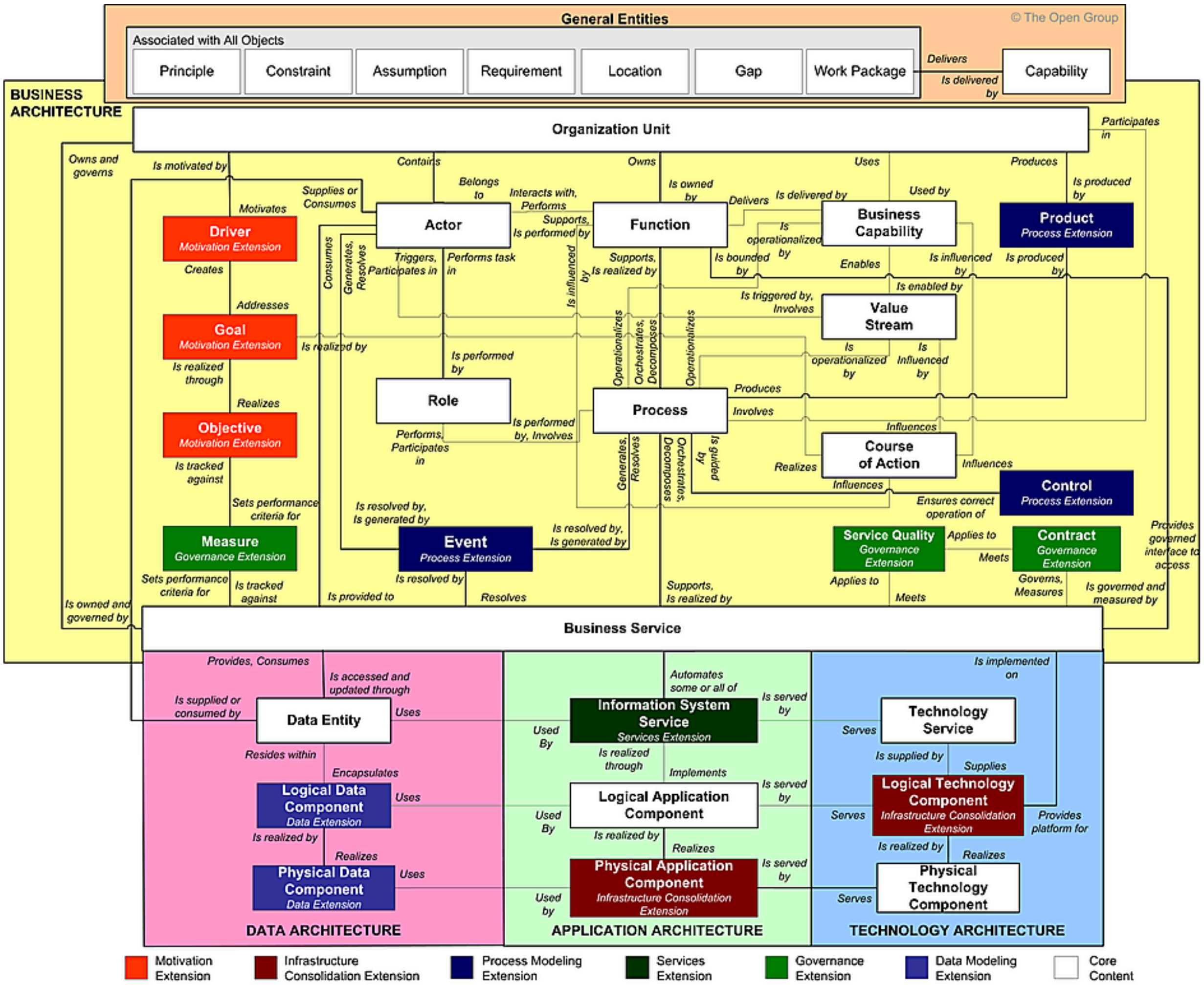
Opportunities, Solutions, & Migration Planning
- Capabilities
- Work Packages
- Architecture Contracts

Implementation Governance
- Standards
- Guidelines
- Specifications


General Entities
© The Open Group
Associated with all entities
Principle
Constraint
Assumption
Requirement
Location
Gap
Work Package
Delivers
Is delivered by
Capability
Owns and governs
Organization Unit
Contains
Owns
Participates in
Uses
Supplies, Consumes
Belongs to
Is owned by
Is used by
Consumes
Actor
Interacts with, Performs
Function
Delivers
Is delivered by
Business Capability
Is influenced by
Supports, Is performed by
Is Operationalized by
Triggers, Participates in
Performs task in
Supports, Is realized by
Is bounded by
Is influenced by
Enables
Is performed by
Is triggered by, Involves
Role
Is enabled by
Operationalizes
Orchestrates, Decomposes
Is Operationalized by
Value Stream
Performs, Participates in
Process
Operationalizes
Is performed by, Involves
Involves
Is influenced by
Influences
Orchestrates, Decomposes
Influences
Course Of Action
Influences
Supports, Is realized by
Provides governed interface to access
Is provided to
Business Service
Is owned and governed by
Provides, consumes
Is realized through
Is implemented on
Is accessed and updated through
Implements
Provides platform for
Is supplied or consumed by
Data Entity
Application Component
Technology Component
Technology Service
Is used by
Uses
Is implemented on
Provides platform for
Implements
Is realized through



Relationship between entitles in full metamodel.



Content Metamodel Extensions :

- Governance Extensions : Extensions to support in-depth operational governance.
- Service Extensions : Extensions to support definition of discrete business and application service.
- Process modelling extension : Extension to support process modelling
- Data Extension : Extension to support data modelling
- Infrastructure Consolidation Extension : Extension to support consolidation of application and technology across location.

- Motivation Extension : Extension to support linkage of drivers, goals, and objectives to organization and service.

Note : During the architecture vision phase of a particular engagement, the scope of the engagement will be used to make a determination on appropriate extensions to be employed in order to adequately address the architecture requirements.

For example the scope of an engagement could be defined as core content, plus the governance extensions.

Governance Extension : Intended to allow additional structured data to be held against objectives and business services, supporting operational governance of the landscape.

The scope of this extension is as follows :

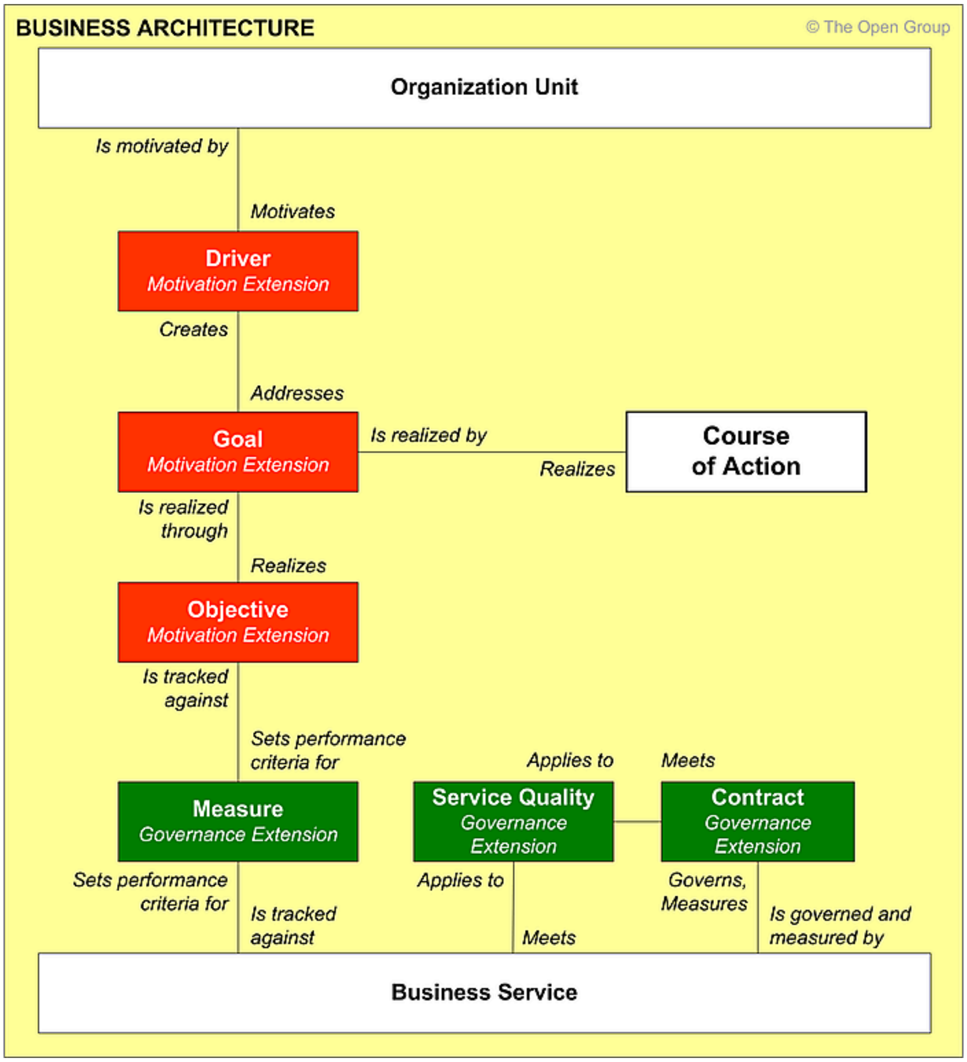
- The ability to apply measures to objectives and then link those measures to services.
- The ability to apply contracts to service communication or service interactions with external users and systems.
- The ability to define re-usable service
- Creation of additional diagrams to show ownership and management of systems.

The extensions should be used in following situations.

- When an organization is considering IT change that will result in significant impact to existing operational and governance model.
- When an organization has granular requirements for service level that differ from service to service.
- When an organization is looking to transform its operational governance practice.
- When an organization has very strong focus on business drivers, goals and objectives and how these trace to service level.

Benefit of using this extension is as follows :

- Service level are defined in a more structured way, with:
    - More detail
    - The ability to re-use service profile across contracts
    - Stronger tracing to business objectives
- Impacts to operations and operational governance model are models are considered in a more structured way, with:
    - Additional diagrams of system and data ownership
    - Additional diagrams of system operation and dependencies on operations processes.



Service extension : The service extension is intended to allow more sophisticated modelling of service portfolio by creating concept of Information System (IS) Service in addition to the core concept of business service.

IS service are directly supported by applications and creating the layer of abstraction relaxes the constraints on business services while simultaneously allowing technical stakeholders to put more formality into IS service catalog.

The scope of extension is as follows:

- Creation of IS Service as an extension of business service.

The extension should be used in the following situations :

- When business has a pre-set definition of its service that does not align well to technical and architectural needs.
- When business and IT use different language to describe similar capabilities.
- Where IT Service is misaligned with business need, particularly around the areas of quality of service, visibility of performance and management granularity.
- Where IT is taking initial steps to engage business in discussion about architecture.

31: ***Architectural Artifacts***

- Architectural artifacts are created in order to describe a system, solution or state of the enterprise.

The "Environment" of a system is the context determining the setting and circumstances of all influences upon a system.
The "environment" of a system includes developmental, technological, business, operational, political, economical, legal, regulatory, ecological and social influences .

A "System" is a combination of interacting elements organized to achieve one or more stated purposes.

The "Architecture" of a system is the fundamental concept or properties of a system in its environment embedded in its elements, relationship and in principles of its design and evolution.

An "Architecture Description" is a work product used to express an architecture, a collection of architecture view and models that together documents the architecture.

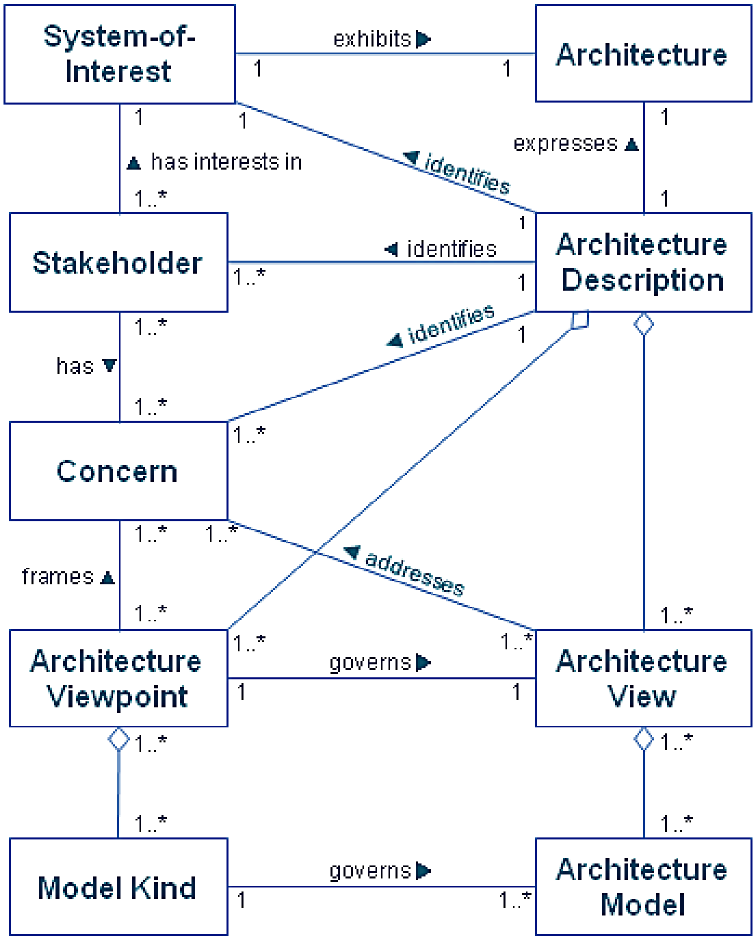
"Stakeholders" are individual, teams, organizations or classes thereof, having interest in a system.

"Concerns" are interest in a system relevant to one or more of its stakeholders. Concerns may pertain to any aspect of the system's functioning, development, or operation, including consideration such as performance reliability, security and evolvability and may determine the accountability of the system.

An "Architecture View" is a representation of system from perspective of a related set of concerns. It consists of one or more architecture models of the system.

An "Architecture Model" is a representation of a subject of interest. A model provides a smaller scale, simplified and/or abstract representation of subject matter.

An "Architecture model" is a representation of a subject of interest. A model provides a smaller scale, simplified and/r abstract representation of the subject matter.



An "Architecture Viewpoint" is a specification of the conventions for a particular kind of architecture view. It can also be called the definition or scheme for that kind of architecture view.
It establishes conventions for constructing, interpreting and using an architecture view to address a specific concern about a system of interest.

A "Model Kind" establishes conventions for a type of modeling.

A "Viewpoint Library" is a collection of the specification of architecture viewpoints contained in the Reference Library Portion of Architecture Repository.

A architecture view is what we see, an architecture viewpoint is where we are looking from - the vintage point or perspective that determine what we see.

Simple example of an Architecture Viewpoint and Architecture View:

Architecture	
Viewpoint Element	Description
Stakeholders	Management Board, Chief Executive Officer
Concerns	Show the top-level relationships between US/UK geographical sites and business functions.
Modelling technique	Nested boxes diagram. Outer boxes = locations; inner boxes = business functions. Semantics of nesting = functions performed in the locations.

-

Architect has a responsibility for ensuring the completeness (fitness for purpose) of the architecture, in terms of adequately addressing all pertinent concern of its stakeholders, and integrity of architecture, in terms of connecting all the various view of each other, satisfactorily reconciling the conflicting concerns of different stakeholders and showing the trade-offs made in so doing.

Architecture view creation process:

- Refer to an existing library of architecture viewpoints
- Select the appropriate architecture viewpoints (based on the stakeholders and concerns that need to be covered by views)
- Generate views of the system by using the selected architecture viewpoints as templates.

-

This approach can be expected to bring the following benefits:

- Less work for architects (Because the architecture viewpoints have already been defined and therefore the views can be created faster)
- Better comprehensibility for stakeholders (Because the architecture viewpoints are already familiar)

  - Greater confidence in the validity of the views (because their architecture viewpoints have a known track record)

- Architecture viewpoint is also normally developed, visualized, communicated and managed using tool, and it is also highly desirable that standard architecture viewpoints (i.e. template or schemas) be developed, so that different tools that deals in the same views can interoperate the fundamental element of an architecture can be re-used and architecture description can be shared among tools.

An architecture viewpoint is a model (or description) of information contained in a view.

Note : each architecture viewpoint is an abstract model of how all the stakeholders of a particular type - all pilots, or all controllers - view the airport system

Tools exist to assist stakeholders, especially when they are interacting with complex models such as the model of an airspace, or the model of air flight

The users of the system have an architecture viewpoint that reflects their concerns when interacting with the system, and the developers of the system have a different architecture viewpoint. Architecture views that are developed to address either of the two architecture viewpoints are unlikely to exhaustively describe the whole system, because each perspective reduces how each sees the system.

The architecture viewpoint of the user is comprised of all the ways in which the user interacts with the system, not seeing any details such as applications or Database Management Systems (DBMS).

The architecture viewpoint of the developer is one of productivity and tools, and doesn't include things such as actual live data and connections with consumers.

In this example, one architecture viewpoint is the description of how the user sees the system, and the other architecture viewpoint is how the developer sees the system. Users describe the system from their perspective, using a model of availability, response time, and access to information. All users of the system use this model, and the model has a specific language.

Developers describe the system differently than users, using a model of software connected to hardware distributed over a network, etc. However, there are many types of developers (database, security, etc.) of the system, and they do not have a common language derived from the model.

These concepts can be summarized as:

- ○ Selecting a key stakeholder
- ○ Understanding their concerns and generalizing/documenting those concerns
- ○ Understanding how to model and deal with those concerns

●

The content metamodel is used as a technique to structure architectural information in an ordered way so that it can be processed to meet the stakeholder needs.

Building blocks are entities of a particular type within the metamodel (for example a business service called "Purchase Order"). Building block carry metadata according to the metamodel, which supports query and analysis. For example business service have a metadata attribute for owner, which allows a stakeholder to query all business services owned by particular organization.

Building blocks may also include dependent or contained entities as appropriate to the context of the architecture (for example, a business service called "Purchase Order" may implicitly include a number of processes, data entities, application components, etc.).

Catalogs are lists of building blocks of a specific type, or of related types, that are used for governance or reference purposes (for example, an organization chart, showing locations and actors).

Matrices are grids that show relationships between two or more model entities. Matrices are used to represent relationships that are list-based rather than graphical in their usage (for example, a CRUD matrix showing which applications Create, Read, Update, and Delete a particular type of data is difficult to represent visually).

Diagrams are renderings of architectural content in a graphical format to allow stakeholders to retrieve the required information. Diagrams can also be used as a technique for graphically populating architecture content or for checking the completeness of information that has been collected.

Recommended artifacts for production in each ADM phases are as follows:

  - Preliminary Phase :
    - Principle catalog
    - Principle
  - Phase A : Architecture Vision
    - Stakeholder Map Matrix : The purpose of the Stakeholder Map matrix is to identify the stakeholders for the architecture engagement, their influence over the engagement, and their key questions, issues, or concerns that must be addressed by the architecture framework

-
  - Value chain diagram : A value chain diagram provides a high-level orientation view of an enterprise and how it interacts with outside world. In contrast with more formal Functional decomposition diagram developed within phase B the value chain diagram focuses on presentational impact.

-
  - Solution Concept diagram : A Solution concept diagram provides a high level orientation of the solution that envisaged in order to meet the objective of architecture engagement. In contrast to more formal and detailed architecture diagram developed in the

following phases, the solution concept represents a "pencil sketch" of the expected solution at the outset of the engagement.

-
  - Business Model Diagram : A model describing the rationale for how an enterprise creates, delivers and captures value.
-
  - Business Capability Map : A family of diagram representing a definitive listing of a particular ability that a business may posses or exchange to achieve a specific purpose.
-
  - Value Stream Map : A family of diagram representing a definitive listing of an end-to-end collection of value-adding activities that create an overall result for a customer, stakeholder or end-user.
-
  - Phase B : Business Architecture
    - Organization/Actor catalog - Purpose is to capture definitive listing of all participants that interact with IT, including users and owners of IT system.
-

The Organization / Actor catalog can be referenced when deveoping requirement in order to test the completeness.

For example, requirements for an application that services customers can be tested for completeness by verifying exactly which customer types need to be supported and whether there are any particular requirements or restrictions for user type.

The Organization/ Actor catalog contains the following metamodel entities

  - Organization unit
  - Actor

  - Location (May be included in this catalog if an independent location catalog is not maintained)
-
  - Driver / Goal/ Objective catalog : The purpose of the Driver/ Goal/ Objective catalog is to provide a cross organizational reference of how an organization meets its drivers in practical terms through goals, objectives and (optionally) measures.
- Publishing a definitive breakdown of drivers, goals, and objectives allows change initiatives within the enterprise to identify synergies across the organization (e.g. multiple organizations attempting to achieve similar objectives), which in turn allow stakeholders to be identified and related change initiative to ab aligned or consolidate. The Driver/Goal/ Objective catalog contains the following metamodel entities:
  - Organization unit
  - Driver
  - Goal
  - Objective
  - Measure (may optionally be included)
-
  - Role Catalog : Listing of all authorization levels or zones within an enterprise.
- If roles are defined, understood, and aligned across organizations and applications, this allows for a more seamless user experience and generally more secure applications, as administrators do not need to resort to workarounds in order to enable users to carry out their jobs.

  In addition to supporting security definition for the enterprise, the Role catalog also forms a key input to identifying organizational changer management impacts, defining job functions and executing end-user training.

  The role catalog contains the following metamodel entities:

  - Role

-

  - Business service/function catalog : *Provides a functional decomposition in a form that can be filtered,* reported on and queries as a supplement to graphical Functional decomposition diagram.

-

The business service/ function catalog can be used to identify capabilities of an organization and to understand the level that governance is applied to the functions of an organization.

This functional decomposition can be used to identify new capabilities required to support business change or may be used to determine the scope of change initiatives, applications, or technology components.

The Business Service/ Function catalog contains the following metamodel entities:

  - Organizational unit
  - Business function
  - Business service
  - Information system service (may optionally be included here)

-

  - Location catalog : **Listing of all locations where enterprise carries out business operation or houses architecturally relevant assets such as data center or end user computing equipment.**

- Maintaining a definitive list of location allows change initiative to quickly define a location scope and to test for completeness when assessing current landscape or propose target solution. For example project to upgrade desktop operating system will need to identify all locations where desktop operating system are deployed.
  Similarly when new system are being implemented a diagram of locations is essential in order to develop appropriate deployment strategy that comprehend both user and application location and identify location related issues such as internationalization, localization, time zone impacts on availability, distance impacts on latency, network impact on

bandwidth and access.
The location catalog contains following metamodel entities:

  - Location

-

  - Process/Event/Control/Product catalog : Provides a hierarchy of processes, events that trigger processes, outputs from processes and controls applied to the execution of processes.

- This catalog provide a supplement to any Process Flow diagram that are created and allows an enterprise to filter report and query across organizations and processes to identify scope, commonalities and impact.
  For example the Process/Event/Control/Product catalog allows an enterprise to see relationship of processes to subprocesses in order to identify the full chain of impact resulting from changing a high level process.

  The process/Event/Control/Product catalog contains the following metamodel entities

  - Process
  - Event
  - Control
  - Product

-

  - Contract/Measure catalog : The contract/measure catalog provides a list of all agreed service contracts and the measures attached to those contracts. It forms the master list of service levels agreed to across enterprise.

- The contract/Measure catalog contains the following metamodel entities.

  - Business service
  - Information system Service (Optional)
  - Contract

  - Measure
-
  - Business capabilities catalog : Definitive listing of particular abilities that a business may posses or exchange to achieve a specific purpose.
  - Value stream catalog : A definitive listing of end-to-end collections of value-adding activities that create an overall result for a customer, stakeholder or end-user.
  - Values stream stages catalog : A definitive listing of end-to-end collections of different stages for the value-adding activities that create an overall result for a customer, stakeholder or end user. It includes the following metamodel entities.
    - Business capability
    - Value Stream
-
  - Business Interaction matrix : The purpose of this matrix is to depict the relationship interactions between organizations and business functions across the enterprise.
- Understanding business interaction of an enterprise is important as it helps to highlight values chain and dependencies across organization. The Business Interaction matrix show the following metamodel entities and relationships:
  - Organization
  - Business function
  - Business Service
  - Business service communication with Business Service relationship
  - Business service is dependent on Business Service relationship.
-

    - Actor/Role Matrix : The purpose of this matrix is to show which actor perform which role, supporting definition of security and skills requirements.
- Understanding of Actor-to-Role relationship is a key supporting tool in definition of training needs, user security settings and organizational change management.
  The Actor/Role matrix shows the following metamodel entities and relationships:
    - Actor
    - Role
    - Actor performs Role relationship.
-
    - Value Stream / Capability matrix : The purpose of this matrix is to show the capabilities required to support each stage of a value stream.
    - Strategy/ Capability matrix : The purpose of this matrix is to show the capabilities required to support specific strategy statement.
    - Capability / Organisation matrix : - The purpose of this matrix is to show the organisation elements that implements each capability. The capability / Organisation matrix include following metamodel entities :
        - Business capability
        - Value stream
        - Organisation unit
-
    - Business foot print diagram

https://pubs.opengroup.org/architecture/togaf9-doc/arch/

32: Architecture Deliverables

Deliverables are typically the contractual or formal work products of an architecture project, it is likely that these deliverables will be constrained or altered by any overarching project or process management for enterprise.

Architecture Building blocks :

- Architecture Contract : Joint agreement between development partner and sponsors of the deliverable on quality and fitness for purpose of an architecture.

Successful implementation of these agreements will be delivered through effective Architecture Governance.

Content :

- Introduction and background
- Nature of agreement
- Scope of the architecture
- Architecture and strategic principles and requirements.
- Conformance requirement.
- Architecture development and management process and roles
- Target architecture measures.
- Defined phases of deliverables.
- Prioritized joint workplan
- Time window (s)
- Architecture delivery and business service.

- Architecture Definition document : Is a deliverable container for the core architectural artifacts created during a project and for important related information.

Architecture Definition document spans all architecture domain (Business, data, application and technology) and also examines all relevant state of architecture (baseline, transition and target)

Architecture definition document is a companion to the architecture requirement specification with complementary objective:

- Architecture definition document provides a qualitative view of the solution and aims to communicate the intent to the architects.
- Architecture requirement specification provides a quantitative view of the solution, stating measurable criteria that must be met during the implementation of the architecture.

- Architecture principles : General rules and guidelines intended to be enduring and seldom amended, that inform and support the way in which an organization sets about fulfilling its mission.

Content :

- Business principles
- Data principles
- Application principles
- Technology principles.

- Architecture repository : Acts as a holding area for all architecture related projects within the enterprise. The repository allows projects to manage their deliverables, locate re-usable assets and publish outputs to stakeholders and other interested parties.

- Architecture Requirement specification : Provides a set of quantitative statement that outline what an implemented project must do in order to comply with the architecture.

Content :

- Success measures
- Architecture requirements
- Business service contracts
- Application service contracts
- Implementation guidelines
- Implementation specification
- Implementation standard
- Interoperability requirements
- IT Service management requirements
- Constraints
- Assumptions.

- Architecture roadmap : lists individual work packages that will realize the target architecture and lays them out on timeline to show progression from the Baseline architecture to Target Architecture.

Content :

- Implementation factor assessment and Deduction matrix
  - Risks
  - Issues
  - Assumptions
  - Dependencies
  - Actions
  - Inputs
- Consolidated gap, solutions, and dependencies matrix including
  - Architecture domain

    - Gap
    - Potential solutions
    - Dependencies

- Any transition architecture

- Implementation recommendation
    - Criteria measures of effectiveness of projects
    - Risks and issues
    - Solutions Building blocks (SBB)

- Architecture vision : It provides summary of changes to the enterprise that will accrue from successful deployment of the Target Architecture. The purpose of Architecture vision is to provide key stakeholders with a formally agreed outcome.

Content :

- Problem description
- Objective of statement of Architecture work
- Summary view necessary for Request for architecture work and the vision 1.0 business, application, data and technology architecture created typically including
    - Value chain diagram
    - Solution concept
- Mapped requirements
- Reference to draft architecture definition document.