Avancier Methods (AM)

CONCEPTS

Mapping generic ArchiMate entities to and TOGAF meta model entities

TOGAF’s buildings blocks – archtiectural entities

- An architectural entity is a building block of architecture description, used in artefacts.

- TOGAF’s meta model of architectural entities is underpinned by two big ideas:
  - Components (the designed structure of a system) offer services (the required behaviour of a system).
  - Logical components are defined before physical (implementation-specific) components.

- The following slides analyse TOGAF’s meta model and text
- *Note that there are many odd things not explored here!*

A classification of 26 TOGAF entities

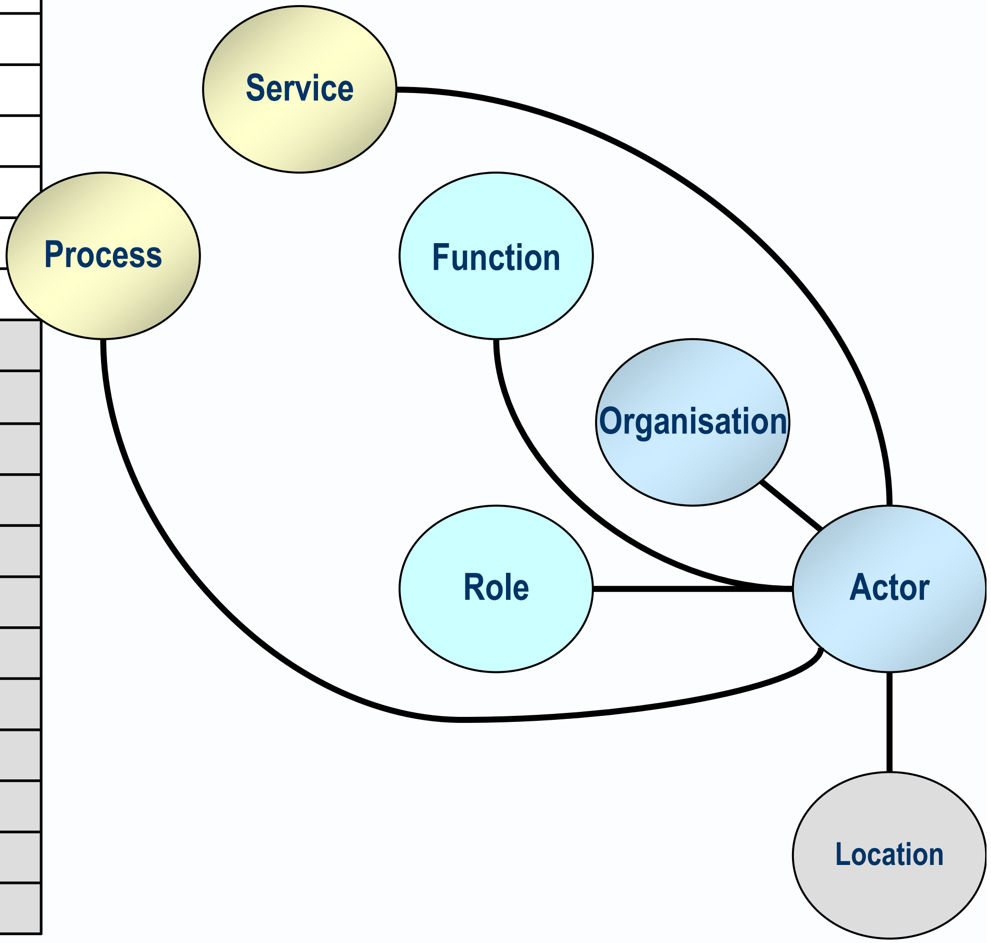
Avancier

Avancier's classification	TOGAF's architectural entities			
Motivation entities	Driver, Goal, Objective, Measure			
Architecture domain	**Business**	**Data**	**Applications**	**Technology**
External behaviour	Product, Service, Service Quality, Contract		Information system service	Platform Service
Internal behaviour	Process, Event, Control			
Logical structure	Capability, Function, Role,	Data Entity, Logical Data Component	Logical Application Component	Logical Technology Component
Physical structure	Location, Organization Unit, Actor	Physical Data Component	Physical Application Component	Physical Technology Component

Business domain relationships (after TOGAF)

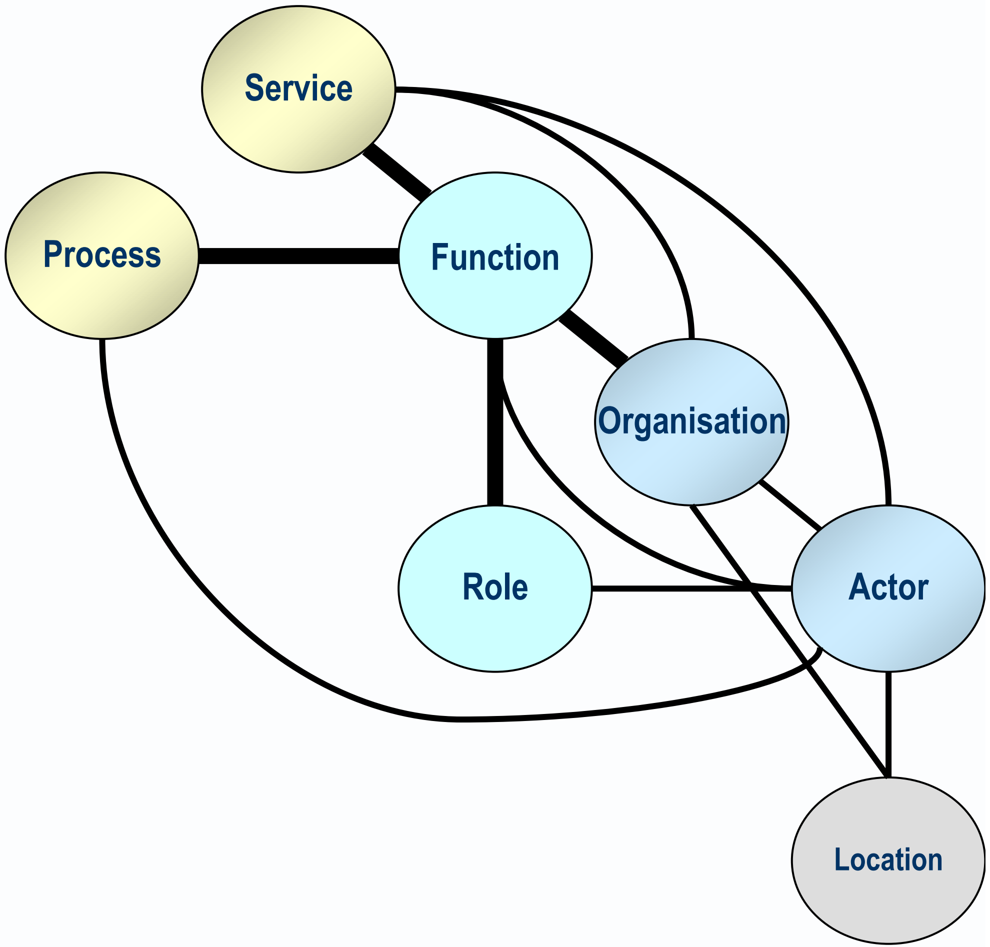
Actor	DECOMPOSES	Actor
Actor	generates	Event
Actor	resolves	Event
Actor	interacts with	Function
Actor	Performs	Function
Actor	operates in	Location
Actor	belongs to	Organization Unit
Actor	participates in	Process
Actor	performs task in	Role
Actor	consumes	Service
Capability	is delivered by	Work Package
Contract	governs and measures	Service
Contract	meets	Service Quality
Control	ensures correct operation of	Process
Driver	DECOMPOSES	Driver
Driver	creates	Goal
Driver	unit motivates	Organization
Event	is resolved by	Actor
Event	is generated by	Actor
Event	is resolved by	Process
Event	is generated by	Process
Event	is resolved by	Service

- Remarkably actor centric
- Should actor be role?

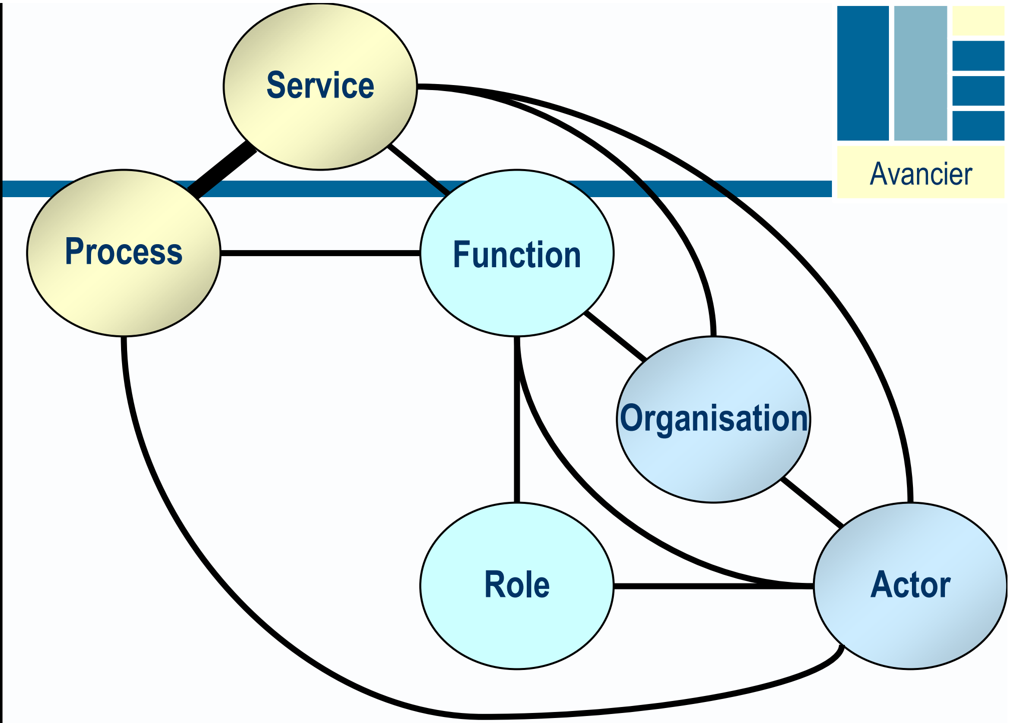


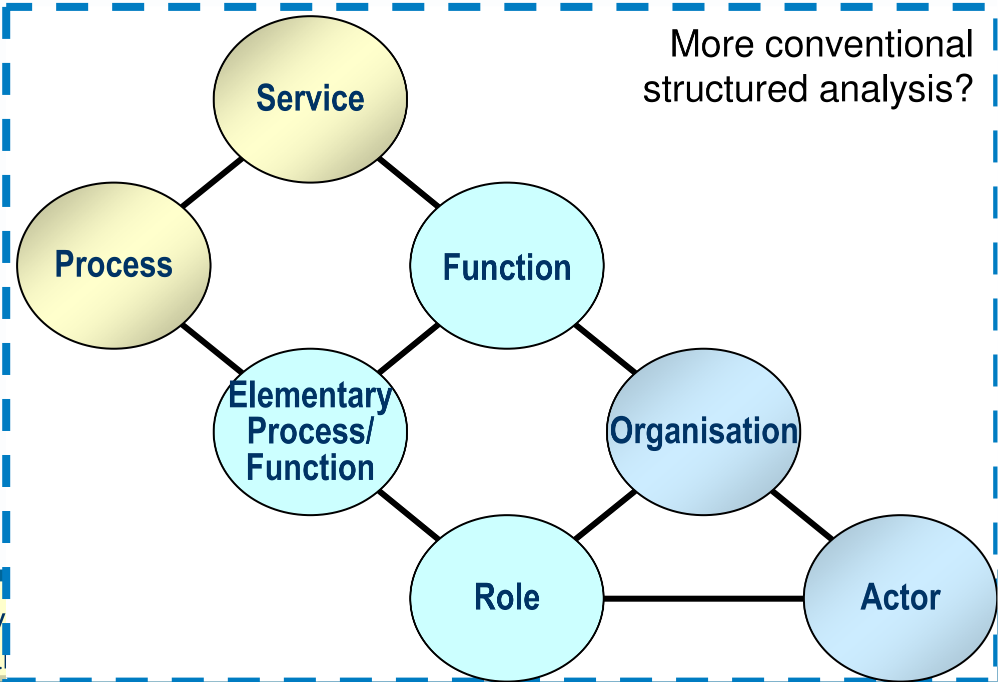
Function	supports	Actor
Function	is performed by	Actor
Function	DECOMPOSES	Function
Function	communicates with	Function
Function	is owned by	Organization Unit
Function	supports	Process
Function	is realized by	Process
Function	can be accessed by	Role
Function	**is bounded by**	**Service**
Goal	addresses	Driver
Goal	DECOMPOSES	Goal
Goal	is realized through	Objective
Location	contains	Actor
Location	DECOMPOSES	Location
Location	contains	Organization Unit
Measure	DECOMPOSES	Measure
Measure	sets performance criteria for	Objective
Measure	sets performance criteria for	Service
Objective	realizes	Goal
Objective	is tracked against	Measure
Objective	DECOMPOSES	Objective
Organization Unit	contains	Actor
Organization Unit	is motivated by	Driver
Organization Unit	owns	Function
Organization Unit	operates in	Location
Organization Unit	DECOMPOSES	Organization Unit
Organization Unit	produces	Product
Organization Unit	owns and governs	Service

- Function here as in conventional “structured analysis”



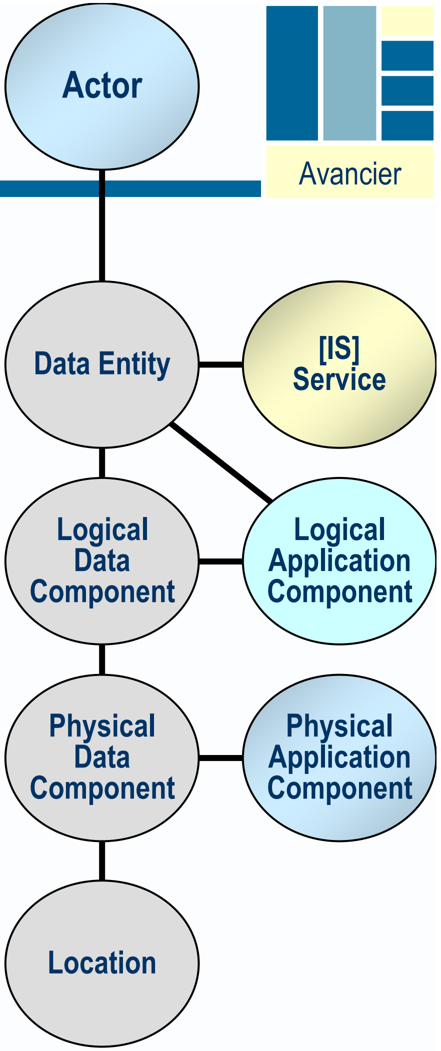
Process	involves	Actor
Process	is guided by	Control
Process	generates	Event
Process	resolves	Event
Process	orchestrates	Function
Process	DECOMPOSES	Function
Process	DECOMPOSES	Process
Process	precedes/follows	Process
Process	produces	Product
Process	orchestrates	Service
Process	DECOMPOSES	Service
Product	is produced by	Organization Unit
Product	is produced by	Process
Role	is performed by	Actor
Role	accesses	Function
Role	DECOMPOSES	Role
Service	is provided to	Actor
Service	is governed and measured by	Contract
Service	resolves	Event
Service	provides governed interface to access	Function
Service	is tracked against	Measure
Service	is owned and governed by	Organization Unit
Service	supports	Process
Service	is realized by	Process
Service	consumes	Service
Service	DECOMPOSES	Service
Service	meets	Service Quality
Service Quality	applies to	Contract
Service Quality	applies to	Service
Work Package	delivers	Capability




More conventional structured analysis?

Data domain relationships (after TOGAF)

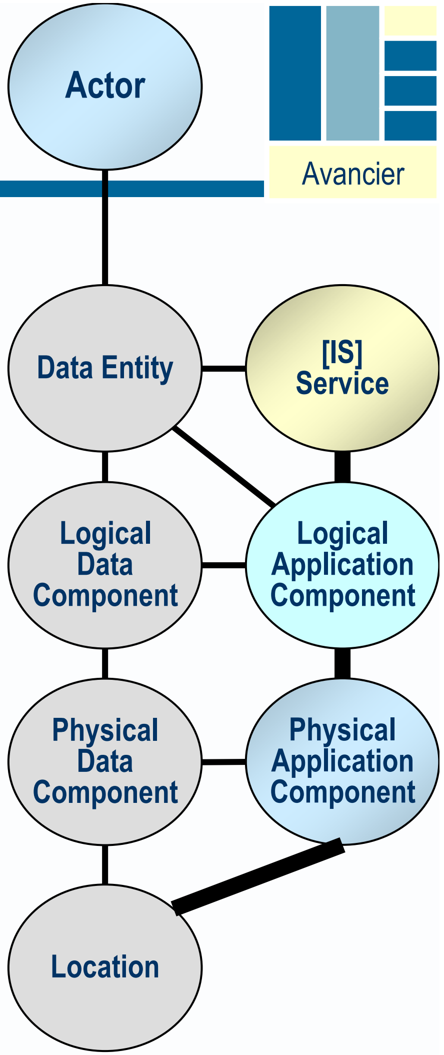
Actor	supplies/consumes	Data Entity
Data Entity	DECOMPOSES	Data Entity
Data Entity	relates to	Data Entity
Data Entity	is processed by	Logical Application Component
Data Entity	resides within	Logical Data Component
Data Entity	is accessed and updated through	Service
Location	contains	Physical Data Component
Logical Application Component	operates on	Data Entity
Logical Data Component	encapsulates	Data Entity
Logical Data Component	is extended by	Physical Data Component
Physical Application Component	encapsulates	Physical Data Component
Physical Data Component	is hosted in	Location
Physical Data Component	extends	Logical Data Component
Physical Data Component	encapsulates	Physical Application Component
Physical Data Component	DECOMPOSES	Physical Data Component
Service	provides	Data Entity
Service	consumes	Data Entity



ApplicationS domain relationships (after TOGAF)

Avancier

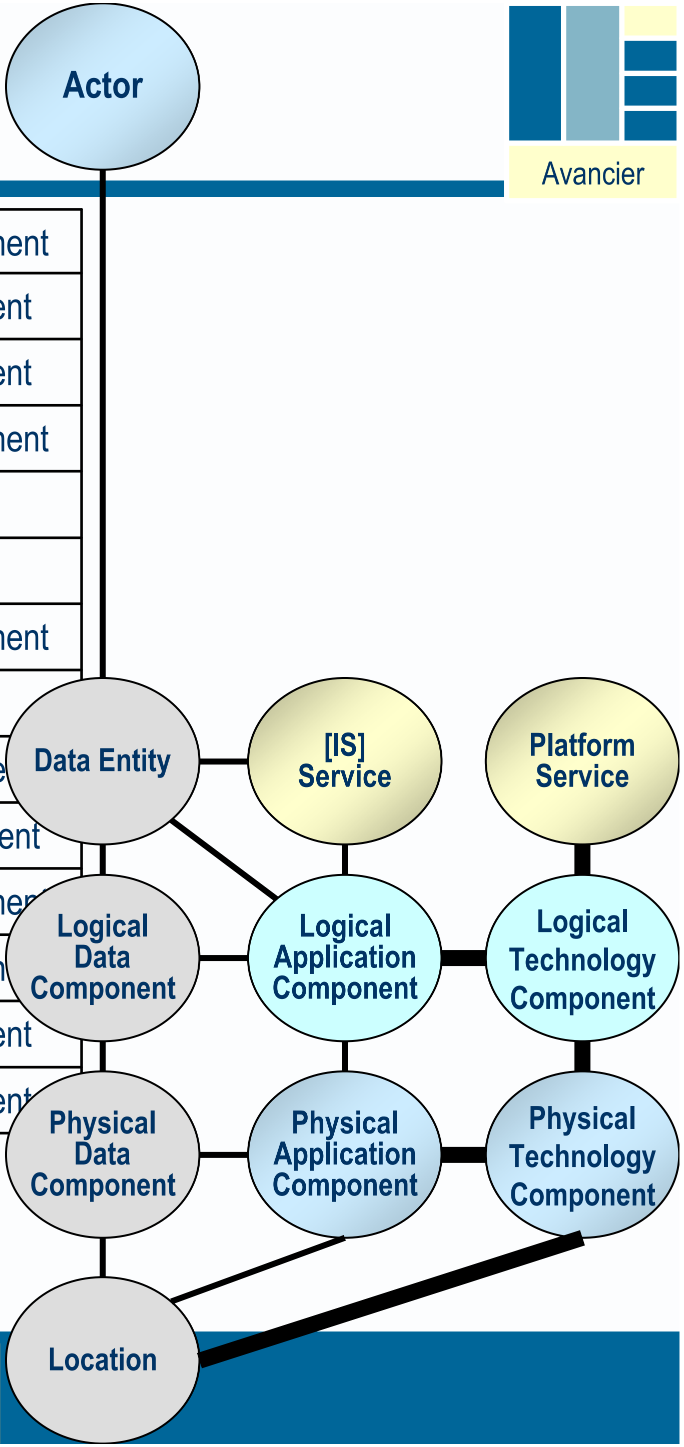
Location	contains	Physical Application Component
Logical Application Component	operates on	Data Entity
Logical Application Component	DECOMPOSES	Logical Application Component
Logical Application Component	communicates with	Logical Application Component
Logical Application Component	is extended by	Physical Application Component
Logical Application Component	implements	Service
Physical Application Component	is hosted in	Location
Physical Application Component	extends	Logical Application Component
Physical Application Component	encapsulates	Physical Data Component
Physical Application Component	DECOMPOSES	Physical Application Component
Physical Application Component	communicates with	Physical Application Component
Service	is realized through	Logical Application Component



Technology domain relationships (after TOGAF)

Avancier

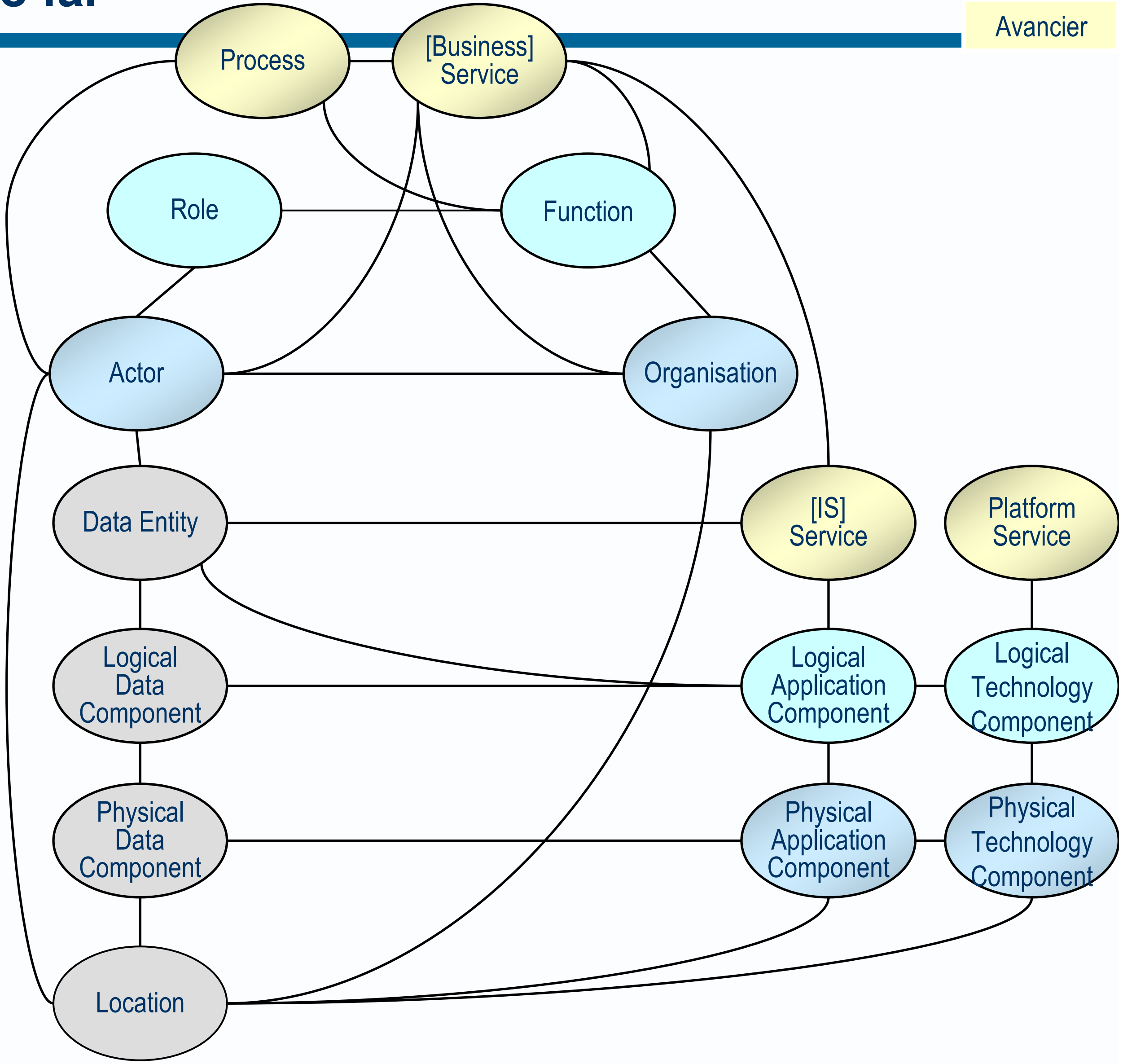
Location	contains	Physical Technology Component
Logical Technology Component	DECOMPOSES	Logical Technology Component
Logical Technology Component	is dependent on	Logical Technology Component
Logical Technology Component	is extended by	Physical Technology Component
Logical Technology Component	supplies	Platform Service
Logical Technology Component	provides platform for	Service
Physical Application Component	is realized by	Physical Technology Component
Physical Technology Component	is hosted in	Location
Physical Technology Component	extends	Logical Technology Compone
Physical Technology Component	realizes	Physical Application Component
Physical Technology Component	DECOMPOSES	Physical Technology Componen
Physical Technology Component	is dependent on	Physical Technology Compon
Platform Service	is supplied by	Logical Technology Component
Service	is implemented on	Logical Technology Component


Actor
Data Entity
[IS] Service
Platform Service
Logical Data Component
Logical Application Component
Logical Technology Component
Physical Data Component
Physical Application Component
Physical Technology Component
Location

Meta model diagram so far

- Model with relationships you can find in TOGAFs explicit meta model

- But not all the relationships implied by the text and artefacts



A model of entities and relationships in TOGAF

- Relationships in the explicit meta model
- Plus 6 more found the in TOGAF artefacts found below!

Role/Application Matrix

Data Entity/Business Function Matrix

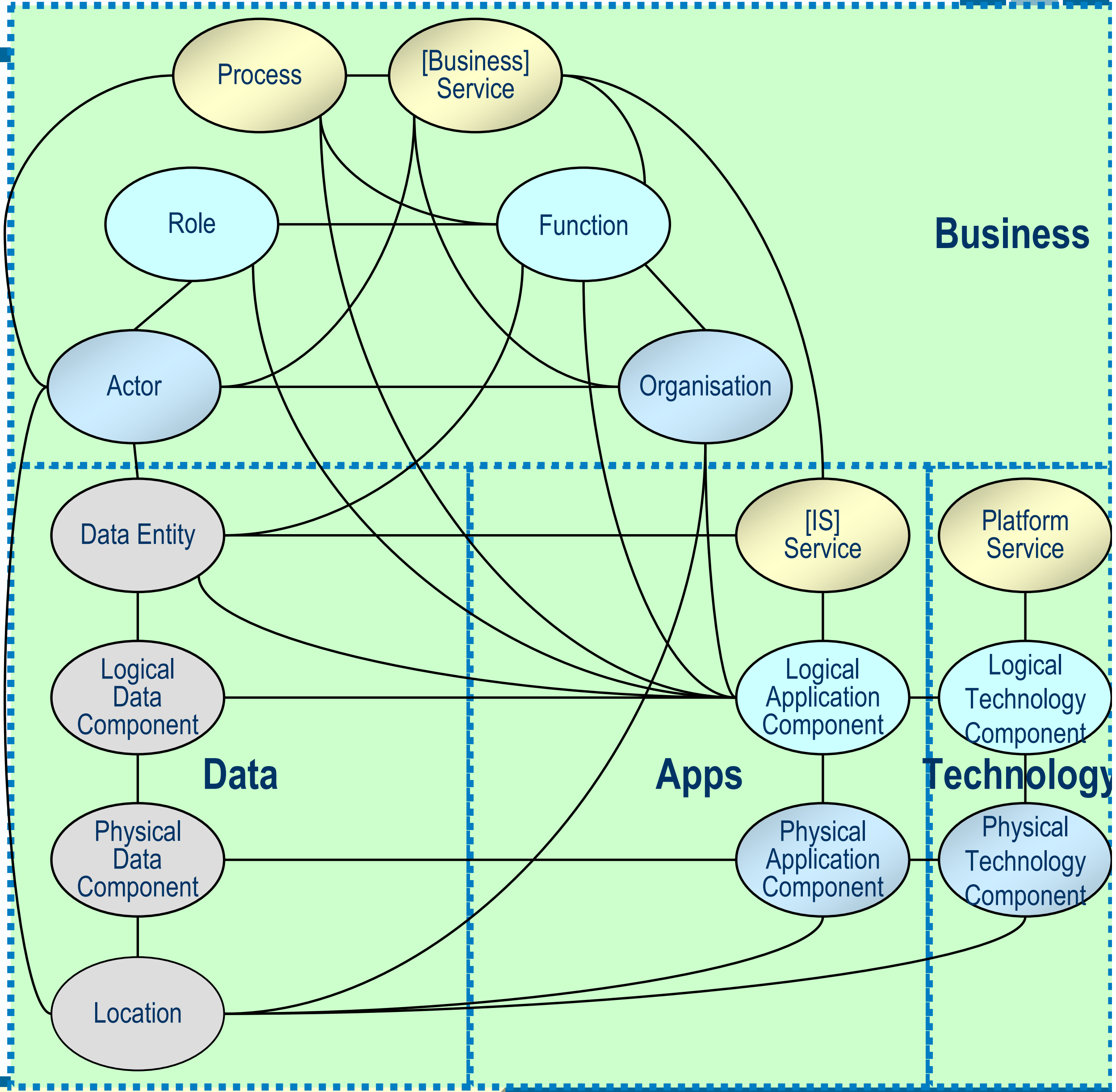
Application/Organization Matrix

Application/Function Matrix

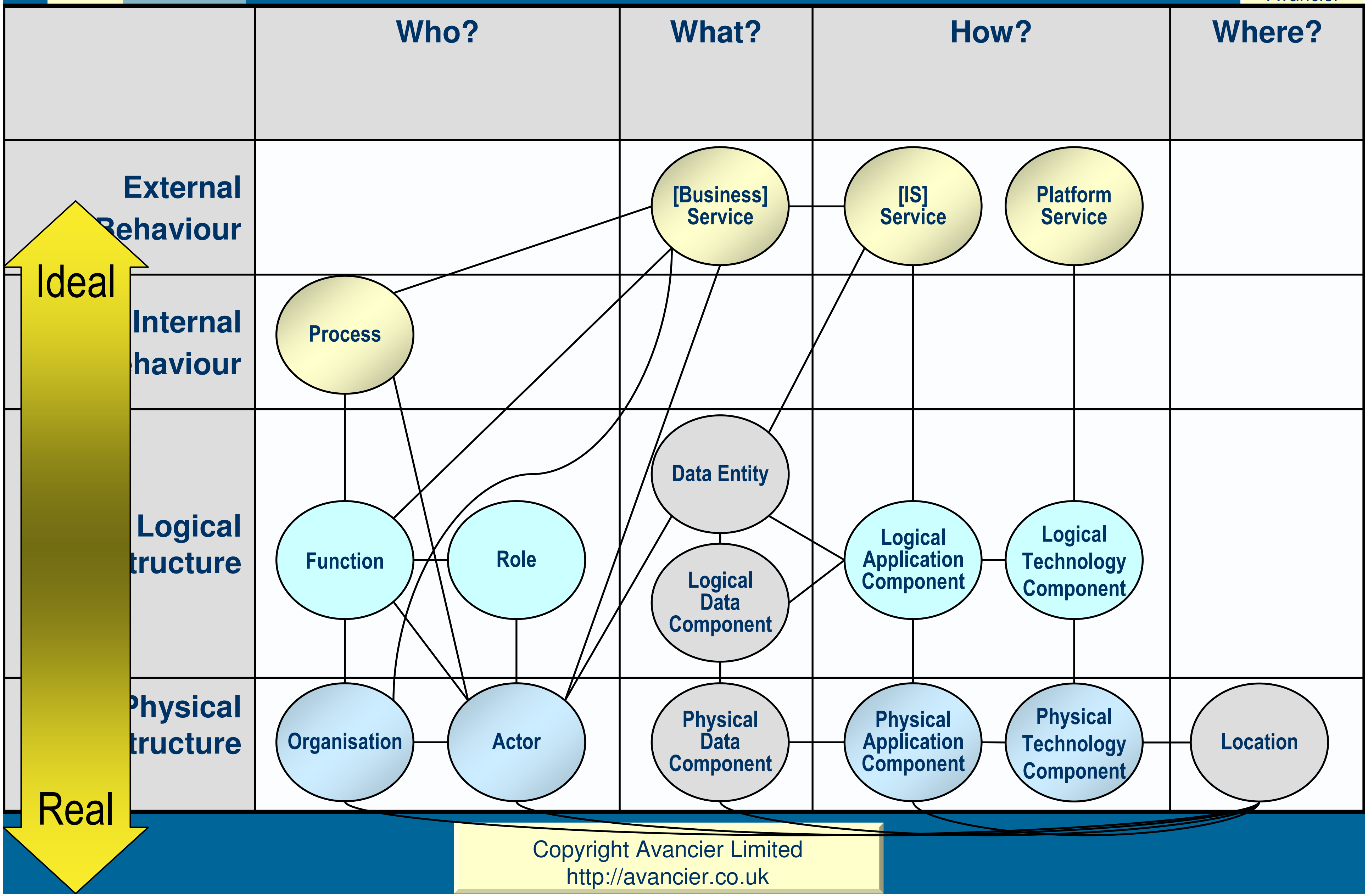
Application Use-Case Diagram

Interface Catalog

Data Flow ???



Mapping TOGAF to Zachman? (sans when and why)



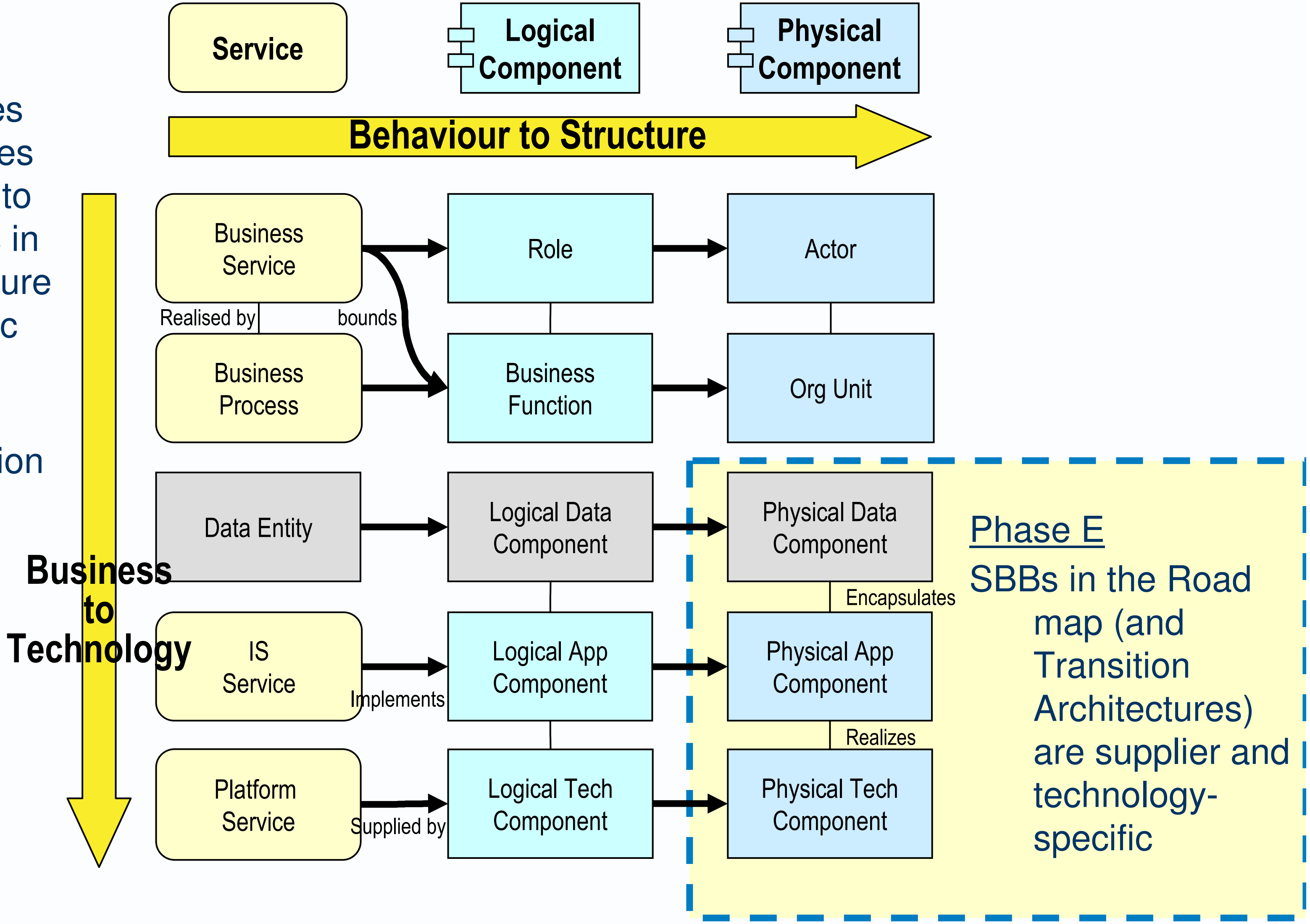
	Who?	What?	How?	Where?
External Behaviour		[Business] Service	[IS] Service Platform Service	
Internal haviour	Process			
Logical tructure	Function Role	Data Entity Logical Data Component	Logical Application Component Logical Technology Component	
Physical tructure	Organisation Actor	Physical Data Component	Physical Application Component Physical Technology Component	Location

Mapping of “logical” to “physical”

Phases B, C, D

Required services and processes are mapped to logical ABBs in the Architecture Definition doc are

Logical means implementation independent

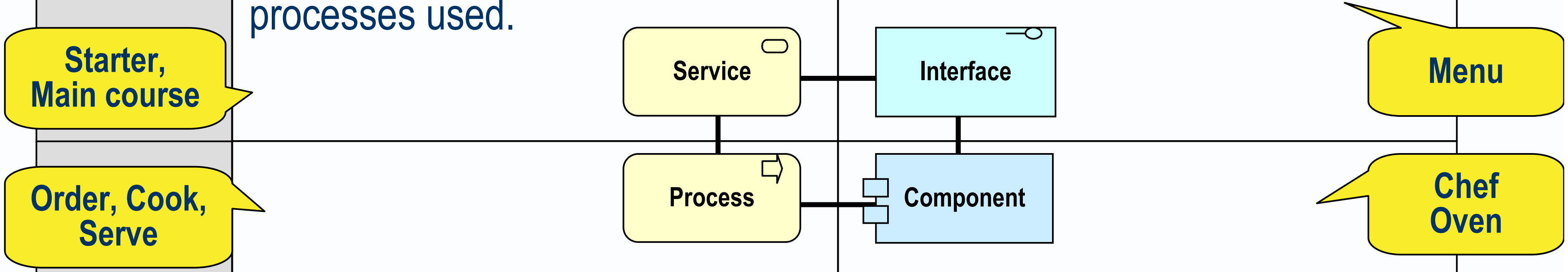


Phase E

SBBs in the Road map (and Transition Architectures) are supplier and technology-specific

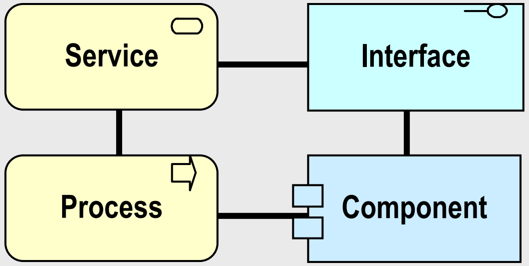
BCS generic meta model of activity systems

	Behaviour	Structure
External	Something consumers want from the system, or the system wants from suppliers. A result of processes, but defined in a service contract without reference to the internal logic of processes used.	A structured collection of services (to the left) that is requested or provided by one or more components (below).
Internal	What happens. A logical sequence of activities that ends up delivering a service at some level of granularity. Executed by components.	A subsystem that does work. A group of related but distinctly invokable activities, encapsulated behind a provided interface.

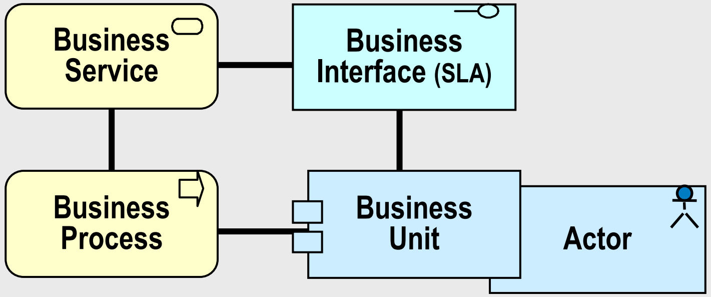


Mapping BCS generic meta model to architecture domains

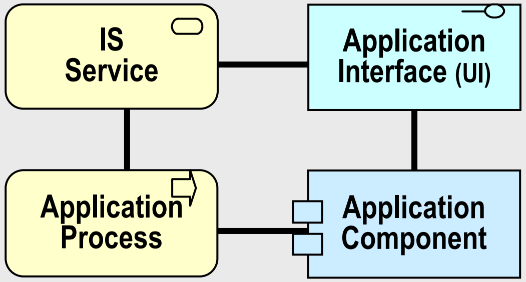
Generic meta model



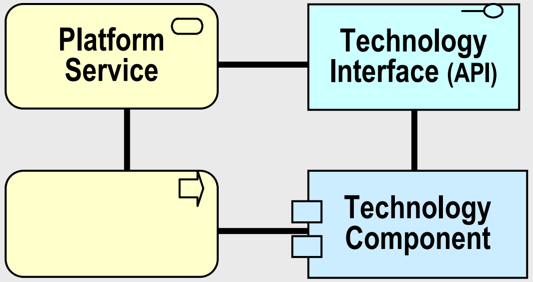
Business meta model



Applications meta model

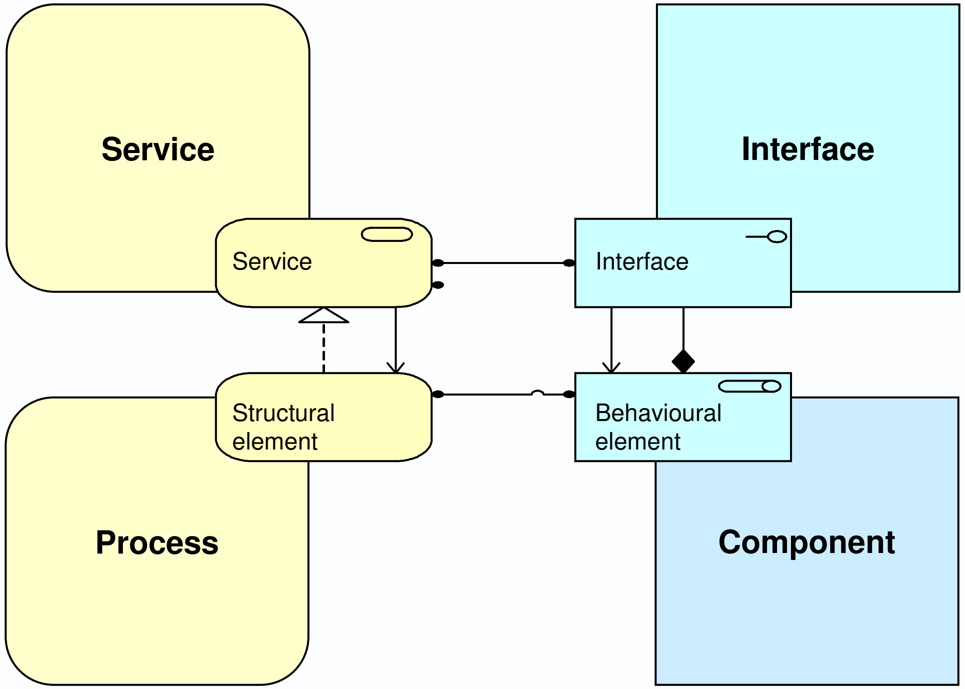


Infrastructure meta model



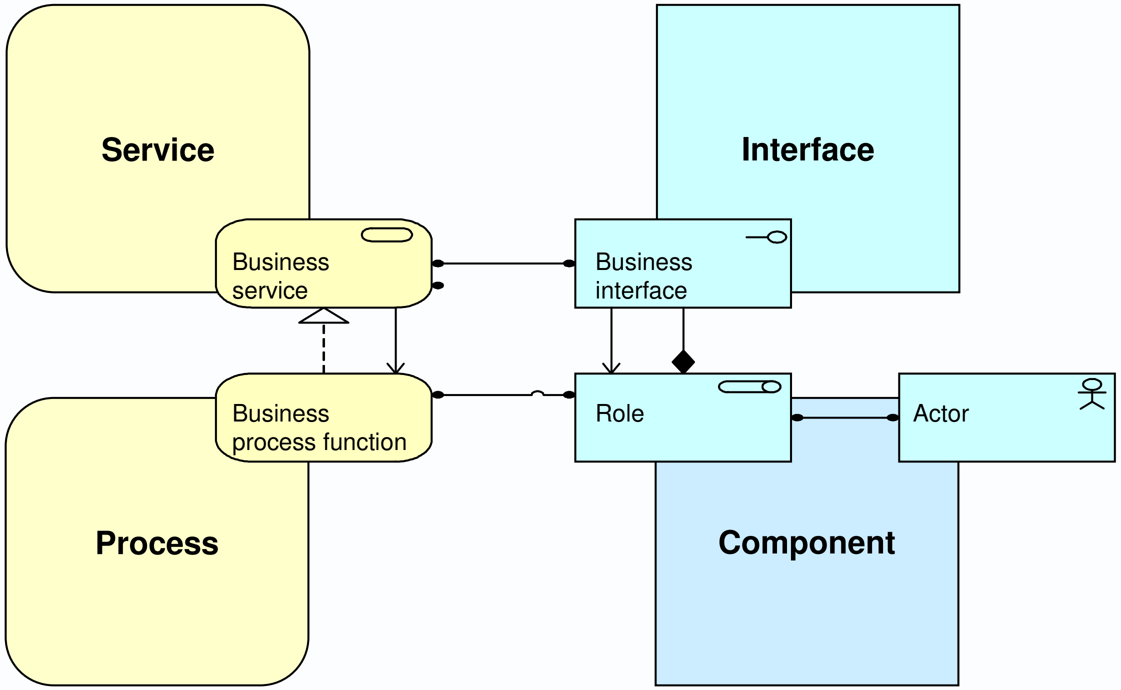
Mapping BCS generic meta model to ArchiMate - Generic

- BCS process = behavioural element
- BCS component = structural element



Mapping BCS generic meta model to ArchiMate - Business

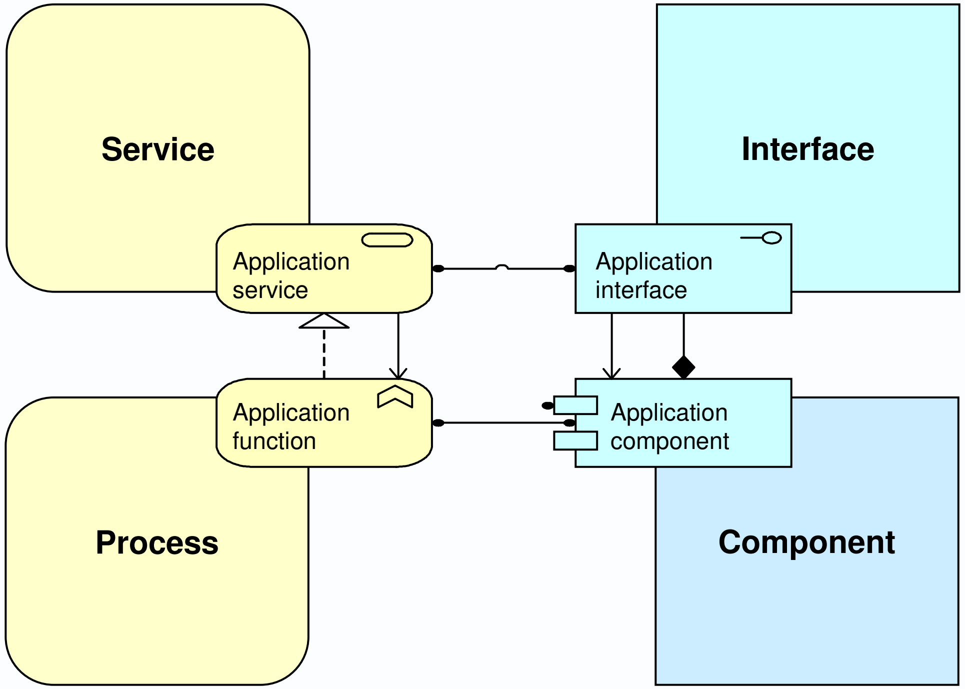
- BCS Process = Process or Function?
- BCS Component = Role and Actor = Logical and Physical?



Mapping BCS generic meta model to ArchiMate - Apps

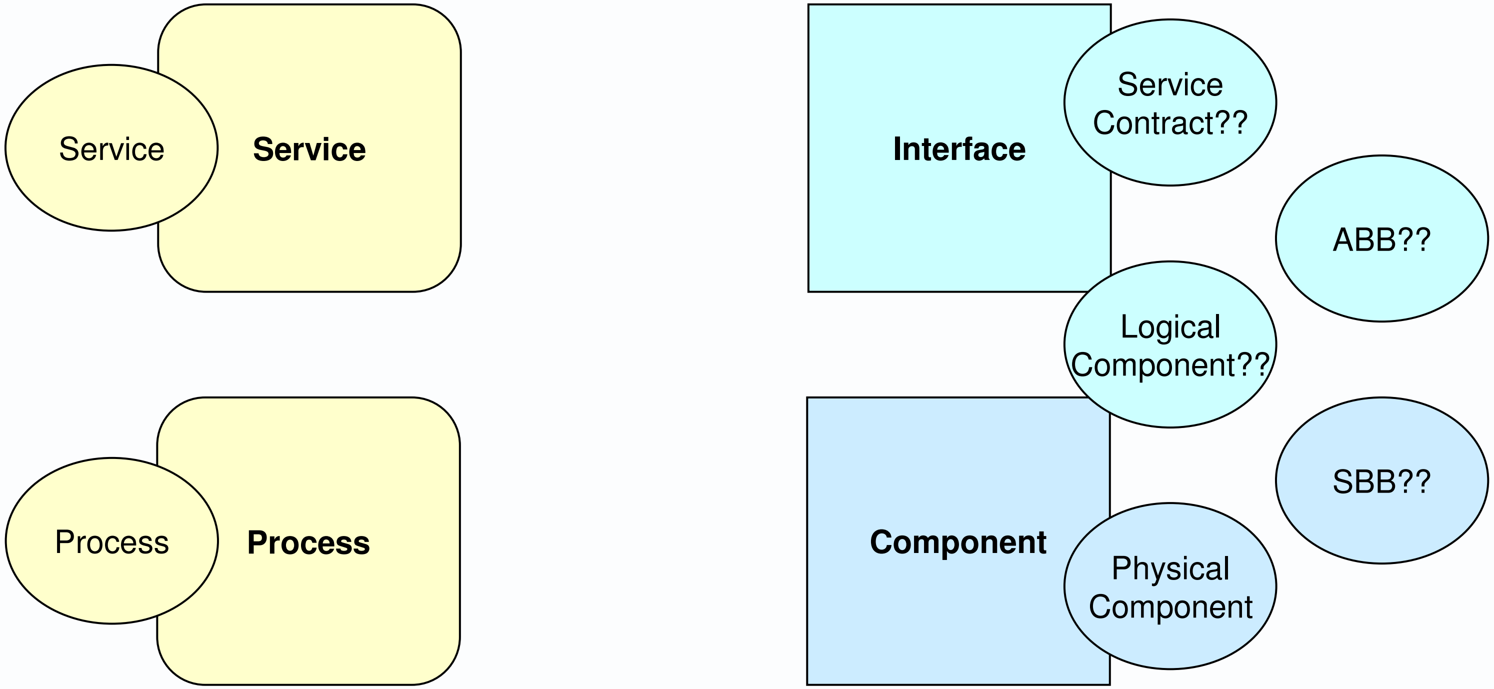
Avancier

- BCS Process = Function?



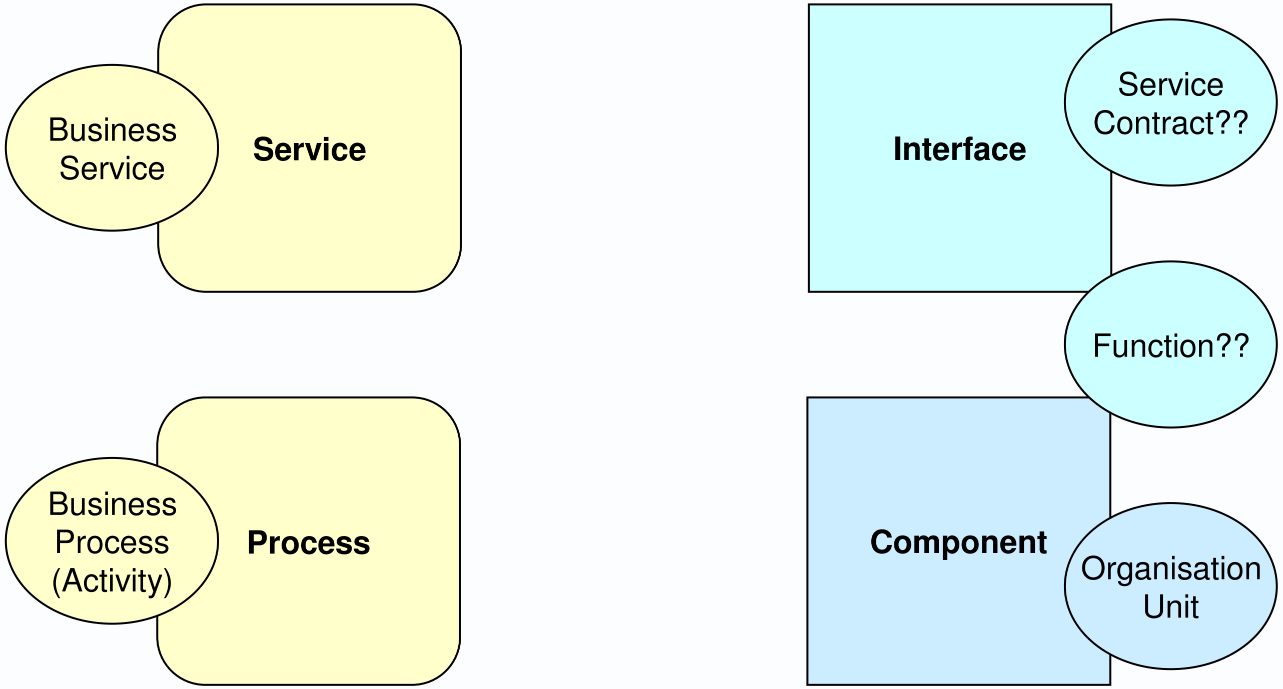
Mapping BCS generic meta model to TOGAF - generic

- Not clear if different authors have the same model in mind



Mapping BCS generic meta model to TOGAF - business

- Does TOGAF’s “service” = service or interface?
- Does TOGAF’s “logical” = technology-independent or external?

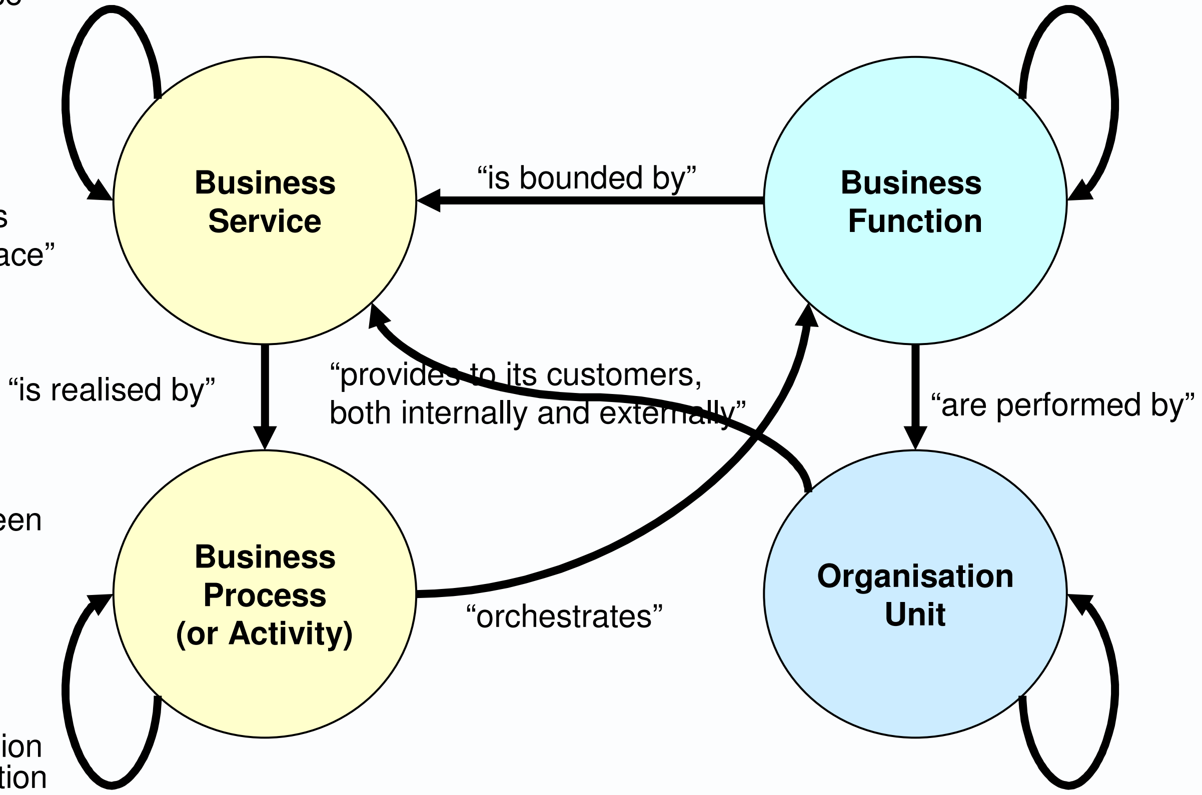


Best fit to TOGAF – business?

Service provides governed interface to access Function
Service is owned and governed by Organization Unit
Service supports Process
Service is realized by Process
Service consumes Service
Service decomposes Service

Function is owned by Organization Unit
Function supports Process
Function is realized by Process
Function is bounded by Service
Function decomposes Function
Function communicates with Function

“a specific function”
“explicit, defined boundaries
“a defined, measured interface”
“contracts with consumers”



“A unit of business capability”

“Correlation of organization and functions”

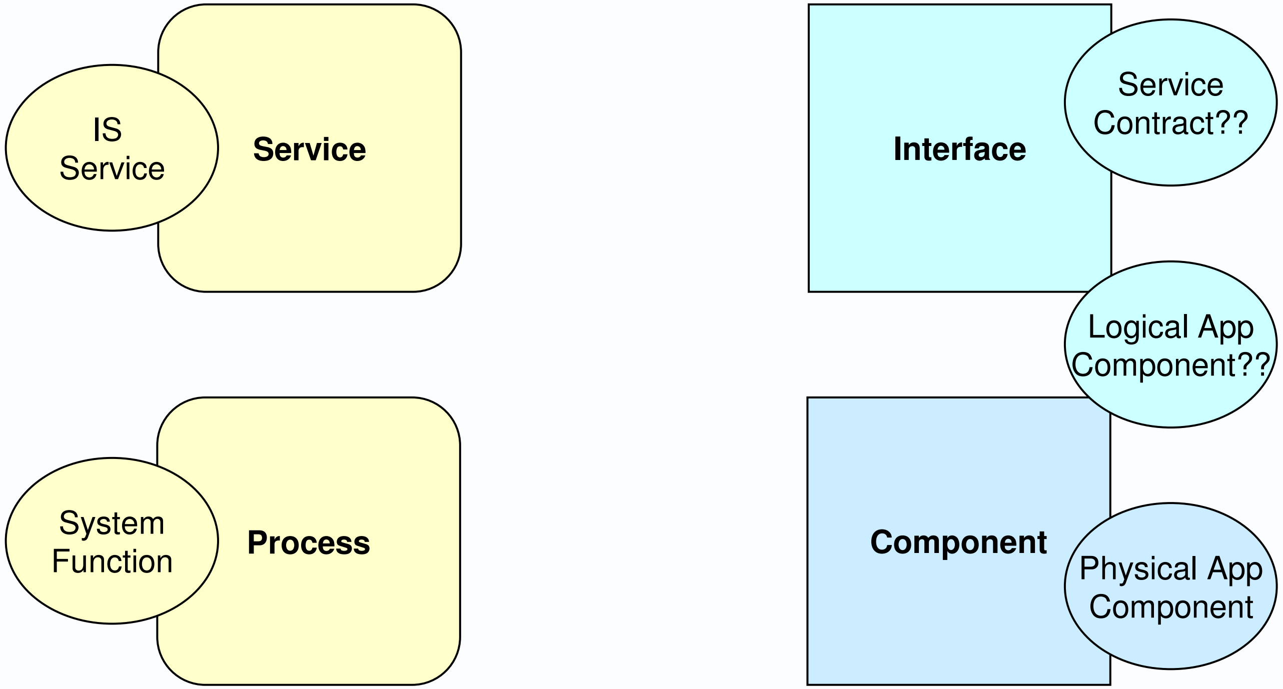
“a flow of interactions between functions and services”
“the flow of execution for a function”

Process orchestrates Function
Process decomposes Function
Process orchestrates Service
Process decomposes Service
Process decomposes Process
Process precedes/follows Process

Organization Unit owns Function
Organization Unit owns and governs Service
Organization Unit decomposes Organization Unit

Mapping BCS generic meta model to TOGAF - apps

- BCS Process = Function

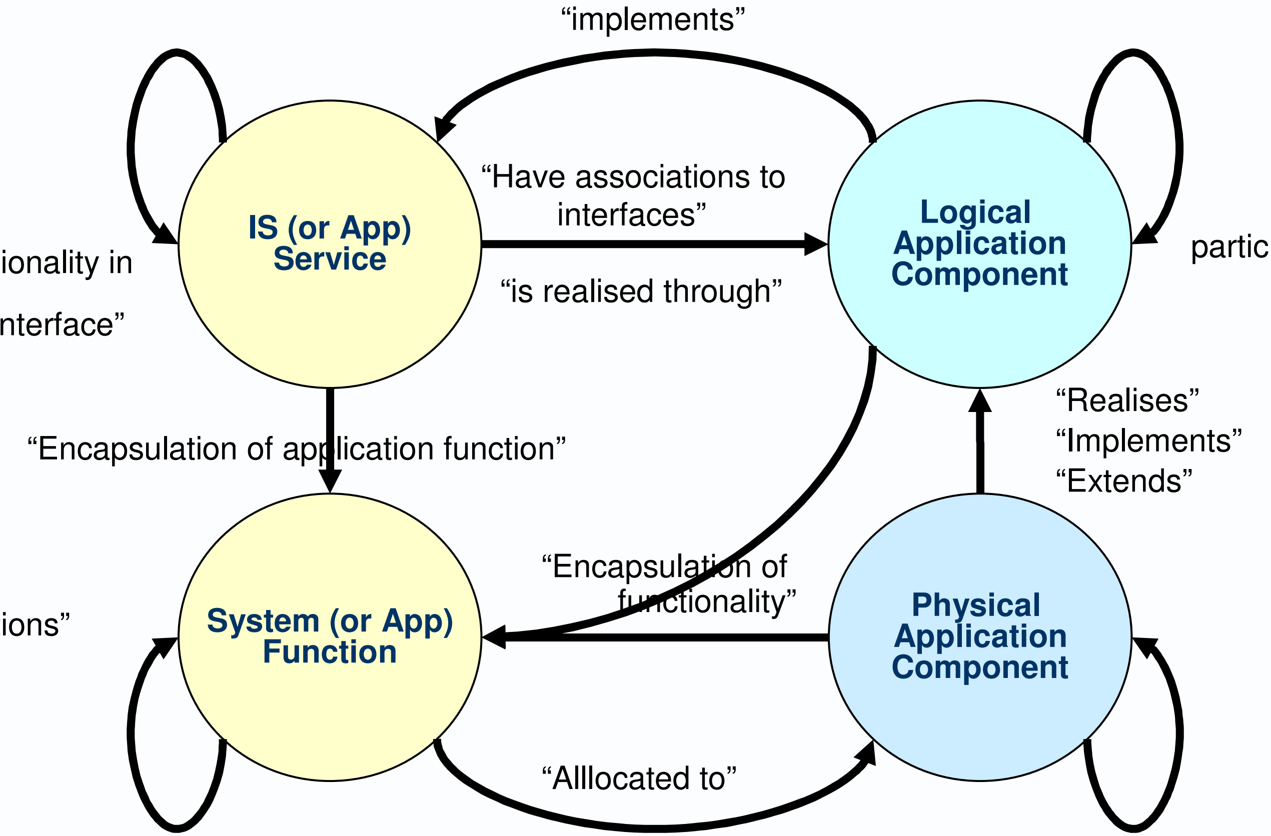


Best fit to TOGAF – apps?

Avancier

Service provides governed interface to access Function
Service is realized through Logical App Component

Logical App Implements Service
Logical App decomposes Logical App
Logical App communicates with Logical App
Logical App is extended by Physical App



“Automated”
“element of behavior”
“provides specific functionality in response to requests”
“has explicitly defined Interface”

“implements”

IS (or App) Service

“Have associations to interfaces”

“is realised through”

Logical Application Component

“independent of a particular implementation”

“Encapsulation of application function”

“Realises”
“Implements”
“Extends”

“Use-Case Models can describe systems functions”

System (or App) Function

“Encapsulation of functionality”

Physical Application Component

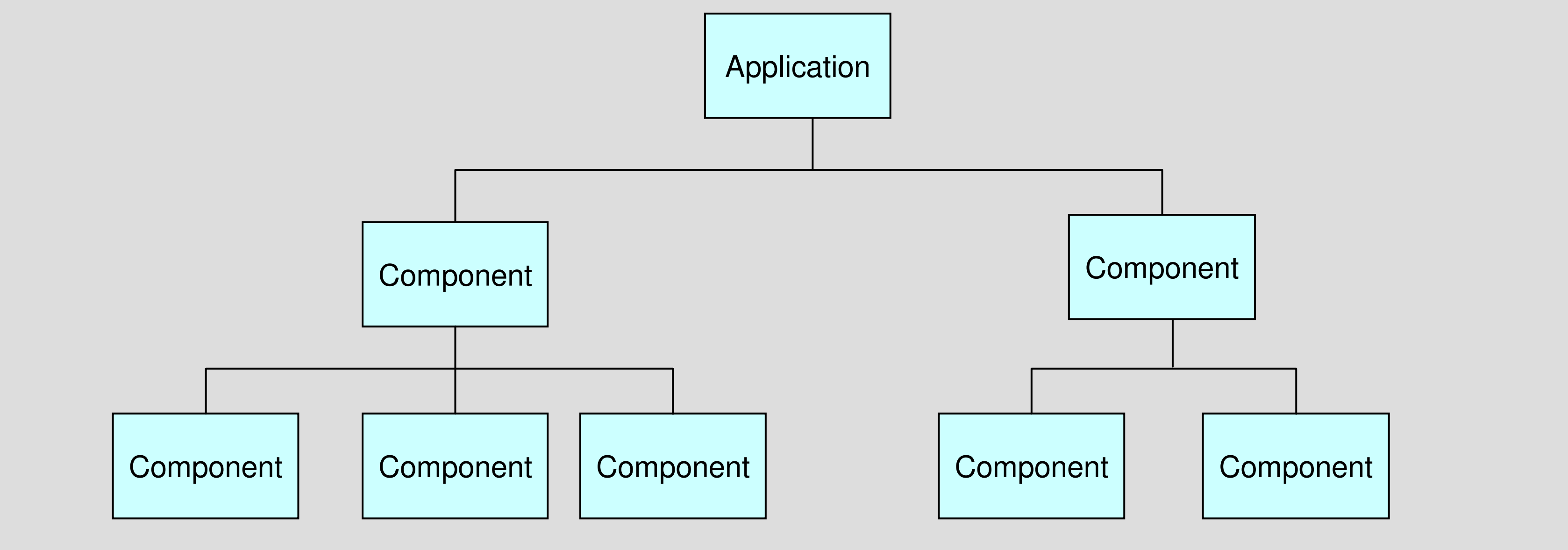
“deployable component of functionality”

“Alllocated to”

Physical App extends Logical App Component
Physical App encapsulates Physical App
Physical App decomposes Physical App
Physical App communicates with Physical App

Footnotes on what TOGAF says

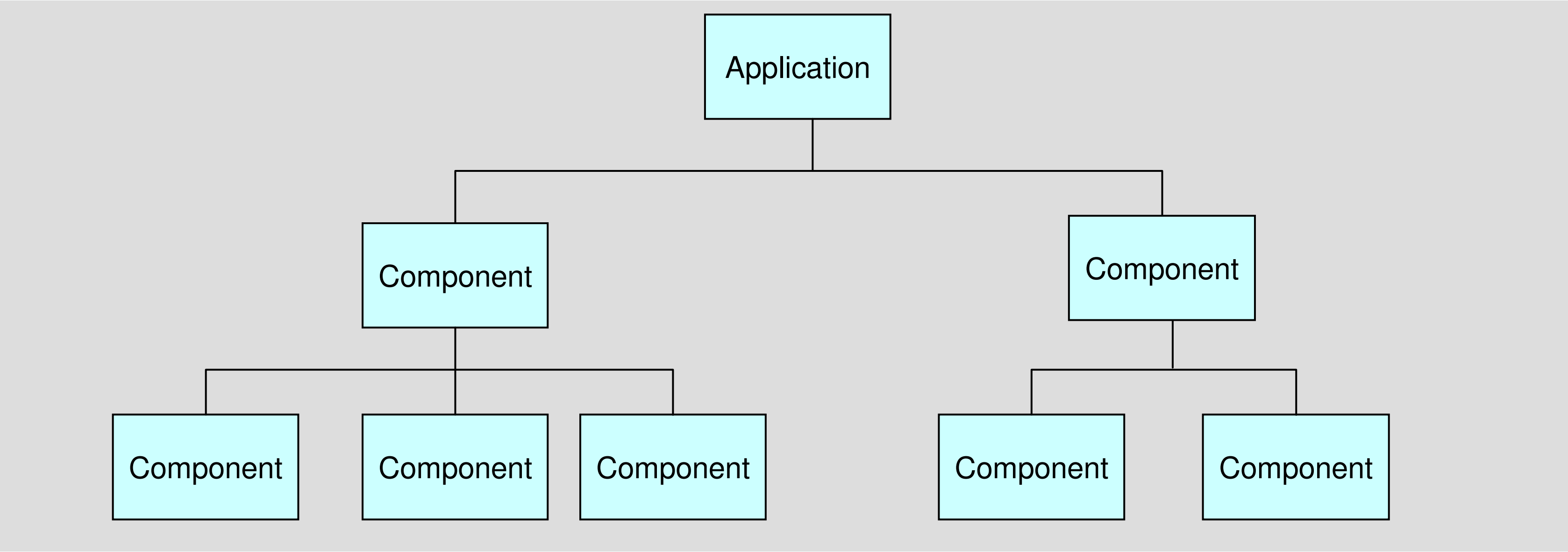
- "The organization's Application Portfolio is captured as a catalog. Catalogs are hierarchical and capture a decomposition of a metamodel entity"
- "**Application Component**: An encapsulation of application functionality that is aligned to implementation structuring."



Structural (component) decomposition in TOGAF

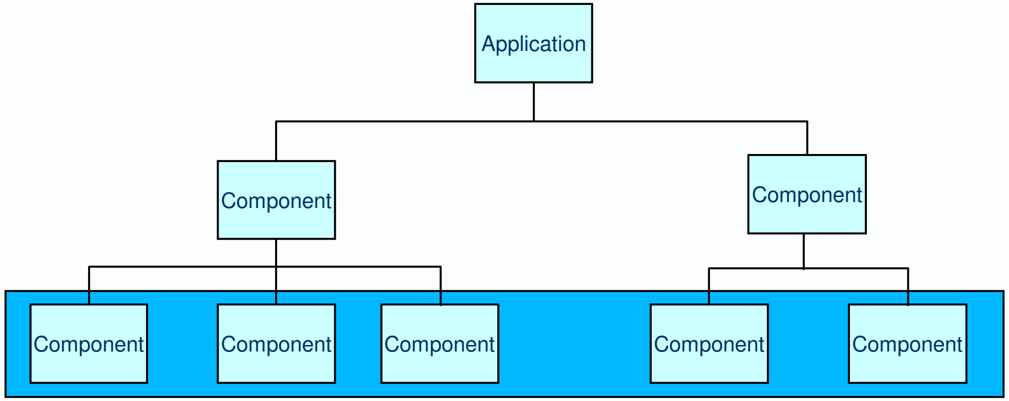
“Catalogs are hierarchical and capture a decomposition of a metamodel entity” TOGAF

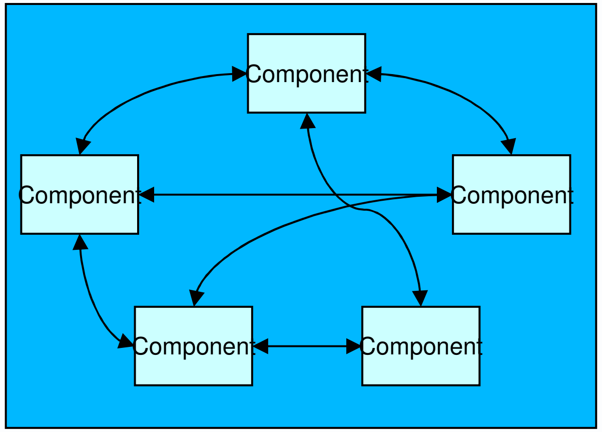
Application Components are components of a structure. They are subsystems – not processes.



Structural views of components

- An Application Communication Diagram is a Data Flow Diagram (rather than a Process Flow Diagram
- Boxes are Application Components (rather than Application Functions)
- Arrows are data flows (rather than control flows)





Behavioural (process) decomposition in TOGAF

- “three approaches can be utilized in sequence to progressively decompose a business.
- **Structured Analysis**
  - Identifies the key business functions within the scope of the architecture, and maps those functions onto the organizational units within the business.
- **Use-case Analysis**
  - The breakdown of business-level functions across actors and organizations allows the actors in a function to be identified and permits a breakdown into services supporting/delivering that functional capability.
- **Process Modeling**
  - The breakdown of a function or business service through process modeling allows the elements of the process to be identified, and permits the identification of lower-level business services or functions.” TOGAF

- TOGAF fudges
  - System v. process decomposition
  - Internal process v. external service
- However, it is clear that
  - Use cases are processes
  - Process models describe processes
- So this is process decomposition, or stepwise refinement (Wirth, 1971)

Application Functions as Use Cases (i.e. processes)

- “**Use-Case Models** can describe either business processes or systems functions, depending on the focus of the modeling effort.” TOGAF

- A system use case
  - is a process with a main path and alternative paths
  - is a high-level Application Function
  - is NOT a component
  - may involve several Application Components

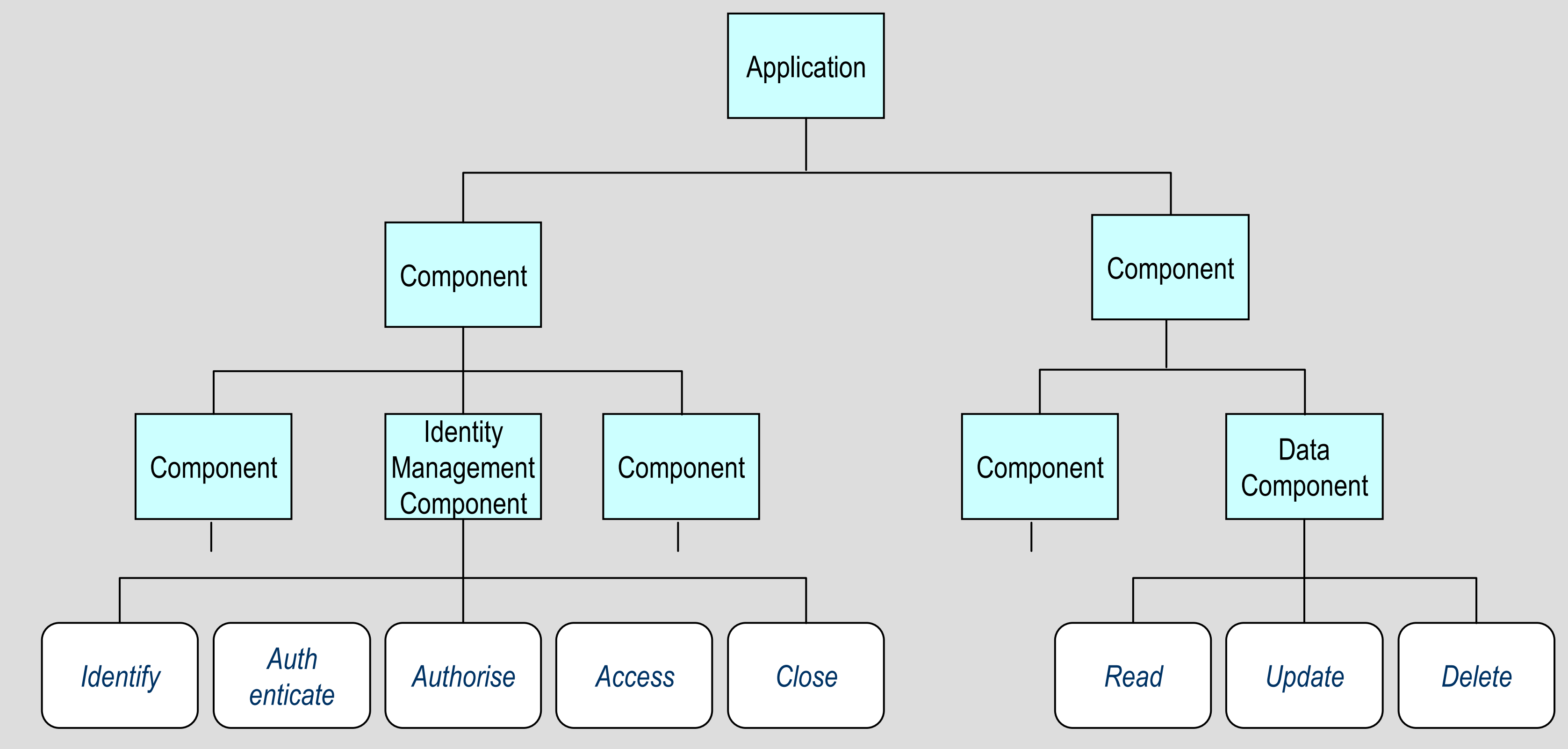
- App Functions can be decomposed into smaller Application Functions, each performed by an Application Component

Structural and behaviour views

"Application Component:
An encapsulation of
application functionality"
TOGAF

Elementary App Functions are

- *Grouped by cohesion criteria into App Components*
- *Orchestrated in sequence by higher App Functions*



Application Services as external view of Application Function

- “Encapsulation of application function as **Application Services**”

- **“Information System Service**:
- has a defined, measured interface
- has contracts with consumers of the service.
- has associations to SOA service interfaces.

- This appears to be an internal-external distinction
- An Application Component
  - describes Application Functions required or performed
  - to deliver one or more IS Services
- An Application Service
  - encapsulates one or more Application Functions,
  - describes a Service recognised by an Application User
  - might be an aggregate of elementary Services