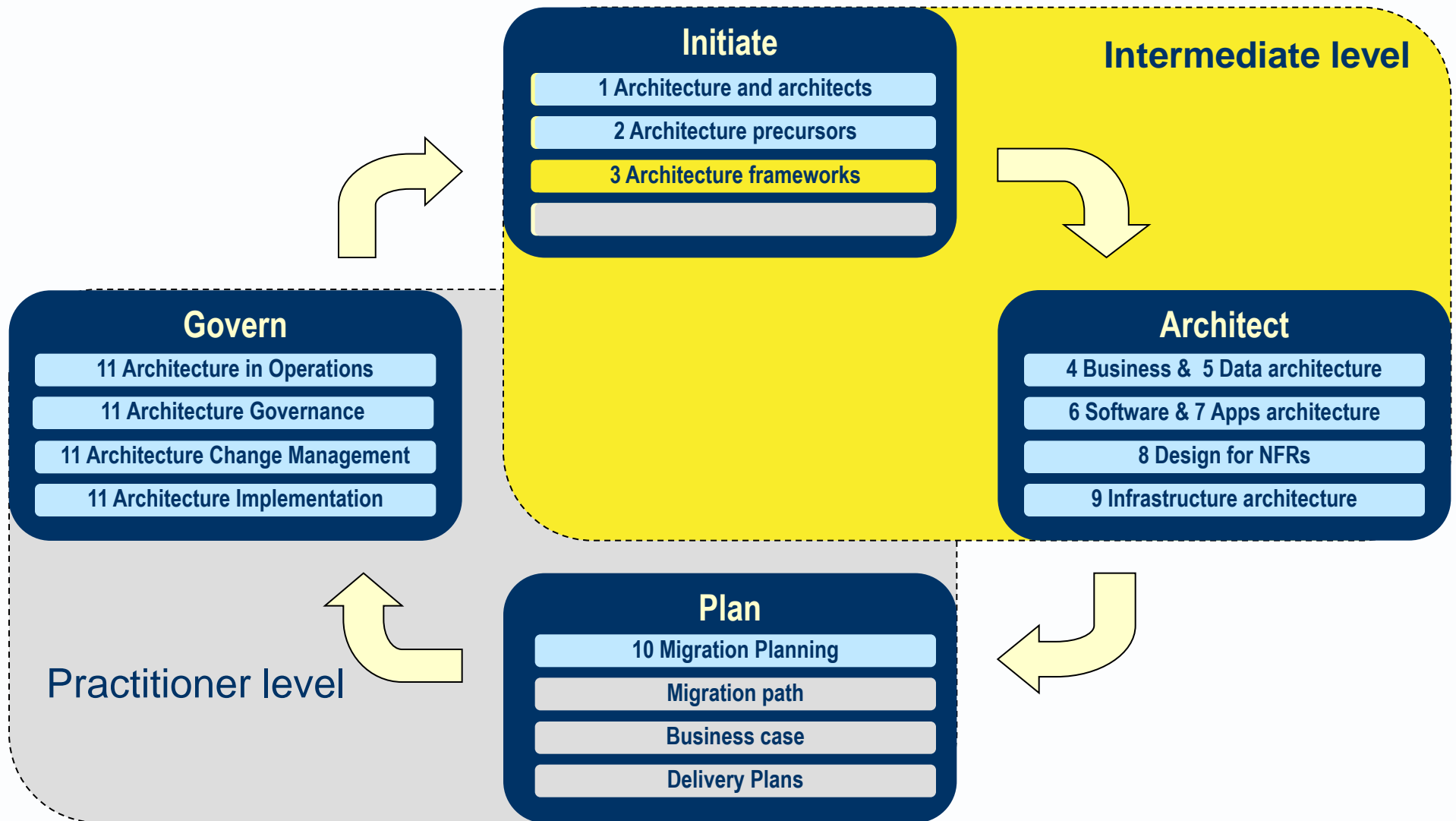


Mapping the reference model to an architecture framework



3.1 Frameworks

- ▶ A comprehensive architecture framework contains advice on
 - processes for architecting,
 - products for architecture definition, and the
 - people involved.

PROCESS

Architecture Development Method (ADM)

- ▶ [The architecture process] in TOGAF® which is centred on a cycle of 8 phases.

Exercise
Copy the diagram

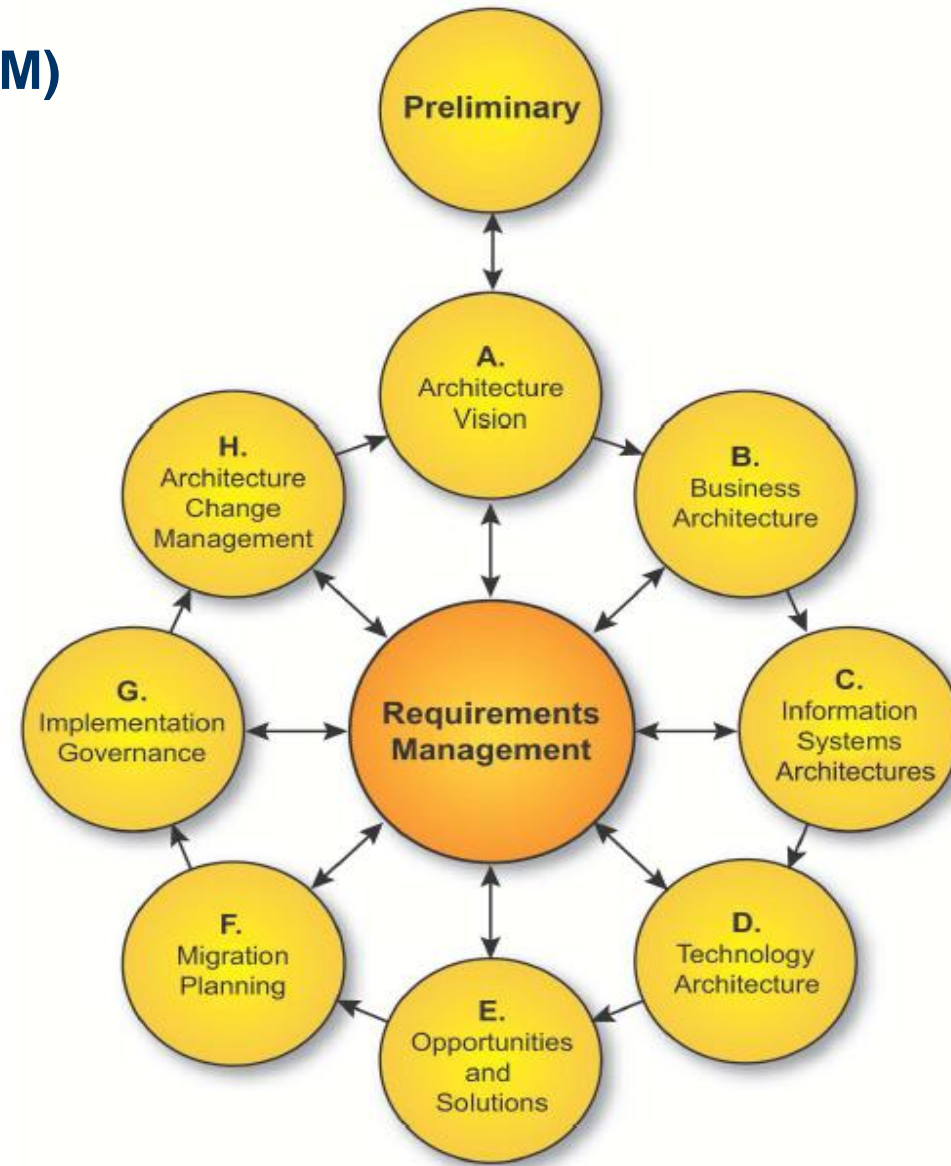
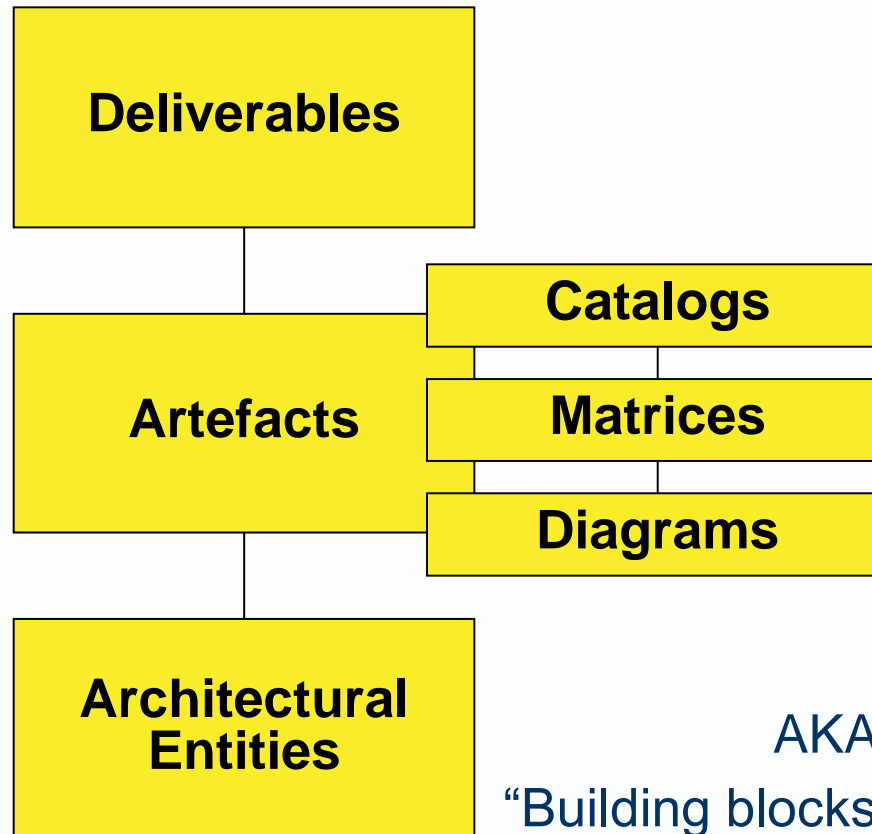


Figure 5-1 Architecture Development Cycle

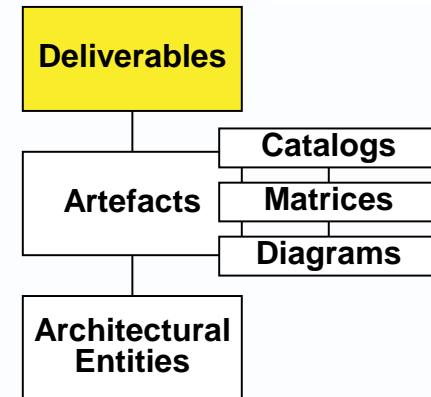
- ▶ [A structure] for organising architecture deliverables, artifacts and entities.



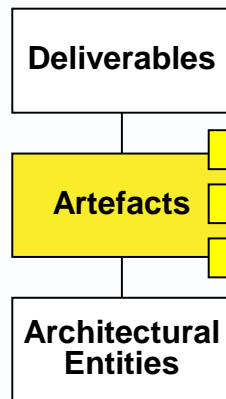
AKA
“Building blocks” in TOGAF

► [An architecture] document, responding to a request for architecture work, requires approval.

- Architecture or solution vision
 - The first response to a request for architecture work.
 - It describes a target just enough to enable options to be compared.
 - It may outline benefits, costs and risks, and work to follow.
- Architecture definition or solution outline
 - [An architecture] that describes the high-level design of a target system or solution.
- Detailed design
 - [An architecture] that describes the low-level design of a target system.
 - Complete enough for building work to start.

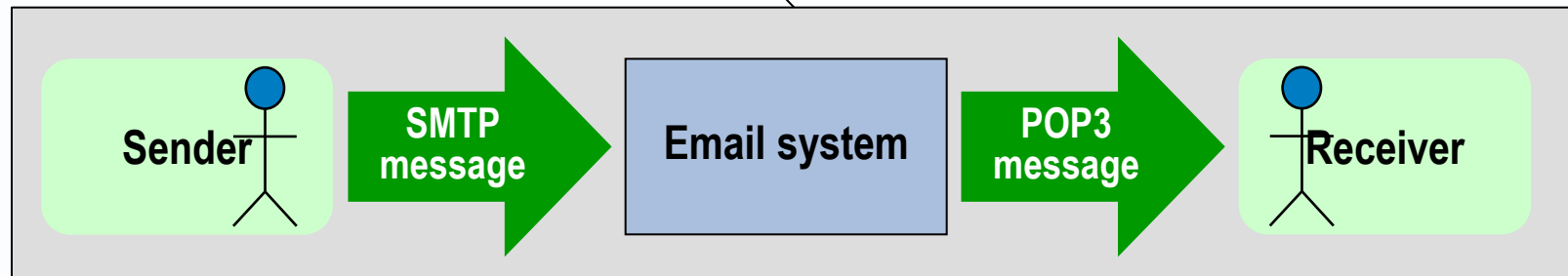


- ▶ A catalogue, matrix or diagram containing architecture entities

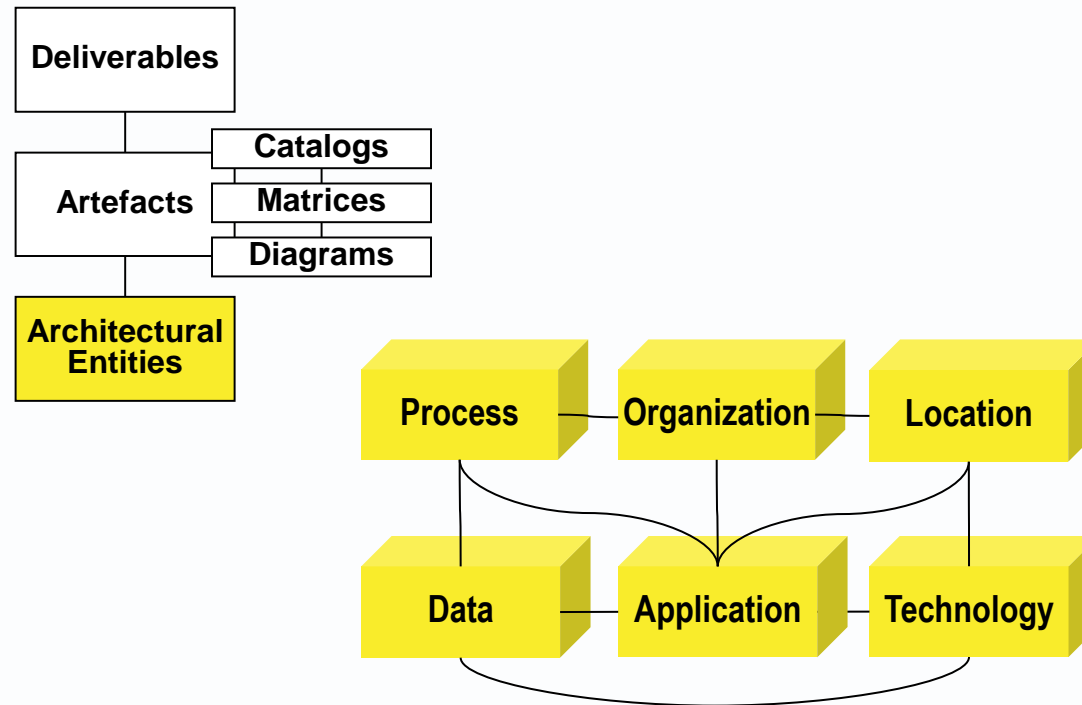


Technology	Qty	Price	
PC			
Printer			
Photocopier			

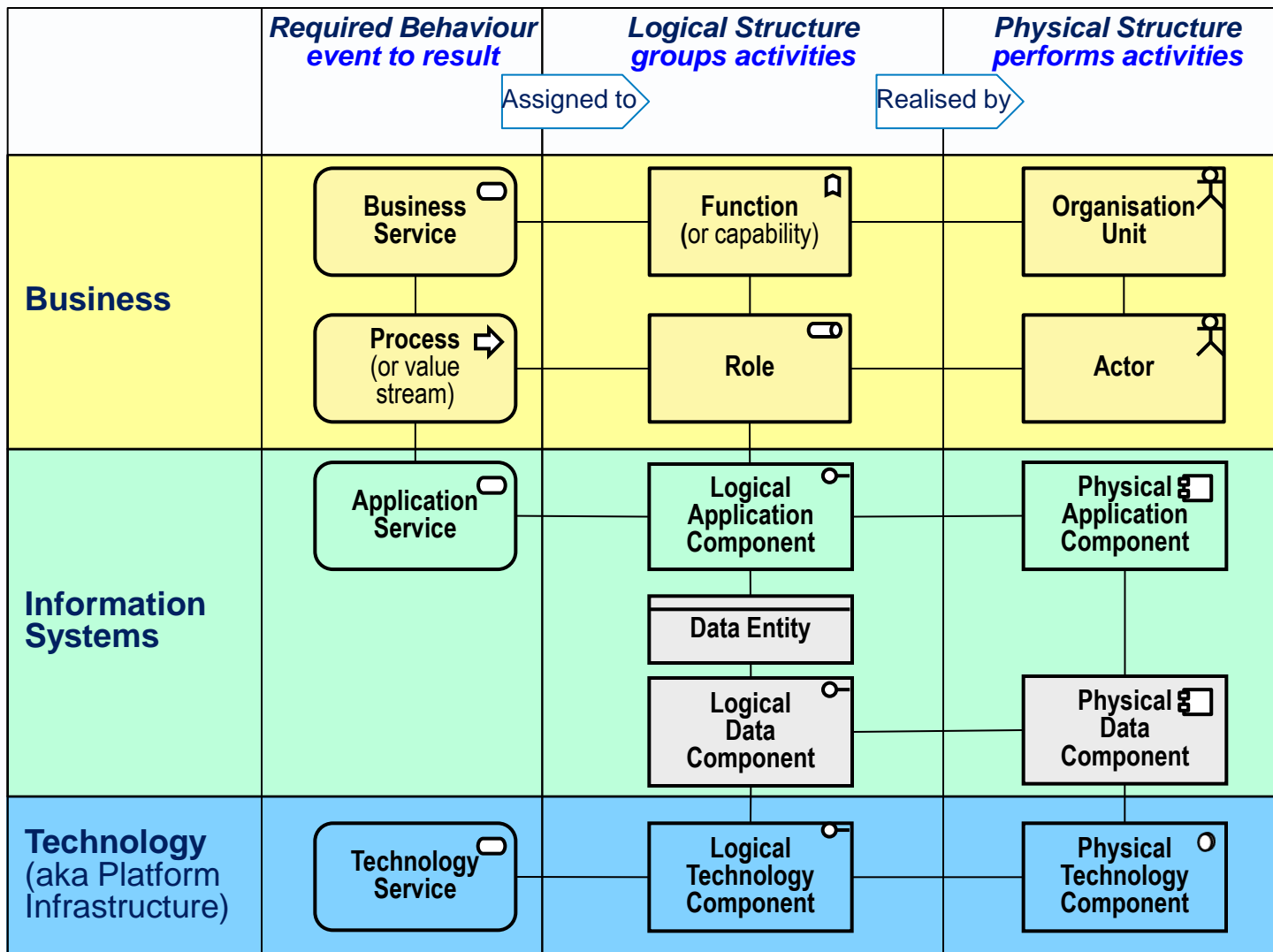
Location Technology	Paris	New York	Hong Kong
PC	10,000	7,000	2,000
Printer	1,000	700	200
Photocopier	100	70	20



- ▶ An element that appears in one or more artifacts, such as process, organization, location, data entity, application and technology (POLDAT).
- ▶ Sometimes called “Building Blocks”

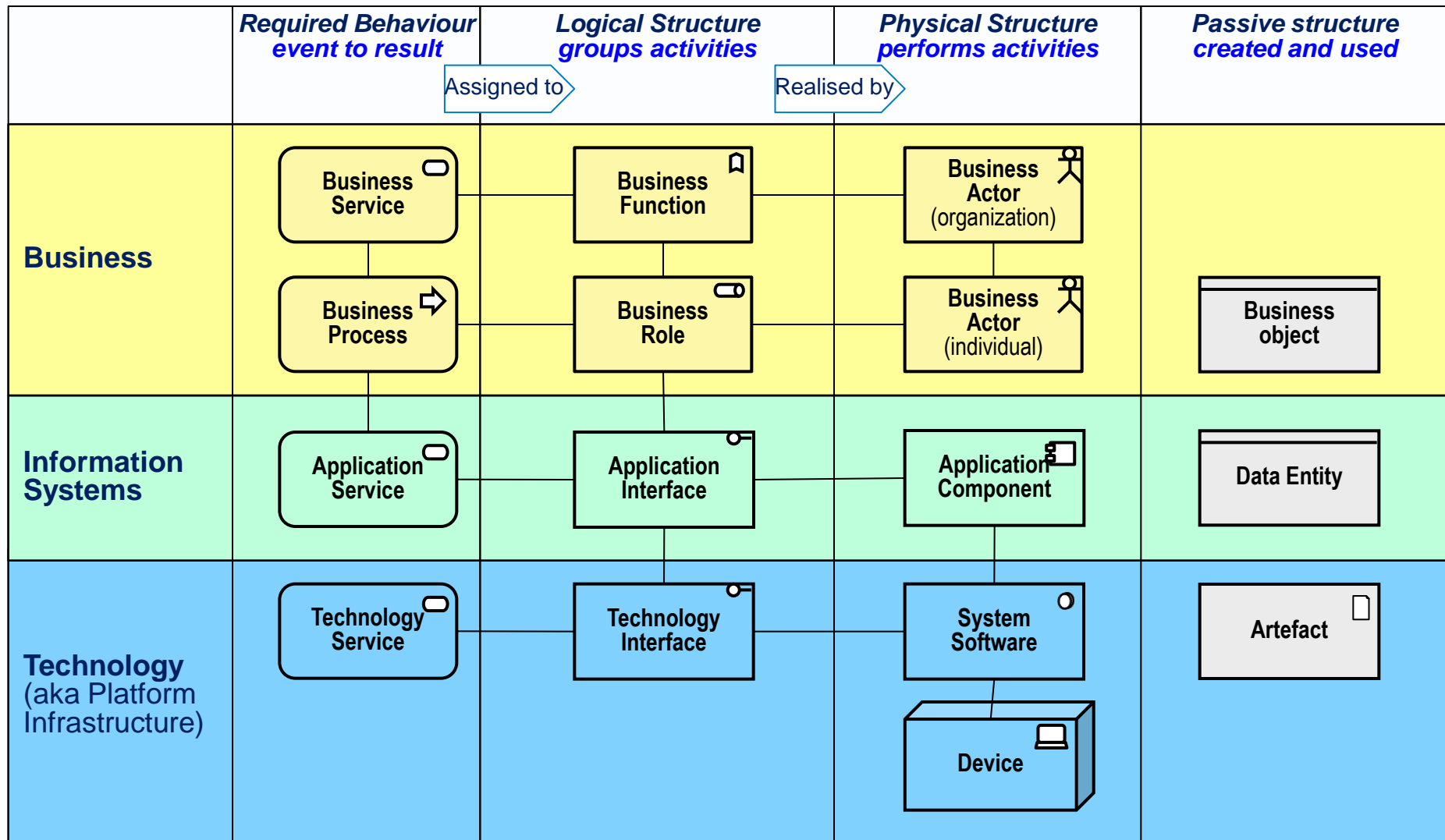


TOGAF: a selection of entities and relationships

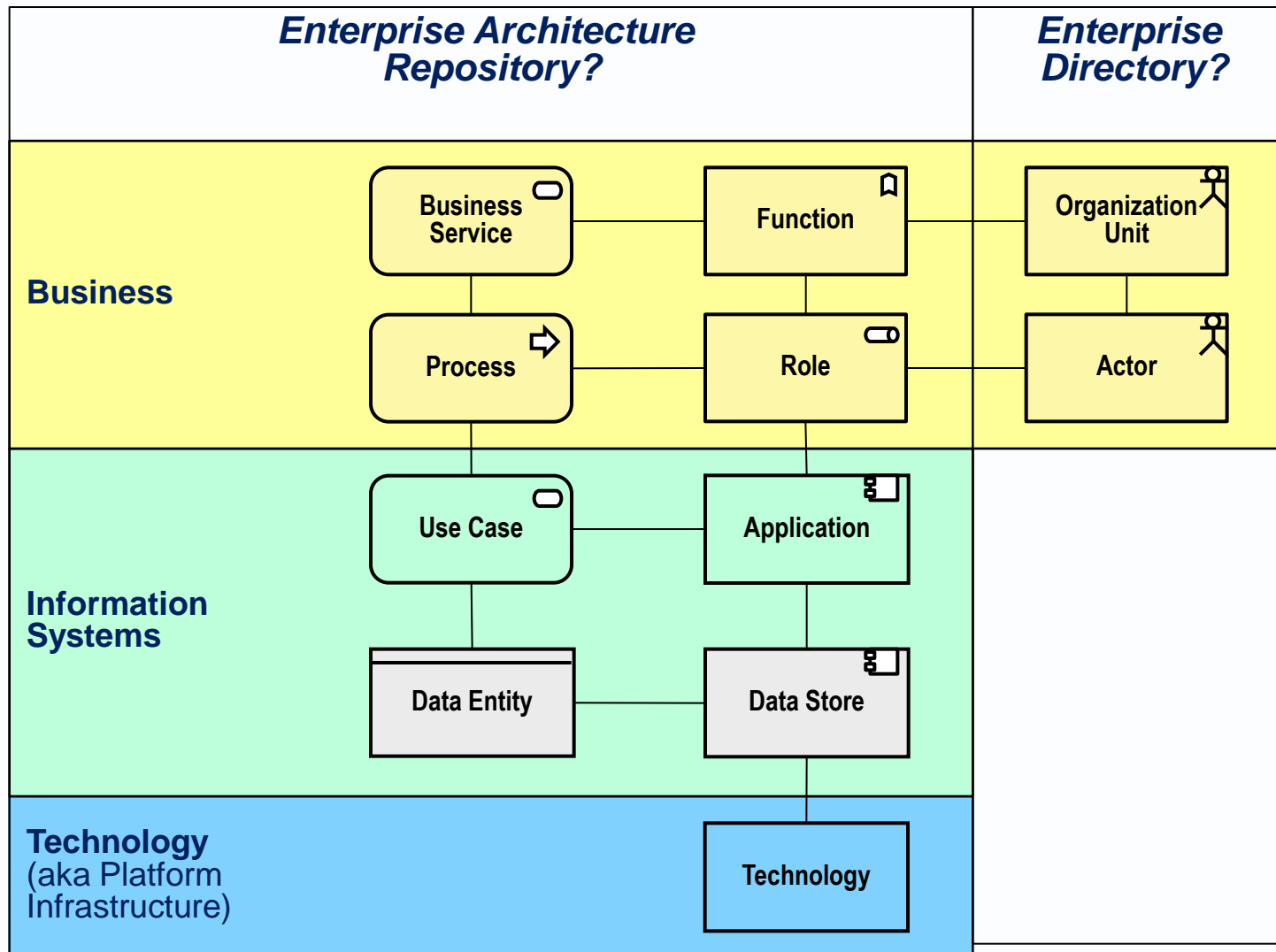


With best-fit
ArchiMate
symbols

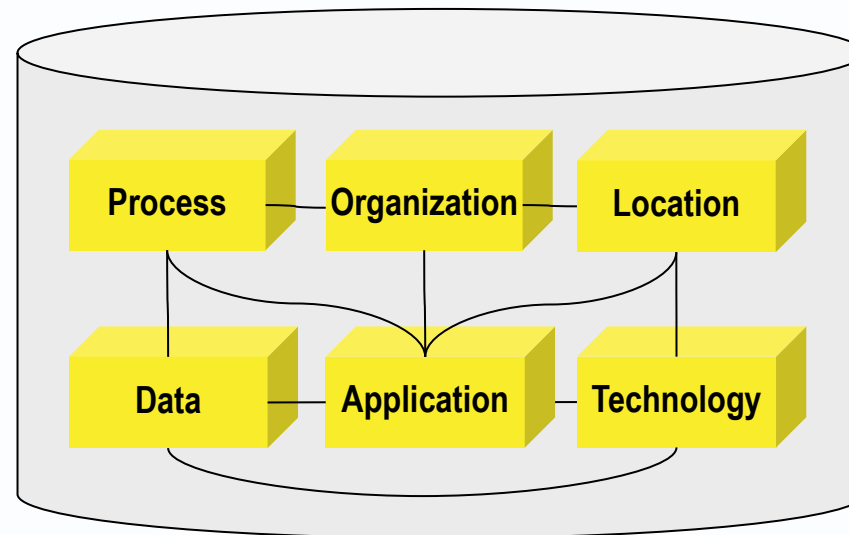
ArchiMate: a selection of entities and relationships



Realistically, can even this much be stored and maintained?



- ▶ [A data store] that stores architecture deliverables, artifacts and entities, and relates architecture entities as defined in a schema or architecture meta model.



- ▶ [A correspondence] drawn between architectural entities in an architecture repository.

	Loc	Loc	Loc
Org	Works at		
Org		Works at	Works at
Org	Works at	Works at	

- ▶ Mappings may be made the purposes of
 - gap analysis,
 - impact analysis,
 - requirements traceability analysis and
 - cluster analysis.

Mappings for gap analysis

- ▶ Gap analysis means looking for
 - items with no relationship,
 - loose ends, black holes and
 - gaps that may require attention.
- ▶ If the correspondence is close to one-to-one, then in a simple table will suffice:

Baseline Office Buildings	Target Office Buildings
London	London
Paris	Paris
New York to be closed	
	Mumbai to be opened

Mappings for gap analysis between states

- ▶ What elements in this architecture **state**
- ▶ don't match or relate to elements in another architecture **state**?

Target Apps Baseline Apps	Billing	CRM	Business Intelligence	Baseline not in target
Billing	Port to new platform			
CRM		Leave as is		
Resourcing			???	Decommission or reconsider
Target not in baseline			Buy or build	

Mappings for gap analysis between domains

- ▶ Which elements in this architecture **domain**
- ▶ don't match or relate to elements in another architecture **domain**?

Technologies	DBMS	Messaging	ESB	Apps with no technology
Applications				
CRM	Supported by	Supported by		
Broker App		Supported by		
Employee Portal			???	Employee Portal
Technologies not used			ESB	

Mappings for traceability analysis

- ▶ Why is this element needed? What objectives does it help to meet?
- ▶ What valued service or output does it help to produce?

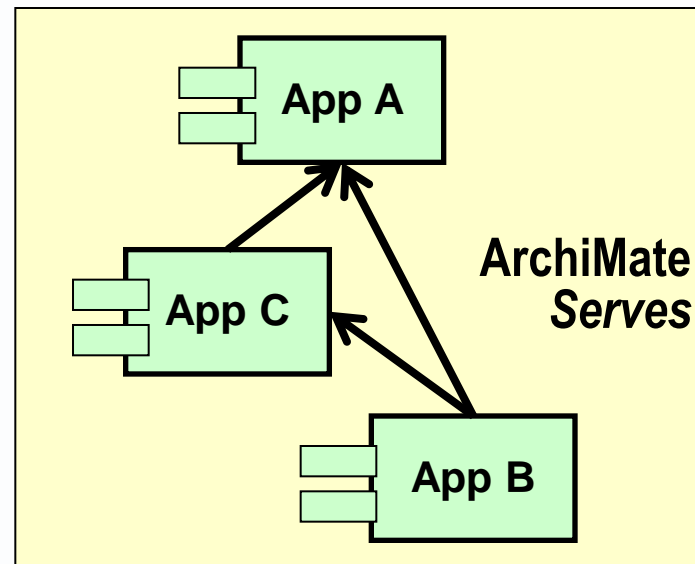
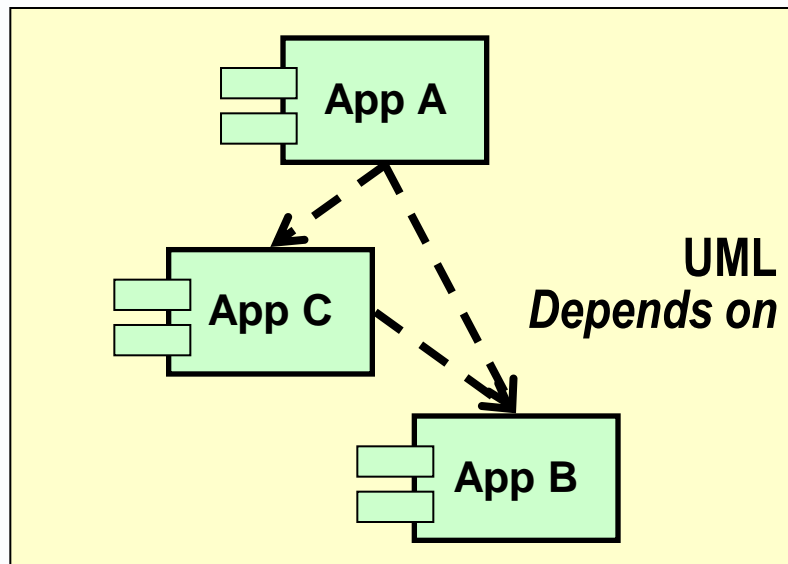
Solution items	Business Intelligence	Sales Mobile device	CRM	Requirements with no solution
Requirements				
Faster Ordering		Satisfied by	Satisfied by	
Remote Working		Satisfied by		
Better Forecasts	Satisfied by			
Lower Sale Cost			???	No solution
Solution items with no requirement				

- ▶ Real life example: 40 solution elements
- ▶ 800 functional requirements (in a hierarchy) + 200 NFRs!

Mappings for change impact analysis

- If we change this element, what related elements may be affected?

	App A	App B	App C
App A		Depends on	Depends on
App B			
App C		Depends on	

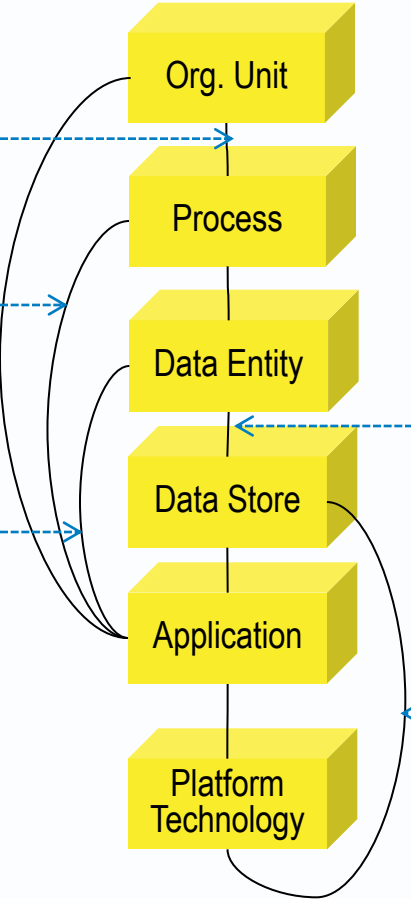
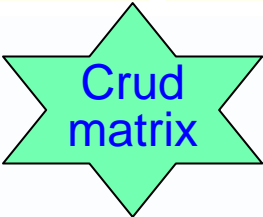


Mappings for change impact analysis

	Process	Process	Process
Org	Performs		
Org		Performs	Performs
Org			Performs

	Process	Process	Process
App	supports		
App		supports	supports
App			supports

	App	App	App
Entity	cru	rud	
Product	crud	ru	r
Asset	r	crud	r



Data Store Data Entity	CRM	ERP	Assets
Customer	x	x	
Product		x	
Asset		x	x

	Tech	Tech	Tech
Data store	uses		
Data store		uses	uses
Data store			

Mappings for cluster analysis

- ▶ Which elements are closely enough related to be grouped in one component or work package?
- ▶ E.g. Which activities create the same data?

Actor or Activity Data element	Billing	Delivery	Sales	Reporting
Customer	Use	Use	Create	Use
Order	Use	Use	Create	Use
Delivery	Use	Create		Use
Invoice	Use	Create		Use
Payment	Create			Use
Report				Create

Actor or Activity Data element	Sales	Delivery	Billing	Reporting
Customer	Create	Use	Use	Use
Order	Create	Use	Use	Use
Delivery		Create	Use	Use
Invoice		Create	Use	Use
Payment			Create	Use
Report				Create

Mappings for cluster analysis

- ▶ Which elements are closely enough related to be grouped in one component or work package?
- ▶ E.g. Which activities create the same data?

		LOGICAL APPLICATION GROUPS		DATA CLASSES	
		PROCESSES			
PLANNING	Develop agency plans	C	C	C	C
	Administer agency budget	C	C	C	C
	Formulate program policies	U	U	C	U
	Formulate admin. policies	U	U	C	C
	Formulate data policies	U	U	C	C
GENERAL MANAGEMENT	Design work processes	U	U	U	C
	Manage public affairs	U	U	U	C
	Manage intorgvt. affairs	U	U	U	C
	Exchange data	U	U	U	C
	Maintain admin. accounts	U	U	U	C
	Maintain prog. accounts	U	U	U	C
	Conduct audits	U	U	U	C
	Establish organizations	U	U	U	C
	Manage human resources	U	U	U	C
	Provide security	U	U	U	C
PROGRAM ADMIN.	Manage equipment	U	U	U	C
	Manage facilities	U	U	U	C
	Manage supplies	U	U	U	C
	Manage workloads	U	U	U	C
	Issue Social Security nos.	U	U	U	C
SUPPORT	Maintain earnings	U	U	U	C
	Collect claims information	U	U	U	C
	Determine elig./entitmt.	U	U	U	C
	Compute payments	U	U	U	C
	Administer debt mgmt.	U	U	U	C
	Generate notices	U	U	U	C
	Respond to prog. inquiries	U	U	U	C
	Provide quality assessment	U	U	U	C

KEY
C = creators of data U = users of data

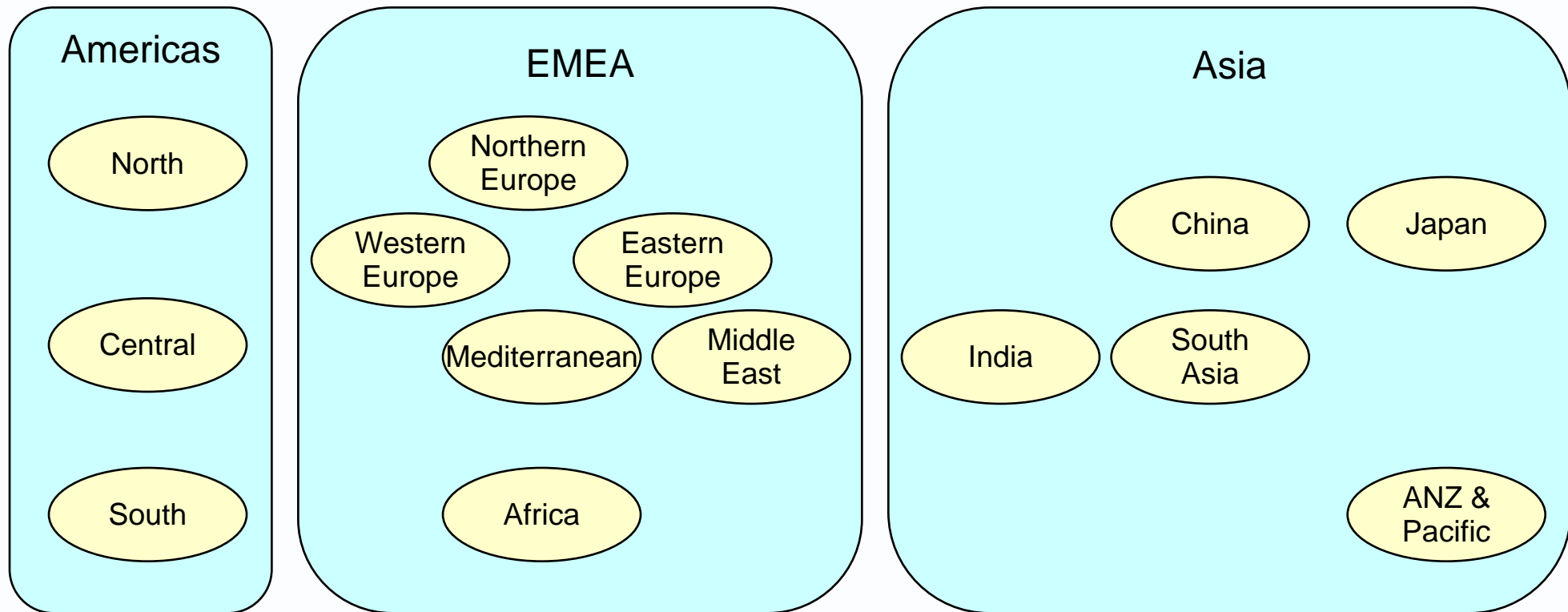
© Minder Chen, 1997-2008

Figure 12-1

Enterprise A

Mappings for cluster analysis

- ▶ Which elements are closely enough related to be grouped in one component or work package?
- ▶ E.g . Which entities are related by **location** and **time-frame**?

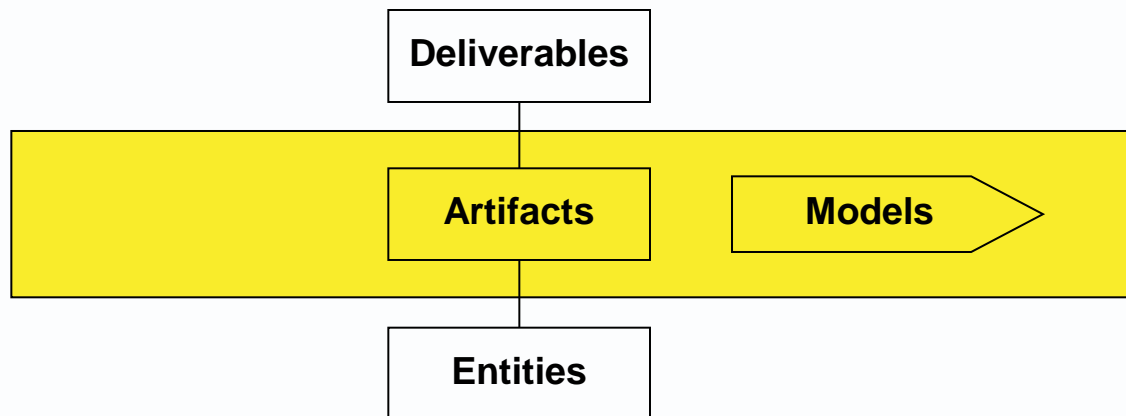


Four reasons to draw *mappings* between architectural entities

- ▶ Gap analysis
 - To identify changes and find potentially missing items
- ▶ Traceability analysis
 - To check deliverables meet goals and solutions solve problems.
- ▶ Impact/Dependency analysis
 - To find the effects of a change
- ▶ Cluster analysis
 - To group cohesive / closely-coupled items into encapsulated components

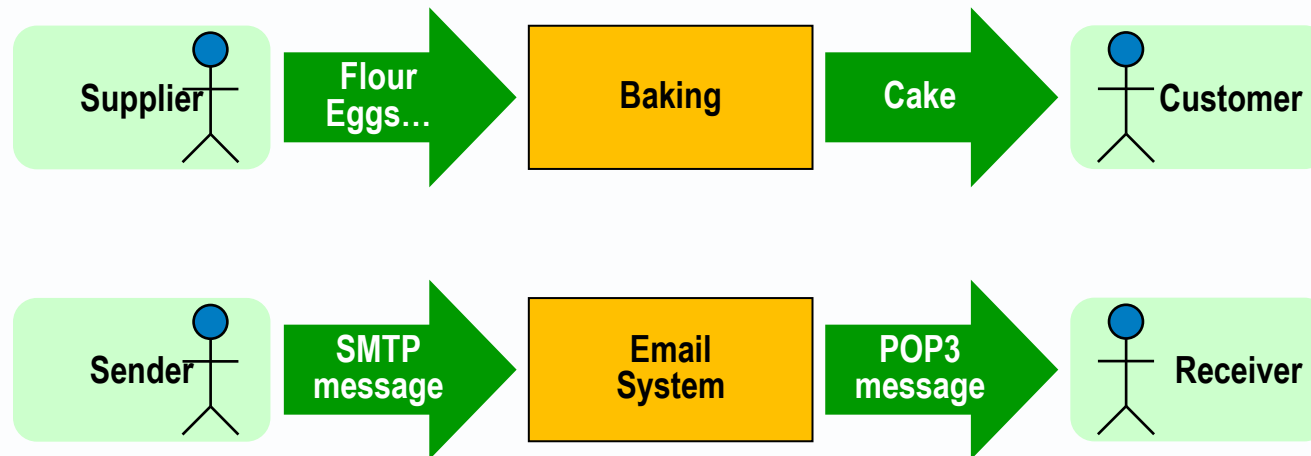
**You'll be
tested later!**

3.2 Architecture models



View (instance or example)


- ▶ [A work product] that shows a part or slice of an architecture.
- ▶ It addresses particular concerns.
- ▶ It can be visual, graphical or textual.
- ▶ It may contain one or more models.




- ▶ As an instance or example of a viewpoint, it conforms to the definition of that viewpoint.

Viewpoint (type or template)

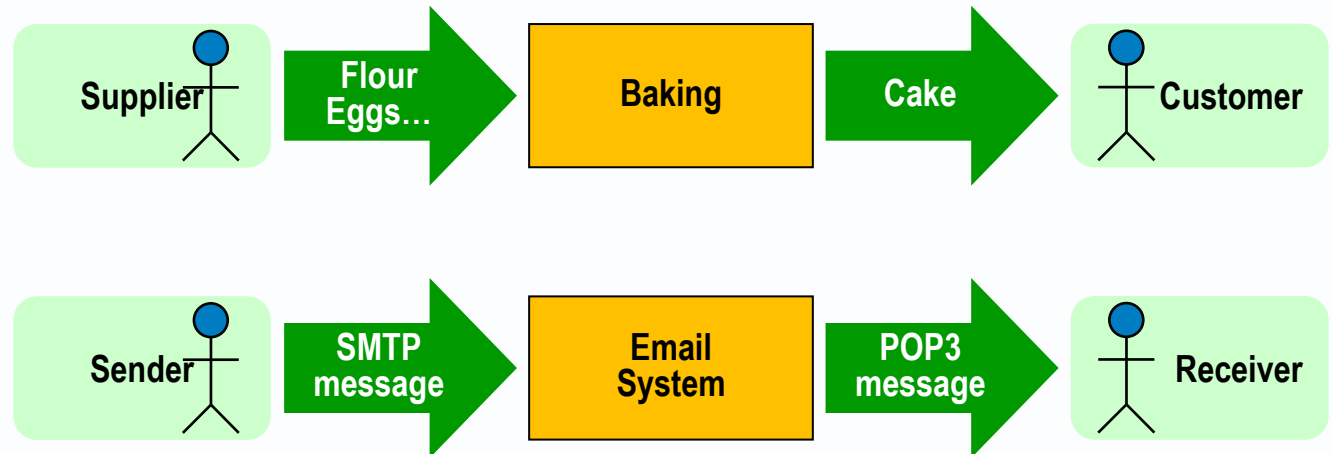
- ▶ [A work product description] that typifies a view and provides a template for defining a view.
- ▶ It defines the conventions for creating and using views to address concerns about a system.
- ▶ It defines:
 - ▶ what – the name of the viewpoint
 - ▶ why - concern(s) that the viewpoint addresses
 - ▶ who - stakeholder(s) who have the concerns
 - ▶ how - model kind(s) used in the view.

What	Who cares?	Why?	How to draw it?
Context diagram	Owners and designers	The I/O scope of the system	 <pre> graph LR S((Supplier)) -- Input --> Sys[System] Sys -- Output --> C((Consumer)) </pre>

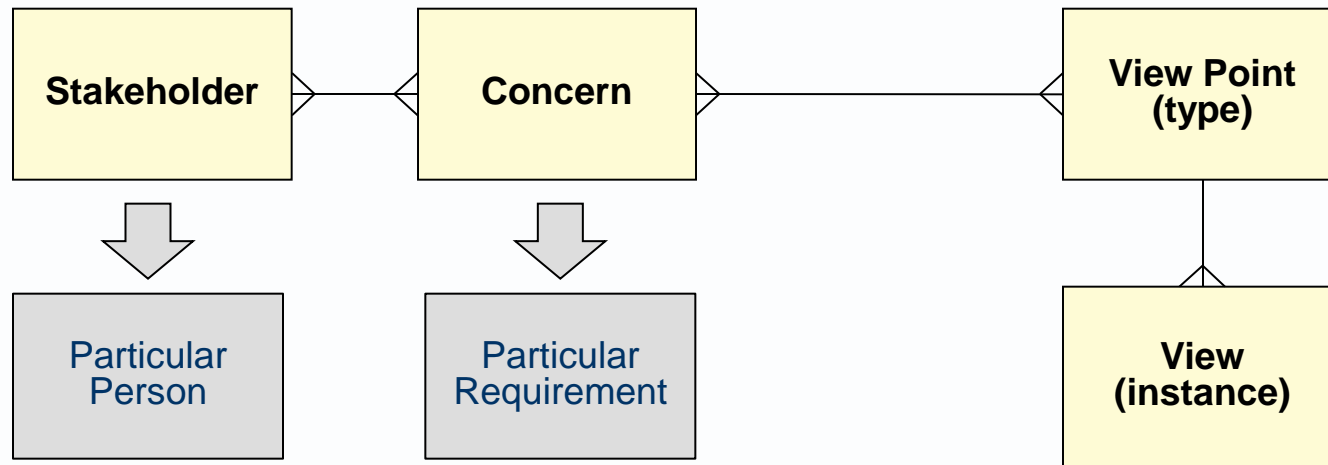
► Viewpoint  Template or type

What	Who cares?	Why?	How to draw it?
Context diagram	Owners and designers	The I/O scope of the system	

► View  Example or instance



A view conforms to one viewpoint



▶ **Model**

- ▶ [A work product] that simplifies or abstracts from a thing or another description.
- ▶ It displays or records some properties of what is modelled.
- ▶ It enables some questions to about it to be answered.
- ▶ Architects build relatively abstract models of systems.

▶ **Model kind**

- ▶ [A work product description] that typifies a model and provides a template for building a model, such as may be found in modelling languages.

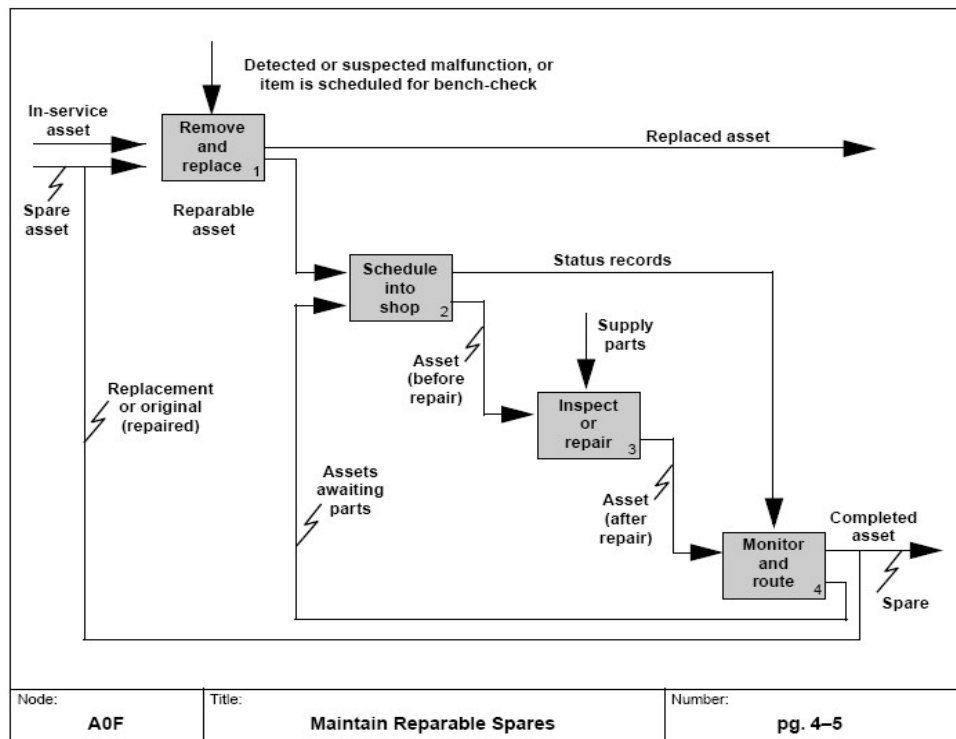
▶ **Modelling language**

- ▶ [A standard] that defines ways to represent architecture entities and the relationships between them, such as IDEF, UML and ArchiMate.

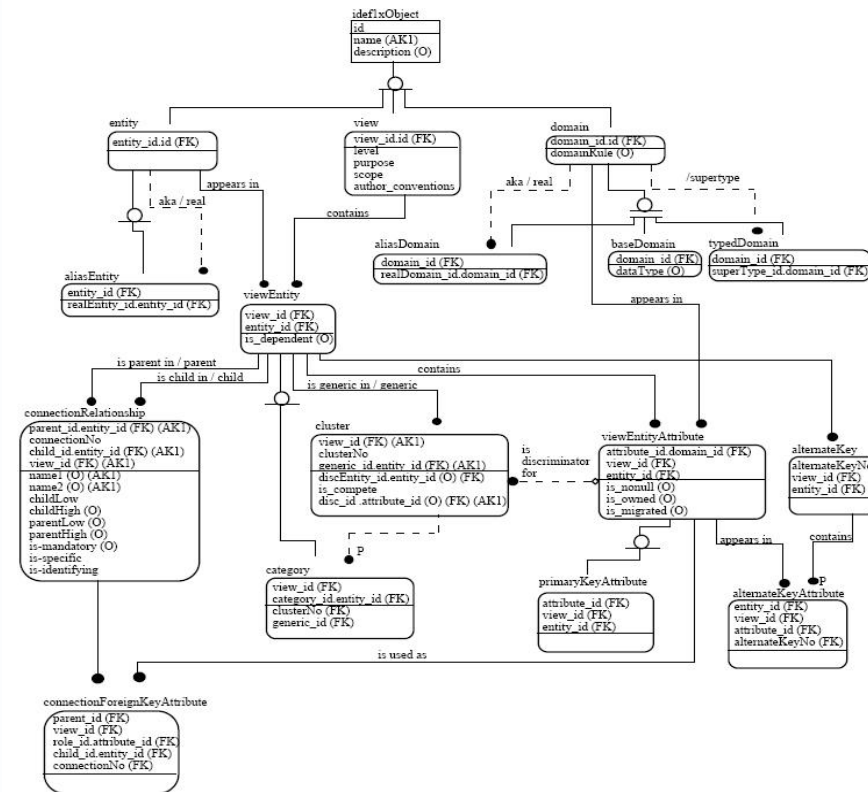
Integration Definition language (IDEF)

► Grew out of 1970s USAF standards, best known for

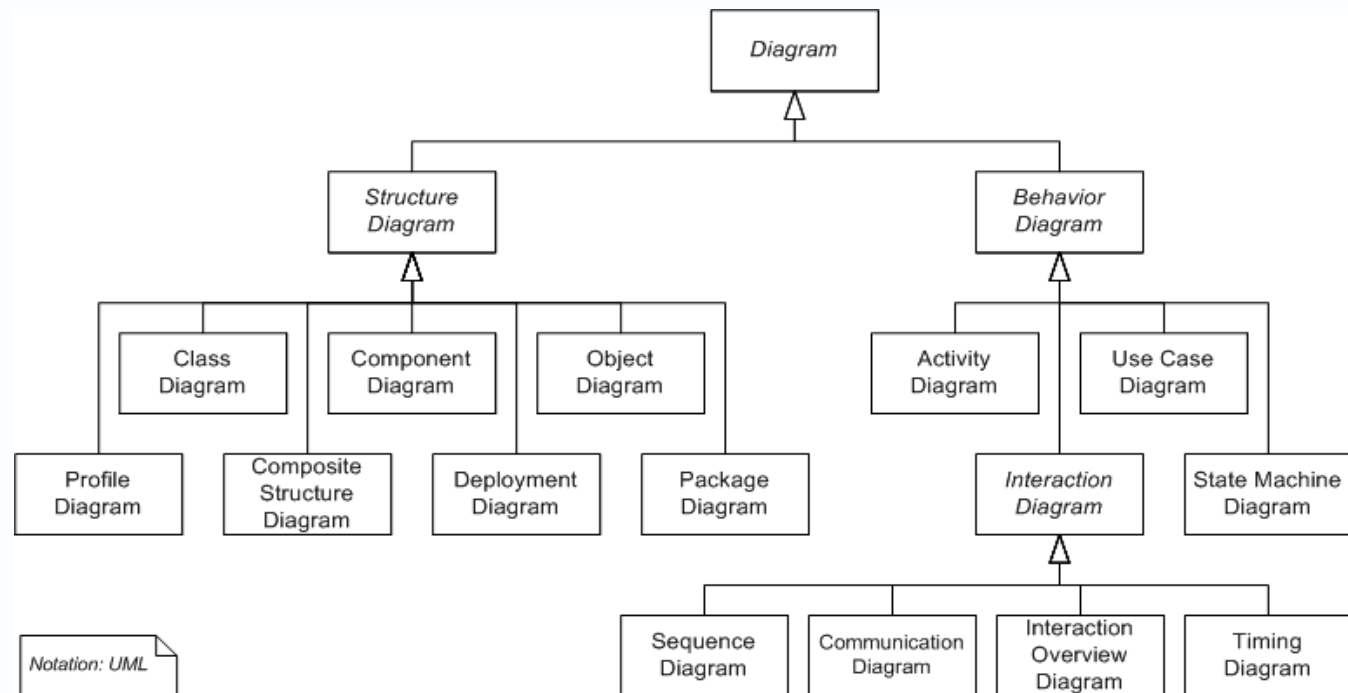
► IDEF 0 Function Models



IDEF 1X Data Models (Wikipedia)

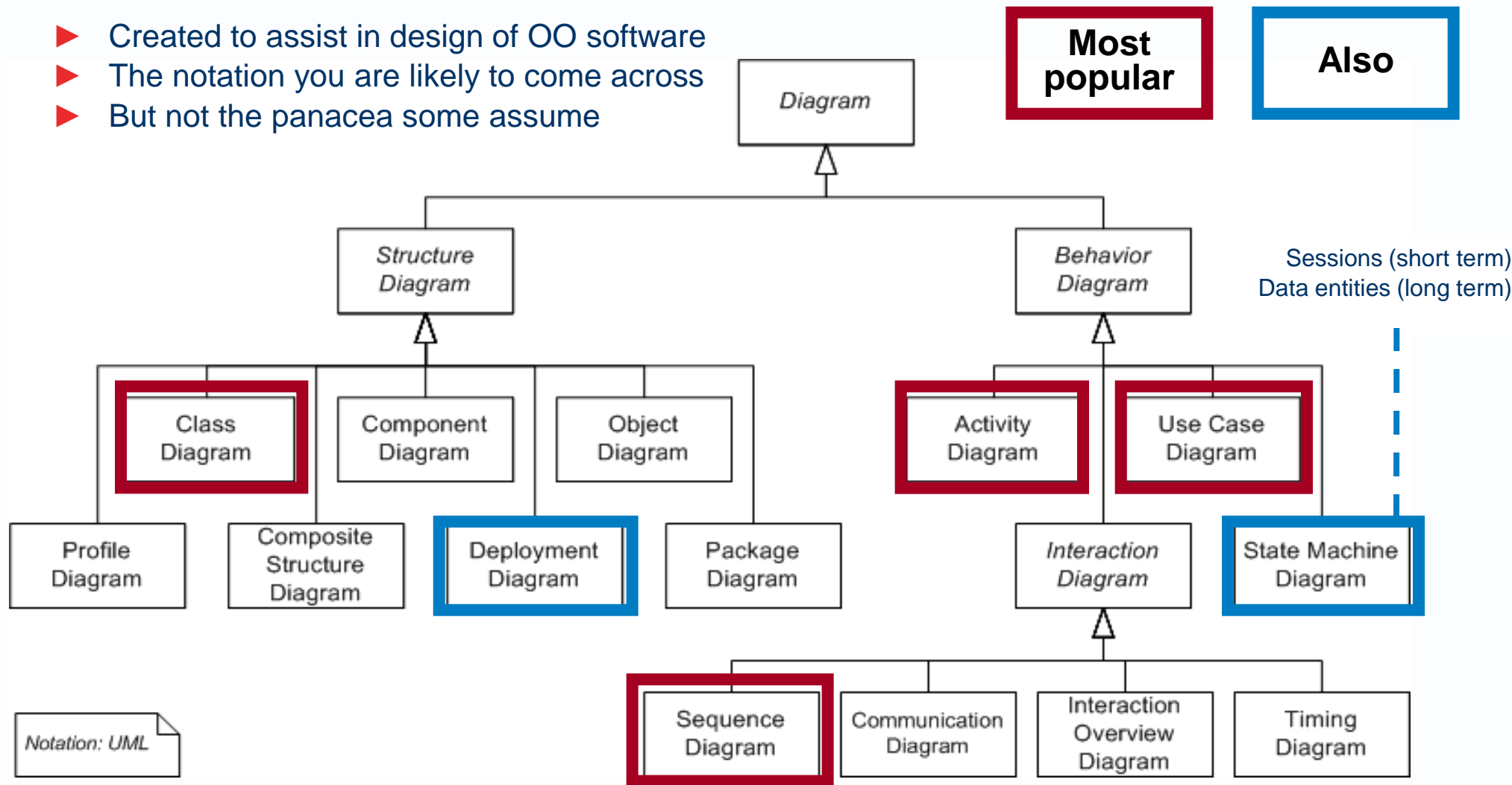


- ▶ [A modelling language] maintained by the Object Management Group.
- ▶ Initially designed to help in OO software design, now used outside of that.
- ▶ Includes *structural* models such as class diagrams and deployment diagrams.
- ▶ Includes *behavioural* models such as use case, activity and sequence diagrams.



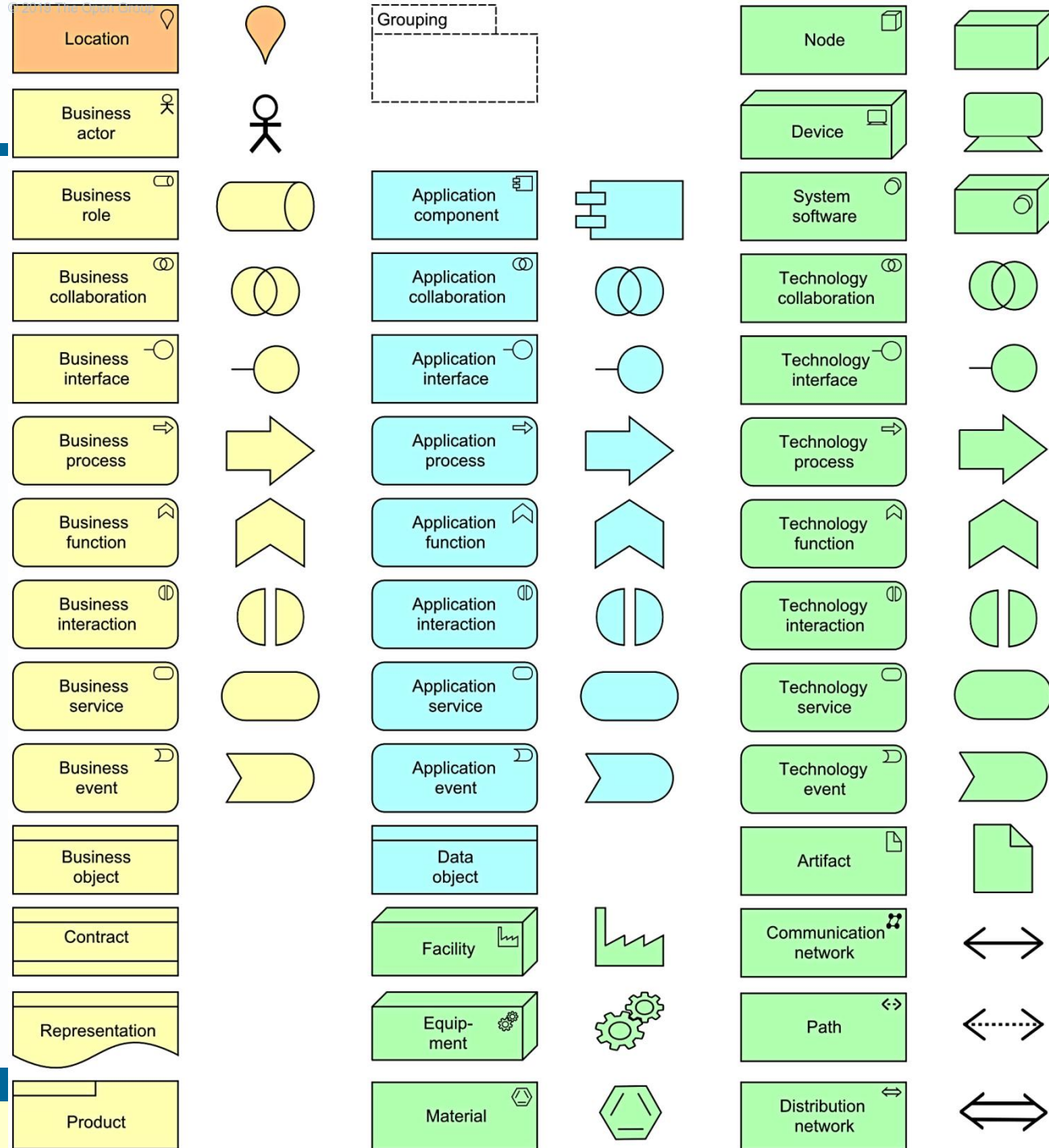
UML (OMG standard)

- ▶ Created to assist in design of OO software
- ▶ The notation you are likely to come across
- ▶ But not the panacea some assume

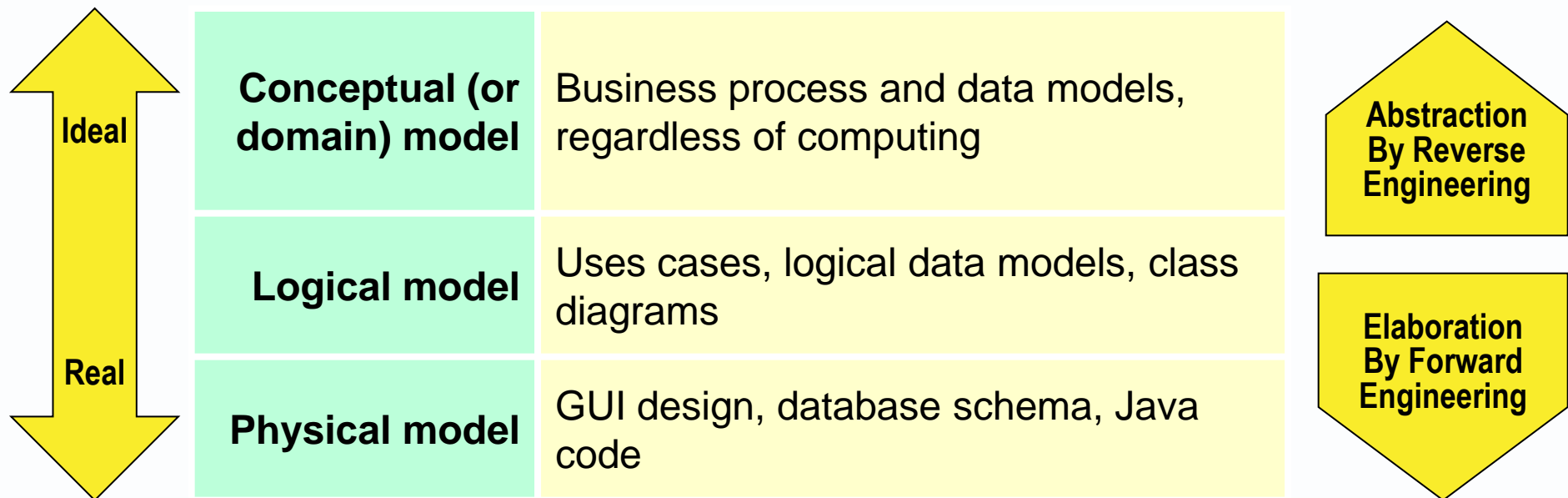


ArchiMate 3.1

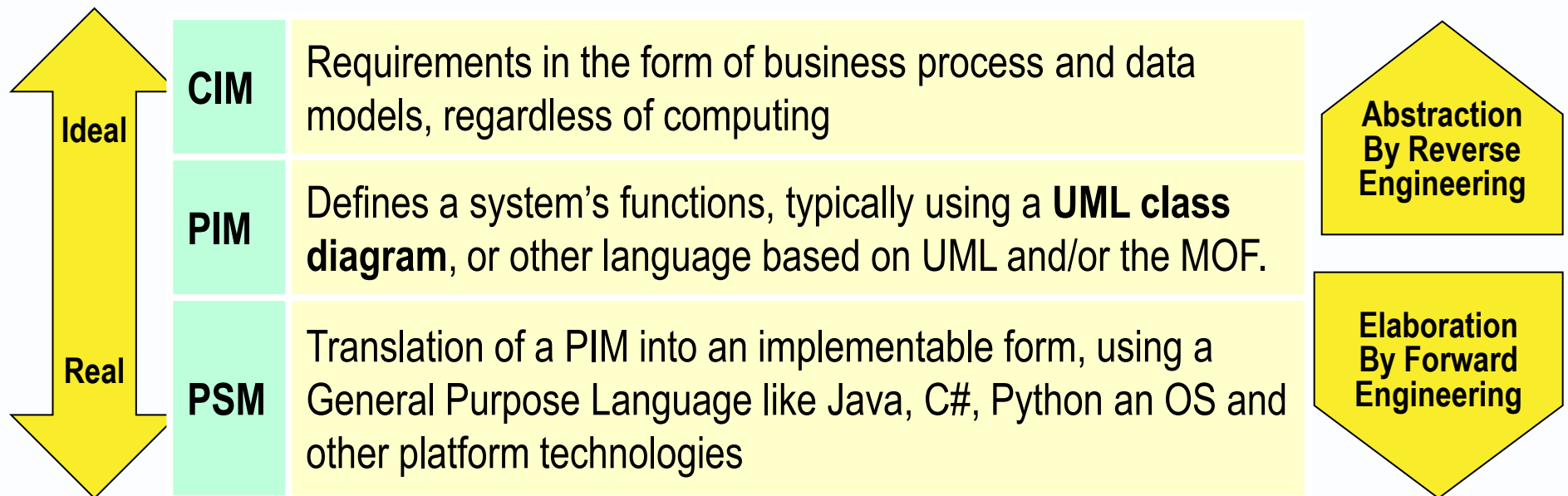
- ▶ [A modelling language] maintained by the Open Group.
- ▶ Components, interfaces and services are representing in boxes using different symbols.
- ▶ It overlaps with UML, but is intended for more abstract architectural design.



- ▶ [A technique] used in methods and tools for forward engineering and reverse engineering, that is, for transforming a conceptual model to a logical model to a physical model, and/or the reverse of that process.



► Model-Driven Architecture from the Object Management Group



- ▶ [A pattern] a generic structure or classification used to create more specific models.
- ▶ It can be a structure of components, processes or data elements.
- ▶ It is sometimes applicable to a particular industry or business domain.
- ▶ It can act as a design pattern.
 - ▶ E.g.
 - APQC for a generic commercial organisation
 - BIAN for banking (one of many banking reference models)
 - TMF for telecoms
 - eTOM – Business Architecture
 - SID – Data Architecture
 - TAM – Applications Architecture
 - SCOR for supply-chain businesses
 - ProAct for retailers
 - FEA for US federal government
 - A long list of industry-specific canonical data model

APQC process classification framework.

This standard hierarchical classification of the functions in a commercial enterprise can provide you with a means to

- Structure baseline activities
- Identify and structure required activities.



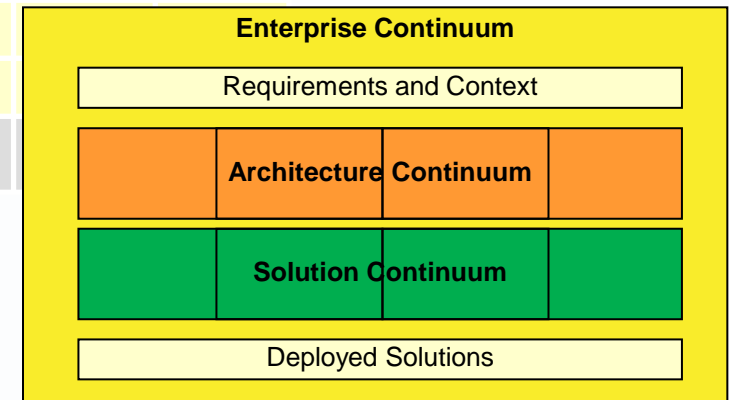
APQC updated and limited to 3 levels

1. UNDERSTAND MARKETS AND CUSTOMERS		
	1.1 Determine customer needs and wants	
	1.1.1 Conduct qualitative assessments	
	1.1.1.1 Conduct customer interviews	
	1.1.1.2 Conduct focus groups	
	1.1.2 Conduct quantitative assessments	
	1.1.2.1 Develop and implement surveys	
	1.1.3 Predict customer purchasing behavior	
	1.2 Measure customer satisfaction	
	1.2.1 Monitor satisfaction with products and services	
	1.2.2 Monitor satisfaction with complaint handling	
	1.2.3 Monitor satisfaction with communication	
	1.3 Monitor changes in market or customer expectations	
	1.3.1 Determine weaknesses of products and services	
	1.3.2 Identify new innovations that meet customer needs	
	1.3.3 Determine customer reactions to new products and services	
2. DEVELOP VISION AND STRATEGY		
	2.1 Monitor the external environment	
	2.1.1 Analyze and understand competition	
	2.1.2 Identify economic trends	
	2.1.3 Identify political and regulatory issues	
	2.1.4 Assess new technology innovations	
	2.1.5 Understand demographics	
	2.1.6 Identify social and cultural changes	
	2.1.7 Understand ecological concerns	
	2.2 Define the business concept and organizational strategy	
	2.2.1 Select relevant markets	
	2.2.2 Develop long-term vision	
	2.2.3 Formulate business unit strategy	
	2.2.4 Develop overall mission statement	
	2.3 Design the organizational structure and relationships between organizational units	
	2.4 Develop and set organizational goals	

3.3 Architecture content classifications

- ▶ A classification framework (aka taxonomy) for documentation.
- ▶ A set of pigeon holes for architecture description artefacts.

Zachman Framework	What	How	Where	Who	When	Why
Scope Contexts						
Business Concepts						
System Logic						
Technology Physics						
Tool components						
Operations – Instance classes						



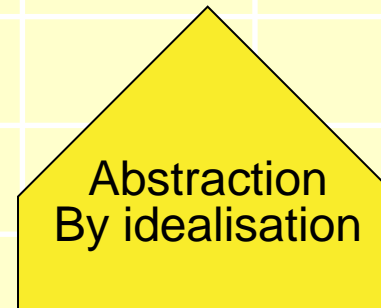
- ▶ A *window* on to an architecture repository
- ▶ A *classification scheme* for reusable architecture assets

Zachman framework

A classification framework for documentation
A set of pigeon holes for architecture description artefacts.

- [A pattern] “A logical structure for classifying and organising the descriptive representations of an Enterprise that are significant to managers and to developers of Enterprise systems.”

Zachman Framework v2		Columns					
Rows - reification		What	How	Where	Who	When	Why
Idealisation-Reification	Stakeholders	Inventory sets	Process Transform'n	Network nodes	Organisation groups	Time periods	Motivation reasons
Scope Contexts	Strategists & theorists						
Business Concepts	Enterprise leaders & owners						
System Logic	Architects & designers						
Technology Physics	Engineers & builders						
Component assemblies	Technicians & implementers						
Operations Instance classes	Workers & participants						




- ▶ The 6 columns, though titled with interrogative questions, are mapped to architectural description facets or elements.
- ▶ The 6 rows are primarily levels of realisation from context to operational systems.
- ▶ But they are also mapped to stakeholder types and architecture domains.
- ▶ Zachman says the rows should not be interpreted as levels of decomposition.

Zachman Framework v2		Columns					
Rows - reification		What	How	Where	Who	When	Why
Idealisation-Reification	Stakeholders	Inventory sets	Process Transform'n	Network nodes	Organisation groups	Time periods	Motivation reasons
Scope Contexts	Strategists & theorists						
Business Concepts	Enterprise leaders & owners						
System Logic	Architects & designers						
Technology Physics	Engineers & builders						
Component assemblies	Technicians & implementers						
Operations Instance classes	Workers & participants						

Enterprise continuum

- ▶ [A pattern] a logical structure in TOGAF for classifying and organising architecture artifacts. It can be drawn as a table or grid.
- ▶ From top to bottom is ideal to real.
- ▶ From left to right is general to specific.

Enterprise Continuum	Foundation	Common systems	Industry	Organisation
	Universal building blocks for system construction	Used in most business domains	E.g. Telecoms or Banking	Your unique business
Context and requirements				
Architecture continuum				
Solution continuum				
Deployed solutions				