Module 6 : Understanding View, Viewpoints

How to Proceed :

A. Part 1 : First read the immediate portion, which is a Summary portion : Part 1 is for reading right now

This portion is important for

1. Understanding TOGAF for practical purposes – Supplement class session understanding with this
2. For Certification purposes, Level 1 and Level 2

B. Part 2 : Go through and workout the exercises in the Part 2 : Module6Questions&Answers. Very helpful for Certification preparation

C. Part 3 : Later when you find more time, do go through portion which says Part 3 : Detailed Courseware. That portion is useful for getting extra grades in Certification and for more proper understanding of TOGAF. Some sections of it are quoted from internet sources and from good authors as discovered by our Participants in earlier courses.

In this Part 3, Case Study and its boxes with samples are for understanding purpose only. Not relevant for Certification.

Part 1 : Summary portion

You may like to first read this Quick Look : Glossary and Acronym

View creation process : The regular work carried out by the Architect, by way of creating Building Blocks, and the activities prior and after the physical creation of such artifacts

Framework : Here, EAF : Enterprise Architecture Framework. The Frameworks that guide the tasks of an EA. Not to be confused with Application Frameworks like Java -Spring etc., or with the way the term Framework is broadly used in IT industry

Solution : Here, it is about the architectural solution, meaning the necessary diagrams and documents and other artifacts that are prepared as part of the activity of an Architect.

Stakeholder : An individual, team, organization, or class thereof, having an interest in a system.

Concern : This may pertain to any aspect of the system’s functioning, development, or operation, including considerations such as performance, reliability, security, distribution, and evolvability and may determine the acceptability of the system.

Building Block : A (potentially re-usable) component of enterprise capability that can be combined with other Building Blocks to deliver architectures and solutions. Note: Building Blocks can be defined at various levels of detail, depending on what stage of Architecture development that 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 a full specification. Building Blocks can relate to "architectures" or "solutions"

Architecture Building Block (ABB) : A portion of the architecture model that describes a single aspect of the overall model.

Solution Building Block (SBB) : A candidate solution which conforms to the specification of an Architecture Building Block (ABB). SBBs are ready to implement architectural details.

Next stage of coding and other work by PMO is known as "Implementation" or "Operational Realization" or "Migration" in TOGAF. So, after **SBB is approved and supplied as "Work Product" by EA Team,** the PMO gets into coding / procurement of software – COTS / getting into subscriptions of Cloud or SAAS - Software As A Service / procurement of hardware / installation / Testing and Trial Run and so on.

Connect these :

1. Stakeholders expresses concern, needs a IT solution
2. Architect approach different Stakeholders with different 'points of view' - **Viewpoints**
3. Architect makes Architectural decisions, produces Building Blocks : Collectively the Building Blocks are known as Architectural **Views**
4. **Views** are presented to appropriate Stakeholders for review and sign-off approval

A **Viewpoint** is the ways to understanding the Stakeholder and Concerns.

That leads to selection of the

right collection of Patterns, Templates, and conventions

The Architect then constructs one type of **View – Building Blocks**

Viewpoint : How to approach the solution :

Only thoughts, line of approach – angle to think from

View is what we Architects produce for stakeholders to 'view" :

Like Diagrams, Catalogues, Matrices ....

View : What (Architecture and Design) that is prepared

: Actual Diagrams, Matrices, Catalogs : Building Blocks

Why different Viewpoints when we approach different Stakeholders ?

For example,

executive management (LOB) will be more interested in a high-level description,

while communication with operational staff will require much more detailed representations

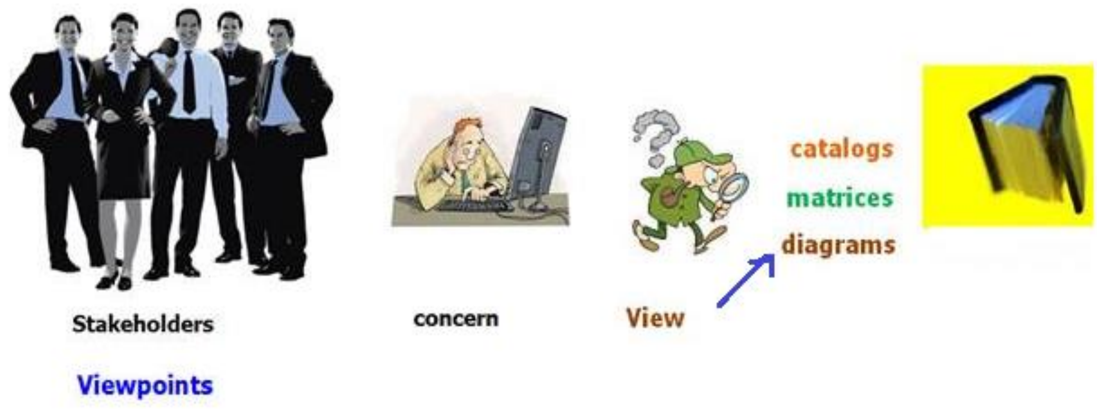
and Security concern may come from many different Stakeholders

For these Viewpoints, multiple Views of the Architecture is prepared and presented in the form of diagrams, documents, or other artifacts.

A Software Architecture is a complex theme, that cannot be described in a simple one-dimensional fashion.

There is no single rendition that is the 'right' one. All the views together convey the architecture.

It depends on your goals and purpose to arrive at the relevant views are : different views appeal to different stakeholders' concerns, and expose different quality attributes.



Stakeholders would have specific concern

View : What a stakeholder expects

in order that the concern is addressed

Relates to the artifact that an

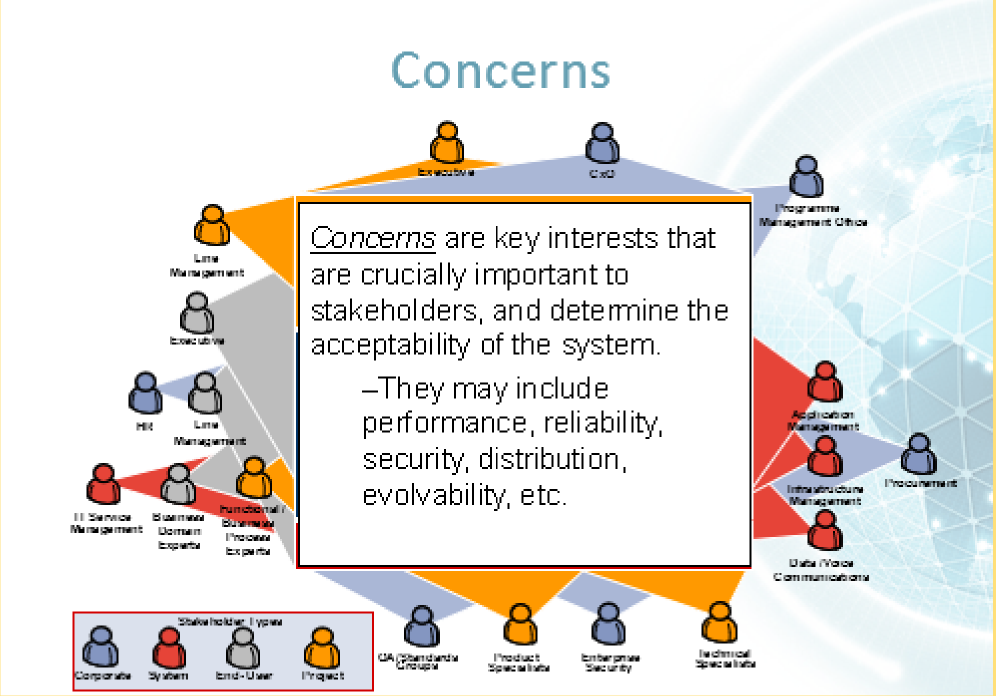
Architect supplies

Concerns determine the acceptability of the system

Concerns are the root of the process of decomposition into requirements

Concerns are represented in the architecture by the requirements

Each stakeholder typically has interests in, or **concerns relative to**, that system



Concerns

Concerns are key interests that are crucially important to stakeholders, and determine the acceptability of the system.

–They may include performance, reliability, security, distribution, evolvability, etc.

Stakeholder Concern

Becomes : Architectural Requirements

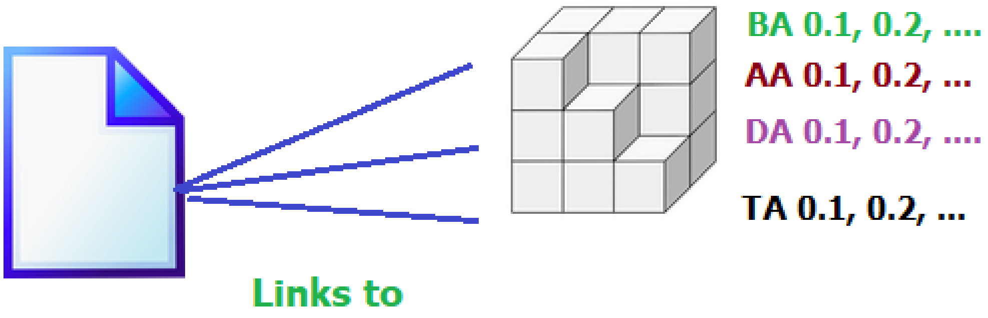
Leads to : Acceptability, Process Decomposition

Forms the core of ARS : Architecture Requirement Specification

ADD : Container of Views

Architecture Definition Document :

Deliverable container
for the core architectural artifacts

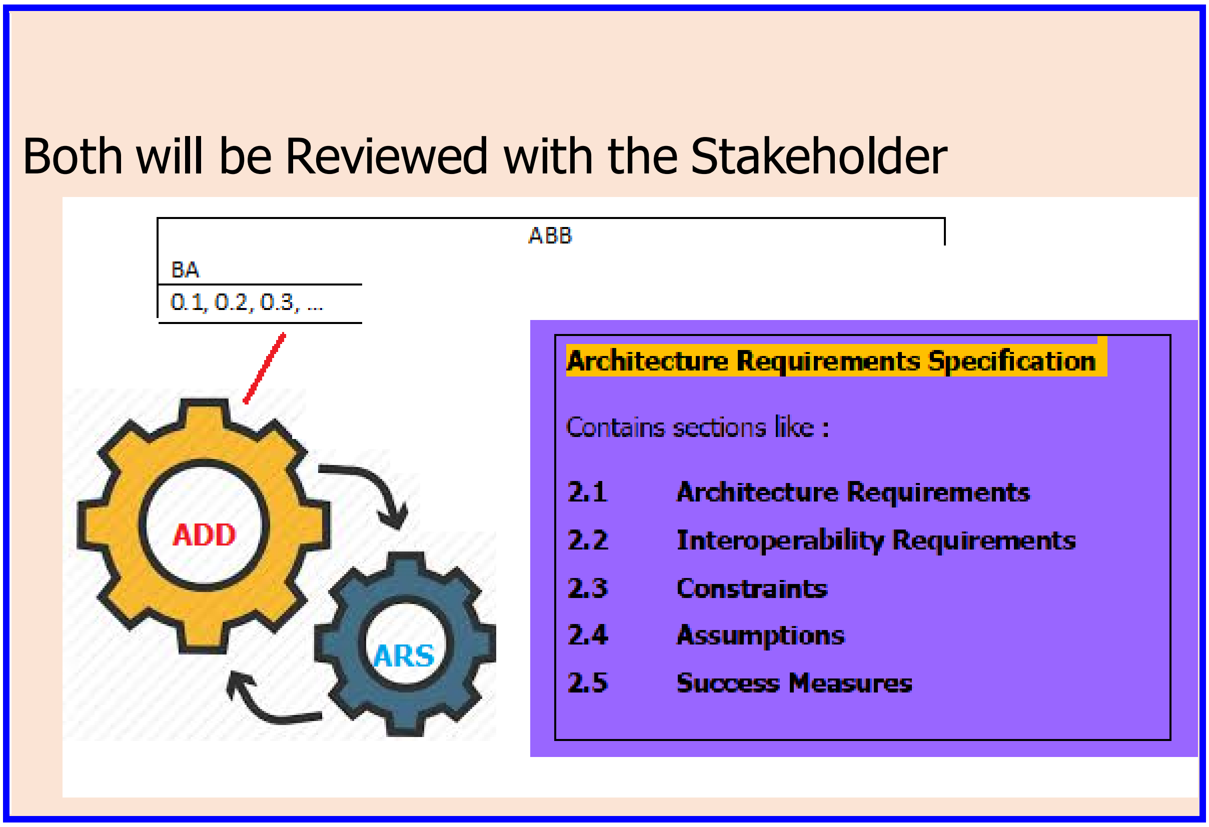


ARS : Reflection of Viewpoints, but in the words of the Architect

Architecture Requirements Specification

Contains sections like :

2.1 Architecture Requirements
2.2 Interoperability Requirements
2.3 Constraints
2.4 Assumptions
2.5 Success Measures



Both will be Reviewed with the Stakeholder

ABB

BA

0.1, 0.2, 0.3, ...

Architecture Requirements Specification

Contains sections like :

2.1	**Architecture Requirements**
2.2	**Interoperability Requirements**
2.3	**Constraints**
2.4	**Assumptions**
2.5	**Success Measures**

Think : Who creates a View, as per TOGAF ? Though what process ?

Answer : The Architect (EA, and other B D A T Architects) create View by way of Building Blocks and Artifacts. The steps taken by them in various ADM Phases becomes the View Creation Process

A RELATED QUESTION, IN LEVEL 1

: ★ ★ ★ 601

Q : In the view creation process, the viewpoints are selected **based** on who's / which **concerns ?**

A. Architect
B. Stakeholder
C. Views
D. Architecture

Answer : B Concerns - > Viewpoints -> addressed through Views – Building Blocks

Explanation :

Concerns come from stakeholders and Architect is expected to address them

See under : **31.2.2 Architecture View Creation Process**

The architect may choose to develop a new architecture viewpoint that will cover the outstanding need, and then generate an architecture view from it.

Whatever the context, the architect should be aware that every architecture view has an architecture viewpoint, at least implicitly, and that defining the architecture viewpoint in a systematic way will help in assessing its effectiveness; i.e., does the architecture viewpoint cover the relevant stakeholder concerns ?

Another Question, in Level 1 :

602

Q : Which **of the following** are all steps of View Creation Process ?

A. Select Key Stakeholders
B. Select Appropriate Viewpoints
C. Refer existing viewpoint from library
D. Create viewpoints from artifacts
E. A, B and D
F. B, C, D
G. A, B, C

Answer : G Viewpoints first then View artifacts created

Explanation :

Understanding these steps is important, since that is what an Architect is expected to do in course of ADM

See under : : **31.2.2 Architecture View Creation Process**

It will often be possible to create the required views for a particular architecture by following these steps :

1. Refer to an existing library of architecture viewpoints

2. Select the appropriate architecture viewpoints (based on the stakeholders and concerns that need to be covered by views)

3. Generate views of the system by using the selected architecture viewpoints as templates

Note that what is created from artifacts is view, not the viewpoint. Afterall, a viewpoint is only a point of the direction to look for.

A Building Block is a **package of functionality** defined to meet business needs across an organization.

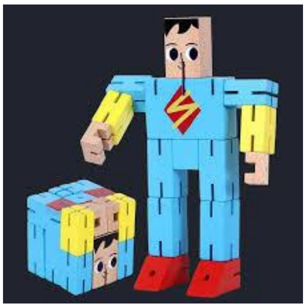
For domain experts and business Architects, it is generally recognizable as "a thing" e.g. a process, an application or data technology, a server and so on.

For Software Engineering experts it is generally recognizable as engineering components, like a Web Server (Application Architecture), a Database Engine (Data Architecture) or (for Infrastructure Technology Architect) a Server Node in a Data Centre or a conceptual Cloud Hosted System and so on

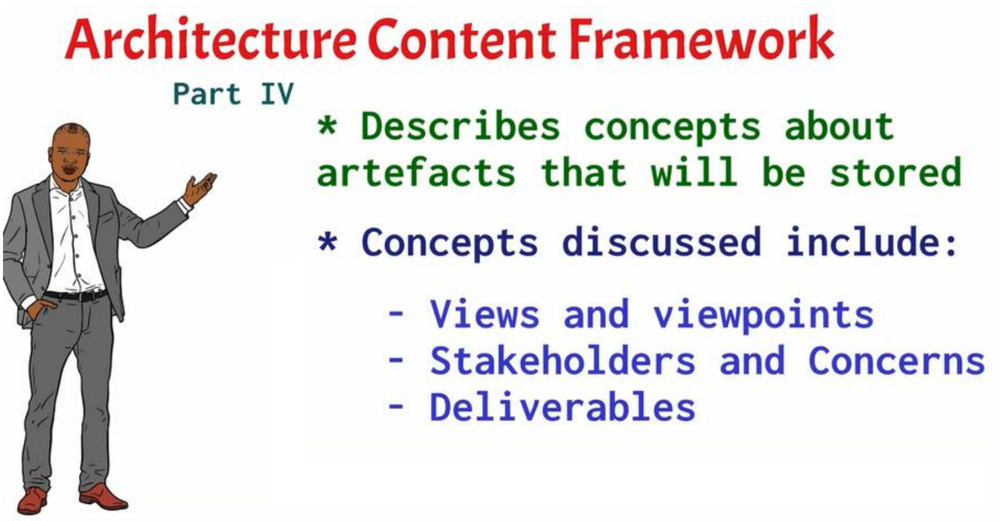
During Architecture development, the BBs can be used to present to business team, for example, the connections among roles, business services and processes.

Through these "business" BB elements, it can be further decomposed into more Software Engineering elements as Application Components, Data BBSs, Infrastructure servers, etc., An "Artifact" is an architectural work product that describes an aspect of the architecture.

An **Artifact** can contain multiple **Building Blocks** and a **Deliverable** can contain multiple **Artifacts**.



Architects executing the ADM will produce a number of outputs as a result of their efforts, such as process flows, architectural requirements, project plans, project compliance assessments, etc. The TOGAF Architecture Content Framework provides a structural model for architectural content that allows major work products to be consistently defined, structured, and presented.



Architecture Content Framework

Part IV

* Describes concepts about artefacts that will be stored

* Concepts discussed include:

  - Views and viewpoints
  - Stakeholders and Concerns
  - Deliverables

The Architecture Content Framework uses the following three categories to describe the type of architectural work product within the context of use :

- A **Deliverable** is a work product that is contractually specified and in turn formally reviewed, agreed, and signed off by the stakeholders. Deliverables represent the output of projects and those deliverables that are in documentation form will typically be archived at completion of a project, or transitioned into an Architecture Repository as a reference model, standard, or snapshot of the Architecture Landscape at a point in time.

- An **Artifact** is a more granular architectural work product that describes an architecture from a specific viewpoint. Examples include a network diagram, a server specification, a use-case specification, a list of architectural requirements, and a business interaction matrix. Artifacts are generally classified as catalogs (lists of things), matrices (showing relationships between things), and diagrams (pictures of things). An architectural deliverable may contain many artifacts and artifacts will form the content of the Architecture Repository.

- A **Building Block** represents a (potentially re-usable) component of business, IT, or architectural capability that can be combined with other Building Blocks to deliver architectures and solutions.

A RELATED QUESTION, IN LEVEL 1

: ★ ★ ★ 608

Q : In TOGAF, what is the difference between an artifact and a deliverable ?

A. An artifact contains one or more deliverables
B. Artifacts and deliverables are synonymous; there is no difference between them

C. Deliverables are prepared by the Project Manager, whereas artifacts are defined by the Architect
D. Deliverables are reusable, whereas artifacts are unique to a given architecture project
E. Deliverables are specified as contractual outputs from a project, whereas artifacts are not

Answer : **E**

Explanation leading to the answer:

Be doubly clear on this distinction. Read portion here a few times till you get the point.

Deliverable : Contractually mandated. Architect delivers, Usually not individual Artifact, but duly grouped Components

Artifact : Any piece (element) of Architectural Value.

Non-Deliverable Artifacts / Components may be optionally prepared and submitted as part of Architectural work.

Both categories are Architectural Work Pieces (Work Products). One is compilatory and mandatory.

Architecture Building Blocks

More of Architecture and High-Level Design

They are Technology Aware,

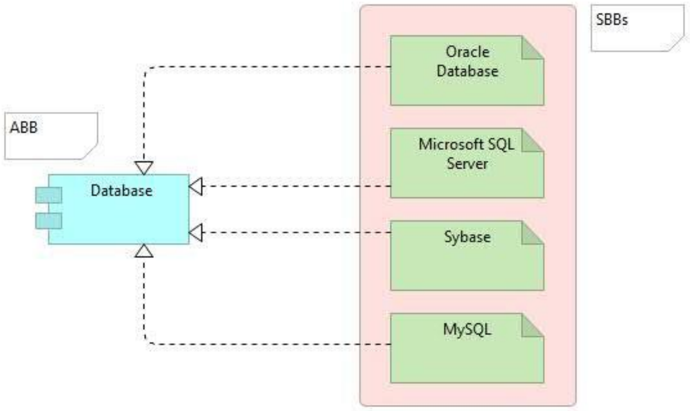
but not specific Product (brand) aware

ABBs directly emerge from captured Requirements. They develop as Architectural descriptions and **land up as ‘What and How” of the Architecture and design.**

An Architecture Requirement is requirement of the highest possible level of granularity for an Architectural Building Block. That is why we say that: ABBs directly emerge from captured Requirements.

However ABBs do not step into precise descriptions and Low Level Design pieces, such as one which is applicable for only one product like Oracle Database 21c with multimodal, multi-workload, and multitenant features.

ABBs are Architecturally feasible Components and their design but have not yet reached “Implementable Architecture” which are ready to be handed over to PMO – Project Management Office by the EA department.



Platform Independent, especially with D A T Segments

ABBs describe generic characteristics and functionalities : So, an ABB has fundamental functionality and attributes : Such as semantic, being unambiguous etc.,

ABBs

Play a major role during Review of Architecture with relevant Stakeholders

Any Architect may refer to ABBs at some stage or other

Use Case as an ABB Sample : aimed at LOB Stakeholders

Component Diagram as ABB : aimed at Application related Stakeholders

Conceptual and Logical Data Diagrams as ABB : aimed at Data Centric Stakeholders. Like DBAs

High Level Infrastructure Diagrams as ABB : aimed at appropriate Stakeholders like Admins

ABBs

concept, abstract, explanatory
Patterns, POCs, Feasibility Studies

White Papers, Plan Documents

Platform Independant Components,
API Components, Library Components
Purchase guidelines, General Specs

Not ready for Coding or Procurement or Installation
But can combine them and move towards solution
with additional Architectural work

Characteristics of ABBs :

Specification Content : ABB specifications include the following as a minimum :

Specification Content of ABB : Fundamental functionality

Functional and Non Functional

Characteristics of ABBs :

Specification Content : ABB specifications include the following as a minimum :

Interfaces : chosen as a set, supplied appropriately

How to connect

Characteristics of ABBs :

Specification Content : ABB specifications include the following as a minimum :

Interoperability and relationship with other Building Blocks

How to work with

Characteristics of ABBs :

Specification Content : ABB specifications include the following as a minimum :

Map to business / organizational entities and policies

Business IT Alignment

Characteristics of ABBs :

Specification Content : ABB specifications include the following as a minimum :

Dependent Building Blocks with required functionality and named user interfaces

All round dependency

The deliverables are represented directly by Platform Independent Models, concept white papers etc., This is the **case for the "ABB" deliverable**, whose aim is to formalize an architecture model.

ABBs

concept, abstract, explanatory
Patterns, POCs, Feasibility Studies

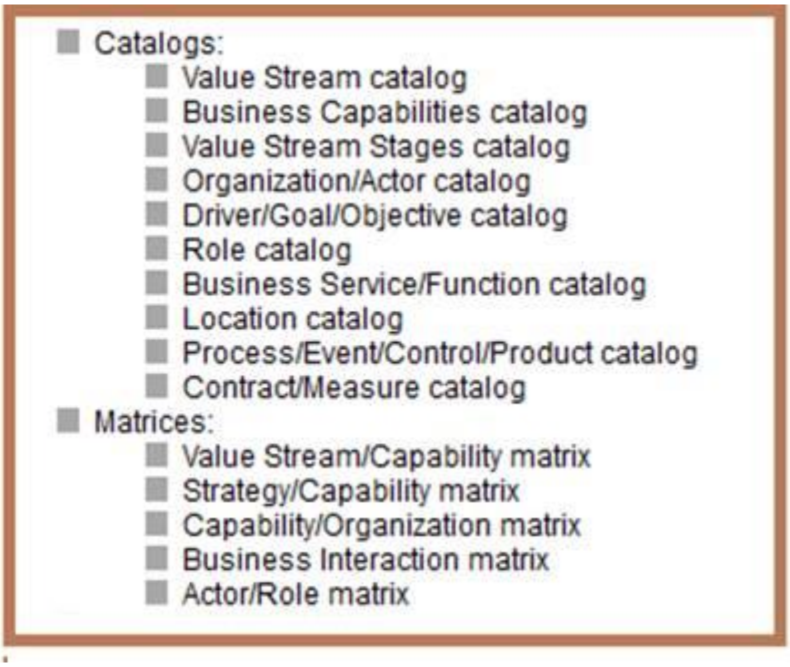
White Papers, Plan Documents

Platform Independant Components,
API Components, Library Components
Purchase guidelines, General Specs

Not ready for Coding or Procurement or Installation
But can combine them and move towards solution
with additional Architectural work

Non-Diagram Building Blocks include :

Business Architecture

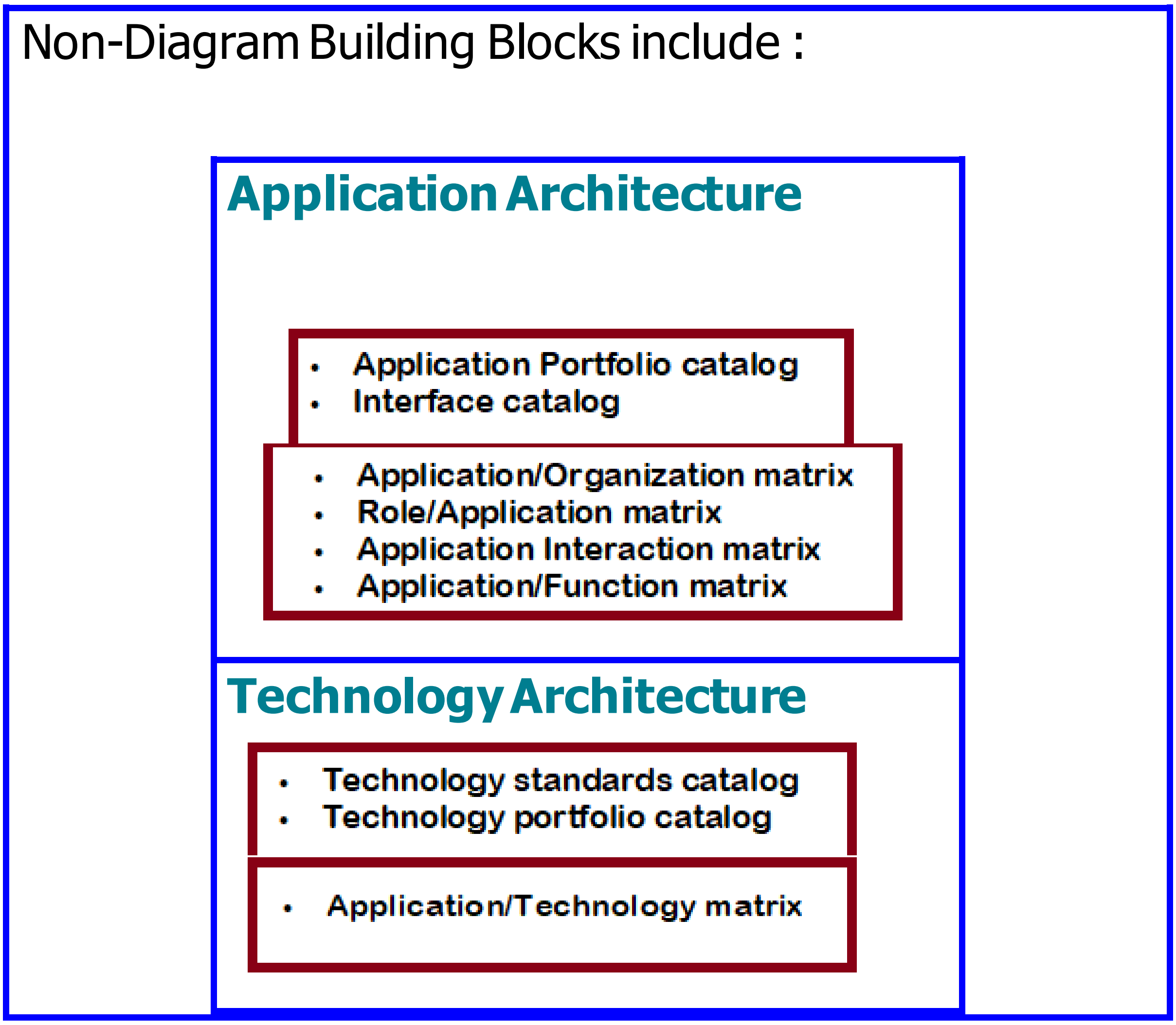


- Catalogs:
  - Value Stream catalog
  - Business Capabilities catalog
  - Value Stream Stages catalog
  - Organization/Actor catalog
  - Driver/Goal/Objective catalog
  - Role catalog
  - Business Service/Function catalog
  - Location catalog
  - Process/Event/Control/Product catalog
  - Contract/Measure catalog
- Matrices:
  - Value Stream/Capability matrix
  - Strategy/Capability matrix
  - Capability/Organization matrix
  - Business Interaction matrix
  - Actor/Role matrix

Data Architecture

- **Data Entity/Data Component catalog**

- **Data Entity/Business Function matrix**
- **Business Service/Information matrix**
- **Application/Data matrix**



Non-Diagram Building Blocks include :

Application Architecture

- Application Portfolio catalog
- Interface catalog

- Application/Organization matrix
- Role/Application matrix
- Application Interaction matrix
- Application/Function matrix

Technology Architecture

- Technology standards catalog
- Technology portfolio catalog

- Application/Technology matrix

One more Characteristic of ABB :

To Direct and guide the development of SBBs

Later on, when the cycle gets into Phase E, Solution centric Architecture, it will have to be in terms of specific development platform, specific pseudo-code etc.,

So, the Platform Specific Model will have to be prepared based on the above ABBs.

Architecture Building Blocks : Summarizing

More of Architecture and High-Level Design

Platform Independent, especially with D A T Segments

Characteristics of ABBs :

1. **Capture architecture requirements;**

😇 e.g., business, data, application, and technology requirements

To Direct and guide the development of SBBs

Capture ???, direct, create ???

Specification Content : ABB specifications include the following as a minimum :

1. **Fundamental functionality**

Functional and Non Functional

2. **Interfaces : chosen as a set, supplied appropriately**

How to connect

3. **Interoperability and relationship with other Building Blocks**

How to work with

4. **Map to business / organizational entities and policies**

Business IT Alignment

5. **Dependent Building Blocks with required functionality and named user interfaces**

All round dependency

Solution Building Blocks

Here we look at Platform Specific Building Blocks

Ready to be handed over to PMO for further work

Can be for coding;

Can be for procurement of a Software or Hardware Product

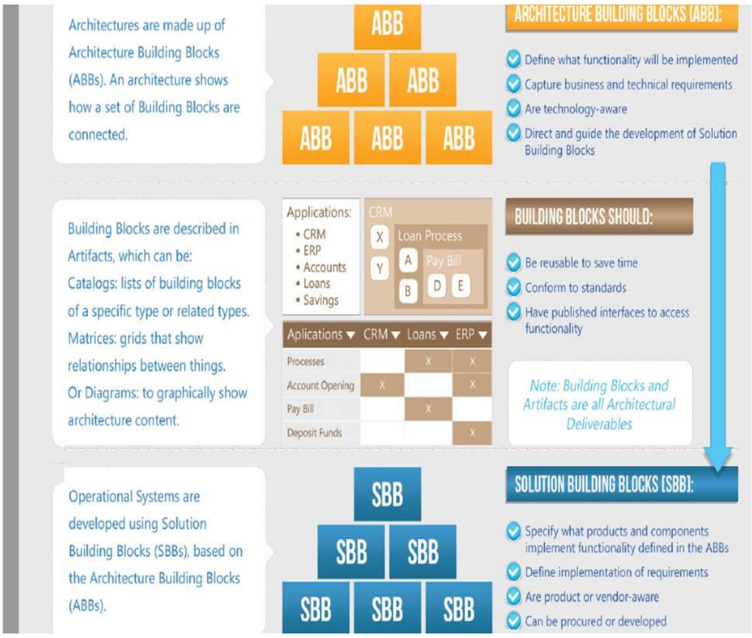
SBBs

Concrete, Implementable

Products - Software and other hardware :
Specs, Design, Models for implementation

COTS, Software as Service : Specs and Design
Designed for Platform Specific Implementation

An Architecture Specification is a document of the highest possible level of granularity on an Solution Building Block.



Architectures are made up of Architecture Building Blocks (ABBs). An architecture shows how a set of Building Blocks are connected.

ARCHITECTURE BUILDING BLOCKS (ABB):

- ✓ Define what functionality will be implemented
- ✓ Capture business and technical requirements
- ✓ Are technology-aware
- ✓ Direct and guide the development of Solution Building Blocks

Building Blocks are described in Artifacts, which can be:
Catalogs: lists of building blocks of a specific type or related types.
Matrices: grids that show relationships between things.
Or Diagrams: to graphically show architecture content.

Applications:
- CRM
- ERP
- Accounts
- Loans
- Savings

Aplications ▼	CRM ▼	Loans ▼	ERP ▼
Processes		X	X
Account Opening	X		X
Pay Bill		X	
Deposit Funds			X

BUILDING BLOCKS SHOULD:

- ✓ Be reusable to save time
- ✓ Conform to standards
- ✓ Have published interfaces to access functionality

Note: Building Blocks and Artifacts are all Architectural Deliverables

Operational Systems are developed using Solution Building Blocks (SBBs), based on the Architecture Building Blocks (ABBs).

SOLUTION BUILDING BLOCKS (SBB):

- ✓ Specify what products and components implement functionality defined in the ABBs
- ✓ Define implementation of requirements
- ✓ Are product or vendor-aware
- ✓ Can be procured or developed

The main relation between ABBs and SBBs is that the first guides the development of the second. In other words, the ABB is an specification of SBBs and it comply with this specification. Essentially the ABB has captured the architecture requirements (business, data, application and technology). The SBB defines the products and components that can be implement the functionality.

ABB is about

Mostly Picked up during
Architectural concept formulation
Rather than Solution design

Say, Most of Business Architecture
And early versions of AA, DA, TA

SBB is about

Mostly Produced during
Solution Design

For Architectural Projects before Phase E
For SOA - Transition Arch Projects : during Phase E

Operational working systems are developed using Architecture of _ BB, which is anyhow based on the Architecture of _BB

Applicable to ABB and SBB :

- ✓ Be to save time
- ✓ Conform to
- ✓ Have published to access functionality

Standards

Reusable

Interfaces

ABBs

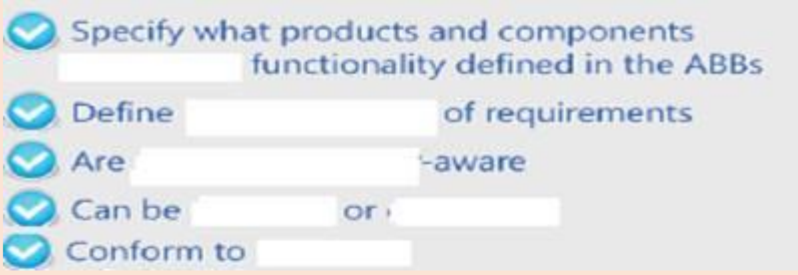
- ✓ Define what will be implemented
- ✓ business and technical requirements
- ✓ Are aware
- ✓ Direct and guide the development of Solution Building Blocks

Technology

Functionality

Capture

SBBs



- ✓ Specify what products and components functionality defined in the ABBs
- ✓ Define of requirements
- ✓ Are -aware
- ✓ Can be or
- ✓ Conform to

implement

implementation

Product or Vendor

Procured or Developed

Standards

Part 2 : Module 6 Questions and Answers

(Also Explanations)

Please answer questions appearing below on a piece of paper and then check the answer and explanation appearing immediately below the questions. Some Questions may be on earlier modules too.

You can choose the ones you want to answer now and keep the rest for a workout on your own later on.

The pictures that appear next to the question is only to break the monotony and has no special meaning.

The star rating gives you a clue of the relative importance of questions, from Certification viewpoint. Three-star questions may appear more often than two star and so on

Think : Who creates a View, as per TOGAF ? Though what process ?

Answer : The Architect (EA, and other B D A T Architects) create View by way of Building Blocks and Artifacts. The steps taken by them in various ADM Phases becomes the View Creation Process

601 In the view creation process, the viewpoints are selected based on who's / which concerns ?

A. Architect
B. Stakeholder
C. Views
D. Architecture

Answer : B

Explanation :

Concerns come from stakeholders and Architect is expected to address them

Concerns - > Viewpoints -> addressed through Views – Building Blocks

See under : **31.2.2 Architecture View Creation Process**

The architect may choose to develop a new architecture viewpoint that will cover the outstanding need, and then generate an architecture view from it.

Whatever the context, the architect should be aware that every architecture view has an architecture viewpoint, at least implicitly, and that defining the architecture viewpoint in a systematic way will help in assessing its effectiveness; i.e., does the architecture viewpoint cover the relevant stakeholder concerns ?

602 Which of the following are all steps View Creation Process ?

A. Select Key Stakeholders
B. Select Appropriate Viewpoints
C. Refer existing viewpoint from library
D. Create viewpoints from artifacts
E. A, B and D
F. B, C, D
G. A, B, C

Answer : G

Explanation :

Understanding these steps is important, since that is what an Architect is expected to do in course of ADM

Viewpoints first, then View artifacts created

See under : : **31.2.2 Architecture View Creation Process**

It will often be possible to create the required views for a particular architecture by following these steps :

1. Refer to an existing library of architecture viewpoints

2. Select the appropriate architecture viewpoints (based on the stakeholders and concerns that need to be covered by views)

3. Generate views of the system by using the selected architecture viewpoints as templates

Note that what is created from artifacts is view, not the viewpoint. Afterall, a viewpoint is only a point of the direction to look for.

603 Which of the following applies to an Architecture Building Block ?

A. It defines the functionality to be implemented.
B. It defines the implementation.
C. It defines what products and components will implement the functionality.
D. It is product or vendor-aware.

Answer : A

Explanation :

ABBs define functionality – not implementation

★ 604 In TOGAF, a constituent of the architecture model that describes a single aspect of the overall model is referred to as

A. An artifact
B. A deliverable
C. An Architecture Building Block
D. A View
E. A Viewpoint

Answer : C

Explanation :

An Architecture Building Block is a constituent of the architecture model that describes a single aspect of the overall model

See :**3.8 Architecture Building Block (ABB)**

A constituent of the architecture model that describes a single aspect of the overall model.

605 Which one of the following statements about Viewpoints is correct ?

A. A Viewpoint is always specific to an architecture
B. A Viewpoint is used to create views in Phases E and F
C. A Viewpoint is used as a template to create a view
D. A Viewpoint is what a stakeholder sees

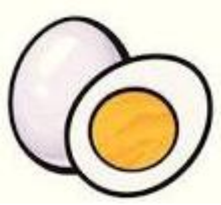
Answer : C

Viewpoint and View are also favourite question focus points

Explanation :

A **Viewpoint** is a collection of Patterns, Templates, and conventions for constructing one type of **View**.

A Viewpoint designates the most appropriate perspective **for a given participant**, and is materialized by a certain number of views of the architecture, in the form of diagrams, documents, or other elements.

★ 606  Which of the following statements is not correct ?

A. A concern might include performance and reliability.
B. A concern is an area of interest.
C. Concerns are key interests of the stakeholders.
D. Concern and requirement are synonymous.

Answer : D

Explanation :

They are not synonymous as concerns are used to derive requirements.

See under : **31.1**

The terms "concern" and "requirement" are not synonymous. A concern is an area of interest. So, system reliability might be a concern/area of interest for some stakeholders.

The reason why architects should identify concerns and associate them with architecture viewpoints, is to ensure that those concerns will be addressed in some fashion by the models of the architecture. For example, if the only architecture viewpoint selected by an architect is a structural architecture viewpoint, then reliability concerns are almost certainly not being addressed, since they cannot be represented in a structural model.

Within that concern, stakeholders may have many distinct requirements: different classes of users may have very different reliability requirements for different capabilities of the system.

607 Which of the following is considered by TOGAF as a MAJOR attribute of a good Building Block ?

A. A Building Block that is re-usable
B. A Building Block meeting business needs
C. A Building Block with public interfaces
D. A Building Block that guides the development of solutions
E. A Building Block that is product-aware

Answer : A

Explanation :

TOGAF places a lot of importance on one specific aspect of every Building Block – it should be as much re-usable as possible. TOGAF considers re-usability an attribute of a good Building Block.

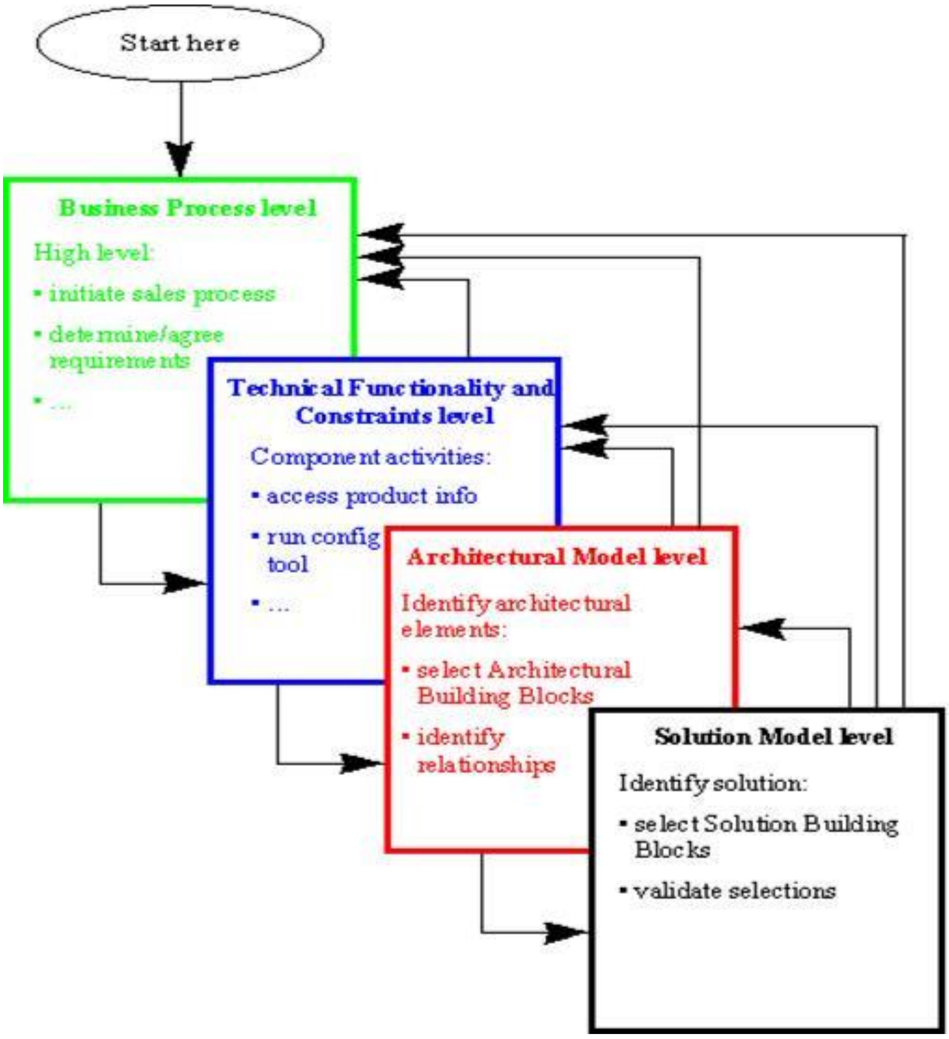
When an architect is developing an architecture, he or she does not work with real components, but works rather with descriptions of idealised components, called Building Blocks. ABB and SBB are terminology used by TOGAF for defining isolatable Building Block Components of Enterprise architecture. Building 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 grouping of functionality such as a customer database and some retrieval tools. Building Blocks at this functional level of definition are described in TOGAF as Architecture Building Blocks (ABBs). Later on, real products or specific custom developments replace these simple definitions of functionality, and the Building Blocks are then described as Solution Building Blocks (SBBs).

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 a full specification.

The level of detail to which a Building Block should be specified is dependent on the objectives of the architecture and, in some cases, less detail may be of greater value (for example, when presenting the capabilities of an enterprise, a single clear and concise picture has more value than a dense 100-page specification).



608 In TOGAF, what is the difference between an artifact and a deliverable ?

A. An artifact contains one or more deliverables
B. Artifacts and deliverables are synonymous; there is no difference between them
C. Deliverables are prepared by the Project Manager, whereas artifacts are defined by the Architect
D. Deliverables are reusable, whereas artifacts are unique to a given architecture project
E. Deliverables are specified as contractual outputs from a project, whereas artifacts are not

Answer : E

Explanation :

Be doubly clear on this distinction. Read portion here a few times till you get the point.

Deliverable : Contractually mandated. Architect delivers, Usually not individual Artifact, but duly grouped Components

Artifact : Any piece (element) of Architectural Value.

Non-Deliverable Artifacts / Components may be optionally prepared and submitted as part of Architectural work.

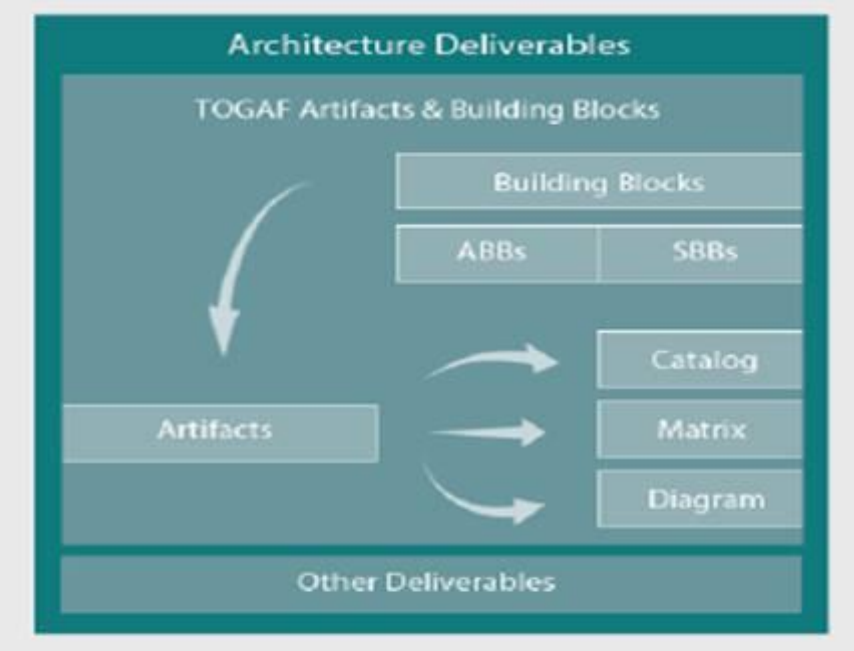
Both categories are Architectural Work Pieces (Work Products). One is compilatory and mandatory.

Contract : What the EA mandated the other Architect. Not referring to any official contract between companies.

■ A **deliverable** is a work product that is contractually specified and in turn formally reviewed, agreed, and signed off by the stakeholders

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

■ An **artifact** is an architectural work product that describes an aspect of the architecture Artifacts are generally classified as catalogs (lists of things), matrices (showing relationships between things), and diagrams (pictures of things). Examples include a requirements catalog, business interaction matrix, and a use-case diagram. An architectural deliverable may contain many artifacts and artifacts.



Arch Artifact, Arch Component, Arch BB : Same ?

Artifact : any arch element

Arch Component : Group of elements with a purpose

Arch BB : Potentially reusable Component

But sometimes TOGAF documentation uses the word artifact to mean a larger Building Block in a loose sense

ABB, SBB same ?

ABB – Architectural Building Block : Conceptual, Platform neutral

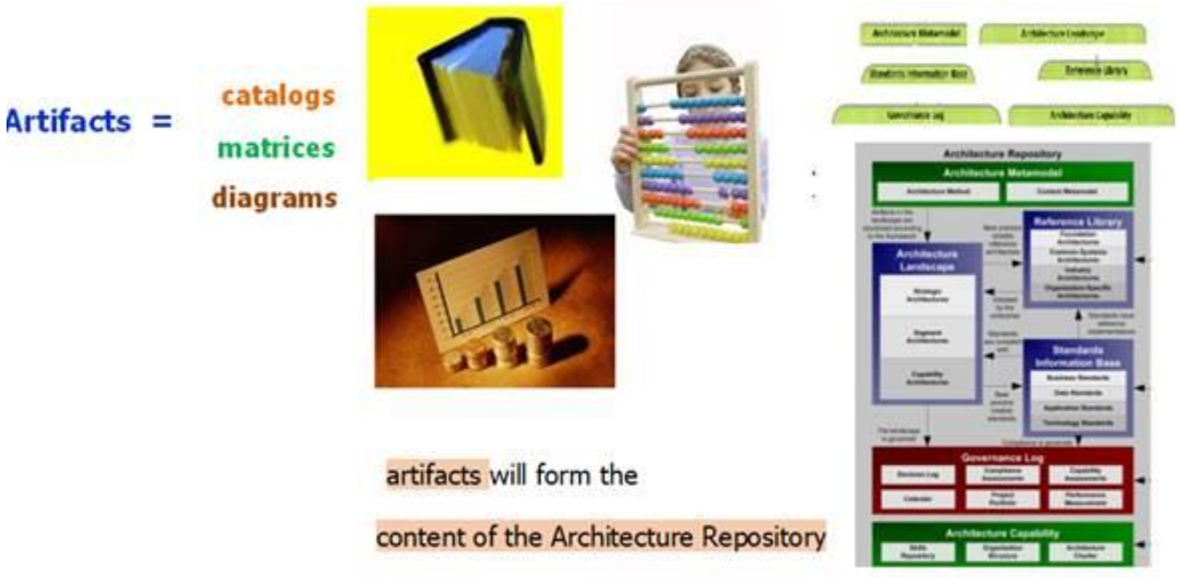
SBB – Solution Building Block : Detailed,

Platform specific, ready to code

or specification for purchase

An architectural deliverable may contain many artifacts, grouped as Building Blocks

and these will form the content of the Architecture Repository



Artifact – One single element : Architectural Work Piece

609 Which of the following is described by the TOGAF Architecture Content Framework as a type of artifact that shows lists of things ?

A. Building Block
B. Catalog
C. Diagram
D. Matrix
E. Deliverable

Answer : B

Explanation :

TOGAF Architecture Content Framework describes Catalog as a type of artifact that shows lists of things. In almost every Phase, TOGAF recommends the kind of catalog that can be produced.

Artifacts are generally classified as catalogs (lists of things), matrices (showing relationships between things), and diagrams (pictures of things). Examples include a requirements catalog, business interaction matrix, and a use-case diagram. An architectural deliverable may contain many artifacts and artifacts will form the content of the Architecture.

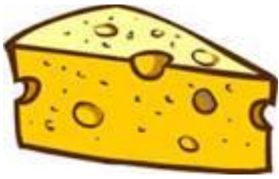
610 Which of the following does the TOGAF document define as architectural work product that describes an aspect of the architecture ?

A. Application Platform Interface
B. Artifact
C. Building Block
D. Deliverable
E. Framework

Answer : B

Explanation :

A building block is a package of functionality and is potentially re-usable. So, building blocks are Components which in turn contain artifacts. On the other hand, artifact is a single element, like an Actor in a Use Case.

611  Which one of the following statements does not correctly describe architecture deliverables ?

A. They are consumed and produced across the ADM cycle
B. They are defined to avoid tailoring the inputs and outputs of the ADM cycle
C. They are typically contractual work products of an architecture project
D. They are usually reviewed and signed off by the stakeholders

Answer : B

Explanation :

TOGAF provides a typical baseline of architecture deliverables in order to better define the activities required in the ADM and act as a starting point for tailoring within a specific organization.

Since 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 the enterprise.

Once the framework has been tailored to the enterprise, further tailoring is necessary in order to tailor the framework for the specific architecture project. Tailoring at this level will select appropriate deliverables and artifacts to meet project and stakeholder needs.

612 Which one of the following is a work product that is contractually specified, formally reviewed, and signed off by the stakeholders ?

A. An artifact
B. A building block
C. A catalog
D. A deliverable
E. A matrix

Answer : D

Explanation :

Deliverable is a work product that is contractually specified and in turn formally reviewed, agreed, and signed off by the stakeholders. Deliverables represent the output of projects and those deliverables that are in documentation form will typically be archived at completion of a project, or transitioned into an Architecture Repository as a reference model, standard, or snapshot of the Architecture Landscape at a point in time.

613 Which among the following statements is NOT correct ?

A. TOGAF recommends the way in which functionality, products, and custom developments are assembled into Building Blocks
B. Every organization must decide for itself what arrangement of Building Blocks works best for it
C. A good choice of Building Blocks can lead to improvements in legacy system integration, interoperability, and flexibility in the creation of new systems and applications
D. An architecture is a composition of a set of Building Blocks and the specification of how those Building Blocks are connected
E. Various Building Blocks in an architecture specify the services required in an enterprise specific system

Answer : A

Explanation :

TOGAF **does not recommend the exact way** Building Blocks are assembled. Every organization must decide for itself what arrangement of Building Blocks works best for it. It may even change a little form project to project at architecture level and vary from solution approach to solution approach

614 Complete the sentence by selecting the applicable pair of words. According to TOGAF, a / an ______ is used to describe the _____ of a stakeholder.

A. activity model, perspective
B. viewpoint, requirements
C. view, concerns
D. Node Connectivity Diagram, interconnections
E. Architecture trade-off analysis, constraints

Answer : C

Explanation :

TOGAF makes it very clear that a view is used to describe the concerns of a stakeholder.

See under : **31.4.2**

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.

Also see under **31.1**

The reason why architects should identify concerns and associate them with architecture viewpoints, is to ensure that those concerns will be addressed in some fashion by the models of the architecture.

615 Complete the sentence. According to TOGAF, a view is a representation of a system from the perspective of the ______ of a stakeholder.

A. concerns
B. constraints
C. interests
D. perspective
E. requirements

Answer : A

Explanation :

View is a representation of a system from the perspective of the concerns of a stakeholder.

The View Creation process is also about selecting the appropriate architecture viewpoints (based on the stakeholders and concerns that need to be covered by views.

See under : **31.1**

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

Views lead to description of the perspective. Perspectives are just 'guides' in the process. **Viewpoint** defines the Perspective.

616 Which of these is not a recommended step to create the required views for a particular architecture ?

A. Refer to any existing libraries of viewpoints

B. Select key stakeholders

C. Analyze their requirement and document them

D. Select appropriate viewpoints

E. Get views of the system using the selected viewpoints as templates

Answer : C

Explanation :

Analyze their requirement and document them is not a step recommended under View Creation Process. It related to couple of steps that are part of Phases B, C and D.

See under : : **31.2.2 Architecture View Creation Process**

It will often be possible to create the required views for a particular architecture by following these steps :

1. Refer to an existing library of architecture viewpoints

2. Select the appropriate architecture viewpoints (based on the stakeholders and concerns that need to be covered by views)

3. Generate views of the system by using the selected architecture viewpoints as templates.

617 What does Architecture Building Blocks do as compared to Solution Building Blocks ?

A. They define how the functionality will be realized through products and components
B. They capture architecture requirements
C. They define the implementation
D. They fulfil business requirements
E. They are product or vendor-aware

Answer : B

Explanation :

Architecture Building Blocks (ABBs) typically describe required capability and shape the specification of Solution Building Blocks (SBBs).

For example, a customer services capability may be required within an enterprise, supported by many SBBs, such as processes, data, and application software.

618 Which one of the following statements about TOGAF Building Blocks is true ?

A. Building Blocks at a functional level are known as Solutions Building Blocks
B. Solutions Building Blocks are selected in Phases A, B, C and D
C. The specification of a Building Block should be loosely coupled to its implementation
D. The gap analysis technique should be used to identify Building Blocks for re-use

Answer : C

Explanation :

On wrong answer choices :

A) Functional level is ABB
B) We get into SBB only ideally at Phase E
D) Reuse possibility is checked at steps prior to : Perform Gap Analysis

Building Block is a favourite topic for Certification questions. You will come across many questions on this topic at different points of this courseware.

Read all about Building Block (ABB, SBB) Characteristics, if you are not clear about this questions and the correct answer.

619 Which one of the following statements about TOGAF Building Blocks is NOT true ?

A. They should not be reused in other enterprise architecture projects
B. They should have stable, published interfaces that allow other Building Block to interoperate with them
C. They are packages of functionality intended to meet the business needs across the organization
D. They have defined boundaries

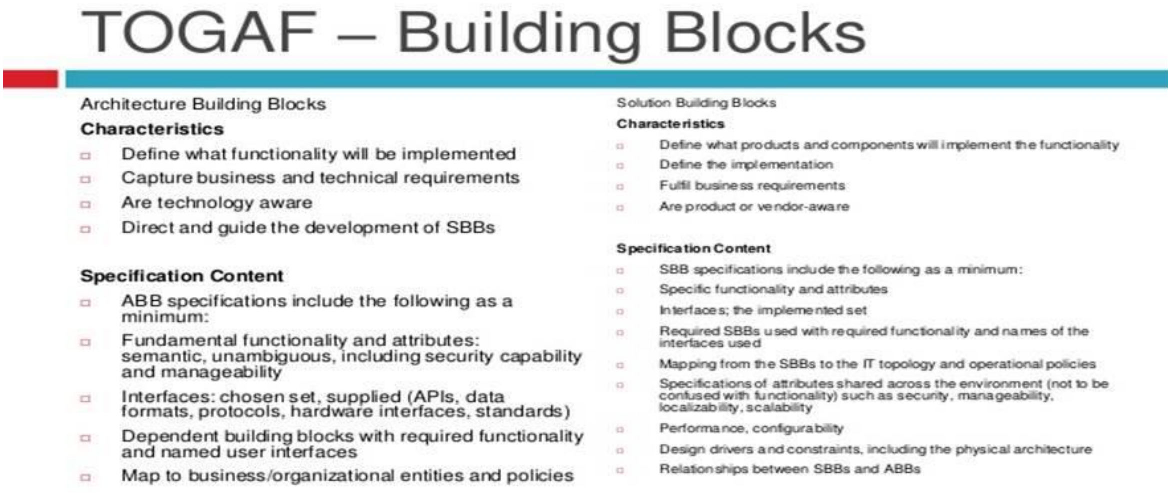
Answer : A

Explanation :

Very idea of preparing Component Building Blocks during entire architecture is to re-use them in other future projects to the extent possible. That is why Building Blocks are defined to be "potentially re-usable".

Get to know facts about Building Blocks. We should see them as ABB and SBB

Here, the term "Architecture" designates a description, and more precisely a logical view, as opposed to the "Solution," which represents a technical implementable reality.



TOGAF – Building Blocks

Architecture Building Blocks

Characteristics

- □ Define what functionality will be implemented
- □ Capture business and technical requirements
- □ Are technology aware
- □ Direct and guide the development of SBBs

Specification Content

- □ ABB specifications include the following as a minimum:
- □ Fundamental functionality and attributes: semantic, unambiguous, including security capability and manageability
- □ Interfaces: chosen set, supplied (APIs, data formats, protocols, hardware interfaces, standards)
- □ Dependent building blocks with required functionality and named user interfaces
- □ Map to business/organizational entities and policies

Solution Building Blocks

Characteristics

- □ Define what products and components will implement the functionality
- □ Define the implementation
- □ Fulfil business requirements
- □ Are product or vendor-aware

Specification Content

- □ SBB specifications include the following as a minimum:
- □ Specific functionality and attributes
- □ Interfaces; the implemented set
- □ Required SBBs used with required functionality and names of the interfaces used
- □ Mapping from the SBBs to the IT topology and operational policies
- □ Specifications of attributes shared across the environment (not to be confused with functionality) such as security, manageability, localizability, scalability
- □ Performance, configurability
- □ Design drivers and constraints, including the physical architecture
- □ Relationships between SBBs and ABBs

★ ★ 620 Which of the following is about a deliverable is correct ?

A. It can be any artifact
B. It has to be a Solution Building Block
C. It cannot be catalog or a matrix
D. It is a work piece (work product) that describes an architecture from a specific viewpoint ?
E. It has to be a diagram

Answer : D

Explanation :

A deliverable is something mandated (specified by the EA to the Solution Architect as a contract of task to be completed)

All other answer choices are incorrect as they distort the fact.

See under : **31.1**

In capturing or representing the design of a system architecture, the architect will typically create one or more architecture models, possibly using different tools. An architecture view will comprise selected parts of one or more models, chosen so as to demonstrate to a particular stakeholder or group of stakeholders that their concerns are being adequately addressed in the design of the system architecture.

621 Which of the following is a work piece (work product) that describes an architecture from a specific viewpoint ?

A. An artifact
B. A Building Block
C. A catalog
D. A deliverable
E. A matrix

Answer : D

Explanation :

Deliverables are typically the contractual or formal work products of an architecture project, They are delivered to specific stakeholders, based on their viewpoint.

622 Which of the following statements is not correct ?

A. A view can be thought of as a template for a viewpoint.
B. A viewpoint defines the perspective from which a view is taken.
C. A viewpoint defines how to construct and use a view.
D. A view is what a stakeholder sees.
E. A view might describe business process for an IT system.

Answer : A

Explanation :

It is the other way round; a viewpoint is considered a template for a view.

See under **31.2.2**

Generate views of the system by using the selected architecture viewpoints as templates.

623 Which of the following statements describing relationships between stakeholders, concerns, views, and viewpoints is correct ?

A. A concern is important to only one stakeholder.

B. A stakeholder identifies one or more concerns.

C. A viewpoint covers one concern.

D. A viewpoint consists of one or more views.

Answer : B

Explanation :

Answer choices A and C should have been worded as one or more stakeholders/concerns

D is incorrect. Viewpoint address a concern which is related the view(s) of stakeholder(s). It does not contain them.

624 Which of the following does not apply to a Building Block ?

A. It is a package of functionality that meets business needs.
B. It has published interfaces to access functionality.
C. It may interoperate with other Building Blocks.
D. It has a specification that is tightly coupled to its implementation.

Answer : D

Explanation :

Building blocks should have a loose coupling to implementation to allow for multiple implementations and re-implementation.

625 Views and Viewpoints are used by an architect to capture or model the design of a system architecture. Which one of the following statements is true ?

A. A view is the perspective of an individual stakeholder
B. A viewpoint is the perspective of an individual stakeholder
C. Different stakeholders always share the same views
D. Different stakeholders always share the same viewpoints

Answer : B

Explanation :

A view is what you see. A viewpoint is where you are looking from – the vantage point or perspective that determines what you see. Diagrams or other Artifacts present the Architecture information from a set of different perspectives (viewpoints) according to the requirements of the stakeholders.

626 Which one of the following best describes the content of an Architecture Building Block ?

A. Defined implementation

B. Fundamental functionality

C. Products and components used to implement the functionality

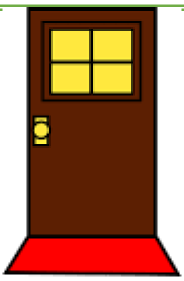
D. Product or vendor-aware

E. Specific functionality

Answer : B

Explanation :

An ABB has fundamental functionality and attributes: semantic, unambiguous, including security capability and manageability.

★ ★ ★ 627  Gap analysis is a key step in validating the architecture in Phase B : Business Architecture. Which one of the following statements is true ?

A. Gap analysis highlights services that are available
B. Gap analysis highlights the impacts of change
C. Gap analysis highlights services that are yet to be procured or prepared
D. Gap analysis identifies areas where the Data Architecture needs to change
E. Gap analysis can be used to resolve conflicts amongst different viewpoints

Answer : C

Explanation :

Gap analysis, apart from looking (and solving for) many types of gaps, also includes highlighting and focussing on services that are yet to be procured or prepared

A key step in validating an architecture is to consider what may have been forgotten or what is yet to be prepared / procured.

628 A statement of need that must be met by a particular architecture or work package is

A. Work product
B. Requirement
C. Constraint
D. Concern

Answer : B

Explanation :

Requirement is the term that applies to a statement of need.

See Section : **3.61 : Requirement**
A statement of need that must be met by a particular architecture or work package.

3.34 Concern
An interest in a system relevant to one or more of its stakeholders.

2.5 Deliverables, Artifacts, and Building Blocks

The Architecture Content Framework uses the following three categories to describe the type of architectural work product within the context of use:

▪ A deliverable

▪ An artifact

▪ A building block

Constraint
A Constraint represents a factor that prevents or obstructs the realization of goals.

In contrast to a requirement, a constraint does not prescribe some intended functionality of the system to be realized, but imposes a restriction on the way it operates or may be realized.

629 Which of the following is defined by TOGAF as a representation of a system from the perspective of a related set of concerns ?

A. Architecture Building Block
B. Capability Architecture
C. Statement of Work
D. View
E. Viewpoint

Answer : D

Explanation :

View is a representation of a system from the perspective of a related set of concerns.

See under : **31.1**

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

630 According to TOGAF, which of the following terms is defined as the key interests that are crucially important to stakeholders ?

A. Concerns
B. Principles
C. Requirements
D. Views
E. Viewpoints

Answer : A

Explanation :

Concerns are interests that are crucially important to stakeholders.

See under : **31.1**

"Concerns" are interests 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 considerations such as performance, reliability, security, distribution, and evolvability and may determine the acceptability of the system.

★ ★ ★ 631 Which of the following does TOGAF describe as a package of functionality defined to meet business needs across an organization ?

A. An application
B. An architecture viewpoint
C. A building block
D. A deliverable
E. A solution architecture

Answer : C

Explanation :

A Building block (especially an ABB) is a package of functionality defined to meet business needs across an organization.

See : **33.2.2 Generic Characteristics (of Building Blocks)**

■ A building block is a package of functionality defined to meet the business needs across an organization

■ A building block has a type that corresponds to the enterprise’s content metamodel (such as actor, business service, application, or data entity)

■ A building block has a defined boundary and is generally recognizable as "a thing" by domain experts

■ A building block may interoperate with other, inter-dependent building locks.

632 Which of the following is used to create architecture models addressing stakeholder concerns ?

A. Catalog
B. Matrix
C. Diagram
D. View
E. Viewpoint

Answer : D

Explanation :

View is used to create architecture models addressing stakeholder concerns

See under : **31.2 Developing Architecture Views in the ADM**

The choice of which particular architecture views to develop is one of the key decisions that the architect has to make.

The architect has a responsibility for ensuring the completeness (fitness-for-purpose) of the architecture, in terms of adequately addressing all the pertinent concerns of its stakeholders; and the integrity of the architecture, in terms of connecting all the various views to each other, satisfactorily reconciling the conflicting concerns of different stakeholders, and showing the trade-offs made in so doing (as between security and performance, for example).

Under **31.4.1 Example of Architecture Views and Architecture Viewpoints** : Example used says : One architecture view can be developed from the architecture viewpoint of the pilot, which addresses the pilot's concerns. An architecture viewpoint is a model (or description) of the information contained in a view.

633 Which of the following relationship between view, viewpoint and architecture is correct ?

A. View can also be called the definition or schema for that kind of architecture Viewpoint
B. An "architecture viewpoint" is a specification of the conventions for a particular kind of architecture view
C. A "Model " establishes conventions for a type of “Model kind”
D. An architecture View references one or more model kinds
E. An architecture Viewpoint incorporates one or more models.

Answer : B

Explanation :

Only answer choice B) is correct, as per TOGAF explanation about View and Viewpoints. All other answer choices give wrong points about View, Viewpoints, Model, and Model kind

See : **Chapter 31**

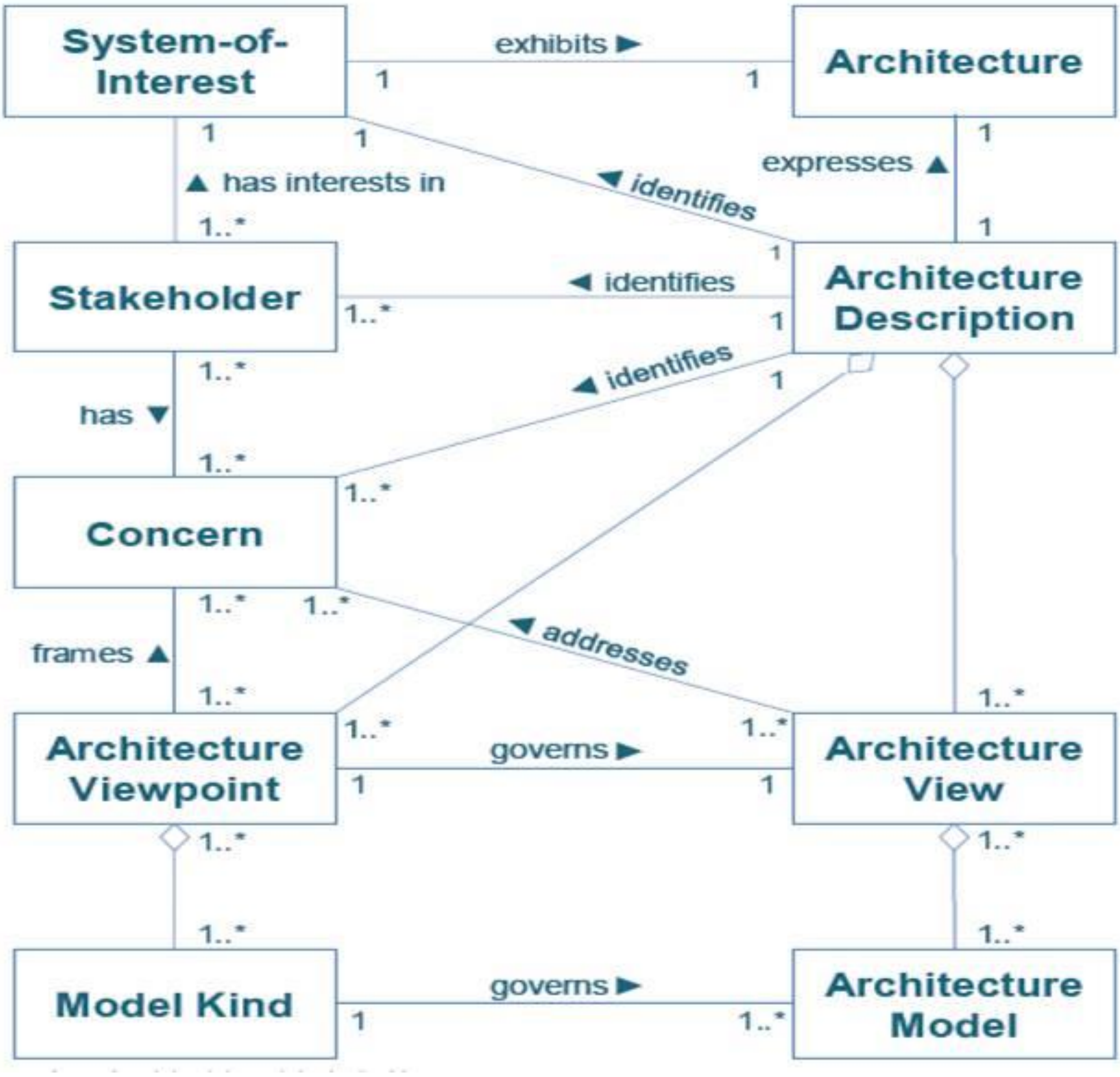
An "architecture viewpoint" is a specification of the conventions for a particular kind of architecture view. It can also be called the definition or schema for that kind of architecture view.

It establishes the conventions for constructing, interpreting, and using an architecture view to address a specific concern (or set of concerns) about a system-of-interest.

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

An architecture viewpoint references one or more model kinds; an architecture view incorporates one or more models.

An architecture view will comprise selected parts of one or more models, chosen so as to demonstrate to a particular stakeholder or group of stakeholders that their concerns are being adequately addressed in the design of the system architecture.



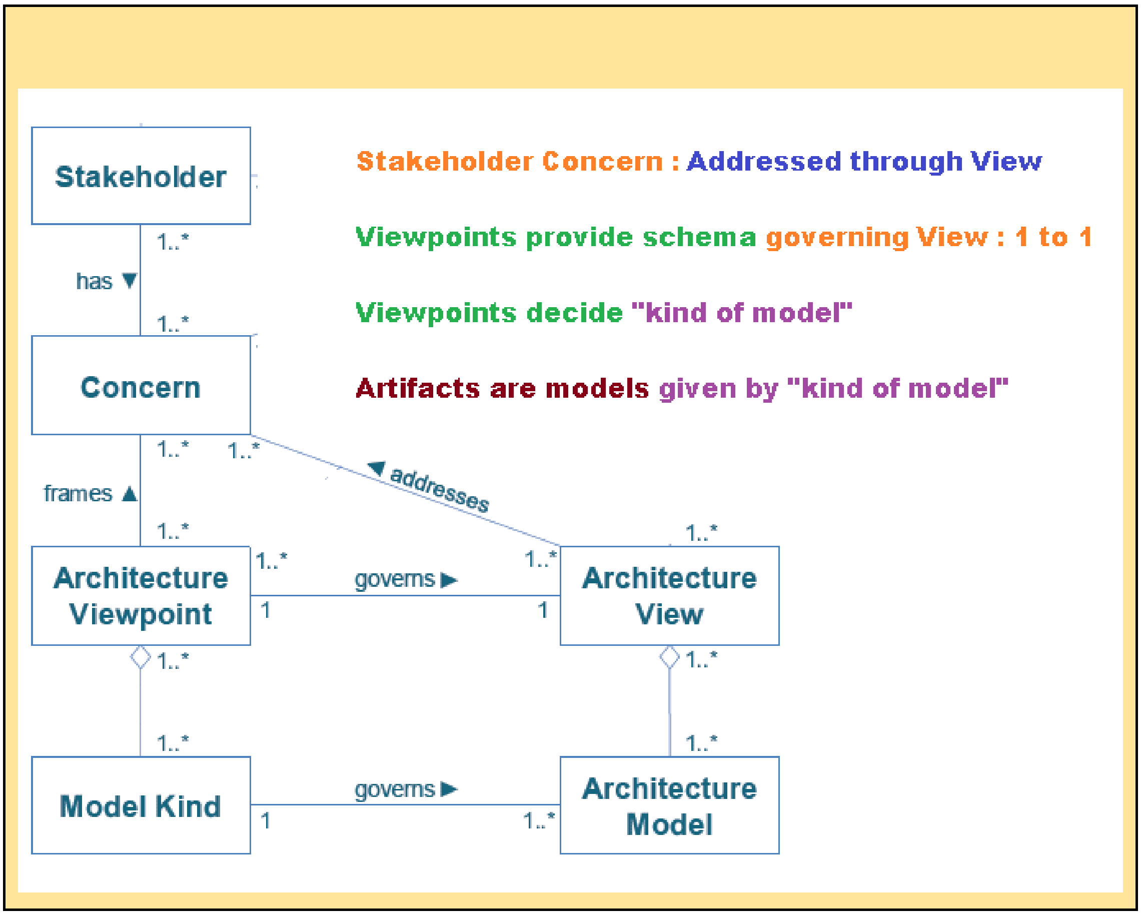
Stakeholders would have specific concern

View : What a stakeholder expects in order to address a concern

Relates to the artifact that an **Architect supplies**

Viewpoint : The definition or scheme for a View considered

towards addressing that concern



Stakeholder Concern : Addressed through View

Viewpoints provide schema governing View : 1 to 1

Viewpoints decide "kind of model"

Artifacts are models given by "kind of model"

634 Not a benefit of View Creation Process

A. Advantages of creating the required views by referring to library , selecting the viewpoints and generating the views
B. Reducing tasks for the architects
C. Offering better comprehensibility for stakeholders
D. Enhancing confidence in the validity of the views

Answer : A

Explanation :

Answer choice A) is not among the benefit of View Creation process as given in TOGAF documentation.

See Section : **31.2.2 : Architecture View Creation Process :**This approach can be expected to bring the following benefits :■ Less work for the 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)

Answer choice A is only about the View Creation Process and not the benefits

Create the required views for a particular architecture by following these steps:

1. Refer to an existing library of architecture viewpoints

2. Select the appropriate architecture viewpoints (based on the stakeholders and concerns that need to be covered by views)

3. Generate views of the system by using the selected architecture viewpoints as templates

635 One that 😎 does not apply to the term : Viewpoint

A. Specification of the conventions for a particular kind of architecture view
B. Definition or schema for that kind of architecture view
C. Establishes conventions for a type of modeling
D. Incorporates one or more models

Answer : D

Explanation :

A View incorporates one or more models; not the Viewpoint.

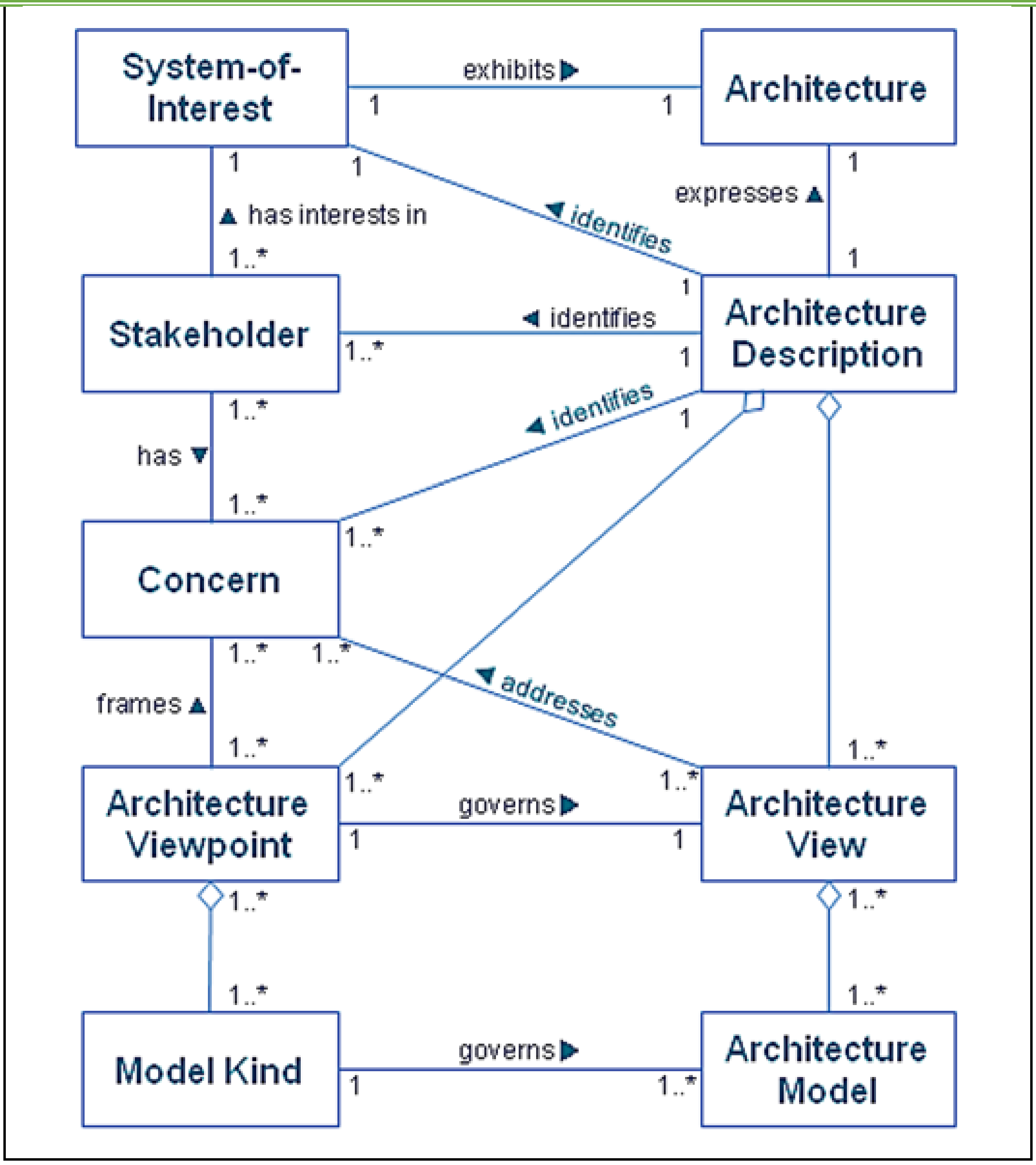
See : **Chapter 31**

An "architecture viewpoint" is a specification of the conventions for a particular kind of architecture view. It can also be called the definition or schema for that kind of architecture view.

It establishes the conventions for constructing, interpreting, and using an architecture view to address a specific concern (or set of concerns) about a system-of-interest.

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

An architecture viewpoint references one or more model kinds; an architecture view incorporates one or more models.


System-of-Interest
Architecture
exhibits ▶
1
1
1
1
has interests in
1..*
identifies
expresses
1
1
1
Stakeholder
Architecture Description
identifies
1..*
1
1..*
has
1..*
identifies
1
Concern
1..*
1..*
1..*
frames
addresses
1..*
1..*
1..*
1..*
Architecture Viewpoint
Architecture View
governs ▶
1
1
1..*
1..*
1..*
1..*
Model Kind
Architecture Model
governs ▶
1
1..*

636 Stakeholders and their concerns are key concepts in TOGAF. Which one of the following statements is false ?

A. Concerns are key interests that are crucially important to stakeholders

B. Concerns should be SMART and have specific metrics

C. Stakeholders can be individuals, teams, or organizations

D. Stakeholders have key roles in, or concerns about, the system

Answer : B

Explanation :

"Concerns" are the key interests that are crucially important to the stakeholders in the system, and determine the acceptability of the system. Concerns may pertain to any aspect of the system's functioning, development, or operation, including considerations such as performance, reliability, security, distribution, and evolvability. The terms "concern" and "requirement" are not synonymous. Concerns are the root of the process of decomposition into requirements. Concerns are represented in the architecture by these requirements. Requirements should be SMART (e.g., specific metrics).

SMART stands for : Specific, Measurable, Actionable, Realistic, and Timebound

Refer to TOGAF 9.2 online documentation while starting to prepare for Level 2 Questions

Very important to get to know parts of TOGAF documentation

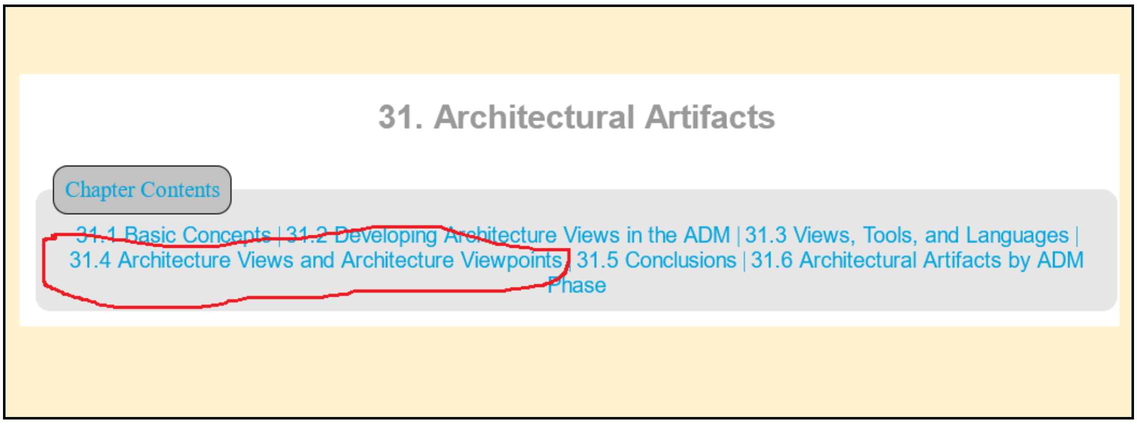
https://pubs.opengroup.org/architecture/togaf9-doc/arch/

You will need this link to be open most of the time in this course

Similar content will **open during your Level 2 Exam**.

Sorry, not during Level 1 Exam

This module may, by itself, may not be in Level 2 question area. But it pays to read the following Chapters, especially the portion marked in red



31. Architectural Artifacts

Chapter Contents

31.1 Basic Concepts | 31.2 Developing Architecture Views in the ADM | 31.3 Views, Tools, and Languages | 31.4 Architecture Views and Architecture Viewpoints | 31.5 Conclusions | 31.6 Architectural Artifacts by ADM Phase

Part 3 : Detailed Courseware

A video on Building Blocks :

https://www.youtube.com/watch?v=pdEfJOgbIH0

To get to know about different Enterprise Architecture Frameworks, watch this video

https://www.youtube.com/watch?v=Bd4LH_nTaAw

EAs receive inputs from Viewpoints

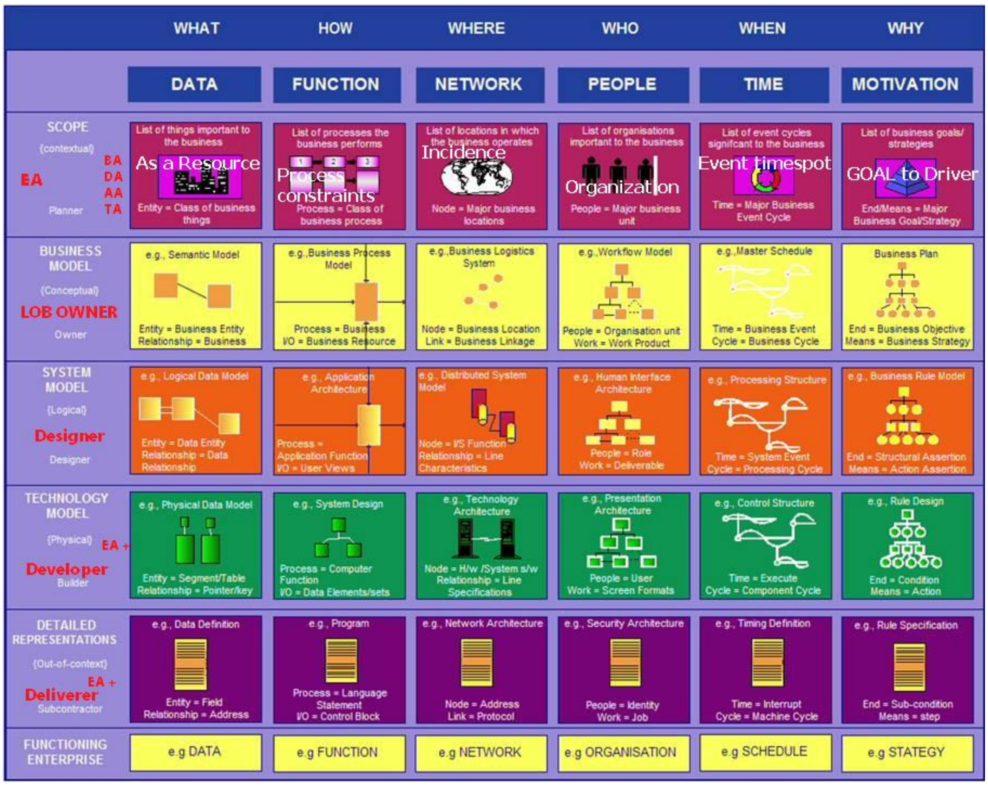
EAs provide output by way of Views

- What does this mean ?

Nice to Know Box

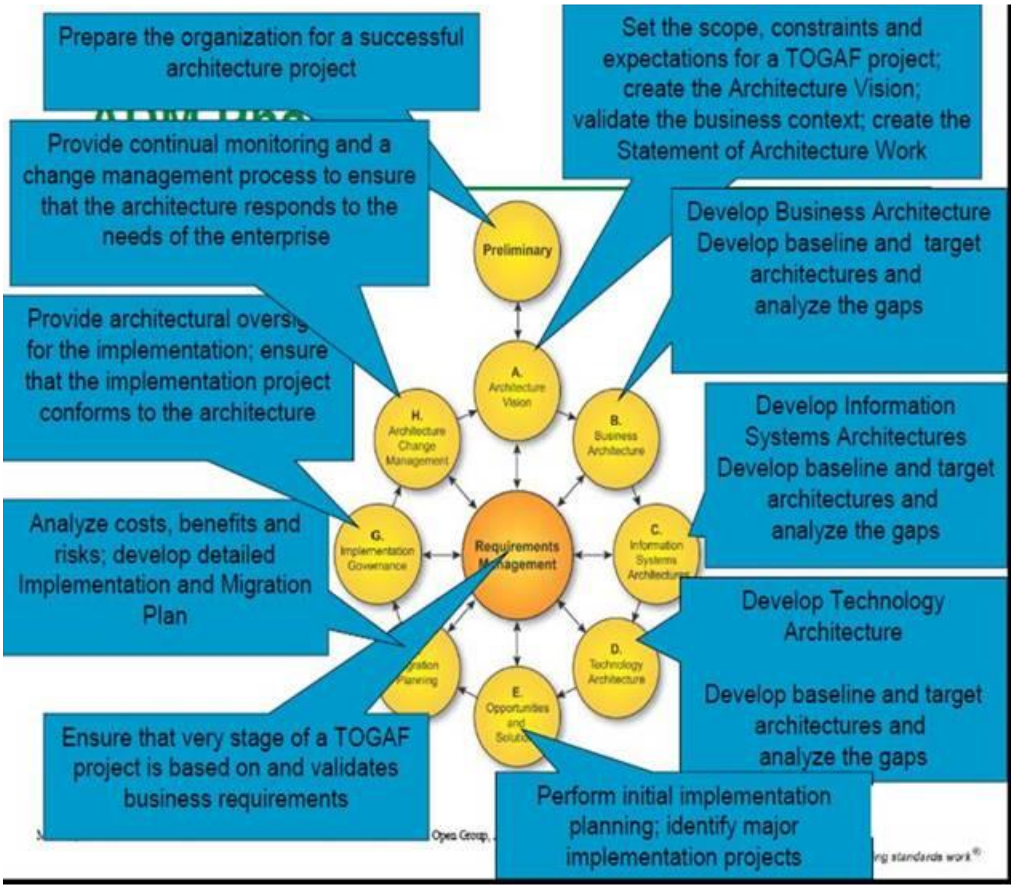
Comparing TOGAF and Zachman Framework

Zachman is a Classification Content Framework



	WHAT	HOW	WHERE	WHO	WHEN	WHY
	DATA	FUNCTION	NETWORK	PEOPLE	TIME	MOTIVATION
SCOPE {contextual} EA BA DA AA TA Planner	List of things important to the business As a Resource Entity = Class of business things	List of processes the business performs Process constraints Process = Class of business process	List of locations in which the business operates Incidence Node = Major business locations	List of organisations important to the business Organization People = Major business unit	List of event cycles signifcant to the business Event timespot Time = Major Business Event Cycle	List of business goals/ strategies GOAL to Driver End/Means = Major Business Goal/Strategy
BUSINESS MODEL {Conceptual} LOB OWNER Owner	e.g., Semantic Model Entity = Business Entity Relationship = Business	e.g.,Business Process Model Process = Business I/O = Business Resource	e.g.,Business Logistics System Node = Business Location Link = Business Linkage	e.g.,Workflow Model People = Organisation unit Work = Work Product	e.g.,Master Schedule Time = Business Event Cycle = Business Cycle	Business Plan End = Business Objective Means = Business Strategy
SYSTEM MODEL {Logical} Designer Designer	e.g., Logical Data Model Entity = Data Entity Relationship = Data Relationship	e.g., Application Architecture Process = Application Function I/O = User Views	e.g., Distributed System Model Node = I/S Function Relationship = Line Characteristics	e.g., Human Interface Architecture People = Role Work = Deliverable	e.g., Processing Structure Time = System Event Cycle = Processing Cycle	e.g., Business Rule Model End = Structural Assertion Means = Action Assertion
TECHNOLOGY MODEL {Physical} EA + Developer Builder	e.g., Physical Data Model Entity = Segment/Table Relationship = Pointer/key	e.g., System Design Process = Computer Function I/O = Data Elements/sets	e.g., Technology Architecture Node = H/w /System s/w Relationship = Line Specifications	e.g., Presentation Architecture People = User Work = Screen Formats	e.g., Control Structure Time = Execute Cycle = Component Cycle	e.g., Rule Design End = Condition Means = Action
DETAILED REPRESENTATIONS {Out-of-context} EA + Deliverer Subcontractor	e.g., Data Definition Entity = Field Relationship = Address	e.g., Program Process = Language Statement I/O = Control Block	e.g., Network Architecture Node = Address Link = Protocol	e.g., Security Architecture People = Identity Work = Job	e.g., Timing Definition Time = Interrupt Cycle = Machine Cycle	e.g., Rule Specification End = Sub-condition Means = step
FUNCTIONING ENTERPRISE	e.g DATA	e.g FUNCTION	e.g NETWORK	e.g ORGANISATION	e.g SCHEDULE	e.g STATEGY

TOGAF is Process Centric Framework



Preliminary
A. Architecture Vision
B. Business Architecture
C. Information Systems Architecture
D. Technology Architecture
E. Opportunities and Solut
ration Planning
G. Implementation Governance
H. Architecture Change Management
Requirements Management
Open Group
ng standards work®

Prepare the organization for a successful architecture project

Set the scope, constraints and expectations for a TOGAF project; create the Architecture Vision; validate the business context; create the Statement of Architecture Work

Provide continual monitoring and a change management process to ensure that the architecture responds to the needs of the enterprise

Develop Business Architecture Develop baseline and target architectures and analyze the gaps

Provide architectural oversig for the implementation; ensure that the implementation project conforms to the architecture

Develop Information Systems Architectures Develop baseline and target architectures and analyze the gaps

Analyze costs, benefits and risks; develop detailed Implementation and Migration Plan

Develop Technology Architecture

Develop baseline and target architectures and analyze the gaps

Ensure that very stage of a TOGAF project is based on and validates business requirements

Perform initial implementation planning; identify major implementation projects

TOGAF provides a comprehensive approach

to the Design, Planning, Implementation Guidance and Governance Oversight

of **Enterprise Architecture**

TOGAF describes an Architecture Development Methodology

without clearly describing the artifacts that are to be developed using the methodology

The Zachman Framework is **not a methodology**; it is a template describing how different abstract ideas are viewed from different perspectives. It is an ontology matrix forming a logical structure for classifying and organizing artifacts.

Zachman is focused on identifying the different viewpoints that might be relevant for different purposes while TOGAF is focused on the process of developing architectures.

Zachman Framework classifies artifacts and models into a Template of Cell

TOGAF is ADM process centric and

we get guidelines in terms of **Views to be produced**

by **looking first from various Viewpoints**

Nice to Know Box

A stakeholder is a person or an organization with an interest in the system.

An Architectural description should identify the stakeholders. A system will have at least one stakeholder, and a stakeholder may be a stakeholder of different systems.

A stakeholder has at least one concern (an interest of a stakeholder), and each concern may be shared by different stakeholders.

Examples of stakeholders are :

– Acquirers are those who decide which system to use

– Assessors are those who check whether a given system conforms to needs or constraints

– Communicators are responsible for training and documentation

– Developers develop the system

– Contributors develop or write documentation

– Committers take decisions in the development process

– Maintainers fix bugs and evolve the system

– Suppliers provide components

– Supporters provide help to users

– System administrators, administer users, and those who configure the system

– Testers test (parts of) the system

– **End Users** use it

Nice to Know Box

The TOGAF framework is designed to be flexible and is used with various Architectural Styles.

Architectural styles differ in terms of focus, form, techniques, materials, subject, and time period. The TOGAF Standard is a generic framework intended to be used in a wide variety of environments. It is a flexible and extensible framework that can be readily adapted to a number of architectural styles.

An organization's Architecture Landscape can be expected to contain architecture work that is developed in many architectural styles. The TOGAF Standard ensures that the needs of each stakeholder are appropriately addressed in the context of other stakeholders and the Baseline Architecture.

When using the TOGAF Standard to support a specific architectural style the practitioner must take into account the combination of distinctive features in which architecture is performed or expressed. As a first step, the distinctive features of a style must be identified.

The second step is determining how these distinctive features will be addressed. Addressing a distinctive style should not call for significant changes to the TOGAF framework; instead it should adjust the models, viewpoints, and tools used by the practitioner.

In Phase B, Phase C, and Phase D the practitioner is expected to select the relevant architecture resources, including models, viewpoints, and tools, to properly describe the architecture domain and demonstrate that stakeholder concerns are addressed.

Dominance of a particular architectural style can direct the practitioner to revisit the Preliminary Phase to make changes to the Architecture Capability or to address a distinctive feature in the expected scope of a single ADM cycle.

Style-specific reference models and maturity models are commonly used tools that support a practitioner.

During the lifetime of the TOGAF Framework many architectural styles have been developed to address key problems facing practitioners and to demonstrate how the TOGAF framework can be made more relevant within defined contexts.

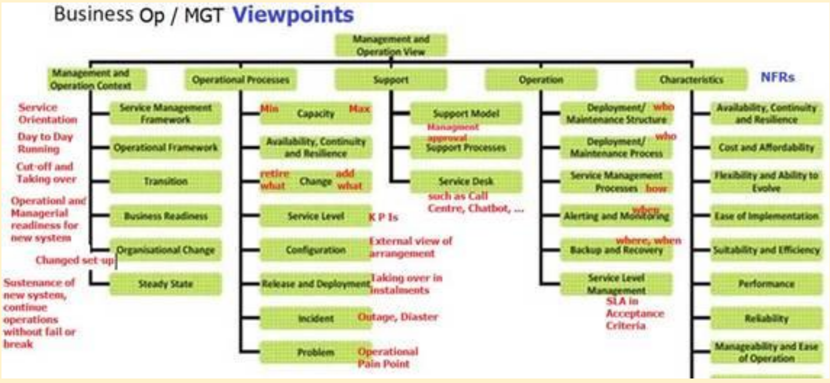
(From TOGAF 10 documentation, Relevant for TOGAF 9.2 also)

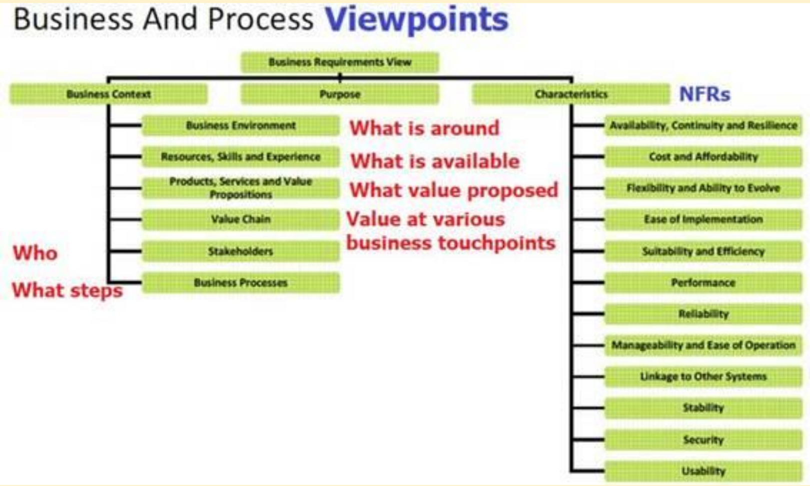
Relating to the **Case Study**

Nice to Know Box

Points of Essence : The Viewpoints shown at various points int eh Case Study gives us the picture of what kind of Stakeholders will be part of this journey of Architecture for the current Portfolios :

Business Architecture Viewpoints :


Business Op / MGT Viewpoints
Management and Operation View
Management and Operation Context
Operational Processes
Support
Operation
Characteristics
NFRs
Service Orientation
Service Management Framework
Day to Day Running
Operational Framework
Cut-off and Taking over
Transition
Operationl and Managerial readiness for new system
Business Readiness
Organisational Change
Changed set-up
Steady State
Sustenance of new system, continue operations without fail or break
Min
Capacity
Max
Availability, Continuity and Resilience
retire what
Change
add what
Service Level
K P Is
Configuration
External view of arrangement
Release and Deployment
Taking over in instalments
Incident
Outage, Diaster
Problem
Operational Pain Point
Support Model
Management approval
Support Processes
Service Desk
such as Call Centre, Chatbot, ...
Deployment/ Maintenance Structure
who
Deployment/ Maintenance Process
who
Service Management Processes
how
Alerting and Monitoring
when
Backup and Recovery
where, when
Service Level Management
SLA in Acceptance Criteria
Availability, Continuity and Resilience
Cost and Affordability
Flexibility and Ability to Evolve
Ease of Implementation
Suitability and Efficiency
Performance
Reliability
Manageability and Ease of Operation


Business And Process Viewpoints
Business Requirements View
Business Context
Purpose
Characteristics
NFRs
Business Environment
What is around
Resources, Skills and Experience
What is available
Products, Services and Value Propositions
What value proposed
Value Chain
Value at various business touchpoints
Stakeholders
Who
Business Processes
What steps
Availability, Continuity and Resilience
Cost and Affordability
Flexibility and Ability to Evolve
Ease of Implementation
Suitability and Efficiency
Performance
Reliability
Manageability and Ease of Operation
Linkage to Other Systems
Stability
Security
Usability

Operations Viewpoint – For LOB stakeholders

Management Viewpoint – For Top Management who would approve

Financial Viewpoint – For Budget sanction

A Viewpoint designates the most appropriate perspective **for a given participant**, and is materialized by a certain number of views of the architecture

Operational Process Viewpoints :

New changes needed : Online eCommerce : Point of view of smooth operations

Service Level therein : Response to Customers : Concern on speed and Web failure

Support Operations Viewpoint : Need some kind of online Service Desk

(Internal) Operations Viewpoints :

What are the alerts and monitoring reports and dashboards needed : Concern of people who are looking into 24 X 7 Operations

How will all data be backed up ?

NFR – Architectural Quality Characteristics as seen from Business Architecture Viewpoint :

Changes will keep coming. What is the business and technology strategy ?

Degree of Scalability and Security possible

Best of User Experience : How to take care of this

Application Architecture Viewpoints :

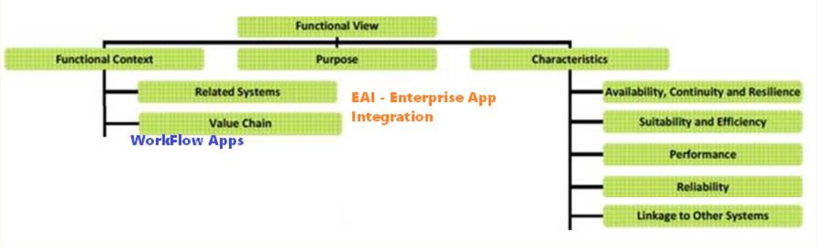
What points of view, of which Stakeholder to focus and proceed during AA work ?

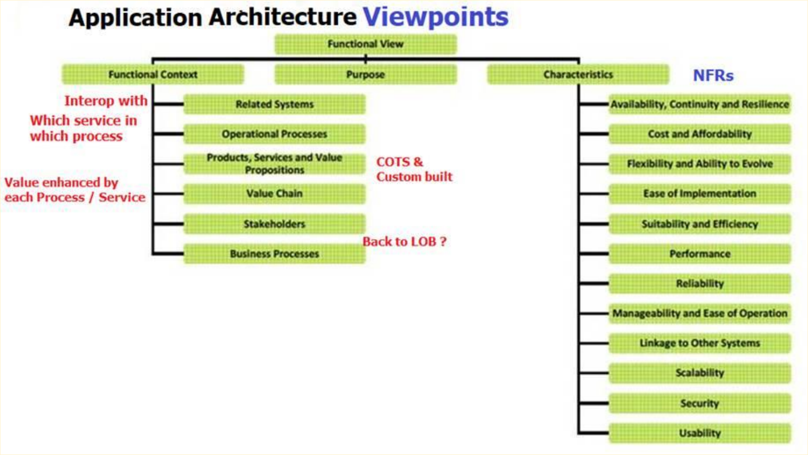
Functional Context

User Experience Operations

NFR related Operations

Immediate focus is on :





Data Architecture Viewpoints :

What points of view, of which Stakeholder to focus and proceed during DA work ?

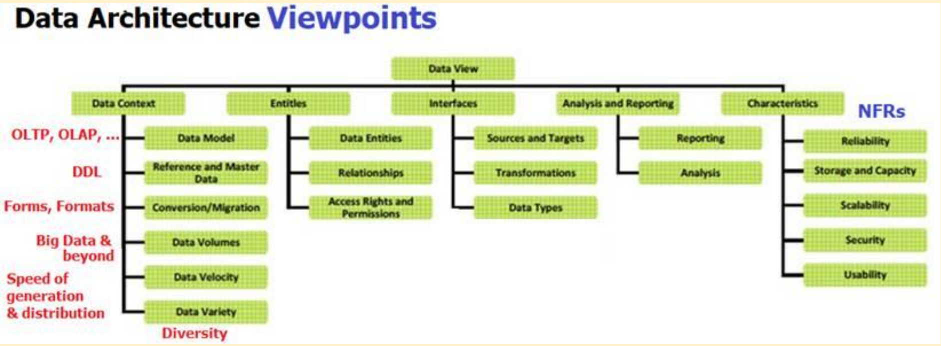
Data Context

Entities and Relationships

Sources, Sinks, Transformations

Data Visualization

Data centric NFRs



Data Architecture Viewpoints

Data View

Data Context
- Data Model
- Reference and Master Data
- Conversion/Migration
- Data Volumes
- Data Velocity
- Data Variety

OLTP, OLAP, ...

DDL

Forms, Formats

Big Data & beyond

Speed of generation & distribution

Diversity

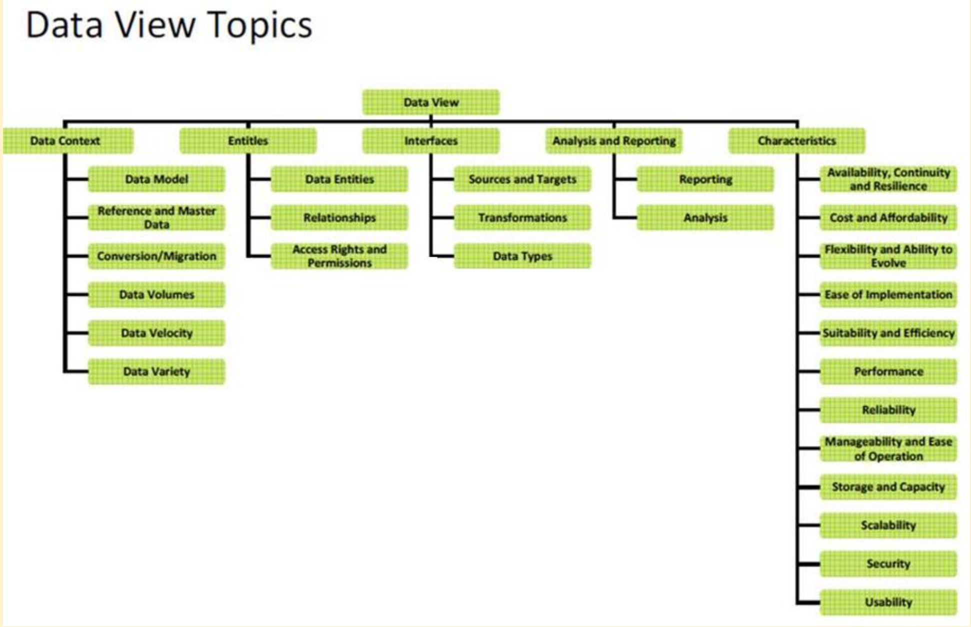
Entities
- Data Entities
- Relationships
- Access Rights and Permissions

Interfaces
- Sources and Targets
- Transformations
- Data Types

Analysis and Reporting
- Reporting
- Analysis

Characteristics
- Reliability
- Storage and Capacity
- Scalability
- Security
- Usability

NFRs



Data View Topics

Data View

Data Context
- Data Model
- Reference and Master Data
- Conversion/Migration
- Data Volumes
- Data Velocity
- Data Variety

Entities
- Data Entities
- Relationships
- Access Rights and Permissions

Interfaces
- Sources and Targets
- Transformations
- Data Types

Analysis and Reporting
- Reporting
- Analysis

Characteristics
- Availability, Continuity and Resilience
- Cost and Affordability
- Flexibility and Ability to Evolve
- Ease of Implementation
- Suitability and Efficiency
- Performance
- Reliability
- Manageability and Ease of Operation
- Storage and Capacity
- Scalability
- Security
- Usability

Data Context Views

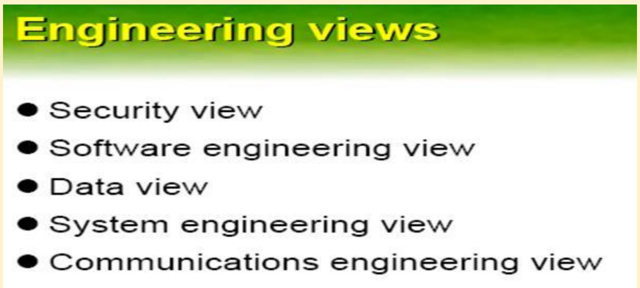
Data Model : Start with OLTP for all transaction operations in the Portfolios

Build up a set of Data Warehouses and Data Marts for the areas of : …

Extend to Big Data Analytics; Aspire on longer horizon for Data Lake, Lake House. AI driven Analytics

Technology Infrastructure Architecture Viewpoints :

To select and develop the relevant architecture Viewpoints that will enable the Technology - Infrastructure Segment Architect to demonstrate how the stakeholder concerns are addressed. Step is meant for EA to point out the Artifact Components – Viewpoints to be created as per TA Target specified. EA gives this list to Technology - Infrastructure (Segment) Architect.



Engineering views

- Security view
- Software engineering view
- Data view
- System engineering view
- Communications engineering view

What points of view, of which Stakeholder to focus and proceed during TA work ?

Context of Tools and Environment

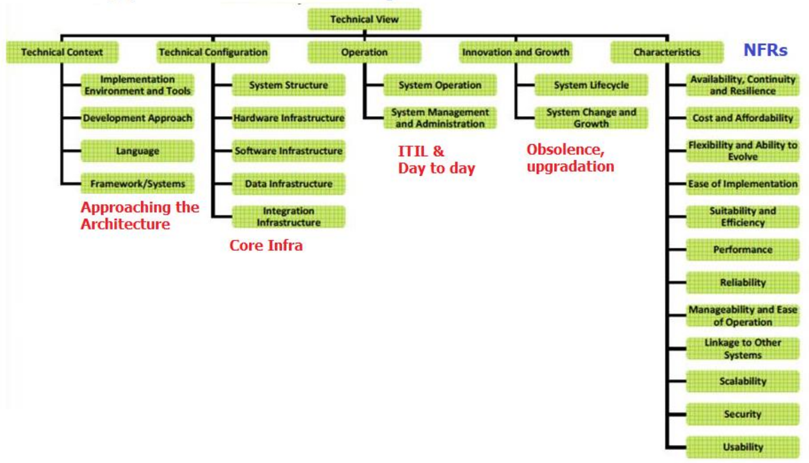
Infrastructure Configuration

Operations and Administration

Lifecycle, Change and Growth

NFRs as a mainstay

Technology Architecture Viewpoints



Nice to Know Box

"Deliverable" contains "artifacts" which describe "Building Blocks".

As 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 the Enterprise

The Architecture Repository is a software tool that stores the important architectural input and output, including Architectures themselves, the elements of which they are composed, standards, references, principles and the Governance Register.

Nice to Know Box

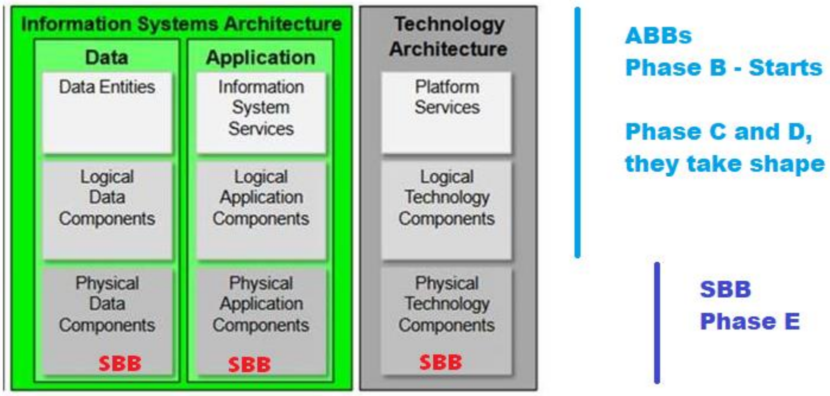
Categories of Building Blocks

ABBs are more of conceptual architectural work that comes out when we get into planning based on the requirement.

ABBs capture the common business requirements and represent the general Architecture resulting from that.

It is not ready to be implemented in any specific environment, language, platform.

It usually stops with some High Level edifices. This is because, they represent the **Logical structure ONLY**.


Information Systems Architecture
Data
Data Entities
Logical Data Components
Physical Data Components
SBB
Application
Information System Services
Logical Application Components
Physical Application Components
SBB
Technology Architecture
Platform Services
Logical Technology Components
Physical Technology Components
SBB
ABBs
Phase B - Starts
Phase C and D, they take shape
SBB
Phase E

SBBs take it forward with Low Level detailing

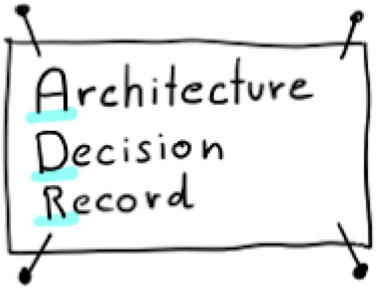
In other words, SBBs become Platform specific

They represent the technical capabilities as specific to and available in a specific environment, coding language and deployment platforms.

Hence they are ready to be implemented and can move to PMO – Project Management Office from the Enterprise Architecture department.

Nice to Know Box

ABBs can be related to **ADRs**



Nice to Know Box

An **Architecture Decision Record** (ADR) is **a document that captures an important architectural decision made along with its context and consequences**. An architecture decision (AD) is a software design choice that addresses a significant requirement.

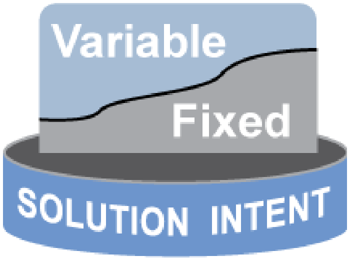
An ADR captures the key options available, the main requirements that drive a decision, and the **design decisions themselves**.

ADRs can also help you run more reliable applications and services. The ADR helps you understand your current state and troubleshoot when there is a problem or technology change in future, sometimes forcing us to change the solution track of solutioning technology product choice.

ADRs also build a collection of engineering decisions **to help future decision choices and deployments. - Reuse potential**

Nice to Know Box

SBBs can be related to Solution Intent Design



Nice to Know Box

The Solution Intent **provides the basic understanding of the current and evolving requirements, design and "intent" -- or larger purpose -- of the solution.**

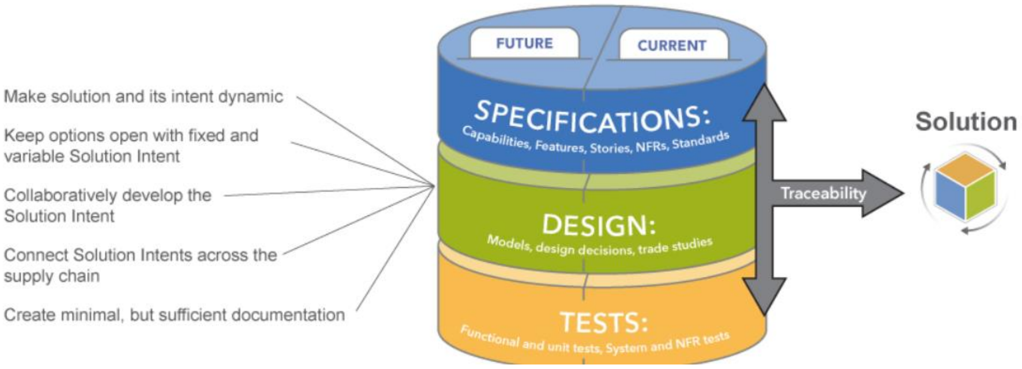
They are just one step ahead of Agile Coding.

The key elements of the Solution Intent : **Product and technology focused Specifications and Designs and possibly test cases**

Nice to Know Box

From : SAFe® | SCALED AGILE®

Solution Intent is the repository for storing, managing, and communicating the knowledge of current and intended Solution behavior. Where required, this includes both fixed and variable specifications and designs; reference to applicable standards, system models, and functional and nonfunctional tests; and traceability.


Make solution and its intent dynamic
Keep options open with fixed and variable Solution Intent
Collaboratively develop the Solution Intent
Connect Solution Intents across the supply chain
Create minimal, but sufficient documentation
FUTURE
CURRENT
SPECIFICATIONS:
Capabilities, Features, Stories, NFRs, Standards
DESIGN:
Models, design decisions, trade studies
TESTS:
Functional and unit tests, System and NFR tests
Traceability
Solution

Nice to Know Box

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 a full specification.

Building Blocks can relate to "architectures" or "solutions".

Architecture Building Blocks (ABBs) typically describe required capability and shape the specification of Solution Building Blocks (SBBs). For example, a customer services capability may be required within an Enterprise, supported by many SBBs, such as processes, data, and application software.

We will be relating this to ADR – Architecture Decision Records

Solution Building Blocks (SBBs) represent components that will be used to implement the required capability. For example, an Azure or AWS product of service is a Building Block that can be described through complementary artifacts and then put to use to realize solutions for the Enterprise.

We will be relating this to Solution Intent Design

From the **Case Study** :

Nice to Know Box

Phase B, Business Architecture lays down the Requirement as seen from business and related needs

Guess the kind of ABBs developed in following Phases of the same Case Study :

Application Architecture

Data Architecture

Technology / Infrastructure Architecture

Then also make a reasonable guess as to who will transform these into SBBs, preferably as reusable, fairly independent Services

Nice to Know Box

Architecture Building Blocks

More of Architecture and High-Level Design

Platform Independent, especially with D A T Segments

Architecture Building Blocks (ABBs) relate to the **Architecture Continuum**, and are defined or selected as a result of the application of the ADM. An ABB has fundamental functionality and attributes : Such as semantic, being unambiguous etc.,

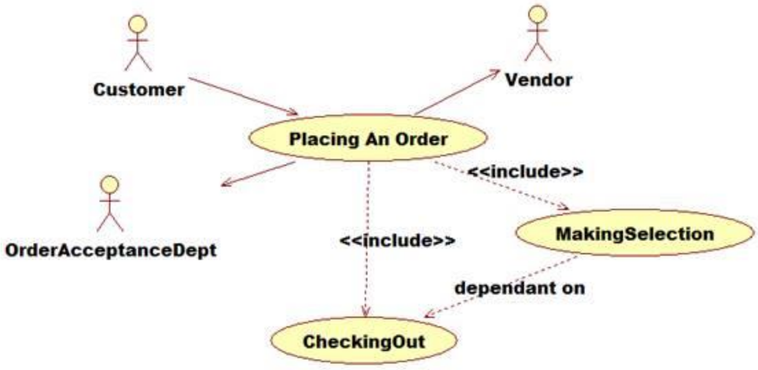
ABBs

Play a major role during Review of Architecture with relevant Stakeholders

Any Architect may refer to ABBs at some stage or other

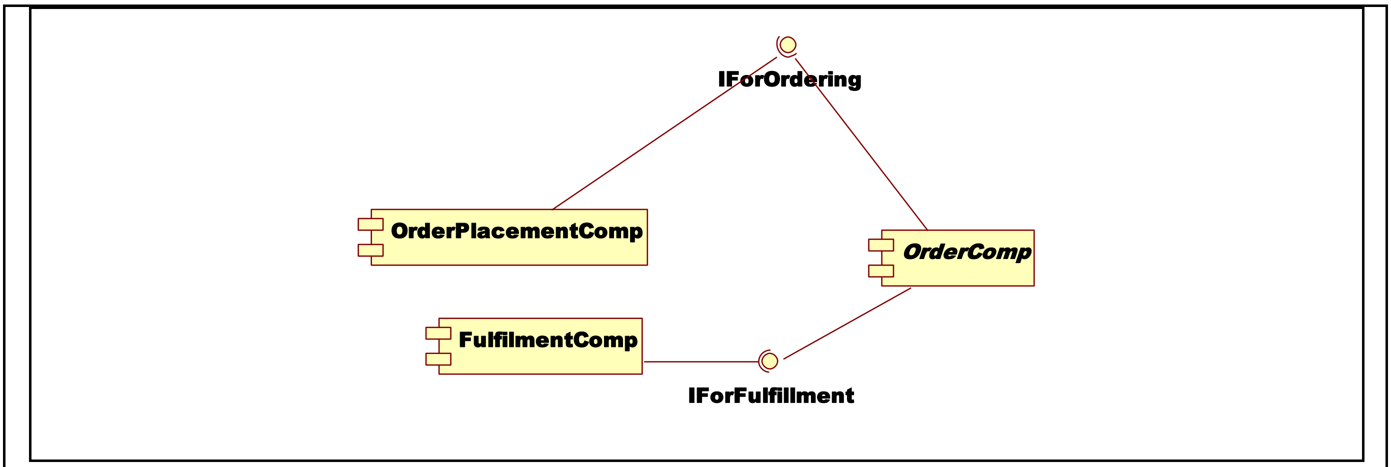
Use Case as an ABB Sample

: aimed at LOB Stakeholders



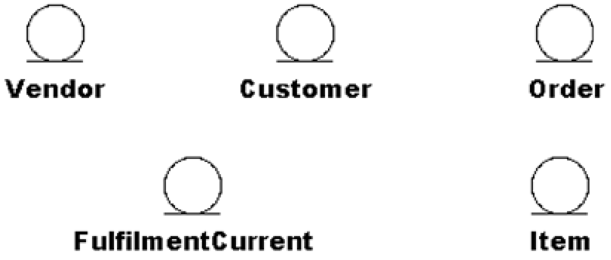
Component Diagram as ABB

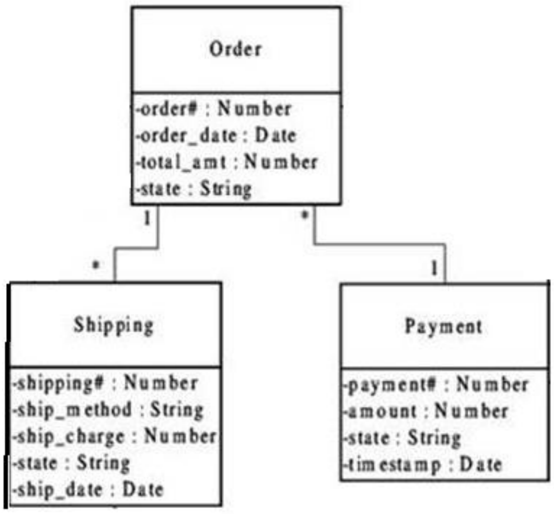
: aimed at Application related Stakeholders



Conceptual and Logical Data Diagrams as ABB

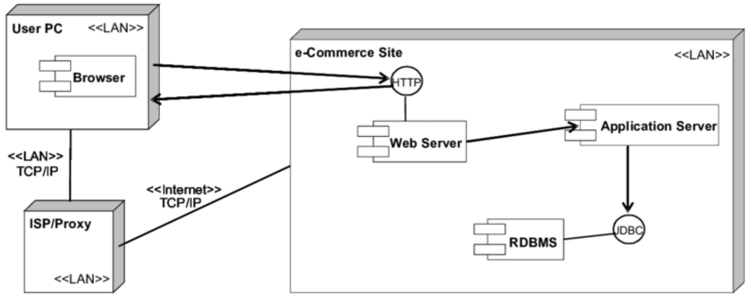
: aimed at Data Centric Stakeholders. Like DBAs





High Level Infrastructure Diagrams as ABB

: aimed at appropriate Stakeholders like Admins



Nice to Know Box

ABB : More look at Patterns than how they get implemented

Infinite Scalability,

By Default Interoperable

Plan ahead towards DevOps as Lifecycle, Agile as last mile strategy

Align with End-to-End Digitization

Each ABB an attempt to raise Capability level

○

Nice to Know Box

The modern market environment also brings demands to enhance system availability and scalability, two characteristics that are drivers in the adoption of MSA – Micro Services Architecture

MSA is an architecture style that defines and creates systems through the use of small independent and self-contained services that align closely with business activities.

An individual microservice is a service that is implemented with a single purpose, that is self-contained, and that is independent of other instances and services. **Microservices are the primary Architecture Building Blocks (ABBs) of an MSA**. As part of an Enterprise Architecture, MSA brings a layer of service capabilities to extend the business capabilities of the enterprise and, where the enterprise is a small self-contained service, MSA may be perceived to be the Enterprise Architecture.

Microservices are the primary ABBs of an MSA. An MSA has the following three key characteristics :

- Service-independence : each microservice is independent of other microservices; each microservice is developed, deployed, and changed independently
- Single responsibility : each microservice is mapped to an atomic (single) business activity for which it is responsible

- Self-containment : a microservice is a self-contained, independent deployable unit encompassing all external Information Technology (IT) resources (e.g., data sources, business rules) necessary to support the unique business activity

Quoting from TOGAF 10 Series Guide : Microservices Architecture (MSA)

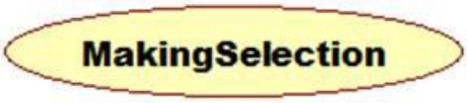
Nice to Know Box

Characteristics of ABBs :

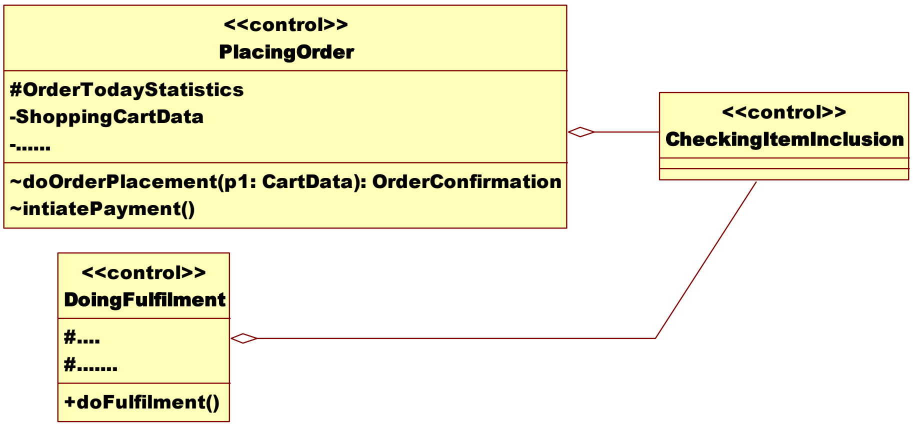
Capture Architecture requirements;

e.g., business, data, application, and technology requirements

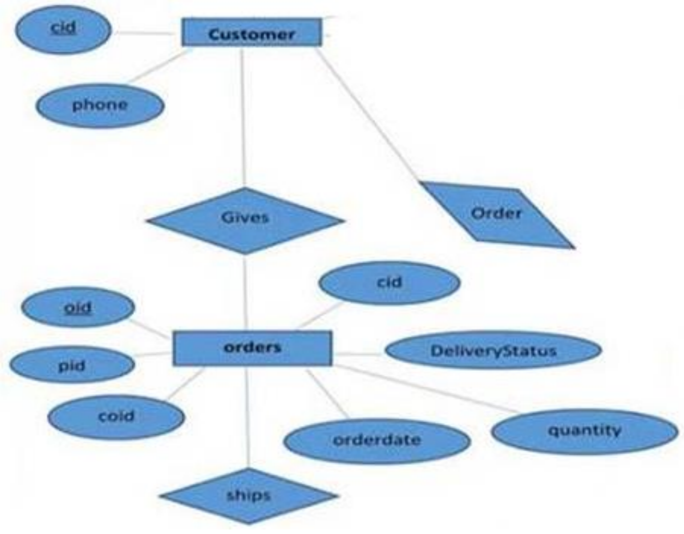
– A Business Requirement as captured here is :



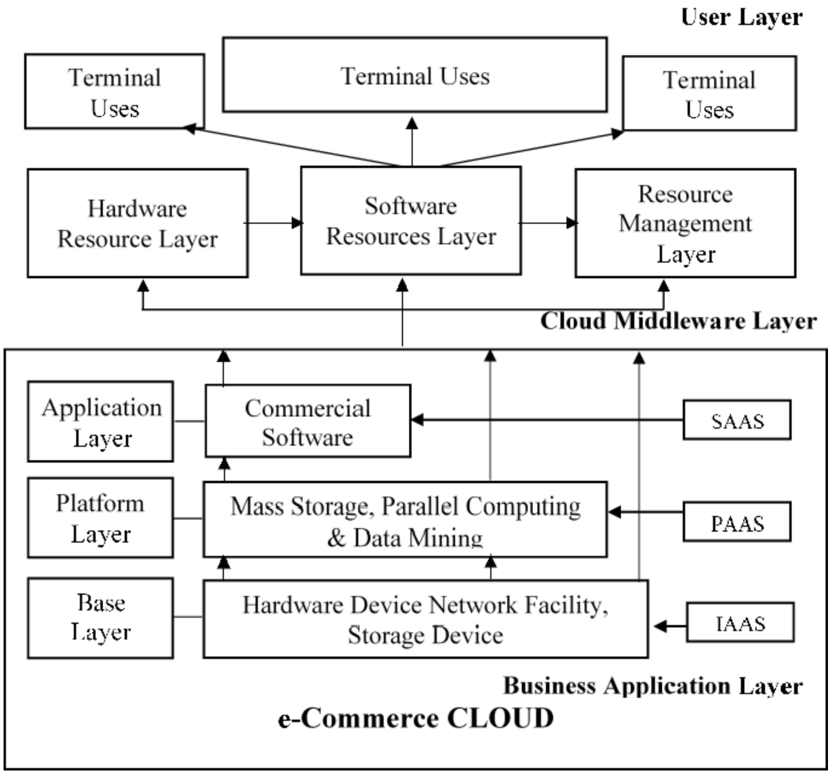
– An Application related (Control) Requirement as captured here is :



– A Data related Requirement as captured here is :



– A Technology – Infrastructure related Requirement as captured here is :



Nice to Know Box

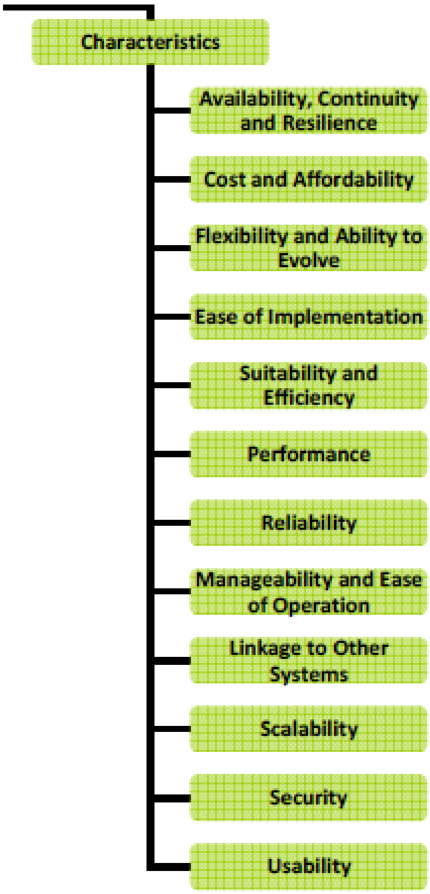
Architecture Building Blocks

Characteristics of ABBs :

Fundamental functionality

Functional and moving to Non Functional

Non Functional : Like Security, Scalability …


Characteristics
Availability, Continuity and Resilience
Cost and Affordability
Flexibility and Ability to Evolve
Ease of Implementation
Suitability and Efficiency
Performance
Reliability
Manageability and Ease of Operation
Linkage to Other Systems
Scalability
Security
Usability

Nice to Know Box

Specification Content of ABB : Business Architecture

Fundamental functionality and attributes : semantic, unambiguous, including security, capability and manageability

CheckingOut

: A sample short format of description would be :

At the time of checkout, final information of items and quantity, price with taxes, delivery indication, delivery mode and such details are shown for final confirmation. Changes in selection are allowed till the time payment option is chosen. Payment is completed with multiple alternate options. Order confirmation is provided through a minimum of two communication channels.

Consequent action involves the point of fulfilment. This can be the warehouse control section where stocking is done and the vendor in other cases. The command to start the fulfilment is triggered therein.

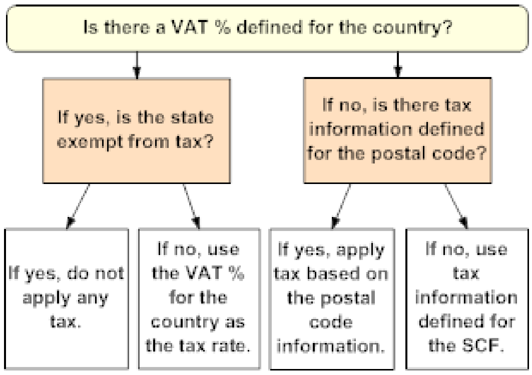
Every order results in updating of a number of records and statistics, These are carried out.

Architect notes in addition would be :

Payment information should be secure

Response time after request shall be xx seconds

Web and System outage must be dealt with alternatives


Is there a VAT % defined for the country?
If yes, is the state exempt from tax?
If no, is there tax information defined for the postal code?
If yes, do not apply any tax.
If no, use the VAT % for the country as the tax rate.
If yes, apply tax based on the postal code information.
If no, use tax information defined for the SCF.

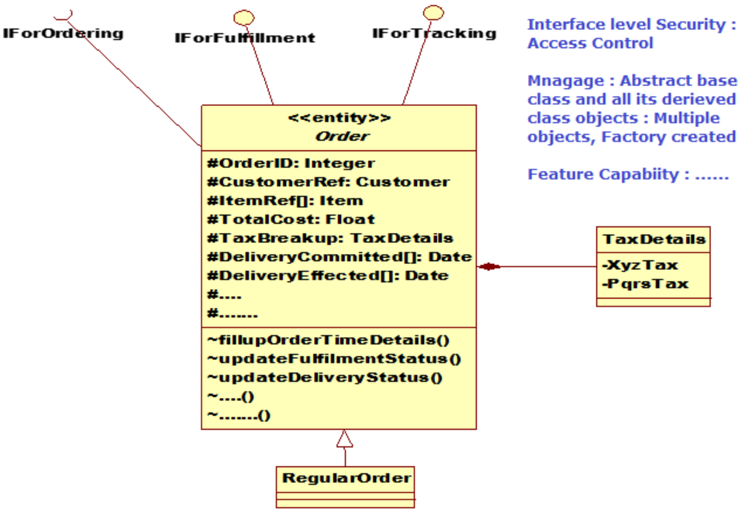
Nice to Know Box

Specification Content of ABB : Application Architecture

Fundamental functionality and attributes : semantic, unambiguous,

including security, capability and manageability

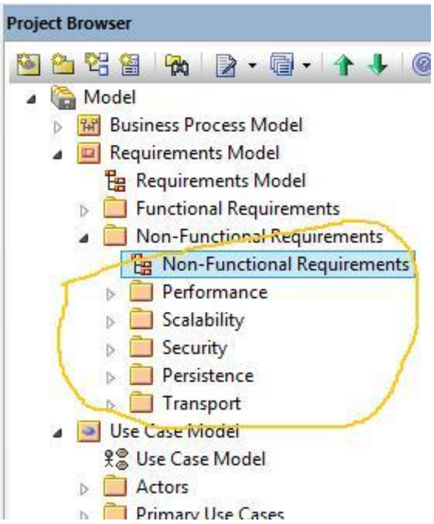
Order Component is specified as :



Interface level Security : Access Control

Mnagage : Abstract base class and all its derieved class objects : Multiple objects, Factory created

Feature Capabiity : ......

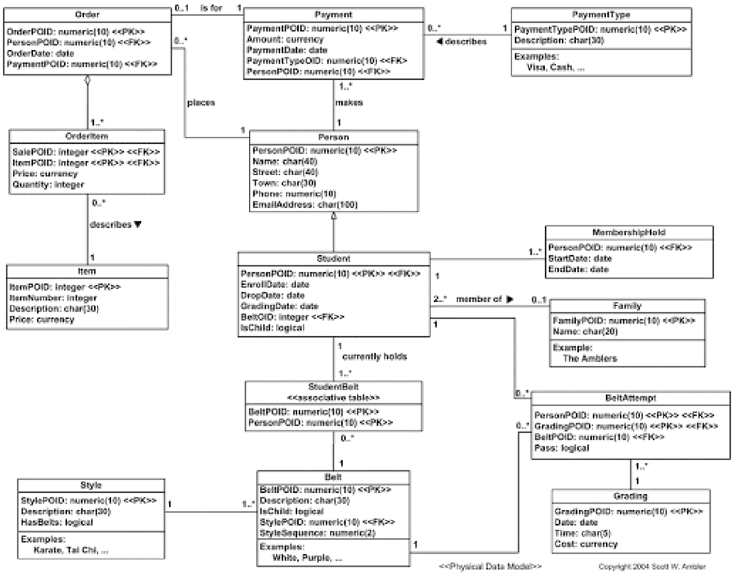


Functional and moving to Non Functional

Nice to Know Box

Specification Content of ABB : Data Architecture

Fundamental functionality and attributes : semantic, unambiguous, including security, capability and manageability

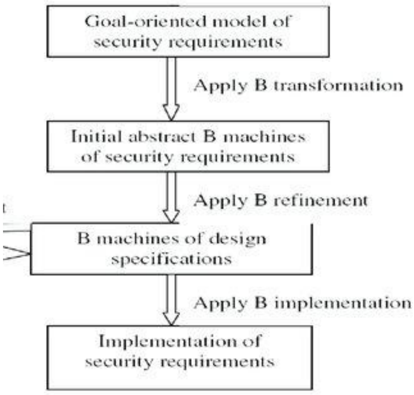


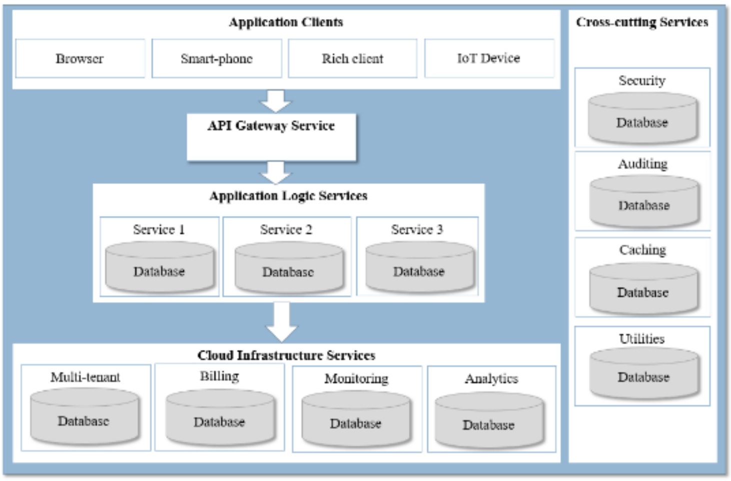
Functional and moving to Non Functional

Nice to Know Box

Specification Content of ABB : Technology Architecture

Fundamental functionality and attributes : semantic, unambiguous, including security, capability and manageability


Goal-oriented model of security requirements
Apply B transformation
Initial abstract B machines of security requirements
Apply B refinement
B machines of design specifications
Apply B implementation
Implementation of security requirements



Functional and moving to Non Functional

Nice to Know Box

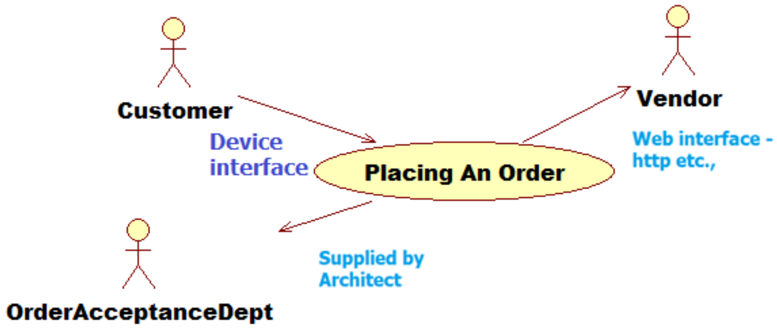
Specification Content : ABB specifications include the following as a minimum : **Interfaces : chosen as a set,** supplied appropriately

How to connect

Nice to Know Box

Specification Content of ABB : Business Architecture

Interfaces : chosen as a set, supplied

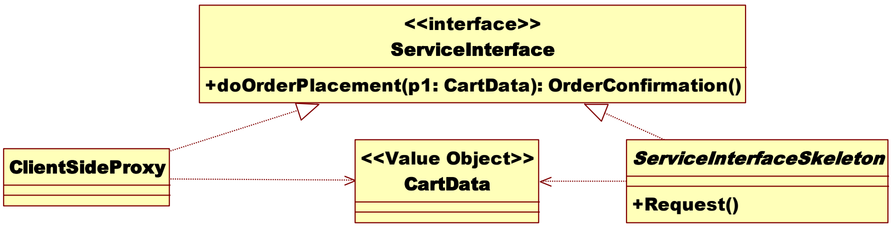


How to connect

Nice to Know Box

Specification Content of ABB : Application Architecture

Interfaces : chosen as a set, supplied

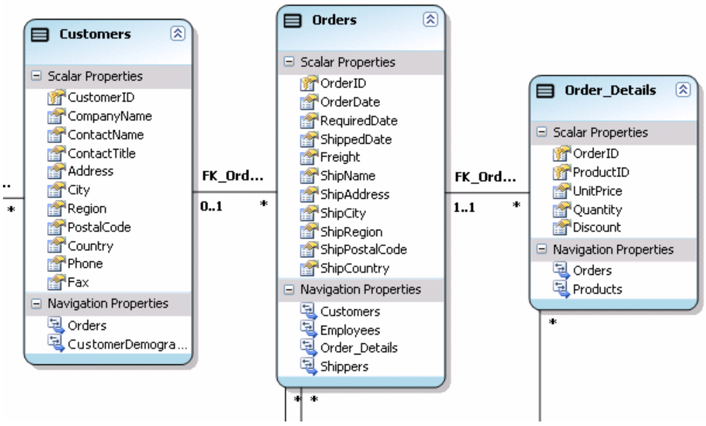


How to connect

Nice to Know Box

Specification Content of ABB : Data Architecture

Interfaces : chosen as a set, supplied

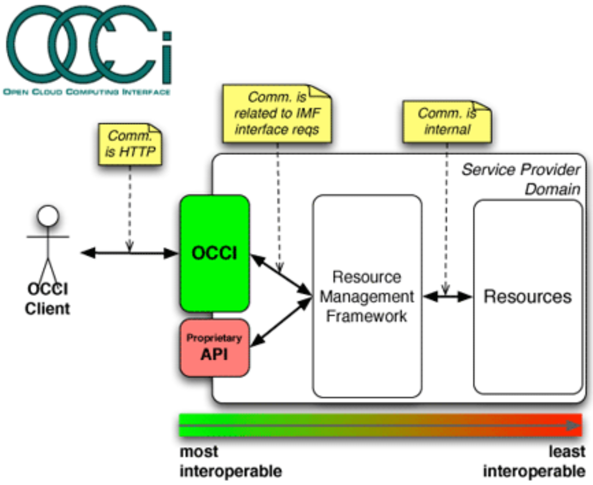


How to connect

Nice to Know Box

Specification Content of ABB : Technology Architecture

Interfaces : chosen as a set, supplied



Open Cloud Computing Interface Architecture

How to connect

.

Nice to Know Box

Architecture Building Blocks

Characteristics of ABBs :

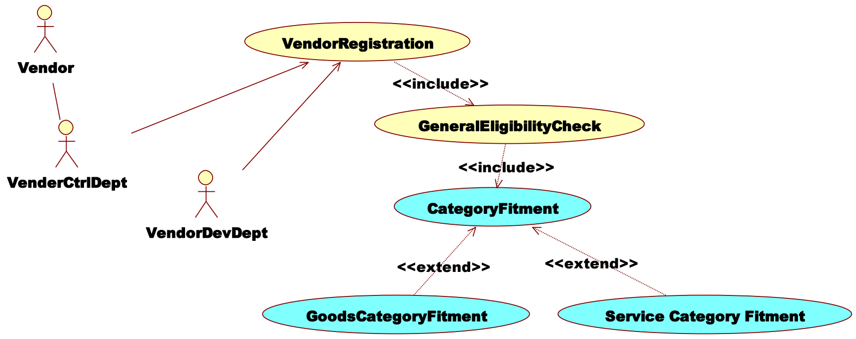
Specification Content : ABB specifications include the following as a minimum : **Interoperability and relationship with other Building Blocks**

How to work with

Nice to Know Box

Specification Content of ABB : Business Architecture

Interoperability and relationship with other Building Blocks



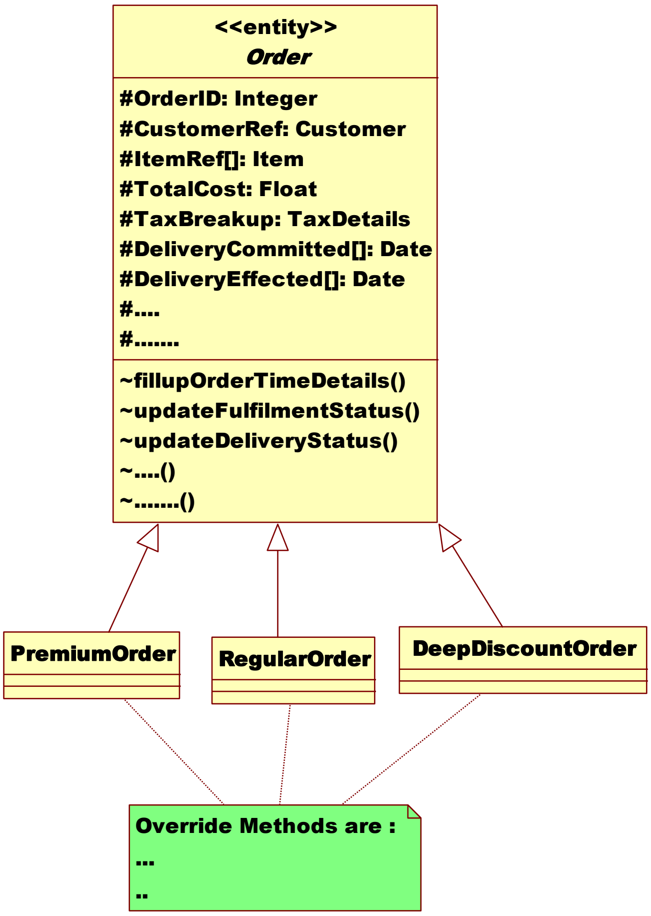
Look at <<include>> and <<extend>> to understand this

How to work with

Nice to Know Box

Specification Content of ABB : Application Architecture

Interoperability and relationship with other Building Blocks



<<entity>>
Order

#OrderID: Integer
#CustomerRef: Customer
#ItemRef[]: Item
#TotalCost: Float
#TaxBreakup: TaxDetails
#DeliveryCommitted[]: Date
#DeliveryEffected[]: Date
#....
#.......

~fillupOrderTimeDetails()
~updateFulfilmentStatus()
~updateDeliveryStatus()
~....()
~.......()

PremiumOrder

RegularOrder

DeepDiscountOrder

Override Methods are :
...
..

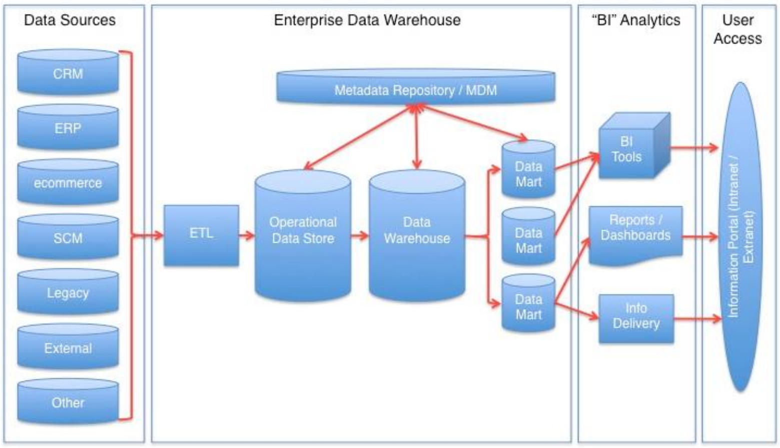
Inheritance, leading to Dynamic Polymorphism

How to work with

Nice to Know Box

Specification Content of ABB : Data Architecture

Interoperability and relationship with other Building Blocks

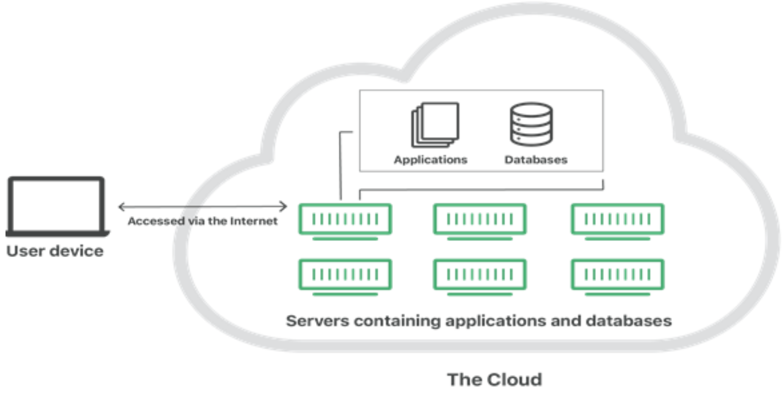


How to work with

Nice to Know Box

Specification Content of ABB : Technology Architecture

Interoperability and relationship with other Building Blocks


Applications
Databases
User device
Accessed via the Internet
Servers containing applications and databases
The Cloud

How to work with

Nice to Know Box

Architecture Building Blocks

Characteristics of ABBs :

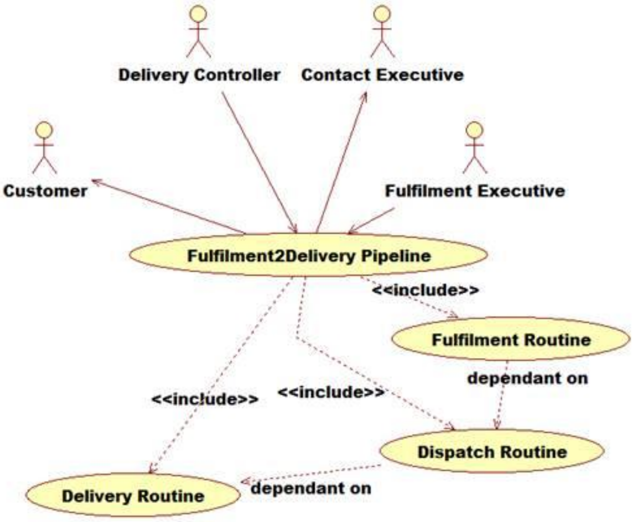
Specification Content : ABB specifications include the following as a minimum : **Map to business / organizational entities and policies**

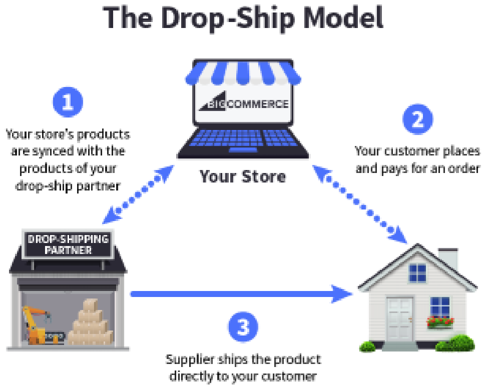
Business IT Alignment

Nice to Know Box

Specification Content of ABB : Business Architecture

Map to business / organizational entities and policies




The Drop-Ship Model
BIGCOMMERCE
Your Store
1
Your store's products are synced with the products of your drop-ship partner
2
Your customer places and pays for an order
DROP-SHIPPING PARTNER
3
Supplier ships the product directly to your customer

Business IT Alignment

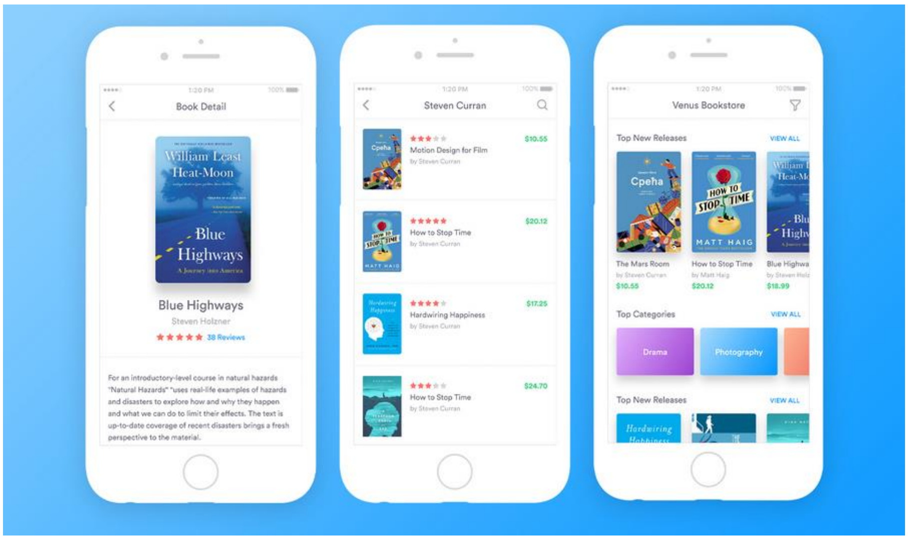
Nice to Know Box

Specification Content of ABB : Application Architecture

Map to business / organizational entities and policies

Vendor **Customer** ***Order***

FulfilmentCurrent **ShippingInsurance** **Item**



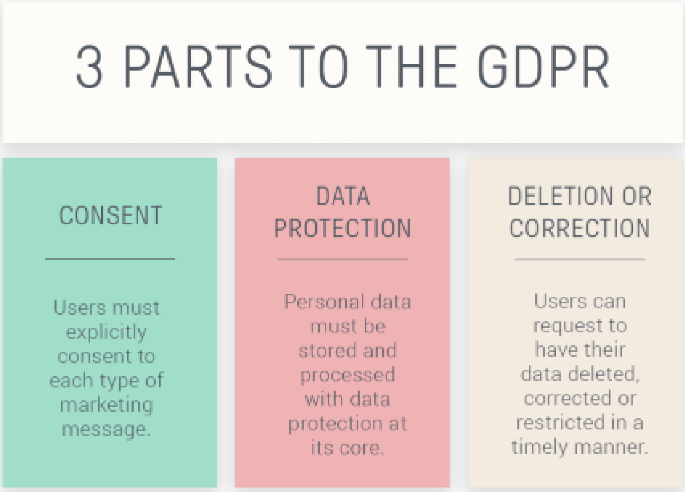
UI / UX Expectations

Business IT Alignment

Nice to Know Box

Specification Content of ABB : Data Architecture

Map to business / organizational entities and policies



3 PARTS TO THE GDPR

CONSENT

Users must explicitly consent to each type of marketing message.

DATA PROTECTION

Personal data must be stored and processed with data protection at its core.

DELETION OR CORRECTION

Users can request to have their data deleted, corrected or restricted in a timely manner.

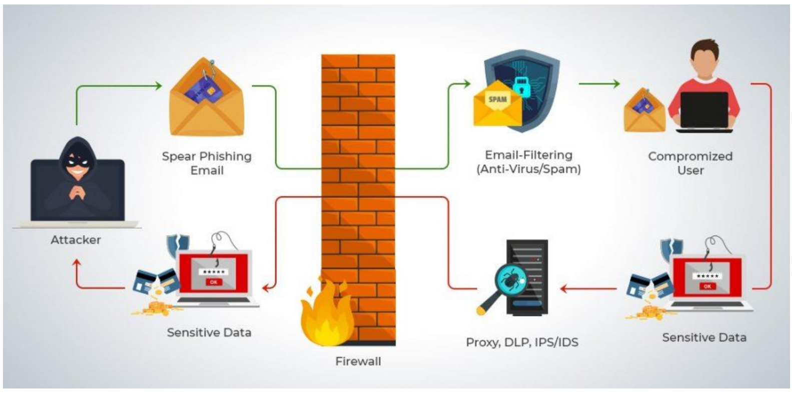
--*

Business IT Alignment

Nice to Know Box

Specification Content of ABB : Technology Architecture

Map to business/organizational entities and policies



Business IT Alignment

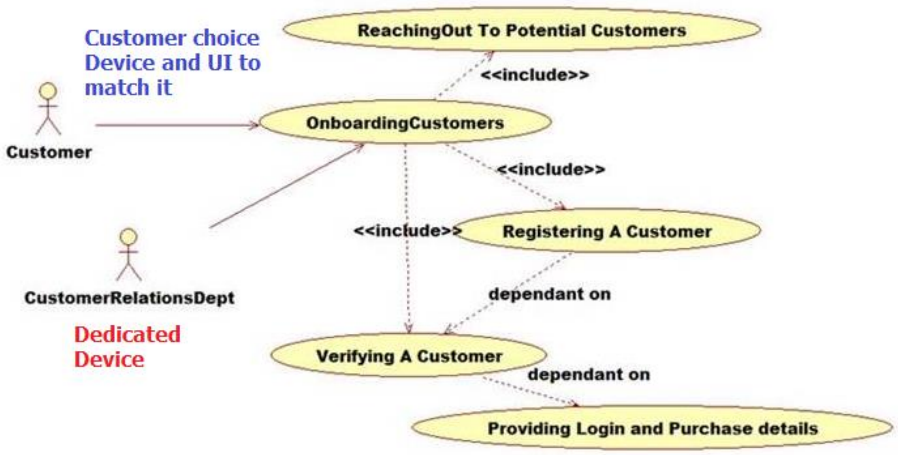
Nice to Know Box

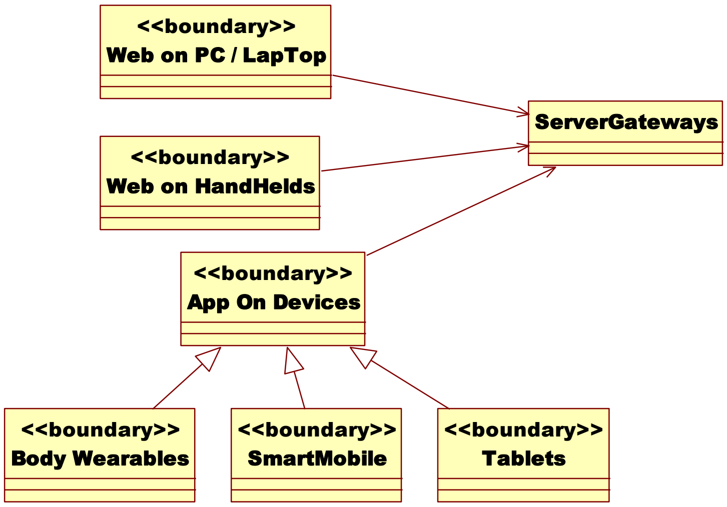
Architecture Building Blocks

Characteristics of ABBs :

Specification Content : ABB specifications include the following as a minimum : **Dependent Building Blocks with required functionality and named user interfaces**

Nice to Know Box

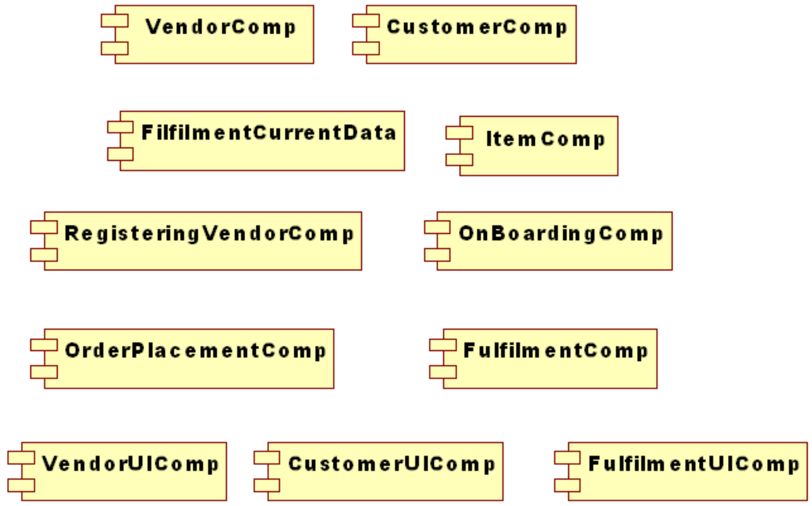


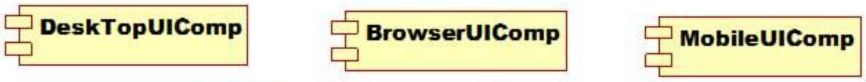


All round dependency

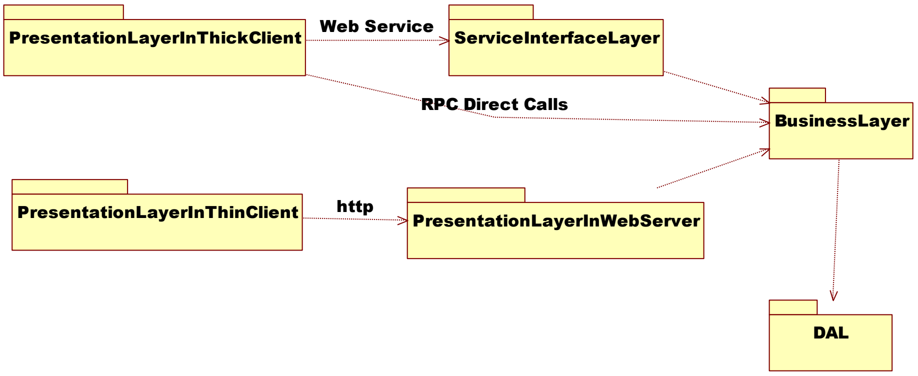
Nice to Know Box

Some AA Building Blocks as ABB



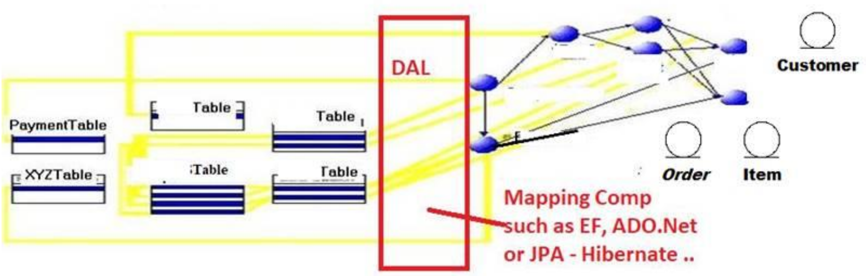


Apply User Interaction Patterns
Apply ....


PresentationLayerInThickClient
Web Service
ServiceInterfaceLayer
RPC Direct Calls
BusinessLayer
PresentationLayerInThinClient
http
PresentationLayerInWebServer
DAL

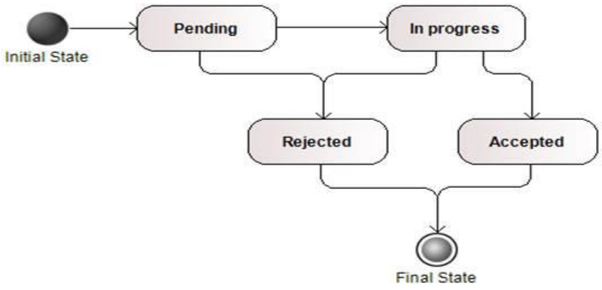
Nice to Know Box

Some DA Building Blocks as ABB



Mapping Comp such as EF, ADO.Net or JPA - Hibernate ..

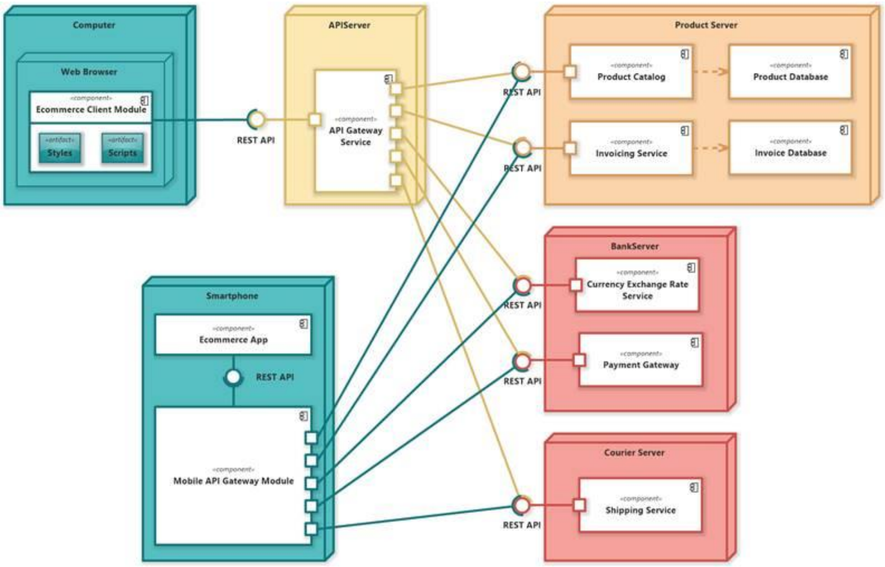
ORM



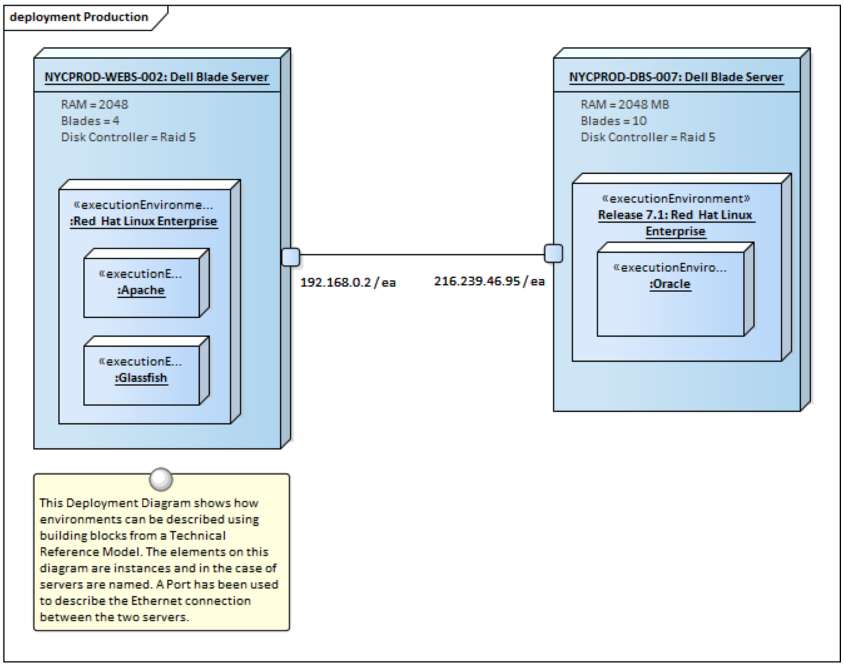
Data Lifecycle Diagram

Nice to Know Box

Some TA Building Blocks as ABB



When does it become a SBB ?



This Deployment Diagram shows how environments can be described using building blocks from a Technical Reference Model. The elements on this diagram are instances and in the case of servers are named. A Port has been used to describe the Ethernet connection between the two servers.

Nice to Know Box

Logical and Physical Building Blocks

A physical component is a material thing that is hired, bought or built to implement / realize specified interfaces (described as "**SBB**" in TOGAF).

A logical component is a material-independent component description – often specified in terms of a service portfolio provided (an "**ABB**" in TOGAF).

The Open Group encourages the specification of business systems in an open, vendor-neutral and portable way.

The ADM is an architecture project management method designed to encourage and facilitate EA-level governance.

To this end, it encourages the open, vendor-neutral and portable specifications for SBBs, which are nothing but ABBs.

It specifies them as vendor-neutral ABBs, each defined by a required service portfolio (and / or a logical data model).

It encourages the EA team to maintain this "open" specification in Architecture Definition documents and an Architecture Repository.

And it applies this service-oriented principle to Business architecture as well as D A T Segment Architectures.

An aim is to increase agility by reducing impact analysis and EA-level architecture change management time.

And to help the Enterprise whenever it wants to, or needs to, migrate from baseline solutions to new solutions.

In procurement, the EA team can evaluate new solutions or products against the current or required ABBs.

If the ABBs are little or not at all affected by a change, then the EA team can readily approve the change.

Good ABB facilitates time-to-market and completeness.

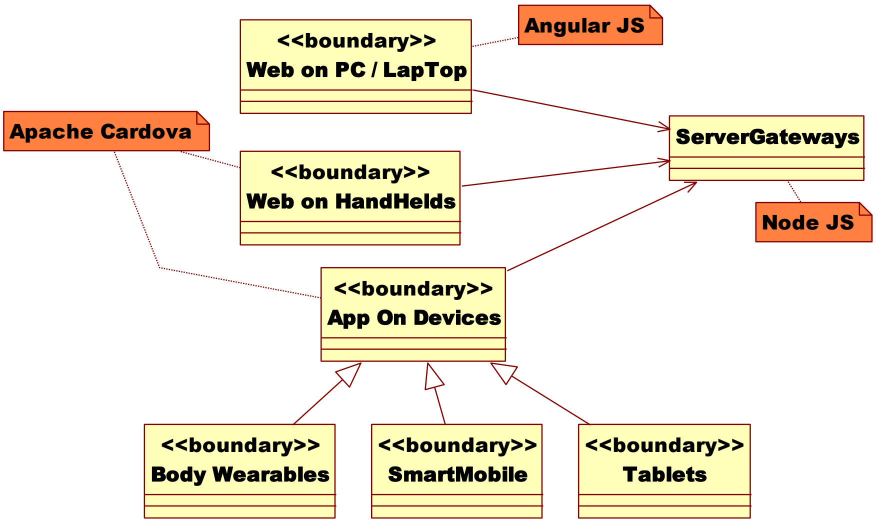
So, TOGAF divides Building Blocks between logical specifications and identifiable hireable / buyable / buildable SBBs.

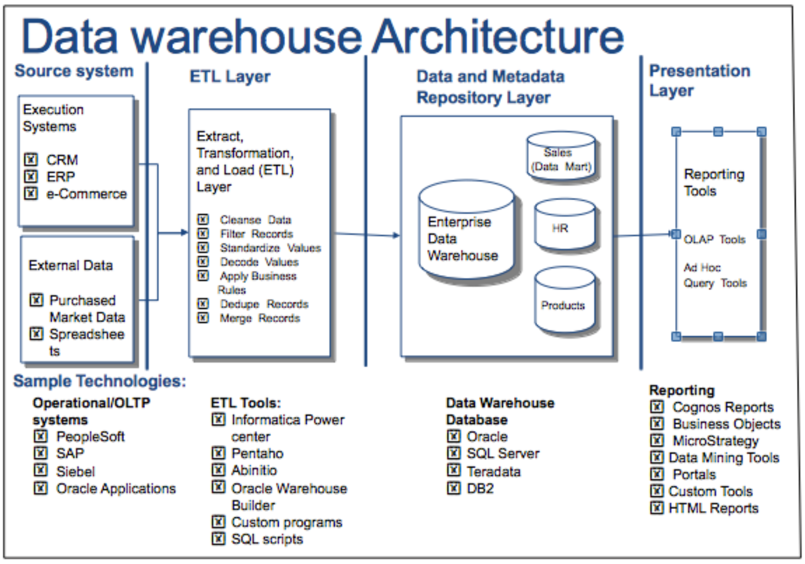
Eventually, the **SBBs are the implemented as physical instances** in the run-time or operational system.

Those **physical instances could be called "operational Building Blocks**".

Nice to Know Box

SBB Samples





Data warehouse Architecture

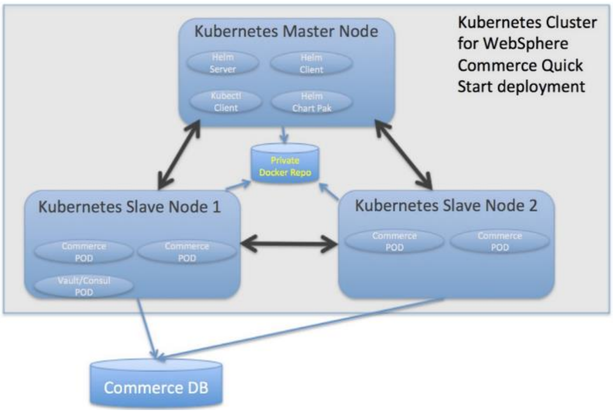
Sample Technologies:

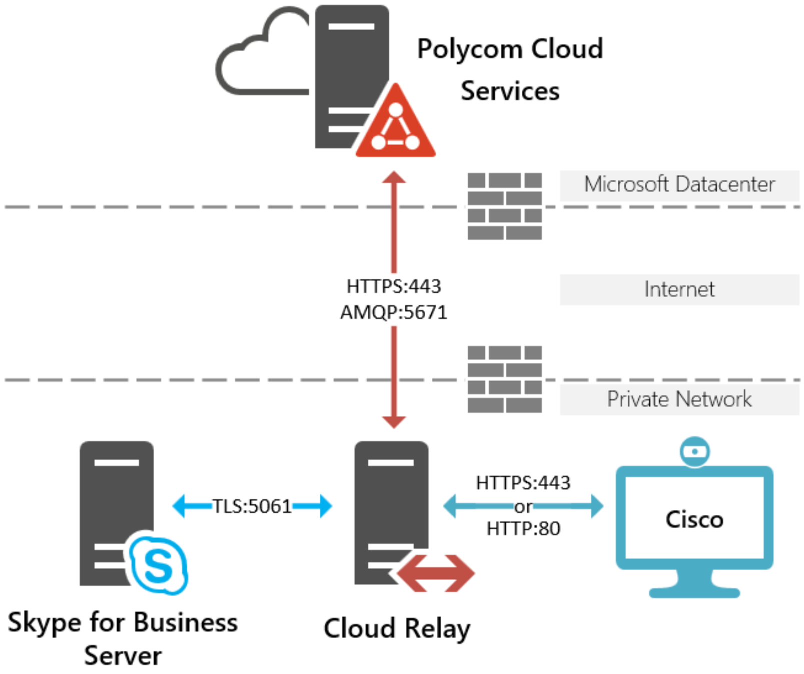
Operational/OLTP systems
- ☒ PeopleSoft
- ☒ SAP
- ☒ Siebel
- ☒ Oracle Applications

ETL Tools:
- ☒ Informatica Power center
- ☒ Pentaho
- ☒ Abinitio
- ☒ Oracle Warehouse Builder
- ☒ Custom programs
- ☒ SQL scripts

Data Warehouse Database
- ☒ Oracle
- ☒ SQL Server
- ☒ Teradata
- ☒ DB2

Reporting
- ☒ Cognos Reports
- ☒ Business Objects
- ☒ MicroStrategy
- ☒ Data Mining Tools
- ☒ Portals
- ☒ Custom Tools
- ☒ HTML Reports





Nice to Know Box

The deliverables are represented directly by Platform Independent Models, concept white papers etc., This is the **case for the "ABB" deliverable**, whose aim is to formalize an architecture model.

ABBs

concept, abstract, explanatory
Patterns, POCs, Feasibility Studies

White Papers, Plan Documents

Platform Independant Components,
API Components, Library Components
Purchase guidelines, General Specs

Not ready for Coding or Procurement or Installation
But can combine them and move towards solution
with additional Architectural work

Points to Ponder

Scenario Approach : Where would it fit ?

Can we say that the following are Viewpoints ?

A diagram prepared by an Architect during ADM ?

No. Diagrams ae Building Blocks, which are Views and not Viewpoints.

A View is what you see. A Viewpoint is where you are looking from - the vantage point or perspective that determines what you see

Solution Concept prepared during Phase A : Architecture Vision

Solution Concept is a Diagram. Diagrams are Building Blocks and which are Views and not viewpoints. A view is a representation of a whole system or part of the system and is what the Architect sees and prepares from the perspective of a related set of concerns.

Request for Architecture Work

No. A Request for Architecture work describes the business imperatives behind the architecture work, thus driving the requirements and performance metrics for the architecture work

This a major document prepared during Preliminary Phase. But this is still a document supplementing a View and is not a Viewpoint that expresses the concern of the (highest possible level) Stakeholders of the Enterprise.

On the other hand it is sent from the sponsoring organization to the architecture organization to trigger the start of an architecture development cycle

It describes the business imperatives behind the architecture work, thus driving the requirements and performance metrics for the architecture work. But it is not a Viewpoint which is supposed to be means to focus on particular aspects of the Architecture

These aspects are determined by the concerns of a stakeholder with whom communication takes place

A catalog that is mandated by the Enterprise Architect and prepared by a down-the-line Architect

No. A catalog is a list of things and is one among the three kinds of Building Blocks. For example, Application catalogue is a list of applications, arranged in hierarchical structure that reflects the business function hierarchy

A lot more of Nice to Know Boxes follow :

TOGAF recommends that SBB specifications include the following as a minimum :

* Specific functionality and attributes

* Interfaces; the physical implemented set specifications

* Required SBBs used with required functionality and names of the interfaces used

* Mapping from the SBBs to the IT topology and operational policies

* Specifications of attributes shared across the environment (not to be confused with functionality) such as security, manageability, localizability, scalability

* Performance, configurability in specific terms

* Design drivers and constraints, including the physical architecture

* Clear relationships between SBBs and ABBs

Structural element types		Structural element instances
Logical ABBs	Physical SBBs	Physical instances in deployed systems (which occupy space and respond when asked to perform behaviours)
A UML *class* is a logical specification of the behaviours (operations, methods) one or more physical objects can perform.	E.g. A class coded in Java or C#	An *object* instantiated or deployed in the software system of interest, along with physical resources it needs, so it can respond to requests.
A **component** is a logical specification of the behaviours(services, uses cases) one or more physical components can perform.	E.g. A named CRM package or DBMS	A component *instance* deployed in the software system of interest, along with physical resources it needs, so it can respond to requests.
A **role** is a logical specification of the behaviours(services, duties) one or more physical human actors can perform.	Named **actors**	A human **actor** employed or deployed in the business system of interest, along with the rewards and physical resources it needs, so it can respond to requests
A **business function** is a logical specification of the behaviours(services, processes) one or more physical organisation units can perform.	Named **organisation units**	An **organisation unit** instantiated in the business system of interest, along with the budget and physical resources it needs, so it can respond to requests.

A Building Block is an encapsulated unit of **functionality or behavior**.

(Or in the case of a data component, it is an encapsulated persistent data structure.)

ABBs = logical component types = vendor and technology neutral service portfolio specifications.

SBBs = physical component types = vendor or technology-specific specifications of what is hired, bought or built to realize the ABBs.

Note that SBB documentation does not necessarily specify the Building Blocks in more detail than ABBs.

TOGAF says specification at the SBB or physical component level is still "considerably abstracted" from implementation.

Ideally, one ABB maps to one SBB.

Ideally, the ABB is a well-defined service portfolio, and a SBB can be found to deliver it.

Or the ABB is a service portfolio reverse-engineered from a known SBB.

Or the ABB is a service portfolio already defined in an “open” interface standard.

In these cases, the SBB may be little more than a vendor/technology name and version number.

If the ABB is the international FTP interface specification, then the SBB might be simply "the FTP server embedded in our Unix OS".

So, that particular FTP server component should be replaceable with minimal EA-level impact analysis and architecture change management.

The same principles apply in the human domain.

Given ABBs in the form of a logical function / capability structure, then the SBBs might be the units of a corresponding "functional organization structure".

The EA team does not worry if the management reorganizes the organization structure, providing it can be mapped to the same logical function/capability structure.

Architecture Building Block is a requirement of what system should do and Solution Building Block is actual implementable design of the solutions.

Figure them out under following Scenarios :

1. Registration – Records identifiers and other general information about a person, place or other entity, typically for registration or enrolment in specific services or programmes and tracking of that entity over time

2. E-Payments – Implement and log financial

transactions receipts and payments online in multiple ways.

3. Workflow management – Help to optimize business processes by specifying the rules that govern the execution of a sequence of activities and the exchange of associated information to orchestrate the process flow from initiation to completion.

4. Case management – Register or enroll users and provide longitudinal tracking of services, often across multiple service categories, departments and locations.

5. Feedback – Provide the ability for consumers and providers of services to send, track and address any issues pertaining to service quality, including any kind grievance redressal.

6. Consent management – Manage a set of policies allowing users to determine the information that will be accessible to specific information consumers, for which purpose, for how long, and whether it can be shared further.

7. Reporting and dashboard – Provide pre-packaged and custom presentations of data and summaries of an organization’s pre-defined key performance metrics, often in a visual format.

Another lot of exercise for you to figure out ABBs and SBBs :

1. eLearning – Support facilitated or remote learning through digital interaction between educators and students.

   2. Collaboration management – Enable multiple users to simultaneously access, modify or contribute to a single activity, such as content creation, through a unified access portal.

   3. Content management – Support the creation, editing, publishing and management of digital media and other information.

   4. Scheduling – Provide an engine for setting up events based on regular intervals or for triggering specific tasks in an automated business process, based on specific combinations of status of several parameters.

   5. Terminology – Provide a registry of definitions and terms with defined nomenclature standards, metadata, synonyms and sometimes a knowledge map for a particular domain of knowledge (e.g. health / travel / hospitality / …) which can be used to facilitate semantic interoperability.

Groups of Building Blocks

Building Blocks can be seen further under four groups :

1. Infrastructure BBs : E.g. - Cloud, SoC, Enterprise App Store, Unified Call Centre
2. Data BBs : E.g. - Digital Identity, Digital Registries, Enterprise Directories
3. Technology BBs : E.g. - Gateways (Integration, Messaging), Mobility Management (MDM), Collaboration Management, Consent Management
4. Application BBs: Analytics, AI, e-Payments, Workflow, Case Management, Dashboard.

It is useful to classify the Building Blocks along these groups to facilitate designing the Logical Architecture of the Enterprise system.

According TOGAF, a Building Block is a package of functionality defined to meet business needs across an organization. It has a type to correspond a TOGAF metamodel (such as actor, business service, data entity, software or infrastructure element). For domain experts, it is generally recognizable as "a thing" e.g. a company, a technology, a server.

Building Blocks has well defined boundaries, interfaces, specification and should be loosely coupled to leverage the reusability to its implementation. Systems are built up from collections of Building Blocks. Having said that, a BB could be different "things". In that case it is used in a software development to demonstrate different perspectives for different stakeholders viewpoints.

On software development the BBs can be used to present to business team, for example, the connections among roles, business services and processes. Through these "business" elements, it can be decomposed in more technical elements as systems, applications, servers, etc. Some examples of Building Blocks are business services as “Purchase Order”, system as “SAP” and infrastructure element as “Oracle Database”.

In layman terms, Architecture is an abstract view of solution. It can be a superset or group of solutions. A solution is a more concrete representation of a problem while architecture represents wider picture of a system.

Architecture Building Blocks : This supposed to be logical, supposed to be a conceptual idea for what we want to achieve within organization, this will guide and support our solution space, yes as you read, this ABB's will set the rules and give some guidelines to how we are going to create our solutions at the corresponding level of Solutions Continuum (foundation, common-system, industry, organization). This architecture Building Blocks cannot be deployed and not even are physicals.

These are more of idea and concept. Less of specifics, Very little of coding platform specifics. Not much on deployment specifics, More of interfacing logical approach.

ABB’s purpose is served when :

Architect captures all needs of Stakeholder properly, analyses and organizes then into conceptual solutions

Validated with co-Architects

Conceptual approach Reviewed with the stakeholders

An Architect would design an Architecture and decompose it into two or more components (= abstractions). One of the components may represent the need to store, for example, customer data. So the architect would write 'a database required to hold customer data'. This is part of architectural design - the abstraction.

At some time in future, the Architect may want to make it concrete. At that time he may write "an Oracle RDBMS or a flat file or an In-memory HSql database) as its concrete implementation.

Solution is an implementable / implemented component whereas Architecture is the description component. Both can be represented at different levels

The Architecture Requirement Specification includes contracts for services offered by business and IS / Application Building Blocks.

Architecture Building Blocks (ABBs) are logical components defined by the services or flows they provide to each other and to external entities.

Solution Building Blocks (SBBs) are physical components hired, bought or built to realise ABBs, and so deliver the required services or flows.

SBBs are defined by the ABBs they realize, along with vendor-specific and implementation-specific details.

The implicit generic meta model is : required services are assigned to logical ABBs are realized by physical SBBs.

The generic meta model is specialized in architecture domain / layers thus.

Abstract / logical Building Blocks (functions, roles, and logical components) cluster behavior types that concrete Building Blocks can be nominated to perform.

Concrete / physical Building Blocks (organization units, actors and physical components) are nominated to realise or perform instances of behavior types.

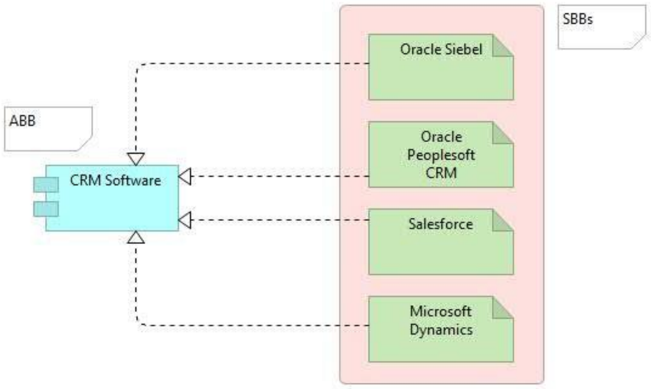
BBs can be classified into Architecture Building Blocks and Solution Building Blocks which are respectively technology-aware and product or vendor-aware. They are defined or selected as a result of the application on Enterprise Architecture frameworks such ADM for TOGAF.

These two types of BBs are used to represents components of business, IT or Architectural Capabilities. For software development it can be used to classify different technologies and decompose that in a group of elements or products.

The main relation between ABBs and SBBs is that the first guides the development of the second. In other words, the ABB is an specification of SBBs and it comply with this specification. Essentially the ABB has captured the architecture requirements (business, data, application and technology). The SBB defines the products and components that can be implement the functionality.

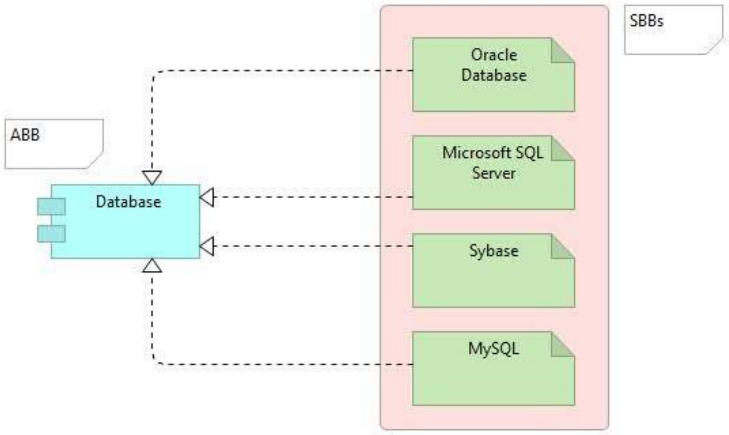
In this example we see as presented the specification for a “CRM Software” (ABB).

Assuming that we have a well-defined interface for this, is possible to determine different market software that implement this specification (SBB) as Oracle Siebel, Peoplesoft, Salesforce and Dynamics.

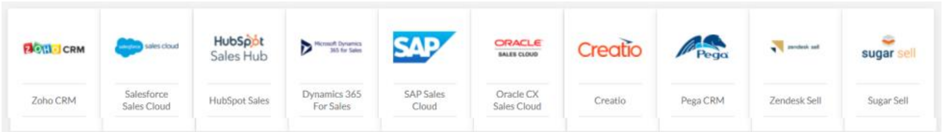


The second example uses Building Blocks although more focused on infrastructure elements.

In this example what is presented is the specification for a “Database” (ABB). In the same way, if we have a well-defined ABB, is possible to identify different market databases that implement this specification (SBB) as Oracle Database, SQL Server, Sybase and MySQL.



The aim is to demonstrate that Building Blocks, apart from being useful for business view, it can be used to assembly different types of architectures for the software development perspective.



According to the Open Group, “The way in which functionality, products, and custom developments are assembled into Building Blocks varies widely between individual architectures. Every organization must decide for itself what arrangement for Building Blocks works best for it.”

Remembering that, an important requirement for Building Blocks is the reusability, so the ABBs and SBBs can be used to setting up different architectures in the future.

TOGAF defines systems elements at several levels of abstraction removed from an implemented system.

"Building Blocks can be defined at various levels of detail, depending on what stage of architecture development has been reached."

Architects describe Building Blocks at several levels of abstraction, depending on what stage of architecture development has been reached.

Phase A : Architecture Vision

Architects gather contextual information (business drivers, principles, goals, visions and plans; system stakeholders and their concerns).

They present, refine and agree this information, getting approval to proceed.

They may name business system components (say Billing, or CRM), with a sketchy description.

"At an early stage, a Building Block can simply consist of a name or an outline description."

Phases B to D : Architecture Development

Architects proceed to define *logical* application and technology components, and the services they provide.

"A Building Block's boundary and specification should be **loosely coupled to its implementation**"

The description of logical ABBs tends to be generalised, logical and coarse-grained.

Ideally, architects specify ABBs as interface definitions, decoupled from specific solutions. E.g.

A Billing function is defined by the 20 business services it provides.

A CRM application is defined by the 15 critical use cases it offers via its interface, and 50 business data entities it maintains.

An RDBMS is defined by conformance to standards such as SQL, ODBC.

General concepts	**Required Behaviours**	**Logical Structures**	**Candidate Physical Structures**
Generic concepts		**ABB**	**SBB**
TOGAF elements	**Service portfolios**	**Functions/Capabilities, Roles**	**Organisation Units, Actors**
Business example	Customer delivery service portfolio	Registered Customer	Specific way technology is leveraged
Business example	Marketing service portfolio Research customer requirements Design campaign Promote product or service Analyse customer responses	Marketing	In-house marketing department Out-sourced marketing department
TOGAF elements	**Service portfolios**	**Logical Components**	**Physical Components**
Application example	Customer relations service portfolio Gather sales leads from social media Promote product or service Capture order details Close deal Maintain a customer profile Forecast sales Send message to sales or service agent	CRM Application	amoCRM Bitrix24 HubSpot Infusionsoft Oracle Siebel Pipedrive Salesforce.com Teamgate CRM

	Display dashboard-style analytics		
Infrastructure Technology example	The footnote contains technology-specific examples of service portfolios, logical and physical components		

Solutions Building Blocks : We can deploy solutions as they are physical, it means solutions can be instantiated within a deployment, this solutions represent the implementation of architectures at the corresponding level of the Architecture Continuum (foundation, common-system, industry, organization). We can build or buy solutions, build in our organization or buy from some vendors.

Fully ready to be translated (even code generated where possible) into code

Will contain even the solution specifics of how it can be deployed after creating the executables

Implicit in TOGAF are the equations : function = logical business component, and organisation unit = physical business component.

The abstract / logical Building Blocks cluster behavior types that concrete Building Blocks can be nominated to perform.

The concrete / physical Building Blocks are nominated to realise or perform instances of those behavior types.

The generic architectural entities discussed above have different generic qualities.

E.g. behaviours run over time, whereas physical components occupy space and do work.

Common attributes of Behaviours e.g. Services, Processes, Value Streams, Scenarios

- name
- start conditions: trigger events, inputs, preconditions for success
- end conditions: outputs, results or values delivered, other post conditions
- duration
- volume or throughout
- performers (Building Blocks).

"Systems concepts include: system-environment boundary, input, output, process, state…." (Principia Cybernetica)

The inputs and outputs of a behavior can include materials/goods as well information/data flows.

The preconditions and post conditions of services and processes can refer to system state, as may be recorded in databases.

Common attributes of Logical Building Blocks
e.g. Roles, Business Functions/Capabilities, Logical Application and Technology Components

name

provided interface(s) that group services offered

required interface(s) that group services needed

behaviours performed

physical Building Blocks that implement the logical Building Blocks

Nice to Know Box

A view is a representation of the whole system from the perspective of a related set of concerns.

The architectural description of a system includes a number of different views. In general, several diagrams and descriptions are used for one view.

A view conforms to a viewpoint. A viewpoint is a collection of patterns, templates, and conventions for constructing one type of view. It defines the stakeholders whose concerns are reflected in the viewpoint and the guidelines, principles, and template models for constructing its views. Viewpoints are defined with specific stakeholders' concerns in mind.

This means that a viewpoint is not related to one specific system; a viewpoint should rather be seen as a certain aspect of systems. Both views and viewpoints will be discussed in more detail.

At a minimum, the stakeholders identified include users, owners, developers and maintainers.

At a minimum, the concerns identified should include the purpose or mission of the system, the appropriateness of the system for use in fulfilling its mission, the feasibility of constructing the system, the risks of system development and operation to users, owners and developers, and maintainability, deployability and evolvability of the system.

This explains the need for many different Viewpoints which are the concerns and point of view as expressed a wide range of Stakeholders.

Nice to Know Box

An "Architecture View" (or just "View") is a representation of a system from the perspective of a related set of concerns.

An "Architecture Viewpoint" (or just "Viewpoint") is a specification of the conventions for a particular kind of architecture view

TOGAF, and also ISO/IEC/IEEE 42010: 2011, encourages architects to define architecture viewpoints explicitly.

As an architect it is our responsibility to ensure the completeness (fitness-for-purpose) of the architecture, in terms of adequately addressing all the pertinent concerns of its stakeholders. Hence it is very important for us to select the appropriate viewpoints to build necessary views of the architecture we are Building. **This is the very first step in each of the Architecture Development Phases (Phases B, C and D) of ADM cycle**.

TOGAF recommends following steps for creating an architecture view :

1. Refer to an existing library of architecture viewpoints

2. Select the appropriate architecture viewpoints (based on the stakeholders and concerns that need to be covered by views)

3. Generate views of the system by using the selected architecture viewpoints as templates.

Always remember, Architecture viewpoints are generic, and an architecture view is always specific to the architecture for which it is created. Every architecture view has an associated architecture viewpoint that describes it, at least implicitly.

For each viewpoint, an architectural description should contain one or more architectural views.

Each view corresponds to one viewpoint. A view is a representation of the system, based on a viewpoint. A view may consist of one or more models. Each view will include introductory information.

In addition, formal or informal consistency and completeness tests may be given, to be applied on the models making up an associated view, evaluation or analysis techniques heuristics, patterns, guidelines and tips. Also, a rationale should be given, that addresses the extent to which the stakeholders and concerns are covered.

Nice to Know Box

EA artifacts provide descriptions of an organization from different perspectives important for the various actors involved in strategic decision-making and implementation of IT systems. They can be considered as key elements and cornerstones of an EA practice. Essentially, an EA practice revolves around using specific sets of EA artifacts for improving communication between different Stakeholders.

- EA artifacts can have different representation formats. Specifically, EA artifacts can be represented in textual, graphical and sometimes tabular formats, or as a mix of these formats. Purely textual EA artifacts contain only plain text. Purely graphical EA artifacts contain only diagrams and models sometimes created using special modelling languages or notations. Purely tabular EA artifacts contain only tables with rows and columns. Mixed EA artifacts can contain the elements of all these representation formats in different proportions

- EA artifacts can provide different levels of detail. Descriptions contained in EA artifacts can range in their granularity from very high-level abstractions (e.g. business or IT capabilities, overarching rules, executive-level considerations, etc.) to pretty low-level details (e.g. specific business activities, IT systems, databases, etc.)

- EA artifacts can cover different organizational scopes. From the perspective of their scopes, coverage of EA artifacts ranges from entire organizations, lines of business and business functions to narrow organizational areas, specific change initiatives and even single IT projects.

- Typically EA artifacts covering wider scopes are less detailed, while EA artifacts covering narrower scopes are more detailed

- EA artifacts can describe different EA domains. EA domains often described by EA artifacts include business, applications, data, infrastructure and security domains, as well as all possible combinations of multiple different EA domains

- EA artifacts can focus on different temporal states of an organization, i.e. describe an organization at different points in time.

-

- All states typically described in EA artifacts can be roughly separated into the current state (now), short-term future state (<1 year), and long-term future state (3-5 years).

- Additionally, some EA artifacts can describe a combination of all these states in different proportions or can even be essentially stateless, i.e. do not focus on specific points in time. For example, some EA artifacts can describe the current state of an organization as well as the planned changes to this state in both the short-term and mid-term future, while other EA artifacts can describe some timeless imperatives for an organization which were relevant in the past, are relevant now and will be relevant in the future

These artifacts are "living documents," which means they evolve to reflect changes in the project.

Nice to Know Box

A Building Block is a package of functionality defined to meet business needs.

Building Blocks have to interoperate with other Building Blocks. A good choice of Building Blocks can lead to improvements in legacy system integration, interoperability, and flexibility in the creation of new systems and applications.

Wherever interoperability is required, the interfaces to a Building Block are published and reasonably stable.

A Building Block is intentionally designed to be cross-functional allowing its generic functionality to be applied to different contexts.

Each Building Block must have the following characteristics :

- Provide a standalone, useful, reusable and implementable capability
- Cross-functional across the value chain by design
- Applicable to multiple use cases
- Interoperable with other Building Blocks
- Use shared digital infrastructure (to the extent feasible)
- Standards-based and
- Designed for scale

Each Building Block must have a clear “Business Owner” and “Technology Owner”. The business owner is responsible for defining the rules and policies essential to effectively manage the Building Block. The technology owner would be responsible for managing the business requirements and technical implementation of these requirements efficiently.

TOGAF divides the Architecture of an Enterprise into Building Blocks - of many shapes and sizes.

Building Block varieties include functions, organization units, roles, human actors and application/technology components.

Building Blocks interoperate by requesting and providing services.

Building Blocks and the services they offer are composed and decomposed.

One large Building Block may offer scores of discrete services.

On the other hand, a coarse-grained Enterprise level service may require scores of interacting Building Blocks.

“A good choice of Building Blocks can lead to improvements in legacy system integration, interoperability, and flexibility in the creation of new systems and applications.” TOGAF.

Nice to Know Box

There are certain important characteristics of Building Blocks which we need to be aware of :

- It is nothing but package of functionality defined to meet the business needs across an organization - so it is important that it corresponds to some function, has some purpose to serve in the overall business process
- It considers implementation and usage, and evolves to exploit technology and standards - however it must be loosely coupled to its implementation, i.e. it must be possible to realize a Building Block in several different ways
- It may be assembled from other Building Blocks or may be a sub assembly of other Building Blocks - i.e. it is potentially re-usable or can be constructed by using other re-usable components and can be replaceable
- Building Blocks has well defined boundaries, interfaces, specification and should be loosely coupled to leverage the reusability to its implementation. Systems are built up from collections of Building Blocks. Having said that, a BB could be different "things". In that case it is used in a software development to demonstrate different perspectives for different stakeholders viewpoints.

Nice to Know Box

Building Blocks and Services

“EA regards the Enterprise as a system, or systems of systems”

“Systems are built up from Building Blocks that interoperate with other Building Blocks.”

“The major architecture work consists of identifying the Architecture Building Blocks required to meet the business goals and objectives.”

“An architecture is a set of Building Blocks depicted in an architectural model.”

“A Building Block’s boundary and specification should be loosely coupled to its implementation."

“For each Building Block, build up a service description portfolio as a set of non-conflicting services.”

"A service is a logical representation of a repeatable business activity that has a specified outcome”

“It is important that the interfaces to a Building Block are published and reasonably stable.”

TOGAF tends to use the terms “interface”, “boundary” and “service portfolio” interchangeably.

It is a principle of The Open Group that clients should invoke services via “open” interfaces.

For each Building Block, build up a service description portfolio as a set of non-conflicting services.

Nice to Know Box

On ABBs

The TOGAF concept of the Architecture Building Block (ABB) is the effective Practitioner's friend.

A good ABB facilitates time-to-market and completeness. As with most TOGAF definitions, knowing that an ABB is "a constituent of the architecture model that describes a single aspect of the overall model" doesn't immediately tell us what they look like in an EA Repository.

An ABB will look like whatever it must be to describe part of the overall architecture – efforts to carefully define the contents and structure of this concept will flounder on the variability and scope of what can be described within an EA Landscape.

A **Building Block is part of a greater whole that accelerates the effective description of the candidate architecture**.

In some cases, it will be a re-usable description of part of the architecture; using it again enables the Practitioner to simply adopt a known successful way to address a problem.

In this case, the ABB is complete in all regards, providing a complete description, and constraints that address repeated requirements.

In other cases, it will not have the constraints and specifications predefined. In this latter case, the components of the description will be complete, but the detail will vary depending upon the requirements.

(From TOGAF 10 documentation, Relevant for TOGAF 9.2 also)